

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ
КАФЕДРА СИСТЕМНОГО АНАЛІЗУ**

КВАЛІФІКАЦІЙНА РОБОТА

на тему: «ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ ДЛЯ ПІДТРИМКИ ТА
АНАЛІТИКИ ДЛЯ ПРОТАВЦІВ НА ТОРГОВЕЛЬНИХ
МАЙДАНЧИКАХ»

на здобуття освітнього ступеня магістра

зі спеціальності 124 Системний аналіз

освітньо-професійної програми Інтелектуальні системи управління

*Кваліфікаційна робота містить результати власних досліджень.
Використання ідей, результатів і текстів інших авторів мають посилання
на відповідне джерело*

(підпис)

Олександр Філатов
Ім'я, ПРІЗВИЩЕ здобувача

Виконав: здобувач вищої освіт
групи САДМ-61

Олександр Філатов

Керівник: Ровіл НАФЄЄВ

Рецензент: _____

Київ 2024

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ**

Кафедра Системного аналізу

Ступінь вищої освіти Магістр

Спеціальність 124 Системний аналіз

Освітньо-професійна програма Інтелектуальні системи управління

ЗАТВЕРДЖУЮ

Завідувач кафедри СА

_____Ровіл НАФЄЄВ

«___» _____20__р.

**ЗАВДАННЯ
НА КВАЛІФІКАЦІЙНУ РОБОТУ**

Філатов Олександр Олександрович

1. Тема кваліфікаційної роботи: Програмне забезпечення для підтримки та аналітики для продавців на торговельних майданчиках

керівник кваліфікаційної роботи Сергій КОЗАЧЕНКО к.е.н. ,
затверджені наказом Державного університету інформаційно-комунікаційних
технологій від «___» __20__р. № _____

2. Строк подання кваліфікаційної роботи «___» _____20__р.

3. Вихідні дані до кваліфікаційної роботи:

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

1. Аналіз вимог до програмного забезпечення: основні визначення та терміни, опис предметного середовища, огляд існуючих рішень, постановка задачі.

2. Моделювання та конструювання програмного забезпечення, моделювання та аналіз програмного забезпечення, засоби розробки, технічні рішення.

3. Аналіз якості та тестування програмного забезпечення.

4. Розгортання та впровадження програмного забезпечення.

5. Перелік ілюстративного матеріалу: *презентація*

6. Дата видачі завдання «___» _____20__р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів кваліфікаційної роботи	Строк виконання етапів роботи	Примітка
1	Вивчення рекомендованої літератури		
2	Аналіз існуючих методів розв'язання задачі		
3	Постановка та формалізація задачі		
4	Розробка інформаційного забезпечення		
5	Алгоритмізація задачі		
6	Розробка програмного забезпечення		
7	Налагодження програми		
8	Виконання графічних документів		
9	Оформлення пояснювальної записки		
10	Подання КР на попередній захист		
11	Подання КР рецензенту		
12	Подання КР на основний захист		

Здобувач(ка) вищої освіти

(підпис)

Олександр ФІЛАТОВ

(Ім'я, ПРІЗВИЩЕ)

Керівник
кваліфікаційної роботи

(підпис)

Ровіл НАФЄЄВ

(Ім'я, ПРІЗВИЩЕ)

РЕФЕРАТ

Текстова частина кваліфікаційної роботи на здобуття освітнього ступеня магістра складається з чотирьох розділів, містить 20 таблиць, 18 рисунків та 15 джерел – загалом 70 сторінок.

Дана кваліфікаційна робота присвячена створенню програмного забезпечення для контролювання товарів одразу з декількох торговельних майданчиків на одній платформі.

Мета: розробити програмне забезпечення для підтримки та аналітики для продавців на торговельних майданчиках.

У розділі аналізу вимог до програмного забезпечення було проведено аналіз предметної області та наявних альтернатив розроблюваному продукту, сформульовано прецеденти використання компонентів системи, висунуто технічні вимоги до ПЗ.

У розділі моделювання та конструювання програмного забезпечення було здійснено опис архітектури системи, наведено інтерфейси їхньої взаємодії, проаналізовано рівень безпеки даних у запропонованому рішенні.

У розділі аналіз якості та тестування програмного забезпечення було описано сценарії тестування та вимоги до взаємодії структурних модулів.

У розділі впровадження та супроводу програмного забезпечення було описано алгоритм розгортання компонентів програмної системи.

КЛЮЧОВІ СЛОВА: ВЕБДОДАТОК, MYSQL, ІНТЕРФЕЙС, СЕРВЕРНА ЧАСТИНА, КЛІЄНТСЬКА ЧАСТИНА, ТОРГОВЕЛЬНИЙ МАЙДАНЧИК, МОДЕЛЬ С4.

ABSTRACT

Text part of the master's qualification work consists of four sections, contains 20 tables, 18 figures and 15 sources – in total 70 pages.

The purpose of the master's qualification work is the creation software for the control of products from several trading platforms on one platform.

Objective: develop software for support and analytics for vendors on trading platforms.

Analysis of the subject area and available software alternatives was conducted in the software requirements analysis section. The use cases of the system components were formulated, technical requirements for the software were derived.

The software modelling and design section describes the architecture of the system components, interfaces of their interaction, and data security of the proposed solution.

The software quality analysis and testing section describes test cases and requirements for the interaction between the structural modules of the system.

In the software implementation and maintenance section, the algorithm for deploying software system components was described.

KEYWORDS: WEB APP, MYSQL, INTERFACE, SERVER SIDE, CLIENT SIDE, TRADING PLATFORM, C4 MODE

**ДЕРЖАВНИЙ УНІВЕРСИТЕТ
ІНФОРМАЦІЙНО-КОМУНІКАЦІЙНИХ ТЕХНОЛОГІЙ
НАВЧАЛЬНО-НАУКОВИЙ ІНСТИТУТ ТЕЛЕКОМУНІКАЦІЙ**

**ПОДАННЯ
ГОЛОВІ ЕКЗАМЕНАЦІЙНОЇ КОМІСІЇ
ЩОДО ЗАХИСТУ КВАЛІФІКАЦІЙНОЇ РОБОТИ
на здобуття освітнього ступеня магістра**

Направляється здобувач Філатов О. О. до захисту кваліфікаційної роботи

(прізвище та ініціали)

за спеціальністю 124 Системний аналіз

освітньо-професійної програми Інтелектуальні системи управління

на тему: «Програмне забезпечення для підтримки та аналітики для продавців на торговельних майданчиках».

Кваліфікаційна робота і рецензія додаються.

Директор ННІТ _____

(підпис)

Владислав КРАВЧЕНКО

(Ім'я, ПРІЗВИЩЕ)

Висновок керівника кваліфікаційної роботи

При написанні магістерського проекту автор показав глибоке систематичне володіння матеріалом у своїй галузі, здатність до здійснення логічних висновків, продемонстрував вміння грамотно та обґрунтовано проводити аналіз джерел, обізнаність у сучасних трендах розробки інформаційних систем, підходів до синтезу даних із різних систем, що свідчить про глибоке володіння автора сучасними знаннями, отриманими під час навчання, а також про уміння застосовувати набуті навички під час вирішення практичних задач. Під час виконання завдань з магістерського проекту, студент активно взаємодіяв з розробниками торговельних майданчиків, проявив високі комунікативні здібності. Розробка програмного продукту здійснена на високому інженерно-технічному рівні. Графічний матеріал проекту виконаний у відповідності креслень вимогам ДСТУ, ЕСКД.

Все це дозволяє оцінити виконану кваліфікаційну роботу здобувача Філатова Олександра Олександровича оцінку «відмінно» та присвоїти йому(їй) кваліфікацію «Магістр з системного аналізу».

Керівник кваліфікаційної роботи _____

(підпис)

Ровіл НАФЄЄВ

(Ім'я, ПРІЗВИЩЕ)

«___» _____ 20__ року

Висновок кафедри про кваліфікаційну роботу

Кваліфікаційна робота розглянута. Здобувач(ка) Філатова О.О. допускається до захисту даної роботи в Екзаменаційній комісії.

Завідувач кафедрою Системного аналізу

(назва)

(підпис)

Ровіл НАФЄЄВ

(Ім'я, ПРІЗВИЩЕ)

ВІДГУК РЕЦЕНЗЕНТА

на кваліфікаційну магістерську роботу

здобувача(ки) вищої освіти Філатов Олександр Олександрович
на тему «Програмне забезпечення для підтримки та аналітики для продавців
на торговельних майданчиках»

Поданий на рецензування магістерського проєкт містить 88 аркушів тексту,
18 ілюстрацій, 20 таблиць, 4 додатки.

Матеріал роботи викладено логічно та послідовно, та відповідає темі та
завданню магістерського проєкту.

В першому розділі проведено аналіз предметної області, визначено
функціональні та не функціональні вимоги до ПЗ, обрано технології для
розробки. В другому розділі проведено конструювання та моделювання ПЗ,
проведено аналіз безпеки даних. Третій розділ присвячений аналізу якості та
тестування ПЗ, що розбувається. В четвертому розділі розкриті питання
впровадження та супроводу ПЗ.

Ілюстративні матеріали кожного розділу повністю та наочно розкривають
зміст роботи, креслення відповідають чинним нормативним документам.

При виконанні проєкту були використанні сучасні технології та програмні
засоби, як Node.js, React.js, MySQL.

До недоліків можна віднести те, що розроблене програмне забезпечення не
може повністю замінити взаємодію з торговельними майданчиками,
внаслідок відсутності в них API до створення оголошень, але це не впливає
на загальну позитивну оцінку роботи. Вважаю, що результати роботи готові
до впровадження.

Висновок: кваліфікаційна робота на здобуття ступеня магістра заслуговує
оцінку "відмінно", а здобувач Булко Юрій Вадимович заслуговує присвоєння
кваліфікації: «Магістр з системного аналізу»

Рецензент:

науковий ступінь, вчене звання

підпис

Ім'я, ПРІЗВИЩЕ

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ	9
ВСТУП	10
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	12
1.1. ОПИС І АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	12
1.2. ВИЗНАЧЕННЯ ПОКАЗНИКІВ АНАЛІТИКИ ЗА ОГОЛОШЕННЯМИ	15
1.3. РОЗРОБЛЕННЯ ФУНКЦІОНАЛЬНИХ ВИМОГ	16
1.4. РОЗРОБЛЕННЯ НЕФУНКЦІОНАЛЬНИХ ВИМОГ ТА ВИБІР ТЕХНОЛОГІЙ ДЛЯ	
1.5.1. <i>Аналіз відомих програмних продуктів</i>	40
1.6. ВИСНОВКИ ДО РОЗДІЛУ	47
2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ	49
2.1. МОДЕЛЮВАННЯ ПРОГРАМНОЇ СИСТЕМИ	49
2.2. АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	50
2.3. КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	53
2.4. АНАЛІЗ БЕЗПЕКИ ДАНИХ.....	59
2.5. ВИСНОВКИ ДО РОЗДІЛУ	62
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО	
ЗАБЕЗПЕЧЕННЯ	63
3.1. АНАЛІЗ ЯКОСТІ ПЗ.....	63
3.2. ОПИС ПРОЦЕСІВ ТЕСТУВАННЯ.....	64
3.2.1. <i>Компоненти що тестуються</i>	64
3.2.2. <i>Підхід до тестування</i>	64
3.2.3. <i>Критерій проходження тесту</i>	65
3.3. ОПИС КОНТРОЛЬНОГО ПРИКЛАДУ	65
3.4. ВИСНОВКИ ДО РОЗДІЛУ	67

4	ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО	
	ЗАБЕЗПЕЧЕННЯ	68
4.1.	РОЗГОРТАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	68
4.2.	РОБОТА З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ	69
	ВИСНОВКИ	79
	СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	80

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

API	– Application programming interface, прикладний програмний Інтерфейс
C4	– Модель графічного запису для моделювання архітектури програмних систем
ОС	– Операційна система
БД	– База даних
UI	– Інтерфейс користувача

ВСТУП

Успішна торгівля на торговельних майданчиках у сучасних умовах вимагає не лише збору інформації про товари та їх прибутковість, але й вміння проводити аналіз цих даних. Продавці, які майструють графіки та розуміють, які позиції продаються найкраще, завжди знаходяться на передовій перед конкурентами. Вони ефективно розпоряджаються ресурсами, уникаючи постачання неефективних товарів та раціоналізуючи витрати.

Надто уважні продавці передбачають попит та слідкують за залишками, щоб уникнути ситуацій, коли на товарі вказується "немає в наявності" та зберегти його видимість для покупців. Без цих вмінь усі зусилля щодо розвитку продажів можуть залишитися безрезультатними.

Торговельні майданчики, відомі також як "класифайди" в сучасному світі, є платформами з оголошеннями від фізичних та юридичних осіб, що групуються за різними тематиками. Серед найпопулярніших українських "класифайдів" варто відзначити: [olx](#), [izi.ua](#), [ria.com](#), [auto.ria.com](#), [besplatka.ua](#) та інші. Ці ресурси переважно спрямовані на взаємодію між споживачами і пропонентами товарів або послуг, тому вони відносяться до типу C2C.

Проте при роботі продавця на подібних торговельних майданчиках виникає проблема використання кількох систем одночасно, потреба у реєстрації на кожній з них, відмінності у функціоналі персональних кабінетів та невідповідності у форматах подання статистики. Ці фактори стають актуальними при прагненні оптимізувати та спростити робочий процес продавця.

Мета даної роботи полягає в розробці програмного забезпечення для підтримки та аналізу діяльності продавців на торговельних майданчиках. Це передбачає розв'язання наступних завдань:

- проведення аналізу існуючих аналогів програмного забезпечення, що використовується для управління продажами на торговельних майданчиках;

- вибір оптимальної архітектури та визначення технологій для реалізації програмного продукту;
- проектування структури бази даних, яка відповідатиме вимогам функціонування програм;
- розробка функціональних та нефункціональних вимог до програмного забезпечення з урахуванням потреб користувачів;
- проведення тестування програмного продукту з метою оцінки його якості та ефективності функціонування.

Ця робота спрямована на створення інструменту, який спростить та оптимізує робочий процес продавців на торговельних майданчиках, забезпечуючи їм зручний інструмент для аналізу, управління та оптимізації їхньої діяльності.

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1. Опис і аналіз предметної області

Програмне забезпечення для підтримки та аналітики продавців на торговельних майданчиках є спеціалізованим вебсайтом, належить торгівій фірмі та призначене для забезпечення швидкого та зручного аналізу оголошень з кількох ресурсів.

Програма вирішує проблему аналізу оголошень одразу з декількох ресурсів.

Цей програмний продукт інтегрує різноманітні ключові функції, які значно полегшують користувачеві роботу та забезпечують зручність та ефективність. Основні аспекти, що були успішно інтегровані в дане програмне забезпечення, включають:

Глобальний збір статистичних даних: Програма здатна збирати та аналізувати статистичні дані з різних джерел, надаючи користувачеві повну картину щодо продуктивності та результативності його оголошень на різних платформах.

Головні аспекти що були інтегровані до даного програмного забезпечення:

- швидкий доступ до декількох ресурсів. Програма надає можливість швидкого та одночасного доступу до різних торговельних майданчиків, спрощуючи процес вивчення та управління різними оголошеннями;
- вибір торговельних майданчиків. Користувач може обирати, з якими саме торговельними майданчиками працювати, що дозволяє налаштовувати роботу програми під свої потреби та уподобання;
- аналітика кожного оголошення. Система надає можливість отримувати детальну аналітику для кожного окремого оголошення, включаючи дані про його популярність, кількість переглядів, дії користувачів та інше;

- деактивація оголошень. Програма дозволяє користувачу вмикати або вимикати оголошення, що є важливою функцією управління активністю оголошень на різних платформах;
- додавання товару до списку проданих. Користувач може легко додавати товари до списку проданих, використовуючи програму, що дозволяє зберігати важливі дані про продажі для подальшого аналізу.

Під'єднання торговельного майданчика на разі підтримує два вебресурси, а саме:

- OLX [1] – це популярна платформа онлайн-оголошень, спрямована на об'єднання людей для купівлі, продажу та обміну товарів та послуг. Основною метою OLX є створення зручної та доступної платформи для взаємодії між користувачами, незалежно від того, чи є вони представниками бізнесу, чи приватними особами. Для того щоб почати торгувати на OLX, користувачам необхідно пройти попередню реєстрацію, яку можна здійснити за допомогою електронної пошти, мобільного телефону або шляхом увійти через облікові записи у соціальних мережах. Після цього, користувачі мають можливість розміщувати свої товари, для чого потрібно заповнити форму подачі оголошення. Важливим етапом у розміщенні оголошення є створення його опису, додавання фотографій та вказання контактних даних. Користувачі мають можливість взаємодіяти з опублікованими товарами, комунікуючи з автором через онлайн-чат на платформі, використовуючи веб-сайт або зв'язуючись за допомогою наданих контактів, таких як номер телефону. Платформа OLX створена з метою спрощення процесу торгівлі для усіх її учасників, надаючи можливість швидко та зручно розміщати, переглядати та взаємодіяти з оголошеннями різних товарів та послуг. Вона є важливим інструментом для обміну товарами та послугами між користувачами з різних сфер життя, від особистого використання до бізнесу.

- AutoRia [2] – це відомий та популярний автомобільний портал, який функціонує як центральний базар автомобілів в Україні. Його основною метою є забезпечення відкритого доступу до різноманітної та корисної інформації про продаж автомобілів у різних регіонах країни. Одним із ключових аспектів AutoRia є його широкий спектр оголошень про продаж авто, які охоплюють різні марки, моделі та типи транспортних засобів. Платформа надає користувачам можливість ознайомитися з різноманітними варіантами автомобілів для продажу, представленими як від приватних власників, так і від офіційних дилерів. Крім того, AutoRia створює сприятливі умови для пошуку та фільтрації автомобілів з урахуванням різних критеріїв, таких як марка, модель, рік випуску, тип кузова, тип пального, об'єм двигуна та інші параметри. Це дозволяє користувачам швидко та зручно знаходити та порівнювати різні варіанти автомобілів згідно з їхніми потребами та уподобаннями. Платформа також надає корисну інформацію про технічні характеристики автомобілів, їхню історію обслуговування та стан, фотографії, відеоогляди, а також контактні дані власників або продавців для зручної комунікації. Окрім оголошень про продаж, AutoRia також надає корисну інформацію, таку як новини автомобільної індустрії, огляди нових моделей авто, поради з технічного обслуговування та експлуатації автомобілів. Це робить платформу важливим ресурсом для всіх, хто цікавиться автомобільною тематикою та шукає інформацію щодо покупки або продажу автомобілів в Україні.

Вхідні дані для системи автоматизації процесів включають отримання спеціальних токенів, необхідних для доступу до API різних платформ класифікованих оголошень. Програмне забезпечення має можливість отримувати ці токени, що дозволяє отримувати дані з цих майданчиків. Для доступу до OIх достатньо лише підтвердити доступ до сервісу через oauth2.

Однак для AutoRia, замість oauth2, необхідно вручну надати деякі ключі, такі як api key та user Id.

Отримання токена є ключовим етапом, оскільки він є результатом проходження авторизації та надає можливість отримати дані в форматі JSON з вибраних торговельних майданчиків. Ці дані відображають актуальну інформацію про оголошення та статистику, що дозволяє користувачеві отримати необхідну інформацію для подальшого аналізу та використання.

Таким чином, система отримує доступ до API різних майданчиків через отримання спеціальних токенів, що дозволяє здійснювати звернення та отримання даних у форматі JSON для подальшого використання в програмному забезпеченні.

1.2. Визначення показників аналітики за оголошеннями

Аналітика за оголошенням містить наступні показники:

- деталізована аналітика переглядів та спостережень: Окрім загальної кількості, додати можливість перегляду статистики за різними періодами (днями, тижнями, місяцями). Також відображати популярність оголошень на різних торговельних майданчиках;
- розширені дані по взяттям номера телефону: Перегляд аналітики, яка показує час взяття номера, конверсію та детальні дані про користувачів, які здійснили виклики. Можливість відслідковувати результативність номерів у різних оголошеннях;
- предиктивні аналітичні функції: Аналіз даних, щоб передбачити, які оголошення можуть потребувати поновлення чи редагування, зокрема, рекомендації до поновлення оголошення перед завершенням терміну дії;
- сегментація аналітики за аудиторією: Розширення засобів для аналізу поведінки користувачів, які взаємодіють з оголошенням, з відображенням даних про їхній профіль та інтереси;

- гнучкі засоби статистичного аналізу: Налаштовувані фільтри для вибору конкретних параметрів аналізу, таких як тип оголошення, місцезнаходження або ціновий діапазон товар

Кожне оголошення має встановлений термін свого дії. Якщо цей термін наближається до завершення, система автоматично надсилає користувачеві сповіщення, які нагадують про можливість оновлення або продовження строку дії оголошення. Для зручності продавців доступний інструмент, що відстежує терміни дії оголошень. Це дозволяє забезпечити своєчасне оновлення або вчасне видалення оголошень, що втратили актуальність.

1.3. Розроблення функціональних вимог

Для задоволення функціональних потреб програмного продукту необхідно врахувати ряд вимог. Між основні функції входить обробка авторизації користувачів, що включає процедури реєстрації, входу та виходу з системи. Крім цього, в системі реалізовано обробник помилок для валідації полів логіну та паролю. Технологія JWT забезпечує унікальний токен для кожного клієнта, що дозволяє серверу ідентифікувати користувача в системі.

До інших ключових функцій відноситься підключення до різних торговельних майданчиків, таких як OLX та AutoRia, з наступним сповіщенням продавця про успішне під'єднання. Після під'єднання, продавець може переглянути список своїх оголошень, сортувати їх за торговельним майданчиком або використовуючи функцію пошуку. Інформація в карточці товару має містити найважливіші деталі.

Також важливою функцією є відображення статистики: користувач може переглядати графіки за критеріями, такими як кількість переглядів, спостережень та взяття номерів телефону. Крім цього, в системі реалізовано вивід найпопулярніших оголошень та тих, у яких закінчується строк дії. Користувач може додавати товари до списку проданих; ця інформація зберігається без можливості подальшого редагування, оскільки вона впливає на загальну статистику.

Це набір функціональних вимог представляє собою основні можливості системи, які визначають способи взаємодії користувачів з програмним забезпеченням. Розширення цих вимог включає:

- можливість реєстрації користувача;
- можливість авторизації користувача;
- можливість відновлення пароля користувача;
- можливість підключення торговельних майданчиків;
- можливість перегляду всіх товарів з підключених торговельних майданчиків;
- можливість фільтрувати товари;
- можливість додавати товар у список проданих товарів;
- можливість перегляду статистики за вибраними товарами;
- можливість перегляду загального прибутку та прибутку за останній місяць;
- можливість перегляду профілю користувача;
- можливість редагування профілю користувача.

Вказані функціональні вимоги можуть бути подальшим чином деталізовані та проілюстровані за допомогою діаграми використання (додаток А) та таблиць (таблиця 1.1 – 1.11) для кращого розуміння користувачами продукту.

Таблиця 1.1 – Варіант використання входу користувача в систему.

Назва	Вхід продавця в систему
Особливості	Продавець може проводити всі дії, які необхідні для входу, а саме: зареєструватися або авторизуватися.
Учасники	Продавець
Передумови	Продавець не авторизований

Таблиця 1.2 – Варіант використання авторизації користувача.

Назва	Авторизація продавця
Особливості	Продавець має можливість увійти в систему через веб-додаток, підтверджуючи свою ідентифікацію та авторизацію.
Учасники	Не авторизований продавець.
Передумови	Продавець повинен мати можливість увійти в систему без попередньої авторизації.
Постумови	Продавець має можливість авторизуватися в програмному забезпеченні для доступу до його функціоналу.
Основний сценарій	Продавець скористався можливістю увійти до програмного забезпечення, натиснувши на відповідну кнопку, що викликала відображення екрану авторизації. У цьому вікні він ввів свої особисті облікові дані, включаючи електронну адресу та пароль, які вже були пов'язані з його обліковим записом. Після введення цих даних, він перейшов до наступного кроку, натиснувши кнопку "Далі", і в результаті система успішно ідентифікувала його як авторизованого користувача.

Таблиця 1.2 – Варіант використання авторизації користувача.

Розширення сценаріїв	Після введення особистих даних для авторизації у програмному забезпеченні, система виконує ретельну перевірку цих даних. Ця перевірка включає аналіз відповідності формату та збереженням в системі даним. Якщо система виявляє будь-яку невідповідність або помилку, наприклад, неправильно введені дані або наявність таких даних у системі, програма виводить повідомлення про помилку для користувача. Це сповіщення зазвичай з'являється у відповідній області інтерфейсу програми, часто у верхній правій частині, щоб забезпечити зручний доступ користувача до цієї інформації. Такий підхід дозволяє користувачу легко та швидко усунути помилку та завершити процес авторизації успішно. Повідомлення включає вказівки щодо того, як усунути проблему або виправити невідповідність. Це буде роз'яснення, яке допомагає користувачеві внести необхідні зміни у свої дані, або вказівки щодо подальших дій.
----------------------	---

Таблиця 1.3 – Варіант використання реєстрації продавця в програмному забезпеченні.

Назва	Реєстрація продавця в програмному забезпеченні.
Особливості	Продавець має можливість зберегти особисту інформацію, таку як електронна пошта та пароль, під час введення їх у програмне забезпечення. Це дозволяє зручно користуватися веб-додатком у майбутньому, оскільки вони можуть бути використані для автоматичного входу без повторного введення кожного разу при використанні системи. Таким чином, продавець має можливість ефективно керувати своїм обліковим записом та використовувати програму без необхідності повторного вводу особистих даних кожен раз під час входу.
Учасники	Неавторизований продавець
Передумови	Продавця не було попередньо внесено до списку зареєстрованих користувачів у системі.
Постумови	Після успішної завершені реєстрації, особисті дані продавця були успішно додані та збережені у системі.

Таблиця 1.3 – Варіант використання реєстрації продавця в програмному забезпеченні.

Основний сценарій	У програмному інтерфейсі введення особистих даних для реєстрації продавця представлено у формі, де можна внести всі необхідні відомості. Після заповнення цих даних система надає можливість переглянути та перевірити їх на вірність та коректність. Підтвердивши правильність введених даних, програма автоматично ініціює відправку листа на вказану електронну адресу, який містить інструкції для підтвердження реєстрації продавця. Цей лист містить посилання або інші відповідні дані для завершення процесу реєстрації.
Розширення сценаріїв	Програмне забезпечення проводить аналіз введених користувачем даних для створення облікового запису продавця. У випадку, якщо система виявляє, що обліковий запис, створений на основі цих даних, вже існує в системі, програма негайно повідомляє користувача про цю ситуацію. Це дозволяє уникнути створення дублюючих акаунтів.

Таблиця 1.4 – Варіант використання оновлення пароля продавця

Назва	Зміна пароля продавця в системі. Новий пароль можна створити або оновити існуючий, забезпечуючи зміну доступу до облікового запису.
Особливості	Продавець має опцію відновлення пароля, що надає можливість повернутися до свого облікового запису у випадку, якщо він забув чи втратив свій пароль для входу в систему.
Учасники	Авторизований продавець
Передумови	На момент передвходу в програмне забезпечення, продавець попередньо підтвердив свою особисту інформацію через відповідні процедури авторизації, забезпечуючи вірність та коректність введених даних. Дана дія дозволила продовжити використання програмного інтерфейсу з використанням вже підтверджених персональних даних.
Постумови	Продавцю надається можливість змінити свій пароль за потреби. Ця опція дозволяє забезпечити безпеку облікового запису шляхом оновлення пароля на більш надійний та запам'ятовуваний.

Таблиця 1.4 – Варіант використання оновлення пароля продавця

Основний сценарій	На сторінці "Налаштування" продавець знаходить та обирає опцію "Оновити пароль". У програмному інтерфейсі з'являється спеціальна форма, де продавець має заповнити свою електронну адресу. Після введення email і натискання кнопки "Відправити", програма забезпечує можливість змінити пароль, надсилаючи спеціальні інструкції на вказаний email. Таким чином, продавець отримує доступ до процедури зміни паролю через отриманий на пошту лист з інструкціями.
Розширення сценаріїв	Коли користувач вводить некоректні дані для процедури відновлення пароля, програмне забезпечення виявляє цю ситуацію та надсилає відповідне повідомлення. Це повідомлення може включати інформацію про неправильно введений електронний адресу або інші невірні дані, щоб користувач міг корегувати помилки та спробувати знову. Цей процес забезпечує чіткість для користувача, надаючи необхідну зворотну інформацію про помилку.

Таблиця 1.5 – Варіант використання підключення торговельних майданчиків.

Назва	Інтеграція з торговельними майданчиками
Особливості	Програма надає продавцю можливість підключити до свого облікового запису один чи навіть кілька торговельних майданчиків. Ця функція дозволяє об'єднати різні платформи продажу товарів або послуг, дозволяючи продавцю зручно керувати своїм асортиментом товарів або послуг усіма цими каналами одночасно через централізований інтерфейс програми. Такий підхід дозволяє ефективно керувати, контролювати та вести торгівлю на різних платформах, спрощуючи процес управління.
Учасники	Продавець
Передумови	Перед встановленням зв'язку з торговельними майданчиками, необхідно, щоб продавець був авторизований та мав попередньо створені облікові записи на обраних платформах продажу.
Постумови	Продавець успішно здійснив підключення до обраних торговельних майданчиків.

Таблиця 1.5 – Варіант використання підключення торговельних майданчиків.

Основний сценарій	Підтвердження успішного підключення до обраного торговельного майданчика є ключовою частиною функціональності програми. Це повідомлення передає важливу інформацію користувачеві про успішну взаємодію з майданчиком, що дозволяє впевнитися у готовності платформи до подальшої роботи. Воно не лише повідомляє про завершення процесу підключення, але й підтримує користувача впевненістю у правильності вибору майданчика для роботи, забезпечуючи комфорт та задоволення від користування програмою.
Розширення сценаріїв	Спроба підключення до обраного торговельного майданчика може спричинити автоматичне видалення створеного облікового запису, і програмне забезпечення повідомляє користувача про будь-які підозрілі дії чи помилки, пов'язані з цією процедурою на сервері.

Таблиця 1.6 – Варіант використання отримання доступних товарів

Назва	Отримання переліку доступних товарів зазначених торговельних майданчиків та їх відображення у програмному інтерфейсі
Особливості	Продавець у програмному забезпеченні має можливість переглядати асортимент товарів, які доступні для нього для подальшої роботи та взаємодії у межах системи. Ця можливість надає йому повний огляд товарів, які він може обробляти та працювати з ними у відповідності з його потребами.
Учасники	Продавець
Передумови	Для доступу до переліку товарів в програмному забезпеченні, продавець має пройти авторизацію. Відображення товарів стає можливим лише у випадку, якщо у продавця існує хоча б один товар, розміщений на одному з торговельних майданчиків, таких як OLX або AutoRia.
Постумови	Продавець отримав можливість перегляду переліку товарів, які стали доступні для нього на обраних торговельних площадках.

Таблиця 1.6 – Варіант використання отримання доступних товарів

Основний сценарій	Програмне забезпечення надає продавцю зручний та повний перелік доступних продуктів, який охоплює асортимент інвентарю з усіх підключених до системи торгових майданчиків, щоб забезпечити йому зручний та ефективний перегляд товарів у єдиному інтерфейсі.
Розширення сценаріїв	Внаслідок вилучення продукту під час переміщення від етапу успішної авторизації користувача на одному з торговельних майданчиків до переходу на сторінку зі списком продуктів, програма сповіщує продавця про цей несподіваний випадок. Це повідомлення може виникнути через виявлення підозрілих дій, що вказують на можливий конфлікт або невідповідність у процесі передачі даних між серверною та клієнтською частинами програми. В подібних ситуаціях програма негайно повідомляє користувача про можливі помилки та вимагає уваги для вирішення проблеми або для подальшої роботи з програмою, щоб запобігти можливим невідповідностям у майбутньому.

Таблиця 1.7 – Варіант використання – додавання товару у список проданих товарів.

Назва	Реєстрація товару в списку реалізованих продуктів.
Особливості	Продавцю надається можливість внести відповідну кількість товарів, які він бажає додати до списку проданих, що дозволяє управляти обсягами продажу в програмному інтерфейсі.
Учасники	Продавець
Передумови	Продавець, який використовує цю систему, повинен спочатку пройти процедуру авторизації, а також володіти щонайменше одним товаром, доступним для продажу на одному з підключених торгових майданчиків. Це вимога, необхідна для коректного використання можливостей програми, що забезпечує ефективну роботу з товаром та забезпечує можливість його відображення та управління через систему.
Постумови	Продавець успішно вніс вказану кількість товарів у список проданих. Це дозволяє продавцю ефективно керувати та відстежувати кількість товарів, які він розмістив у продажі, надаючи зручність у веденні обліку

Таблиця 1.7 – Варіант використання – додавання товару у список проданих товарів.

Основний сценарій	Коли продавець обирає товар для продажу та активує опцію "Продати", система відображає модальне вікно для введення кількості проданих одиниць товару. Після вказання кількості, продавець переходить до наступного етапу, натиснувши кнопку "Далі". Цей процес дозволяє користувачеві точно визначити кількість продукції, що продається, та продовжити операцію продажу.
Розширення сценаріїв	Виникнення ситуації, коли продукт видаляється між успішною авторизацією продавця на одному з торговельних майданчиків та переходом на сторінку з переліком продуктів, може призвести до відображення системою відповідної помилки у разі неуспішної валідації. Цей процес може надати продавцю зрозуміння про несправність деяких операцій, що відбулися у проміжку часу, та може забезпечити інформацію щодо подальших кроків для коректного відновлення даних.

Таблиця 1.8 – Варіант використання – деактивування продукту.

Назва	Відключення певного товару або вимкнення його активності.
Особливості	Продавцю надається можливість виключити вибраний товар із списку доступних для продажу. Після цієї дії товар не буде активним для покупців.
Учасники	Користувач
Передумови	Щоб отримати доступ до функціоналу, продавець повинен авторизуватися у системі та мати принаймні один товар на платформі, такий як OLX або AutoRia. Це вимога для коректного використання послуг і продажів через систему, а також для забезпечення належного функціонування і взаємодії з іншими торговельними майданчиками.
Постумови	Після вибору продукту і натискання кнопки «Додати до списку проданих» продавець має можливість вказати необхідну кількість одиниць продукту, яку він бажає додати до списку проданих. Це дозволяє зберігати відомості про кількість реалізованих товарів і зручно контролювати їх наявність у списку доступних товарів для продажу.

Таблиця 1.8 – Варіант використання – деактивування продукту.

Основний сценарій	На етапі вибору продукту, коли продавець вирішує деактивувати його, відбувається важливий крок: він натискає на кнопку «Deactivate», що призводить до відображення модального вікна, яке попросить його підтвердити цю дію перед подальшими кроками. Цей підтверджувальний етап гарантує свідоме та відповідальне управління станом продукту в системі, дозволяючи продавцеві впевнено керувати своїм асортиментом.
Розширення сценаріїв	Відстеження змін стану продукту може відбуватися у проміжку часу між успішною авторизацією продавця на різних платформах торговельних майданчиків і переходом на сторінку, що містить перелік доступних товарів. Цей процес дозволяє програмному забезпеченню сповіщати про можливі помилки у валідації отриманих даних, що в свою чергу допомагає уникнути можливих проблем у функціонуванні системи. Додатково, ця функція забезпечує сповіщення користувача про будь-які непередбачені зміни в статусі товарів.

Таблиця 1.9 – Варіант використання – перегляд статистики.

Назва	Отримання асортименту товарів, які доступні.
Особливості	Продавцеві відкритий доступ до перегляду статистики не лише проданих, але й активних товарів, що дозволяє відстежувати стан реалізації товарів та їх доступність для продажу в системі. Це надає продавцеві інформацію про ринкову активність та взаємодію з товаром.
Учасники	Продавець
Передумови	Для доступу до системи необхідно, щоб продавець був авторизований, а також мав хоча б один товар у продажу на одному з торгових майданчиків, наприклад, OLX чи AutoRia. Це обов'язкова умова для взаємодії з платформою та можливості працювати з продуктами.
Постумови	На екрані програми відображається інформація, що містить статистику товарів для продавця. Ця інформація включає в себе деталі про кількість та характеристики товарів, які є доступними для продажу. Такий зручний звіт дозволяє продавцю відстежувати активні товари та їх статуси.

Таблиця 1.9 – Варіант використання – перегляд статистики.

Основний сценарій	На екрані програми відображається ретельна інформація щодо продуктів, яка включає статистику товарів, доступних для продажу на усіх підключених торгових платформах. У цій відображеній інформації також присутня детальна специфікація щодо кількості та особливостей товарів, які були успішно реалізовані через платформи OLX і AutoRia. Це сприяє зручності та контролю для продавця, дозволяючи зберігати повний огляд та управління продуктовим асортиментом.
Розширення сценаріїв	В разі видалення товару після успішної авторизації продавця на платформі торговельного майданчика, але перед переходом на сторінку зі списком продуктів, програмне забезпечення може показати відповідне повідомлення про помилку валідації даних у системі. Це сповіщення надасть продавцеві інформацію про те, що інформація про товари не може бути оновлена, дозволяючи йому усвідомити проблему та вжити заходів

Таблиця 1.10 – Варіант використання – перегляд профілю користувача.

Назва	Отримання доступних продуктів
Особливості	На веб-платформі продавці можуть легко отримати доступ до власного профілю, де вони можуть переглядати та керувати своїми особистими даними та інформацією про продукти. Це включає в себе можливість переглядати та редагувати особисті дані, які дозволяють відобразити актуальну інформацію про себе, таку як контактні дані, адреси електронної пошти, та інші зміни у своєму профілі. Додатково, продавці мають можливість переглядати свою активність, статистику продажів та взаємодіяти з іншими функціями платформи, що полегшує їхню роботу та контроль над аккаунтом.
Учасники	Продавець
Передумови	Продавець авторизований.
Постумови	Продавець отримав дані профілю.
Основний сценарій	Система показує користувачу особисті дані та інформацію про налаштування профілю, дозволяючи змінювати її для покращення керування обліковим записом
Розширення сценаріїв	Система демонструє оповіщення про відповідну помилку.

Таблиця 1.11 – Варіант використання – фільтрація товарів.

Назва	Фільтрація товарів
Особливості	Продавець має можливість фільтрувати товари за вибраним сервісом.
Учасники	Продавець
Передумови	Продавець повинен бути авторизованим.
Постумови	Продавець відфільтрував товари за вибраним сервісом.
Основний сценарій	Продавець натискає на кнопку «Filter» на сторінці списку всіх продуктів та обирає торговельний.
Розширення сценаріїв	Система демонструє продавцеві повідомлення про відповідну помилку.

1.4. Розроблення нефункціональних вимог та вибір технологій для розробки

Нефункціональні вимоги для продавця у розробленому програмному забезпеченні є досить простими. Тобто для комфортного користування програмним забезпеченням користувачеві потрібен звичайний пристрій який має вихід в інтернет. В такому випадку можна розробити SaaS вебдодаток з

використанням бібліотеки React.js [4]. Таким чином програмне забезпечення отримує наступні переваги:

- кросплатформність;
- добре працює з SEO;
- продуктивність;
- зручна підтримка.

Також React.js використовує компонентно-орієнтований підхід який дає можливість з легкістю редагувати існуючі компоненти та здатність перевикористати код, тим самим це перетворює React розробку в процес покращення написання програмного забезпечення. Ті компоненти, які були створені у роботі над програмним забезпеченням, не мають додаткових залежностей. Тобто, ніщо не заважає їх перевикористати.

Бізнес-логіка розміщена на серверній стороні програмного забезпечення. Для цього можна використати Node.js [5]. Ця платформа була обрана як основна, тому що вона є наразі найпопулярнішою для написання серверної частини програмного забезпечення, використовує мову програмування Javascript [6]. Її перевагами є:

- швидкість та продуктивність – в його не блокуючі операції введення-виведення роблять середовище одним із найшвидших доступних варіантів. Код працює досить швидко та значно покращує все середовище виконання, а це своєю чергою пов'язано з його роздільною системою. Також він працює на движку JavaScript V8 який розроблений Google. Серверна частина, яка написана за допомогою Node.js має сумісність plug and play що сприяє швидкості та продуктивності розроблюючого додатку;
- ефективне кешування – це одна з ключових переваг Node.js. Оскільки, він має досить потужну здатність кешувати дані. В той час коли до додатка надходять запити, вони мають здатність кешуватися у пам'яті програми;

- гнучкість – гнучкість програмування це одне з найбільших переваг Node.js. Оскільки коли вносять зміни в Node.js додаток, це може вплинути лише на один вузол. І це чудово не тільки для початкової збірки, але й для подальшого обслуговування. Також це стосується того, коли поєднується JSON з Node.js, таким чином досить легко обмінюватися даними між клієнтським сервером та вебсервером. Також можна використовувати API для вбудовування TCP, HTTP, DNS тощо;
- здатність до масштабування проєкту;
- величезне ком'юніті розробників.

Node.js також дозволяє підключати інші модулі, написані іншими людьми на різних мовах програмування. Для цієї задачі створені різні пакетні менеджери, але в даній роботі було використано npm package manager. Оскільки він дає змогу комфортно керувати кількома версіями коду та його залежностями.

Сервер створено на основі Express.js [7], через те, що він має велику площу і являє собою дуже тонкий шар абстракції над даним вузлом для створення додатків на серверній стороні програмного забезпечення. Він повністю підключається і має величезну спільноту. Також він побудований з використанням передової та прогресивної технології (ES6), за допомогою якої з'являються очевидні переваги перед старими фреймворками.

У даного програмного забезпечення немає жорстких вимог що до масштабованості бази даних оскільки всі дані зберігаються виключно в базах торговельних майданчиків, а зв'язок відбувається через їхнє API. В такому випадку було вирішено вибрати реляційну СУБД MySQL [8].

Головною метою було забезпечити цілісність інформації в базі даних. Потрібно приділити увагу персональним даним користувачів. Для конфіденційних даних користувача використовується кілька кроків шифрування.

- по-перше, ключ шифрування даних шифрується за допомогою AWS Encryption SDK використовуючи AES у режимі GCM з 256 бітами розмір ключа;
- по-друге, ключ, який використовувався для шифрування ключа шифрування даних, зберігається в AWS KMS;
- по-третє, зашифрований ключ і дані кодуються в base64 і зберігаються в базі даних.

Тобто таким чином вдалося запобігти порушенню даних і витоків даних, оскільки, коли зловмисник скомпрометує дані одного користувача, неможливо отримати доступ до даних інших користувачів, використовуючи скомпрометовані, ключ шифрування, що зберігається в AWS KMS, і ключ шифрування даних який захищає дані користувача.

Мінімальні технічні параметри сервера:

- CPU – 64bit x86;
- RAM – 4 GB.

Рекомендовані параметри сервера:

- CPU – Multi Core 64bit x86;
- RAM – 8 GB or higher.

Технічний стек являє собою (React.js, Node.js, MySQL). Ці технології роблять програмне забезпечення кросплатформним, що означає, що воно може працювати на будь-якій операційній системі - Linux, MacOS або Windows. Однак, враховуючи його особливості, оптимальним варіантом для роботи серверної частини є Linux, оскільки він є менш вимогливим до ресурсів, має продуману файлову систему та є більш вартісним варіантом.

Логіка серверної частини програмного забезпечення ґрунтується на CRUD операціях, які відображають базові функції створення, читання, оновлення та видалення даних. Це забезпечує простоту та ефективність взаємодії з базою даних.

Для забезпечення безпеки даних, всі дані обов'язково мають передаватись в зашифрованому вигляді через мережу. Доступ до бази даних ззовні повинен бути обмежений лише за відповідним секретним ключем.

Для забезпечення безпеки та віддаленого доступу до сервера використовується протокол SSH (Secure Shell), який забезпечує шифрування та захищену зв'язок з сервером.

Для розгортання програмного забезпечення було обрано хмарний сервіс Amazon [9] AWS. Amazon - це класичний IaaS (Infrastructure as a Service) хостинг додатків, який надає віртуальне середовище для розгортання програмного забезпечення та сервісів, необхідних для його роботи. Обираючи AWS, отримуємо можливість вибору мови програмування, платформи інтернет-додатків, операційної системи, бази даних та інших необхідних сервісів, що спрощує процес міграції та зберігає можливість для створення нових рішень. Аналіз альтернатив

Після ретельного аналізу ринку та вивчення аналогів, таких як merchantspring.io та moneyplace.io, виявлено, що вони спеціалізуються на зборі інформації з маркетплейсів. Ці платформи орієнтовані на тип продажу B2C – "business to consumer", що означає взаємодію між бізнесом і кінцевим споживачем. Це форма продажу, коли товари або послуги продаються безпосередньо кінцевому користувачеві, і цей тип операцій може охоплювати різні процеси продажу без посередніх постачальників чи посередників.

У контексті власної розробки програмного забезпечення для підтримки та аналітики для продавців на торговельних майданчиках, основний акцент робиться на типі C2C – "customer to customer". Це тип взаємодії, де продавець та покупець виступають в якості кінцевих користувачів та безпосередньо укладають угоди один з одним через торговельний майданчик.

Специфіка цього програмного продукту полягає в тому, що він орієнтований на сприяння взаємодії та аналізу для користувачів, які активно торгують на майданчиках типу C2C. Він надає можливості аналізу оголошень,

статистики продажів та підтримки управління товарами безпосередньо між користувачами, не втручаючись у процеси B2C.

Це програмне забезпечення дозволяє продавцям ефективно аналізувати та управляти своїми продуктами на різних торговельних майданчиках типу C2C, спрощуючи процес торгівлі та надаючи засоби для управління та аналітики, спеціалізовані саме для цього типу взаємодії між користувачами.

1.5.1. Аналіз відомих програмних продуктів

Найбільш популярні платформи для збору статистичних даних:

Moneyplace

Moneyplace шукає популярні та маржинальні товари на Ozon, Aliexpress, Wildberries, СберМегаМаркет, Яндекс.Маркет та KazanExpress, інтерфейс зображено на рисунку 1.1.

ЗБРОГ ЭДЖЭНСИ ООО

Главная / Продавцы / **Продавец**

Выручка

9,733,297

Разница со вчерашним днем:

84% ↑

Продаж

12,143

Разница со вчерашним днем:

133% ↑

Состояние склада

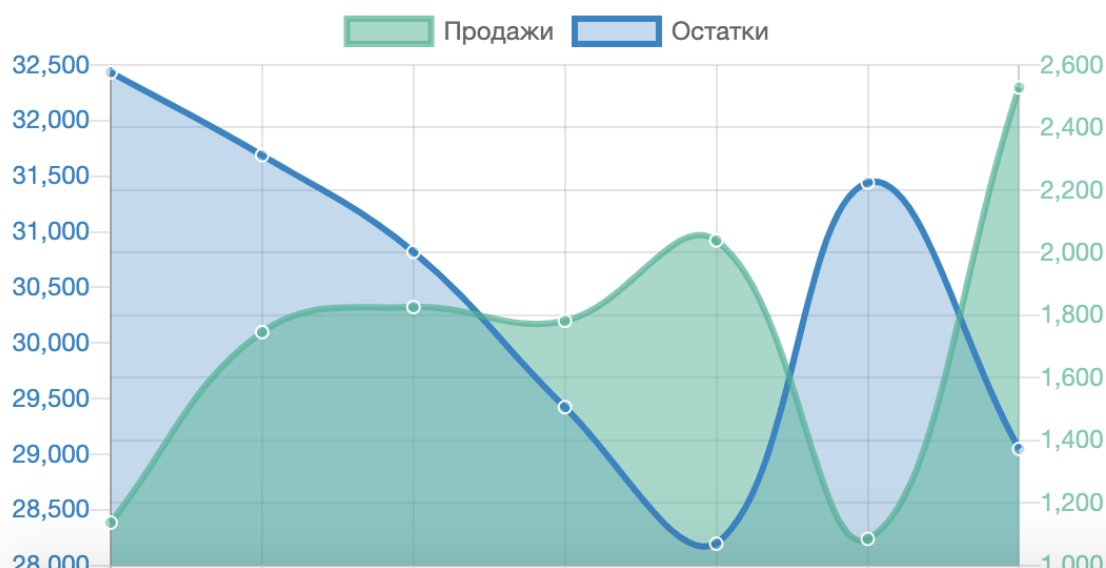


Рисунок 1.1 – Интерфейс Moneyplace

Даний сервіс виявляє новинки та тренди на ринку, знаходить товари, які показують стабільний попит. Він не лише допомагає знаходити ніші для бізнесу, але й ефективно аналізує існуючі магазини. Підключення до API магазинів дозволяє проводити глибокий аналіз продажів. Система відображає статистику щодо обороту, середнього чеку та залишків товарів. Крім того, сервіс контролює та розраховує поставки, надсилаючи рекомендації стосовно оптимальних дат поставок через повідомлення. Доступні різні тарифні плани: вартість користування коливається від 1000 до 30000 гривень на місяць. Кожен пакет має

свої особливості, такі як кількість користувачів, періоди аналізу, обсяги завантажених даних, можливість скористатися послугами персонального аналітика.

Необхідно відзначити, що серед недоліків даного продукту - його неспеціалізація на українському ринку. Це означає, що сервіс може не мати належного аналізу або підтримки функцій, специфічних для українського бізнесу. Однак, незважаючи на це, зазначений сервіс є потужним інструментом для аналізу ринку та управління продажами, який може бути корисним для підприємств, які діють на інших ринках.

Маяк

Сервіс Маяк є потужним інструментом для виявлення найпопулярніших товарів на платформі Wildberries. Цей сервіс здійснює аналіз кількості продажів у інших магазинах, досліджує пропущені можливості продажу, проводить детальний аналіз статистики виручки та обсягів продажів. Він також вивчає динаміку придбання товарів у різних категоріях та в пошукових запитах на платформі Wildberries. Інтерфейс цього сервісу показано на рисунку 1.2.

Використовуючи дані від інших продавців, Маяк дозволяє визначити найбільш популярні товари та ті, які потенційно можуть стати хітом продажів на платформі Wildberries. Завдяки аналізу статистичних показників та динаміки продажів, сервіс надає корисну інформацію для бізнесу, допомагаючи приймати обґрунтовані рішення стосовно асортименту товарів та стратегій продажів на платформі.

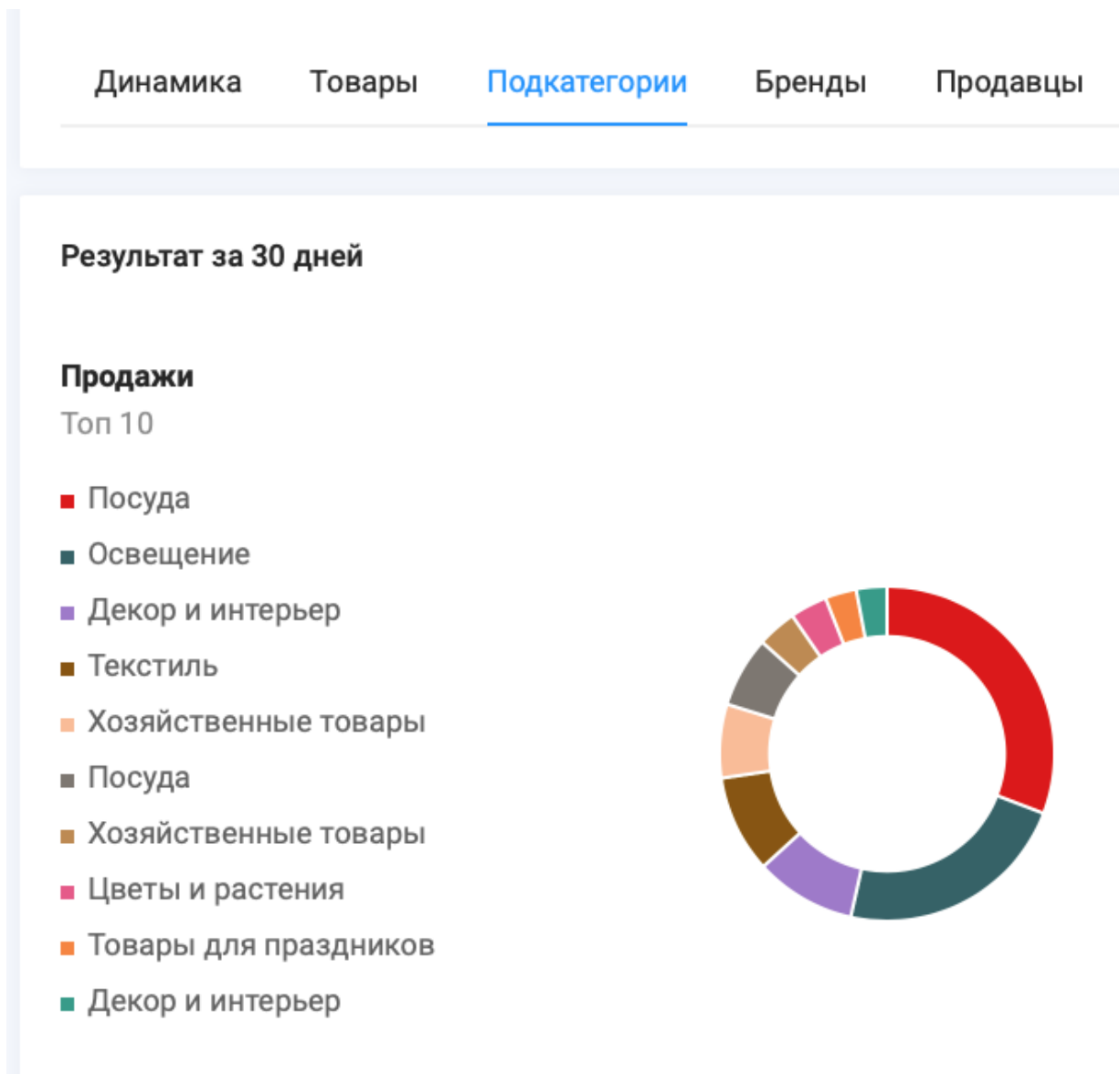


Рисунок 1.2 – Интерфейс Маяк

Сервіс є зручним у використанні завдяки його розширенню для браузера Chrome, яке надає основні дані статистики прямо у картці товару на платформі Wildberries. Крім того, сервіс пропонує користувачам бот у Telegram, який у відповідь на надіслане посилання на картку товару, надсилає графіки з даними про виторг, замовлення за останні 30 днів та залишки.

Сервіс пропонує безкоштовний тариф із обмеженим функціоналом, який є ідеальним для початківців з обмеженим бюджетом, щоб визначити нішу. Доступний також стандартний тариф за 1000 гривень на місяць. Для більш

досвідчених продавців доступний тариф "Професійний", який пропонує вибір ключових слів для описів товарів серед своїх ключових функцій.

Серед недоліків даного продукту можна відзначити відсутність спеціалізації на українському ринку, що може призвести до нестачі аналітики або підтримки функцій, специфічних для українського бізнесу. Також слід зазначити, що цей продукт підтримує тільки один торговельний майданчик, що є найменшим показником порівняно з конкурентами.

Shopstat

Shopstat – це безплатний сервіс аналітики збирає дані для аналізу з Wildberries та Ozon. Яндекс.Маркет та AliExpress у розробці. Показує аналітичні дані щодо будь-яких товарів, інтерфейс зображено на рисунку 1.3.

WB

Wildberries

OZON

Ozon

AliExpress

Aliexpress

Планировщик заказа

Данные на: 23.07.2020 10:18:23

Остатки по товарам

Бренды

Товар

Наименование	Категория	Бренд	SKU	Ос-ки, шт	Ос-ки, дней	Продаж	К заказу
Набор автомо...	Автотовары	Castlelady	123456	34	6	34	12 011
Аптечка перв...	Автотовары	Castlelady	123456	34	24	341	12 011
Трос буксиров...	Автотовары	Castlelady	123456	12	2	12	12 011
Компрессор	Автотовары	Castlelady	123456	34	6	34	12 011
Набор авто...	Автотовары	Castlelady	123456	34	24	341	12 011

Рисунок 1.3 – Интерфейс Shopstat

Даний сервіс надає можливість відстежувати зміни цін, вплив залишку товару на прибуток, а також визначати топові товари за певний період. Повна статистика категорій фільтрується за різними критеріями, такими як рейтинг, виторг, кількість, ціна та залишок. Сервіс дозволяє аналізувати попит без необхідності проведення тестових поставок товарів, перевіряти торгові гіпотези та визначати потенційну нішу на ринку.

Інформація про географію продажів за різними складами демонструється з урахуванням розмірів і кольорів, що дозволяє легко розрахувати оборотність та налаштувати постачання у регіонах, де товар користується популярністю.

Важливо відзначити, що інформація про продажі оновлюється щомиті, тому будь-який запит на статистику принесе лише актуальні дані. Однак серед недоліків даного продукту є те, що він не спеціалізується на українському ринку, що може призвести до обмежень у доступній аналітиці або підтримці функцій, специфічних для українського бізнесу.

Merchantspring

На платформі Merchantspring реалізована система для управління продуктивністю, яка охоплює низку важливих онлайн-ринків, включаючи Amazon, Walmart, Shopify та понад 75 інших ринків по всьому світу.

Платформа надає можливість одночасно вести діяльність на кількох відомих ринках, що сприяє покращенню ефективності та розширенню аудиторії продажів. Користувачам доступні інструменти для управління та аналізу продажів, моніторингу запасів, прогнозування попиту, оптимізації цін та багато іншого.

Важливою перевагою платформи є її широкий охоплення різноманітних ринків, що дає змогу підприємствам вести бізнес у різних країнах світу через один інтерфейс. Однак, слід зазначити, що така широка функціональність може вимагати додаткового часу та ресурсів для оволодіння платформою, особливо враховуючи велику кількість ринків та можливостей, які вона надає. Інтерфейс цієї платформи представлений на рисунку 1.4.

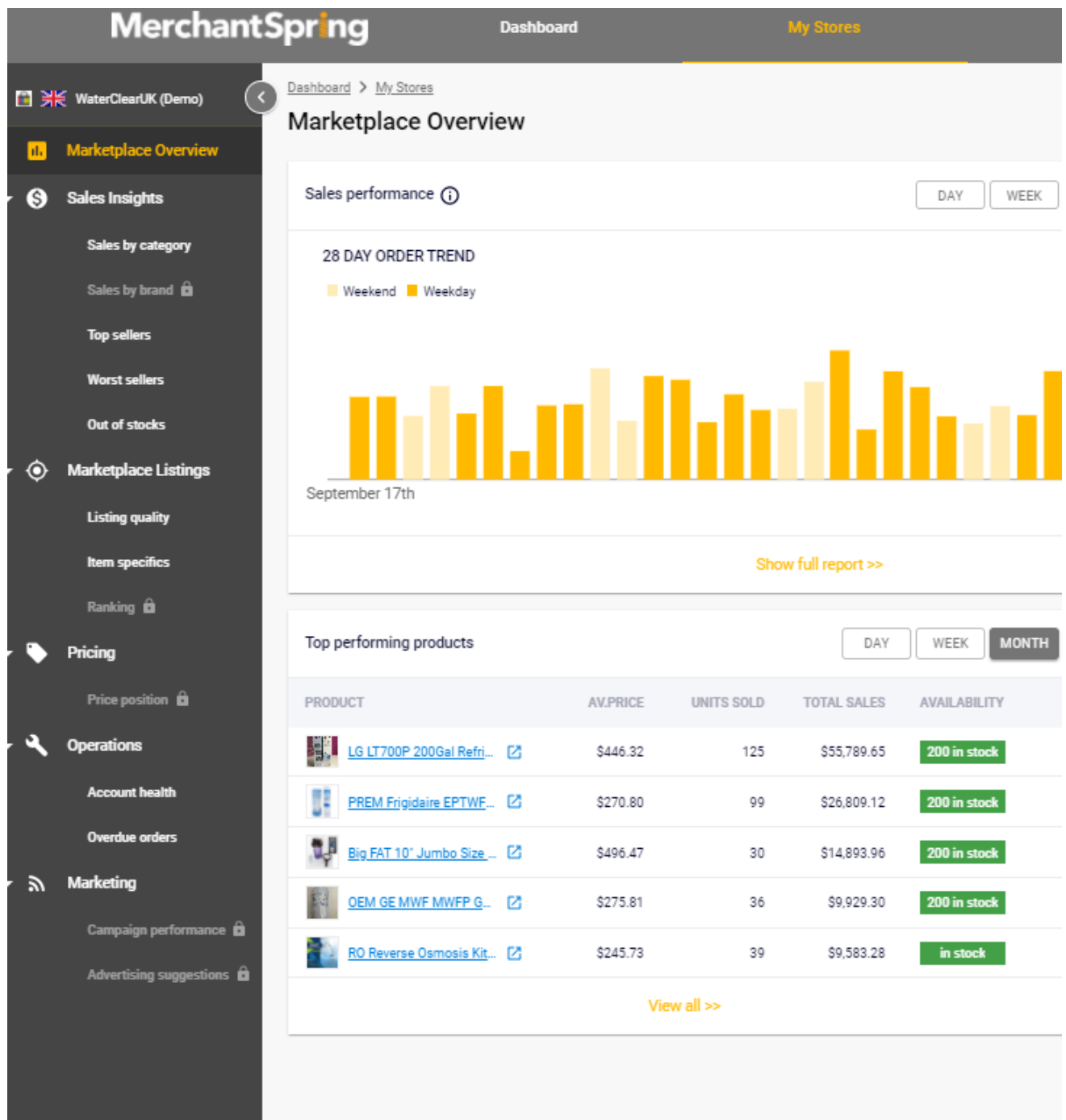


Рисунок 1.4 – Інтерфейс Merchantspring

Ця платформа надає можливість стежити за тенденціями продажів, оцінювати якість списку товарів, а також вимірювати коефіцієнти конверсії трафіку через зручний мобільний центр та систему сповіщень.

Порівняно з іншими подібними платформами, Merchantspring, хоч і конкурує з ними, проте має свої унікальні особливості. Відмінність полягає в

тому, що ці конкуруючі платформи не орієнтуються на український ринок та не спеціалізуються на типі продажу C2C.

Зазначеною проблемою є відсутність спеціалізації продукту на українському ринку, що може стати чинником обмеження для користувачів з цієї локації. Додатково, складність архітектури може викликати труднощі у розумінні та використанні продукту для деяких користувачів.

Зокрема, важливо відмітити, що веб-застосунок цієї платформи є сумісним з усіма сучасними браузерами, що підвищує його доступність для широкого кола користувачів. Крім того, переважна частина бізнес-логіки розташована на серверній частині програмного забезпечення, що може сприяти покращенню продуктивності та забезпеченню стабільності роботи платформи.

1.5. Висновки до розділу

У розділі проведено глибокий аналіз вимог до програмного забезпечення. Серед розглянутих альтернатив не було знайдено жодного рішення, яке в повній мірі задовольняло б усі висунуті функціональні та нефункціональні вимоги. Багато з аналогів не забезпечують вихід на український ринок, не мають необхідного типу торговельних майданчиків C2C, тому розроблене програмне забезпечення стає актуальним та унікальним у своїй специфікації.

Важливим аспектом є те, що розроблене програмне забезпечення орієнтоване саме на український ринок та має властивий йому тип торговельних майданчиків C2C. Це дає можливість ефективно використовувати його для обміну товарами та послугами між користувачами на теренах України, що створює перевагу перед конкурентами.

В даному розділі було чітко визначено необхідний перелік програмних засобів, які будуть використовуватися для реалізації даного проєкту. На підставі аналізу була обрана клієнт-серверна архітектура, де для забезпечення серверної логіки було обрано Node.js, а для клієнтської частини використано React.js. У ролі системи управління базами даних було обрано MySQL, яка забезпечить надійне та швидке зберігання та обробку інформації. Використання такого

технічного стеку забезпечує надійність, швидкість та оптимальність у роботі програмного забезпечення.

Усі ці аспекти дозволяють упевнено стверджувати про актуальність та відмінність розробленого програмного продукту на фоні конкурентів, а також про його відповідність вимогам українського ринку С2С.

2 МОДЕЛЮВАННЯ ТА КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Моделювання програмної системи

Для ефективного моделювання програмної системи обрано модель С4 [10], яка є простим та ефективним методом графічного зображення архітектури програмних систем. Основна ідея моделі С4 полягає у структурній декомпозиції системи на контейнери та компоненти, використовуючи методи, такі як Unified Modeling Language (UML) або Entity Relation Diagrams (ERD).

Цей підхід дозволяє узагальнено та зрозуміло представити складні структури програмних систем через ієрархічне подання їх складових частин. Його основна перевага полягає в універсальності та лаконічності відображення різноманітних аспектів системи, що полегшує сприйняття та спілкування між учасниками

Точки огляду організовані відповідно до їх ієрархічного рівня:

- діаграми контексту (рівень 1): вони показують систему у масштабі та її взаємовідносини з користувачами та іншими системами;
- діаграми контейнерів (рівень 2): вони розбивають систему на взаємопов'язані контейнери. Контейнер — це підсистема, що виконується і розгортається;
- діаграми компонентів (рівень 3): вони поділяють контейнери на взаємопов'язані компоненти та зв'язують компоненти з іншими контейнерами або іншими системами;
- діаграми коду (рівень 4): вони надають додаткові відомості про дизайн архітектурних елементів, які можна порівняти з кодом.

Було створено діаграми для перших трьох рівнів.

Діаграма контексту системи (додаток Б) зображає продавця, основну систему та торговельні майданчики з якими взаємодіє розроблене програмне забезпечення.

Торговельні майданчики є сторонніми сервісами тому вони були винесені за межі програмного забезпечення.

Діаграма контейнерів програмного забезпечення (додаток В). На цьому рівні видно, які контейнери взаємодіють з розробленим програмним забезпеченням. Веб-інтерфейс взаємодіє лише з серверною частиною, тоді як серверна частина має зв'язки з базою даних та сторонніми сервісами, такими як OLX та AutoRia, завдяки безкоштовному API цих систем.

Діаграма компонентів програмного забезпечення (додаток Д). На цьому ієрархічному рівні можна побачити, що вся бізнес-логіка програмного забезпечення знаходиться на серверній стороні програмного забезпечення. Тобто показано як авторизація та відновлення пароля комунікують з “security” компонентом який своєю чергою шифрує дані у три етапи. Перший етап це приведення до однакової довжини за допомогою алгоритму SHA-512 [11]. Другий етап це шифрування даних за допомогою argon2. Та останнім кроком своєю чергою став алгоритм AES-256 [12] за допомогою ключа який зберігався на AWS KMS – (Amazon Key Management Service). Також демонструється за яким принципом торговельні майданчики комунікують з розробленим програмним забезпеченням. Тобто щоб отримати будь-яку інформацію про продавця на тому чи іншому торговельному майданчику потрібна відповідна авторизація. При успішній авторизації access токени будуть зберігатися в localStorage. Але через те, що дія цього токена відносно мала (близько 20 хв) потрібно його постійно оновлювати. Тож було прийняти рішення зберігати токен для оновлення у базі даних оскільки він має значно більший термін дії (близько 1,5 року) відповідно будить досить довго не змінний.

2.2. Архітектура програмного забезпечення

На рисунку 2.1 зображена схема клієнт-серверної архітектури системи. На схемі демонструється як взаємодіє кожен з додатків та як фігурує бізнес-логіка у кожному з процесів.

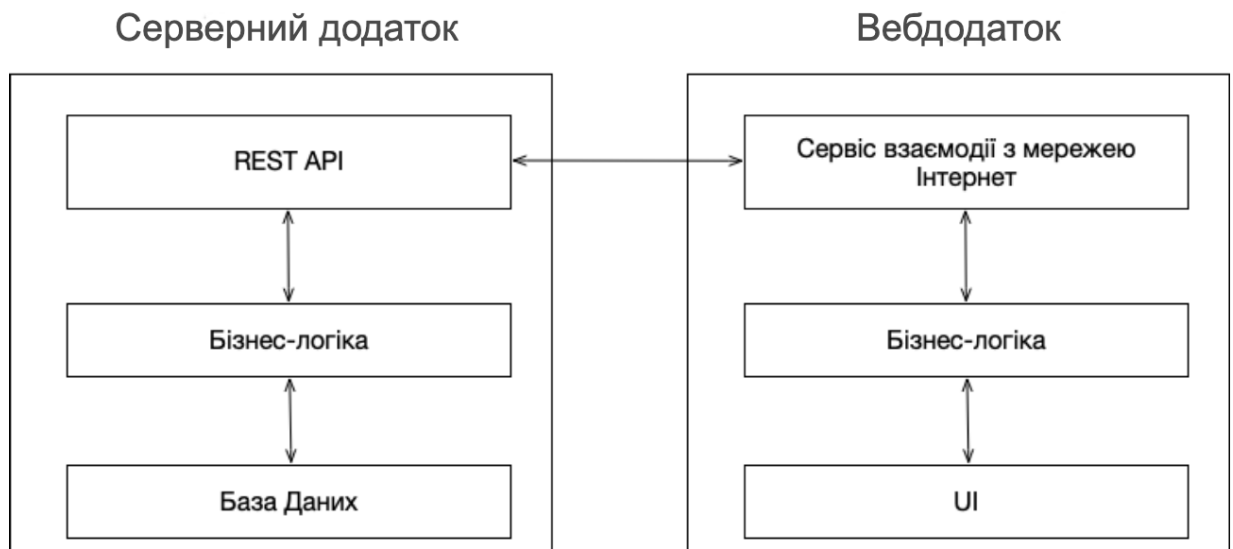


Рисунок 2.1 – Схема клієнт-серверної архітектури системи

Взаємодія між компонентами системи здійснюється за допомогою HTTP запитів до REST API, які підтримуються сервером. REST (Representational State Transfer) – це спосіб створення API за допомогою протоколу HTTP.

Технологію REST API застосовують скрізь, де користувачеві сайту або вебпрограмі потрібно надати дані з сервера. Наприклад, при початковому натисканні іконки з відео на youtube, REST API проводить операції та запускає ролик із сервера у браузері. Зараз це найпоширеніший спосіб організації API. Він витіснив раніше популярні методи SOAP і WSDL.

При реалізації буде виконано 7 основних принципів роботи з REST API:

- відділення клієнта від сервера (Client Server). Клієнт – це безпосередній інтерфейс вебсайту або ж програми. У REST API код запитів залишається на стороні покупця, а код доступу до інформації – на стороні серверної частини. Це спрощує керування API та дозволяє просто виносити інтерфейс користувача на іншу платформу і виділяє ймовірність чимось іншого масштабувати серверне заощадження даних;
- відсутність запису стану клієнта (Stateless). Сервер не повинен зберігати інформацію про стан (проведені операції) клієнта. Кожен запит від вебдодатку має містити тільки інформацію, яка була б потрібна для отримання даних від серверної частини;

- кешування (Cachable). У даних запиту має бути зазначено, чи потрібно кешувати дані або ж іншими словами зберігати у спеціальному буфері для частих запитів. Якщо така вказівка є, клієнт отримає право звертатися до цього буфера за потреби;
- єдність інтерфейсу (Uniform Interface). Усі дані мають проходити через 1 URL адресу стандартними за протоколами, наприклад, HTTP. Це полегшує архітектуру сайту або програми й робить взаємодію з сервером більш зрозумілою;
- багаторівневність системи (Layered System). У RESTful сервері можуть розташовуватися різних рівнях, у своїй кожен сервер взаємодіє лише з найближчими рівнями та пов'язаний запитами коїться з іншими;
- передача коду на запит (Code on Demand). Сервери можуть надсилати покупцеві код. Наприклад артільний код програми або вебсайт стає складніше тільки при необхідності;
- початок від нульової позначки (Starting with the Null Style). Замовник розуміє лише 1 точку входу на сервер. Наступні можливості взаємодії забезпечуються сервером. REST API базується на протоколі передачі гіпертексту HTTP (Hypertext Transfer Protocol). Це стандартний протокол в інтернеті, створений для передачі гіпертексту. Також за допомогою HTTP можна надсилати будь-які інші типи даних.

У розробленому REST API є 4 HTTP методи, за допомогою яких клієнт відправляє на сервер відповідний тип запитів:

- GET – отримання інформації про дані або список об'єктів;
- POST – додавання або зміна даних;
- PUT – оновлення даних;
- DELETE – видалення даних;

Такі запити ще можна назвати ідентифікаторами CRUD, які контролюють основну частину операцій між клієнтом та серверною частиною програмного забезпечення, приклад продемонстровано на рисунку 2.2.

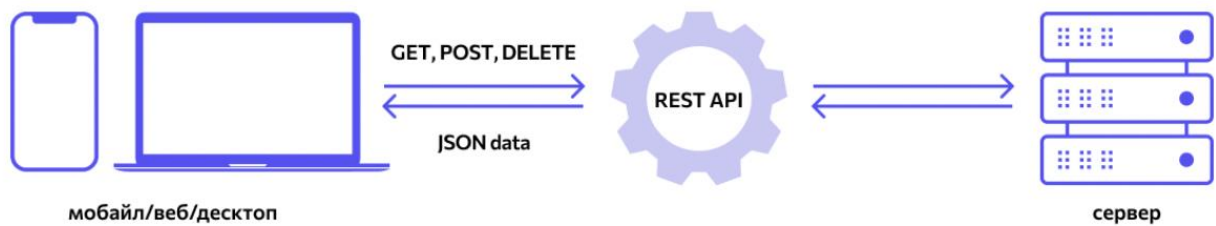


Рисунок 2.2 – Архітектура REST API

На даному рисунку можна чітко побачити як клієнт комунікує з сервером за допомогою REST API. Важливо зазначити що сама архітектура REST не прив'язана до конкретних технологій і протоколів, але в реаліях сучасного Інтернету, побудова RESTful API майже завжди має на увазі використання HTTP і будь-яких поширених форматів представлення ресурсів, наприклад JSON, або, менш популярного сьогодні, XML.

2.3. Конструювання програмного забезпечення

Для взаємодії з клієнтським застосунком використовують наступні інтерфейси:

- authentication – інтерфейс для процесів авторизації, реєстрації, відновлення та зміни пароля;
- statistics – інтерфейс для отримання статистики продуктів;
- products – інтерфейс для підключення торговельних майданчиків та виводу продуктів;
- account – інтерфейс для отримання даних профілю.

За допомогою інтерфейсу можна вказати, що саме повинен виконувати клас, що його реалізує, але не як це робити. Спосіб реалізації вибирає сам клас. Інтерфейси не здатні зберігати дані стану. Інтерфейси - це один із механізмів реалізації принципу поліморфізму "один інтерфейс, кілька методів"

Таблиця 2.1 – Методи інтерфейсу Authentication.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
registration	email, password	HTTP Code	Зареєструвати користувача у системі	001 002
login	email, password	access token	Увійти у застосунок із вказаними email та паролем	001
reset_password	email		Оновити пароль за вказаним email	003

Таблиця 2.2 – Коди помилок інтерфейсу Authentication.

Код помилки	Системна назва помилки	Значення помилки
001	INVALID_CREDENTIALS	Помилка в авторизації. Не правильний email або пароль.
002	EMAIL_ALREADY_IN_USE	Email вже використовується для іншого користувача.
003	INVALID_EMAIL	Користувача з таким email не існує.
004	NOT_AUTHENTICATED	Користувач не аутентифікований.

Таблиця 2.3 – Методи інтерфейсу Statistics.

Назва методу	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
GetActiveStatistics	productid (optional), olx_access_token та/або ria_access_token	Масив статистики	Отримати дані статистики з торговельних майданчиків	101 102 103
GetSoldStatistics	productid (optional)	Масив статистики	Отримати дані статистики проданих товарів.	101

Таблиця 2.4 – Коди помилок інтерфейсу Statistics.

Код помилки	Системна назва помилки	Значення помилки
101	PRODUCTS_NOT_EXIST	Повинен бути хоча б один продукт для виведення статистики.
102	RIA_TOKEN_NOT_EXIST	Невірний токен при намаганні стягування даних з торговельного майданчика.
103	OLX_TOKEN_NOT_EXIST	Невірний токен при намаганні стягування даних з торговельного майданчика.

Таблиця 2.5 – Методи інтерфейсу Products.

Назва методу	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
CreateSession	riaUserId, riaToken або olx_auth_code	HTTP Code	Авторизація з сервісами OLX та/або AutoRIA за допомогою систем API.	201 202
GetProducts	olx_access_token та/або ria_access_token	Масив усіх продуктів незалежно від торговельного майданчика	Отримання усіх товарів з попередньо підключених торговельних майданчиків.	201 202
StopProduct	olx_access_token та/або ria_access_token, productId	HTTP Code	Деактивувати товар на тому чи іншому торговельному майданчику	203
SoldProduct	olx_access_token та/або ria_access_token, productid, count	HTTP Code	Додати продукт у список проданих товарів. Ця діє не може бути повернена до попереднього значення.	203 204

Таблиця 2.6 – Коди помилок інтерфейсу Authentication

Код помилки	Системна назва помилки	Значення помилки
201	INVALID_OLX_TOKEN	Помилка в авторизації до торговельного майданчика OLX. Не правильний access_token.
202	INVALID_RIA_TOKEN	Помилка в авторизації до торговельного майданчика AutoRia. Не правильний access_token.
203	INVALID_PRODUCT	Продукту з таким ідентифікатором не існує
204	OUT_OF_STOCK	Даного продукту не має в наявності.

Таблиця 2.7 – Методи інтерфейсу Account.

Назва метода	Аргументи	Значення, яке повертається	Призначення методу	Коди помилок
GetProfile	token	Об'єкт даного юзера	Переглянути дані профілю поточного користувача	301
UpdateProfile	token	HTTP Code	Оновити дані профілю такі як фото або пароль	301 302 303

Таблиця 2.8 – Коди помилок інтерфейсу Account.

Код помилки	Системна назва помилки	Значення помилки
301	USER_NOT_EXIST	Користувача не знайдено не вірний токен
302	IMG_WRONG_FORMAT	Не вірний формат картинки
303	IMG_SIZE_LIMIT	Ліміт розміру картинки перевищено

На рисунку 2.3, планується створення бази даних, яка забезпечить зберігання важливої інформації для користувачів системи, включаючи їхні дані реєстрації на торговельних майданчиках. Ця база даних також буде фіксувати дані про продаж товарів для подальшого оновлення пропозицій.

Структура бази даних може включати таблиці, що зберігають інформацію про користувачів, їх аутентифікаційні дані, а також інформацію про реєстрацію на різних торговельних майданчиках. Крім того, можуть бути таблиці, які відображають дані про продажі, включаючи деталі про товари, кількість проданих одиниць, ціни, дати тощо. Зв'язки між таблицями можуть бути налаштовані для забезпечення цілісності даних та взаємозв'язків між різними частинами інформації.

Проектування бази даних важливе для забезпечення ефективного зберігання та управління інформацією про користувачів, реєстрацію та дані про продажі, що дозволить системі працювати з даними ефективно та забезпечити потрібну функціональність для взаємодії з різними компонентами програмного забезпечення.

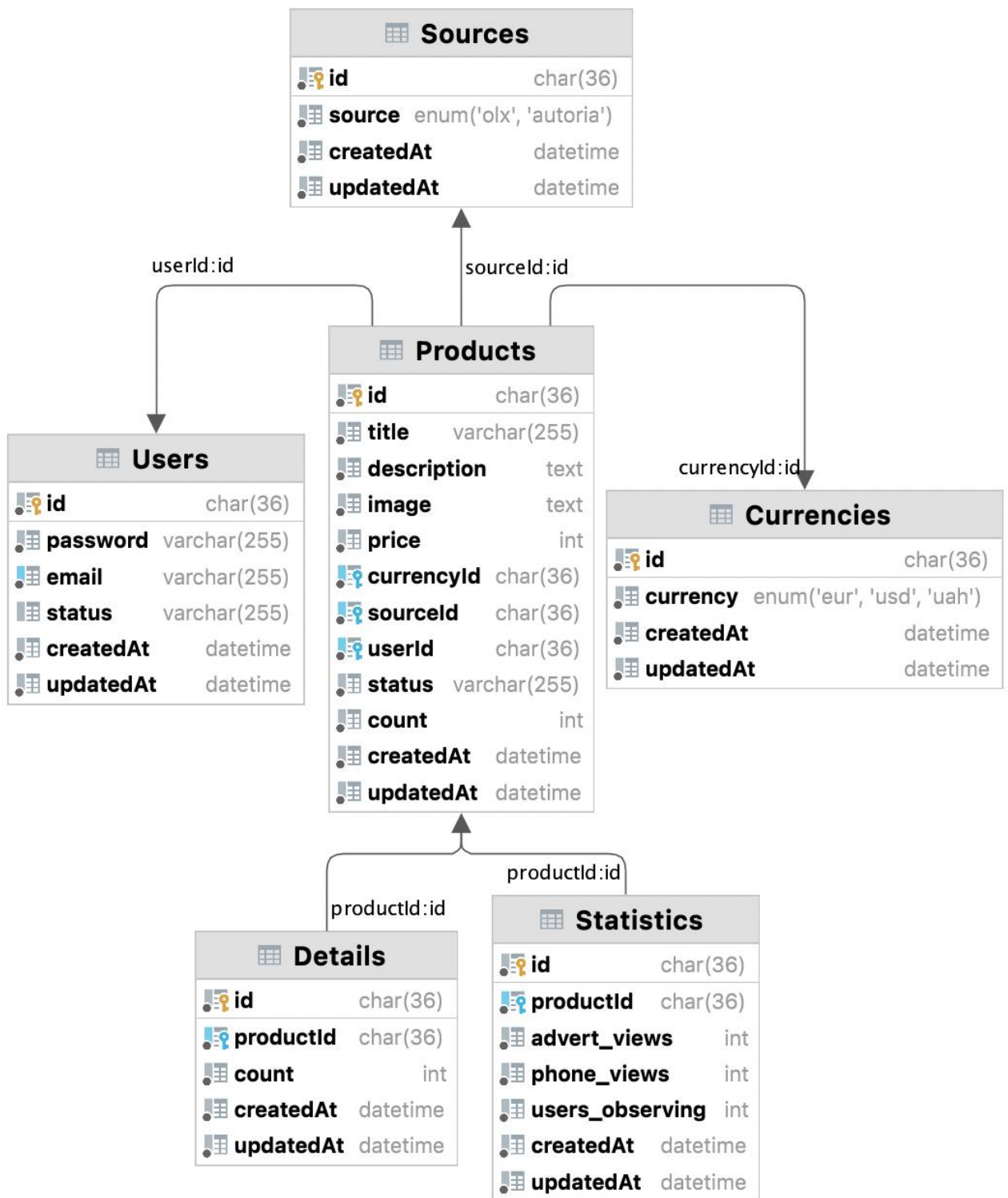


Рисунок 2.3 – Схеми бази даних

2.4. Аналіз безпеки даних

Захист основної інформації, такої як грошові потоки та особисті дані користувачів, є надзвичайно важливим завданням у процесі розробки

програмного забезпечення. Уникнення порушень безпеки має стратегічне значення для будь-якої компанії.

Зламане програмне забезпечення може не лише негативно вплинути на репутацію компанії, але й призвести до втрати даних, яка може бути серйозним фінансовим ударом для бізнесу. Розголос подій інцидентів такого роду може змусити втрачати довіру клієнтів, що може відбутися навіть у випадках, коли компанія здатна відновитися після кібератаки.

Особливо для компаній зі статусом та репутацією, збитки від можливих кібератак можуть бути надзвичайно значними. Інтернет-ресурси та вебсайти цих компаній часто виступають як обличчя бренду, тож будь-які вразливості чи проблеми з безпекою можуть серйозно підірвати довіру клієнтів та споживачів.

Отже, забезпечення безпеки даних, їх шифрування та захист від несанкціонованого доступу мають першочергове значення в розробці програмного забезпечення, особливо для компаній, які ведуть бізнес онлайн та стикаються зі значними обсягами особистих даних користувачів та грошових транзакцій.

Потенційні загрози:

- злом інформаційного середовища (операційна система, вебсервер, середовище програмування, база даних). Це включає в себе можливість злому операційної системи, вебсервера, середовища програмування або бази даних через експлойти, вразливості або несанкціонований доступ;
- злом системи керування корпоративним порталом. Це загроза для централізованої системи керування, яка може бути в складних мережевих середовищах, внаслідок чого можуть зламати важливі функції, отримати несанкціонований доступ до корпоративної інформації або змінювати налаштування системи;
- зламування сторонніх вебдодатків. Ця загроза стосується сторонніх додатків, що використовуються разом з основним програмним забезпеченням. Якщо вони не мають належних заходів безпеки, це може викликати потенційні вразливості для вашої системи;

- атаки типу Denial-of-Service (DoS) або Distributed Denial-of-Service (DDoS). Це атаки, які мають на меті перевантажити серверні ресурси або мережу, щоб заборонити доступ користувачів до ваших служб;
- фішинг. Це соціальний вид атаки, при якому атакувач намагається здобути конфіденційну інформацію, представляючись як достовірний джерело;
- злам індивідуальних акаунтів користувачів. Атакувач може намагатися отримати доступ до облікових записів користувачів для отримання конфіденційної інформації або злому системи через несанкціонований доступ;
- злом мережі. Можливість несанкціонованого доступу до мережі, що може призвести до зміни налаштувань мережі або отримання конфіденційних даних.

Тобто серед переліченого до розробленого програмного забезпечення можна віднести лише перший пункт, оскільки корпоративного порталу не існує а, що до сторонніх веб додатків, до яких якоюсь мірою можна віднести підключені торговельні майданчики, розроблене програмне забезпечення відповідальність за цілісність їх даних не несе.

Ступінь загроз можна розділити на три рівні ризику:

- мінімальний – отримання доступу до не конфіденційної інформації, до якої не санкціоновано доступ, можливість створення косметичних проблем та перешкод у роботі проєкту;
- середній рівень – отримання часткового доступу до конфіденційної інформації, частковий обхід системи авторизації розширює повноваження;
- високий рівень – повний обхід системи авторизації, отримання необмеженого доступу до системи або додатку, можливість запуску несанкціонованих додатків, можливість перегляду або заміни конфіденційної інформації.

Розроблене програмне забезпечення відрізняється від використання рівнів, оскільки в ньому роути чітко розділені на авторизованих та не авторизованих користувачів, не дозволяючи доступ до неконфіденційної інформації.

Авторизація відбувається через JSON web token (JWT), ускладнена трьома рівнями шифрування паролів: SHA 512, Argon2, AES 256. Це надійний захист, що унеможлиблює атаки "brute force" або "dictionary".

Для безпечної передачі даних на сервер програмне забезпечення використовує захищене HTTPS з'єднання, де зворотній проксі-сервер nginx виступає посередником. Протокол HTTPS - розширення HTTP з підтримкою шифрування, що забезпечує захищені транспортні механізми через SSL і TLS. Використання цих технологій дозволяє забезпечити високий рівень конфіденційності та безпеки під час передачі даних між користувачем і сервером, унеможливлюючи несанкціонований доступ та перехоплення інформації.

2.5. Висновки до розділу

В цьому розділі ретельно розглянута модель програмного забезпечення, використовуючи модель C4 для подробиного моделювання архітектури системи. Розглянуті три типи діаграм: діаграма контексту, діаграма контейнерів та діаграма компонентів, що спрощують уявлення про структуру програмної системи на різних рівнях.

Крім цього, докладно проаналізовано структуру бази даних. Описано архітектуру клієнтської та серверної частин програмного забезпечення, визначено їхні функції та способи взаємодії. Також вибраний та детально описаний механізм для безпечного обміну даними між клієнтом та сервером, що є критичним для забезпечення безпеки та конфіденційності під час передачі інформації.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1. Аналіз якості ПЗ

Процес тестування є критичним етапом в життєвому циклі розробки програмного забезпечення. Він сприяє підтвердженню працездатності програмного продукту та перевірці його відповідності вимогам. У контексті розробки даного програмного забезпечення, тестування є важливим у зв'язку зі специфікою бізнес-логіки, що передбачає взаємодію з різними торговельними майданчиками.

Тестування програмного забезпечення - це процес, який включає перевірку дотримання функціональних та нефункціональних вимог. Це може бути як ручне, так і автоматизоване виконання певних алгоритмів, спрямованих на оцінку різних властивостей системи.

Основна мета тестування - виявлення помилок, пропусків чи невідповідностей до вимог, що були визначені на етапі проектування програмного продукту.

На сьогоднішній день тестування є важливою складовою розробки програмного забезпечення, займаючи від 50% до 65% від загальних витрат на проєкт.

На ринку існує кілька підходів до випробування програмного забезпечення, але успішність тестування залежить від специфіки самого продукту, його цільової аудиторії та функціональності.

Процес випробування програмного забезпечення включає:

- формування списку компонентів для тестування;
- опис проходження тестів та визначення тестових випадків;
- створення звіту про завершення випробування системи та звіту про виявлені проблеми.

Успішне тестування вимагає систематичного підходу та уважної уваги до деталей, забезпечуючи надійність та безпеку програмного продукту.

3.2. Опис процесів тестування

3.2.1. Компоненти що тестуються

Зробимо перелік компонентів:

- авторизація користувача;
- реєстрація користувача;
- під'єднання до API торговельних майданчиків;
- перегляд всіх товарів;
- вивід статистики за товарами;
- призупинення оголошення;
- додавання товару у список проданих.

3.2.2. Підхід до тестування

Підхід до тестування складається з наступних рівнів тестування:

- модульне тестування. Цей рівень тестування виконується розробниками під час написання коду. Він орієнтований на перевірку окремих частин коду (модулів), використовуючи сторонній інструмент AVA [13]. Модульні тести перевіряють правильність роботи окремих функцій, методів або об'єктів. Вони служать основою для перевірки працездатності окремих частин програми;
- інтеграційне тестування. На цьому рівні перевіряється взаємодія між компонентами програми, які вже пройшли модульне тестування. Тобто після тестування окремих частин програми їх об'єднують та перевіряють взаємодію за певним планом тестування. Використовується також сторонній інструмент AVA для реалізації цих тестів;
- UI тестування. Цей вид тестування належить до важливих складових етапів розробки програмного продукту. Цей рівень тестування відтворює поведінку реального користувача, виконуючи дії, аналогічні тим, які виконує людина під час взаємодії з програмою. В процесі UI тестування перевіряється не лише коректність роботи

інтерфейсу, а й взаємодія між всіма компонентами програми. Даний вид тестування дозволяє оцінити зручність інтерфейсу для користувача, а також виявити можливі проблеми, які можуть виникнути під час його використання. Ретельне UI тестування забезпечує впевненість у коректності роботи програми та забезпечує позитивне враження користувача від взаємодії з продуктом. Важливим аспектом UI тестування є врахування різноманітних сценаріїв взаємодії користувача з програмним інтерфейсом. Це включає не лише перевірку основних функцій програми, а й аналіз відповідності дизайну та зручності користувацькому досвіду.

Кожен з цих рівнів відповідає за певний аспект програми та спрямований на виявлення можливих помилок чи невідповідностей до вимог. Сукупність цих тестів забезпечує надійність та функціональність програмного забезпечення.

3.2.3. Критерій проходження тесту

Тести вважаються успішно пройденими, якщо не виявлено жодних помилок або вони не були знайдені на будь-якому з рівнів тестування. Крім того, вони повинні відповідати всім вимогам, визначеним у технічному завданні. Успішне проходження тестів є важливим етапом, оскільки це гарантує, що програмне забезпечення відповідає зазначеним критеріям якості та функціональності.

3.3. Опис контрольного прикладу

Завдання контрольного прикладу - перевірити правильність виконання сценарію додавання продукту до списку, який користувачі вважають проданим. В таблиці 3.1 наведено послідовний опис тестування, включаючи дії користувача, очікувані та отримані результати.

Цей тип тесту допомагає підтвердити правильність реакції системи на певні дії користувача, дозволяючи виявити будь-які неполадки чи невідповідності у функціонуванні програмного продукту. Такі випробування

важливі для перевірки функціональності програми в реальних умовах та впевненості у коректності її роботи для кінцевого користувача.

Таблиця 3.1 – опис контрольного прикладу

Дія продавця	Очікуваний результат	Реальний результат
Продавець підключає будь-які торговельні майданчики.	На екрані демонтується повідомлення про успішне підключення.	Продемонстровано повідомлення про успішне підключення.
Продавець заходить на сторінку з продуктами.	На екрані демонтується список усіх продуктів, які належать користувачу.	На екрані продемонстровано список усіх продуктів, які належать користувачу.
Продавець обирає продукт зі списку продуктів та натискає на кнопку «Продано».	На екрані демонтується модальне вікно у якому користувачеві пропонується вибрати кількість проданих товарів.	На екрані продемонстровано модальне вікно у якому користувачеві пропонується вибрати кількість проданих товарів.
Продавець обирає кількість проданих товарів та натискає на кнопку «Зберегти».	На екрані демонтується успішне повідомлення про збереження наданої інформації	На екрані продемонстровано успішне повідомлення про збереження даних
Продавець натискає на кнопку «Статистика».	На екрані демонтується оновлена статистика.	На екрані продемонстрована оновлена статистика.

3.4. Висновки до розділу

В цьому розділі було детально розглянуто тест якості програмного забезпечення, визначені ключові аспекти, що піддаються тестуванню. Зокрема, було детально описано підхід до проведення тестування, включаючи різноманітні види тестів, а також зазначено основні критерії успішності тестування. У додаток до цього, розглянуто контрольний приклад тестування, у якому зазначено послідовність дій продавця, а також описано очікувані та отримані результати в процесі виконання тестування.

Звідомлення про тестування якості програмного забезпечення виявляється важливим етапом у розробці програм, оскільки воно сприяє виявленню помилок, неузгодженостей та відхилень від вимог ще на ранніх етапах розробки. Цей процес допомагає покращити надійність, функціональність та загальну якість продукту перед його випуском на ринок, забезпечуючи високий рівень задоволеності користувачів. Основною метою тестування є не лише виявлення помилок, але й підтвердження відповідності програми вимогам та очікуванням користувачів.

Нехай цей огляд відображає ключові аспекти, які допомагають у вдосконаленні якості програмного продукту та сприяють його успішному впровадженню.

4 ВПРОВАДЖЕННЯ ТА СУПРОВІД ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1. Розгортання програмного забезпечення

Для розгортання програмного забезпечення необхідним аспектом є попереднє встановлення пакетного менеджера NPM та наявність Node.js 16 версії. У випадку з Linux вони можуть бути встановлені за допомогою NVM [14].

Програмне забезпечення було розгорнуто за допомогою хмарного сервісу Amazon:

- Amazon EC2. Сервіс який надає віртуальну серверну платформу, системи зберігання даних та балансувальник навантаження. Було вибрано заздалегідь налаштований сервер з попередньо встановленою операційною системою Linux. Сервіс також надає можливість створювати образи або використовувати власну ОС. Також для забезпечення безпеки було обмежено доступ до серверів EC2 за IP-адресами;
- Amazon RDS. Сервіс який надає віртуальну базу даних, розташовану на виділеному сервері. При цьому платформа налаштована та оптимізована під обрану БД, а саме MySQL. Був вибраний мінімальний розмір дискового простору під базу даних – 5 Тб.

Для безперервної підтримки програми в робочому стані та для змоги перезавантажувати її без простоїв, а також забезпечувати виконання загальних завдань системного адміністрування був вибраний диспетчер процесів робочого середовища PM2 [15]. Також перевагою PM2 є якісний вивід логів та помилок якщо такі наявні.

Таким чином для першого запуску програмного забезпечення на сервері знадобиться:

- 1) розгорнути EC2 та RDS;
- 2) створити зв'язок з EC2 за допомогою протоколу SSH;
- 3) встановити Node.js та NPM;

- 4) зклонувати код з GitHub та внести у змінні оточення RDS налаштування;
- 5) запустити програмне забезпечення за допомогою PM2.

Для оновлення програмного забезпечення на сервері знадобиться:

- 1) створити зв'язок з EC2 за допомогою протоколу SSH;
- 2) зтягнути нові дані з GitHub;
- 3) оновити за допомогою PM2.

4.2. Робота з програмним забезпеченням

При запуску веб додатка пропонується авторизуватися або зареєструватися в додатку, як показано на рисунку 4.1.

Вхід

ДАЛІ

Рисунок 4.1 – Авторизація у вебдодатку

Для того, щоб приєднати торговельні майданчики необхідно перейти на вкладку «Trading Platforms» та вибрати для підключення бажаний сервіс, як показано на рисунку 4.2.

Виберіть Торговельний Майданчик



АВТОРИЗУВАТИСЯ ЧЕРЕЗ OLX



АВТОРИЗУВАТИСЯ ЧЕРЕЗ AUTORIA

Рисунок 4.2 – Сторінка для підключення торговельних майданчиків

При підключенні сервісу OLX потрібно надати дозвіл, як показано на
рисунок 4.3.

Додаток Analytic Subsystem

Дозволити доступ до вашого аккаунту?

alexfilatov380 (alexfilatov380@gmail.com)

Додатку будуть доступні:

читання даних профілю, оголошень, повідомлень
редагування даних профілю, оголошень,
повідомлень

Ви можете змінити налаштування доступу в вашому
Профілі.

Вибравши "Дозволити", ви надаєте програмі Analytic
Subsystem доступ до свого облікового запису OLX і
дозволяєте їй діяти від вашого імені відповідно до Угоди
користувача сервісів OLX та умов використання додатку.

ДОЗВОЛИТИ

ВІДМІНА

Рисунок 4.3 – Авторизація через сервіс OLX

При підключенні сервісу AutoRia необхідно ввести свої унікальні секрети, які можна знайти у своєму обліковому записі, та ввести їх у форму (рис. 4.4).

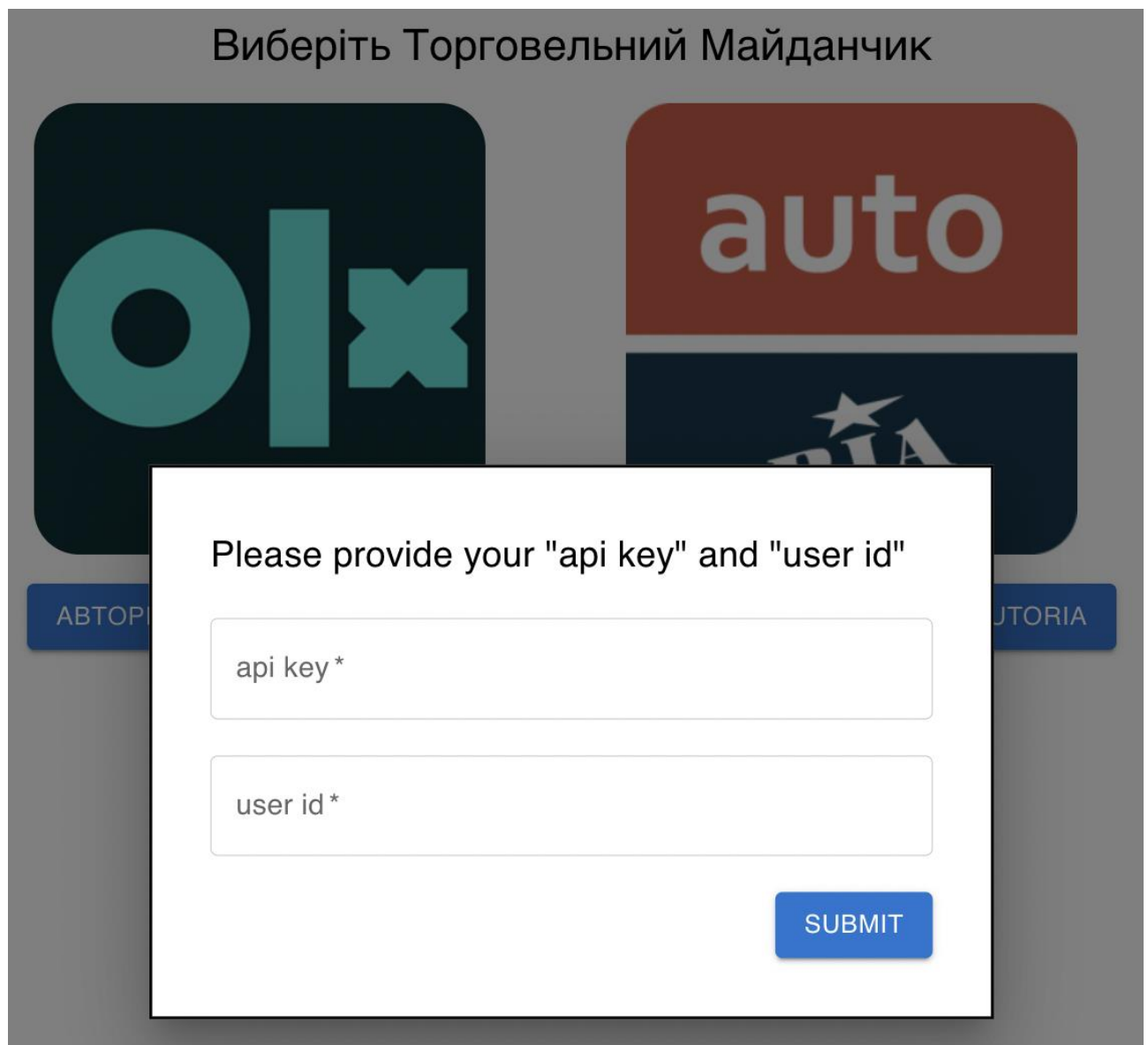
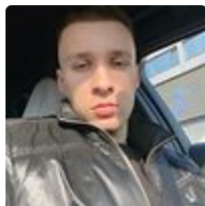


Рисунок 4.4 – Авторизація через сервіс AutoRia

На рисунку 4.5 зображена сторінка профілю AutoRia де можна знайти секрети для доступу до товарів.

Мой профиль



Email	alexfilatov380@gmail.com
Ваш ID (user_id)	11267851
Api Key	9T6v4q... Показати
Зареєстрований	2022-05-18 11:15:01

[Редагувати](#)[Змінити пароль](#)

Рисунок 4.5 – Сторінка з ключами доступу до AutoRia

Після підключення торговельних майданчиків користувач зможе побачити та відфільтрувати всі свої продукти на головній сторінці вебдодатка (рис. 4.6).



Iphone 11 256gb black

Торговельний Майданчик: OLX
 Ціна: 1000 USD
 Продано: 12 шт.
 Статус: active
 Дата Створення: 05/14/2022
 Термін Придатності: 06/13/2022

Статистика: 10 0 0

[ПЕРЕЙТИ ДО РЕСУРСУ](#)



Honda CR-V 2007

Торговельний Майданчик: AutoRIA
 Ціна: 10000 USD
 Продано: 1 шт.
 Статус: active
 Дата Створення: 05/18/2022
 Термін Придатності: 06/18/2022

Статистика: 353 0 2

[ПЕРЕЙТИ ДО РЕСУРСУ](#)

Рисунок 4.6 – Демонстрація карток товарів

На сторінці статистики користувачу буде продемонстрований рейтинг товарів (рис. 4.7) та загальний прибуток (рис. 4.8).

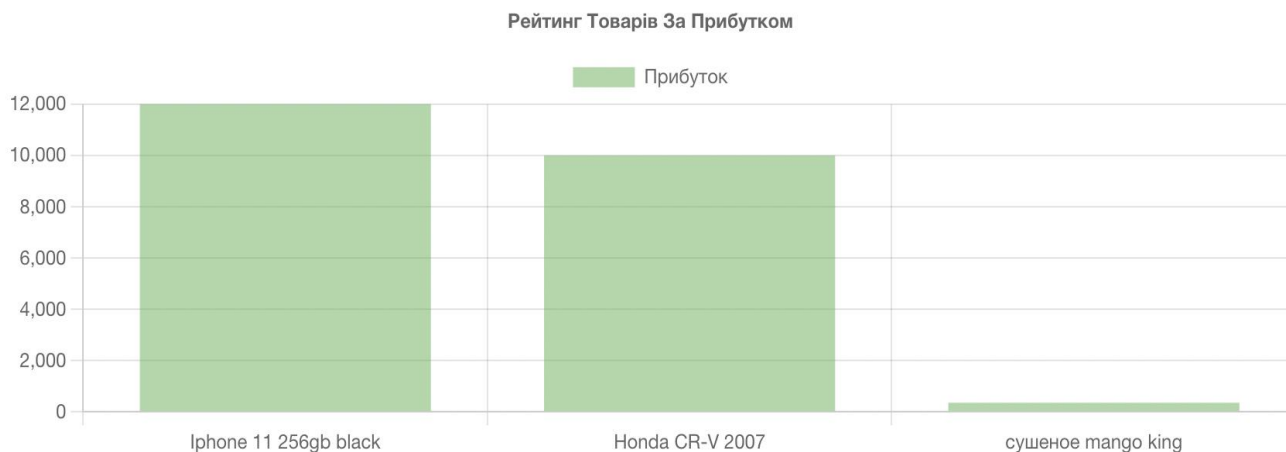


Рисунок 4.7 – Рейтинг товарів

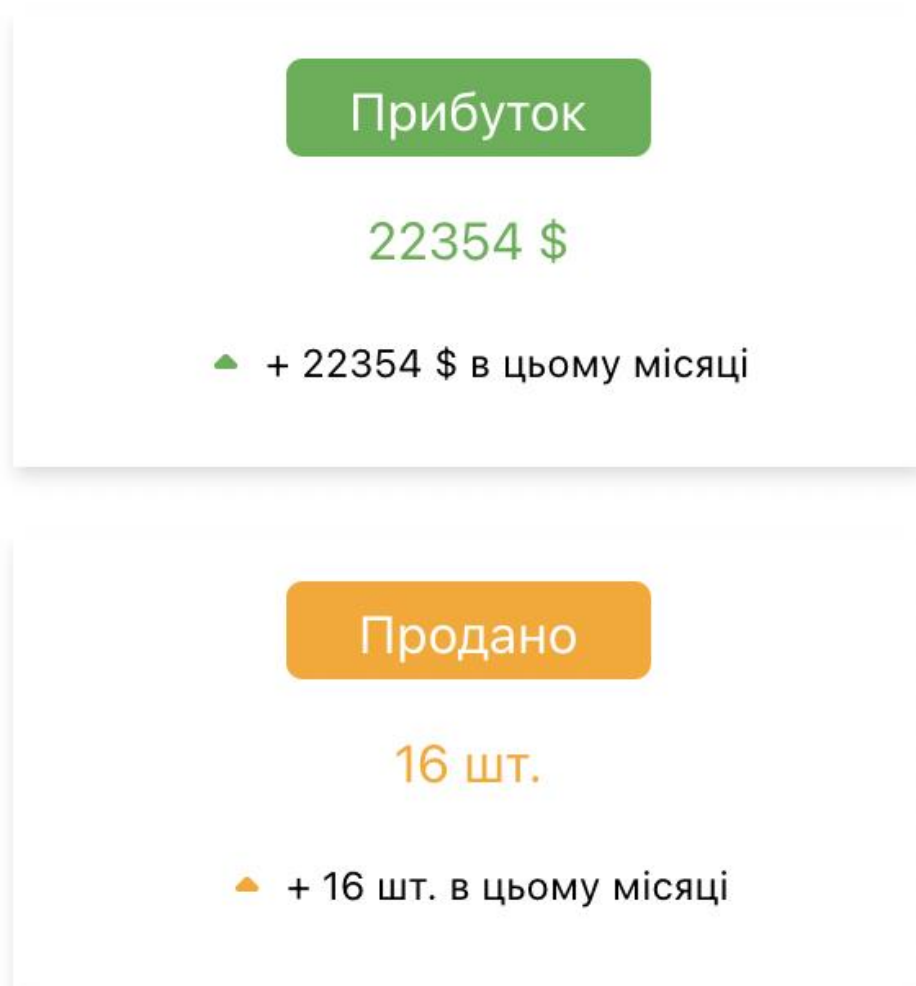


Рисунок 4.8 – Рейтинг товарів

Також користувач зможе скористатися на сторінці статистики лінійний графік по кожному з товарів в якому він зацікавлений попередньо вказавши діапазон (тиждень, місяць або рік), як показано на рисунку 4.9.

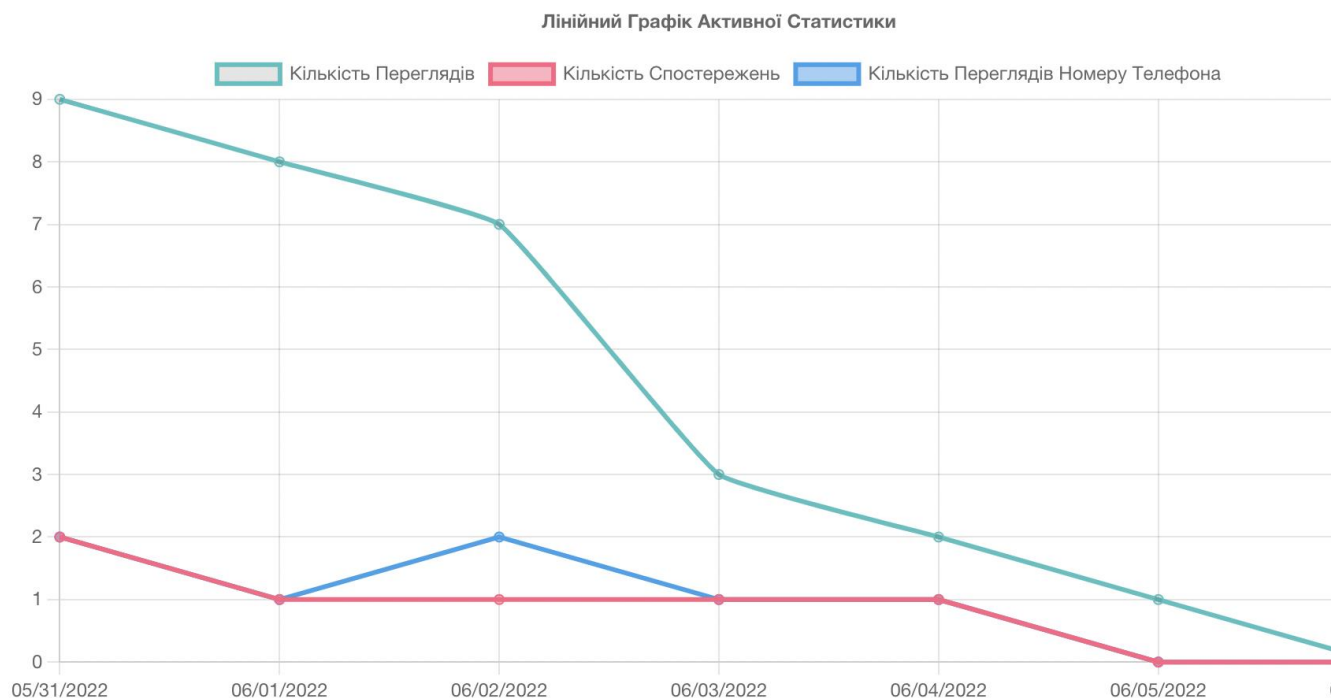


Рисунок 4.9 – Лінійний графік активності товару

Користувач має можливість переглядати активну статистику за трьома пунктами це кількість переглядів (рис. 4.10), кількість спостережень та кількість переглядів номеру телефона.

Кількість Переглядів

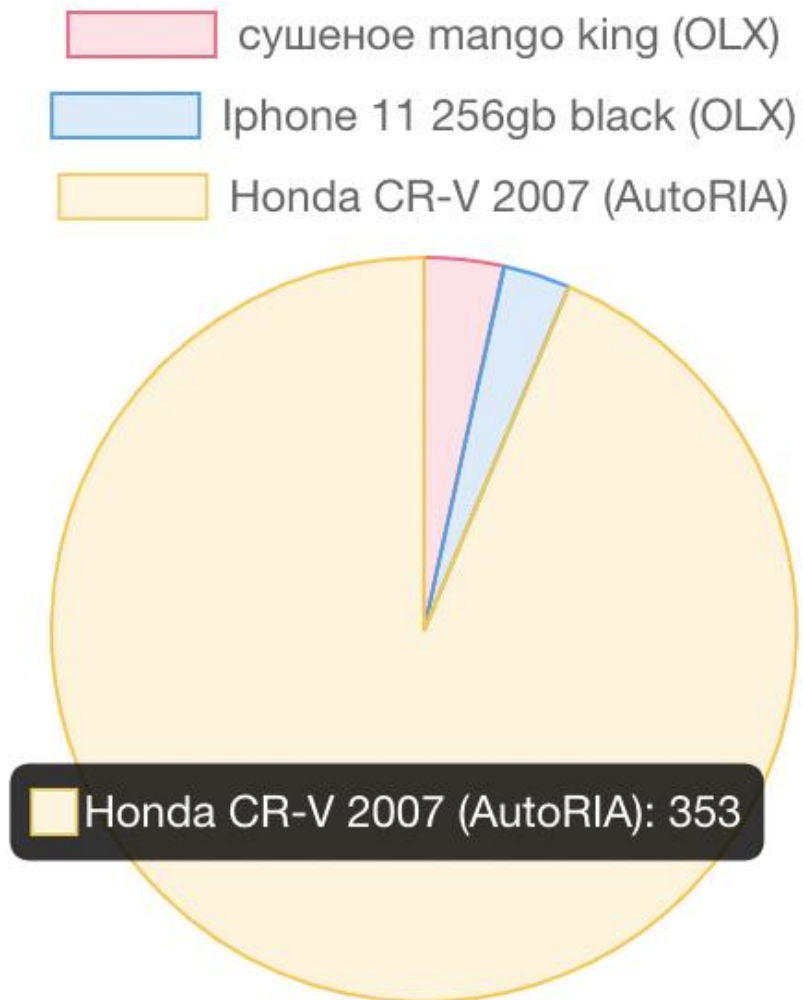


Рисунок 4.10 – Діаграма за кількістю переглядів

Щоб переглянути власний профіль треба натиснути на лівій боковій панелі кнопку «Налаштування» (є можливість зміни мови), рисунок 4.11.

Мова

Українська 🇺🇦 ▼



Персональна Інформація

email

alexfilatov380@gmail.com

password

ВИЙТИ

Рисунок 4.11 – Профіль користувача

4.3. Висновки до розділу

Програмне забезпечення було розгорнуте на хмарному сервісі Amazon. Працездатність програмного забезпечення було проведена шляхом виконання контрольного тесту. Програмне забезпечення повністю готове для впровадження.

ВИСНОВКИ

Задача розробки програмного забезпечення, що відповідає потребам продавців у розміщенні та аналізі своїх оголошень на торговельних площадках, виявилася ключовою. Новостворена програма не лише спрощує роботу продавців, а й об'єднує дані з різних платформ, що полегшує аналітику та управління їхнім бізнесом. Аналіз аналогів дозволив виділити проблеми наявних програмних продуктів, тим самим виправити їх у власній розробці.

Основною складовою успішності розробленого програмного забезпечення стало визначення оптимальних технологій для його створення: Node.js, React.js та MySQL. Ці технології були обрані з урахуванням їх ефективності та сумісності для досягнення мети проекту. Використання моделі С4 для моделювання програмного забезпечення дозволило зрозуміло та систематично представити його архітектуру та інтерфейси.

Тестування виявило ряд недоліків, які були виправлені, підвищивши загальну якість програмного продукту. Вибір хмарного сервісу Amazon для розгортання дозволив забезпечити ефективну та надійну інфраструктуру для програми. Контрольний тест підтвердив працездатність програми та її готовність до впровадження на ринку.

Цей шлях розробки програмного забезпечення пройдений з урахуванням важливих етапів та кроків, що забезпечують високу якість, функціональність та готовність продукту до впровадження в реальні умови експлуатації.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. OLX API [електронний ресурс] URL: <https://developer.olx.ua/api/doc/>
2. AutoRia API [електронний ресурс] URL: <https://developers.ria.com/docs/>
3. Gerardus Blokdyk. JSON Web Token: Third Edition / CreateSpace Independent Publishing Platform, 2018. 134 с.
4. Carlos Santana Roldan. React 17 Design Patterns and Best Practices / Birmingham: Packt Publishing, 2021. 394 с.
5. Mario Casciaro, Luciano Mammino. Node.js Design Patterns: Design and implement production-grade Node.js applications using proven patterns and techniques / Birmingham: Packt Publishing, 2020. 660 с.
6. Zachary Shute. Advanced JavaScript: Speed up web development with the powerful features and benefits of JavaScript / Birmingham: Packt Publishing, 2019. 332 с.
7. Express.js [електронний ресурс] URL: <https://expressjs.com/>
8. Silvia Botros, Jeremy Tinley. High Performance MySQL: Proven Strategies for Operating at Scale / Newton: O'Reilly Media, 2021. 388 с.
9. Amazon [електронний ресурс] URL: <https://docs.aws.amazon.com/>
10. The C4 model for visualising software architecture [електронний ресурс] URL: <https://c4model.com/>
11. SHA-512 Hashing Algorithm [електронний ресурс] URL: <https://komodoplatform.com/en/academy/sha-512/>
12. Advanced Encryption Standard (AES) [електронний ресурс] URL: <https://www.techtarget.com/searchsecurity/definition/Advanced-Encryption-Standard/>
13. AVA [електронний ресурс] URL: <https://github.com/avajs/ava/>
14. NVM [електронний ресурс] URL: <https://ionicframework.com/docs/v3/developer-resources/using-nvm/>
15. PM2 [електронний ресурс] URL: <https://pm2.keymetrics.io/>

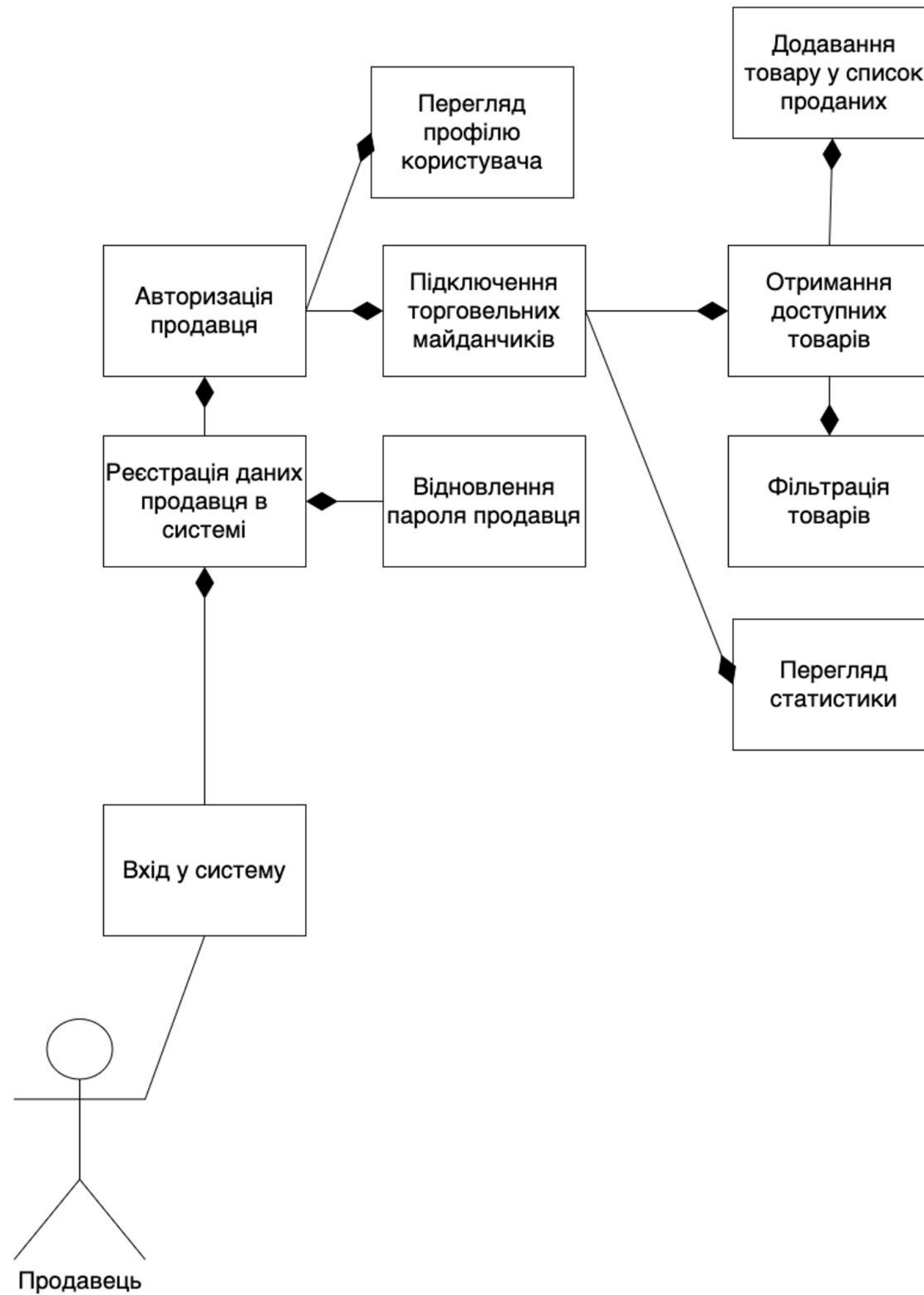
ДОДАТОК А

Програмне забезпечення для підтримки та аналітики для продавців на
торговельних майданчиках

Схема структурна варіантів використання

Аркушів 1

Київ – 2024



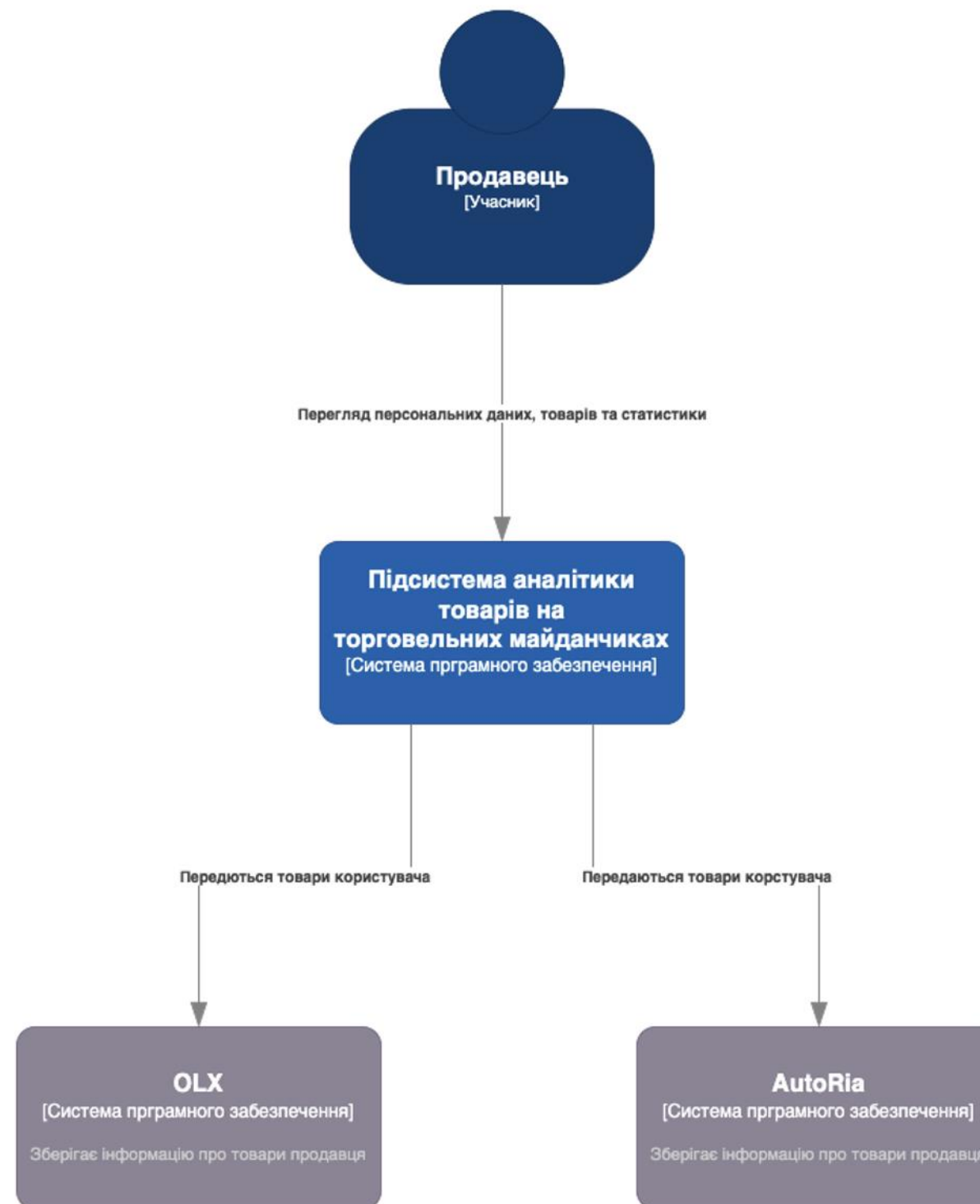
ДОДАТОК Б

**Програмне забезпечення для підтримки та аналітики для продавців на
торговельних майданчиках**

**Схема структурна
контексту програмного
забезпечення**

Аркушів 1

Київ – 2024



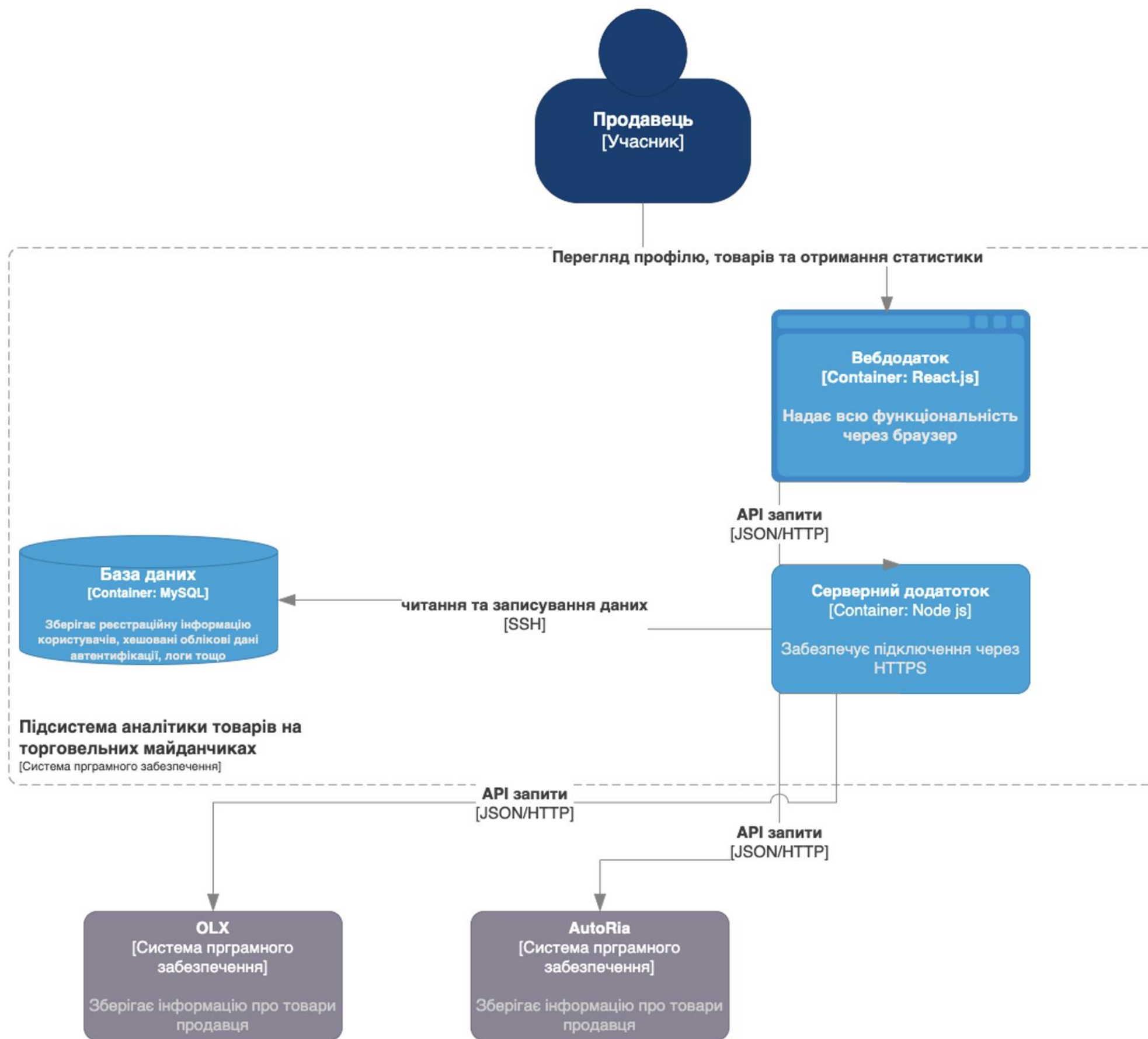
ДОДАТОК В

Програмне забезпечення для підтримки та аналітики для продавців на
торговельних майданчиках

Схема структурна
контейнерів програмного
забезпечення

Аркушів 1

Київ – 2024



ДОДАТОК Д

Програмне забезпечення для підтримки та аналітики для продавців на
торговельних майданчиках

Схема структурна
компонентів програмного
забезпечення

Аркушів 1

Київ – 2024

