

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ імені ІГОРЯ СІКОРСЬКОГО»
Факультет інформатики та обчислювальної техніки
(повна назва інституту/факультету)

Кафедра інформатики та програмної інженерії
(повна назва кафедри)

«До захисту допущено»

Завідувач кафедри

_____ Едуард ЖАРІКОВ
(підпис) (ім'я прізвище)

“ ” _____ 2023 р.

Дипломний проєкт

на здобуття ступеня бакалавра

за освітньо-професійною програмою «Інженерія програмного забезпечення
комп'ютерних систем»

спеціальності «121 Інженерія програмного забезпечення»

на тему: Веб-застосунок для продажу права власності на віртуальні об'єкти з
використанням свідоцтва створеного на основі технології блокчейн.
Веб-застосунок підтримки процесів продажу віртуальних об'єктів.

Виконав студент IV курсу, групи ІТ-83
(шифр групи)

Василець Данило Богданович

(прізвище, ім'я, по батькові)

(підпис)

Керівник ст.в. Головченко М.М.

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

(підпис)

Консультант
з графічної
документації

ст.в. Головченко М.М.

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

(підпис)

Рецензент к.т.н., доц., доц. каф. ОТ, Волокита А.М.

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

(підпис)

Засвідчую, що у цій дипломній роботі
немає запозичень з праць інших
авторів без відповідних посилань.

Студент _____
(підпис)

Київ –2023

Національний технічний університет України
«Київський політехнічний інститут імені Ігоря Сікорського»

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

Рівень вищої освіти – перший (бакалаврський)

Спеціальність – 121 Інженерія програмного забезпечення

Освітньо-професійна програма – Інженерія програмного забезпечення
комп'ютерних систем

ЗАТВЕРДЖУЮ

Завідувач кафедри

(підпис) Едуард ЖАРІКОВ
(ім'я прізвище)

“ ____ ” _____ 2023 р.

ЗАВДАННЯ
на дипломний проєкт студенту

Васильцю Данилу Богдановичу
(прізвище, ім'я, по батькові)

1. Тема проєкту Веб-застосунок підтримки процесів продажу віртуальних об'єктів.

керівник проєкту Головченко Максим Миколайович, ст.в.
(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом по університету від «02» квітня 2023 р. №2160-С

2. Термін подання студентом проєкту « 19 » червня 2023 року

3. Вихідні дані до проєкту: технічне завдання

4. Зміст пояснювальної записки

1) Аналіз вимог до програмного забезпечення: аналіз існуючих технологій та успішних ІТ-проєктів, аналіз вимог до програмного забезпечення, постановка задачі.

2) Конструювання програмного забезпечення: моделювання та аналіз програмного забезпечення, архітектура програмного забезпечення, конструювання програмного забезпечення, аналіз безпеки даних.

3) Аналіз якості та тестування програмного забезпечення: аналіз якості ПЗ, опис процесів тестування.

5. Перелік графічного матеріалу

1) Схема структурна варіантів використання _____

2) Креслення вигляду екранних форм _____

3) Плакат структури бізнес-процесів _____

6. Консультанти розділів проекту

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

7. Дата видачі завдання «10» березня 2023 року _____

Календарний план

№ з/п	Назва етапів виконання дипломного проєкту	Термін виконання етапів проєкту	Примітка
1	Вивчення рекомендованої літератури	19.03.2022	
2	Аналіз існуючих методів розв'язання задачі	01.04.2022	
3	Постановка та формалізація задачі	25.05.2022	
4	Розробка інформаційного забезпечення	11.06.2022	
5	Алгоритмізація задачі	03.08.2022	
6	Обґрунтування вибору використаних технічних засобів	10.12.2022	
7	Розробка програмного забезпечення	20.01.2023	
8	Налагодження програми	14.05.2023	
9	Виконання графічних документів	15.05.2023	
10	Оформлення пояснювальної записки	23.05.2023	
11	Подання ДП на попередній захист	02.06.2023	
12	Подання ДП рецензенту	12.06.2023	
13	Подання ДП на основний захист	19.06.2023	

Студент

(підпис)

Данило ВАСИЛЕЦЬ

(ініціали, прізвище)

Керівник

(підпис)

Максим ГОЛОВЧЕНКО

(ініціали, прізвище)

АНОТАЦІЯ

Пояснювальна записка дипломного проєкту складається з чотирьох розділів, містить 52 таблиць, 18 рисунків та 22 джерела – загалом 72 сторінки.

Дипломний проєкт присвячений розробці веб-застосунку для підтримки процесів продажу віртуальних об'єктів.

Мета роботи – спростити процеси продажу та покупки віртуальних об'єктів та надати користувачу можливість підтверджувати своє право власності на віртуальні об'єкти.

Об'єкт дослідження: веб-застосунок підтримки процесів продажу та покупки віртуальних об'єктів

Предмет дослідження: процеси продажу, покупки та зберігання віртуальних об'єктів.

У першому розділі було досліджено предметну область та галузі застосування програмного забезпечення. Також було проведено аналіз ключових вимог до програмного забезпечення.

Розділ другий, присвячений моделювання програмного забезпечення, детальному опису загальної структури та основних компонентів застосунку і проведення аналізу безпеки даних.

У третьому розділі було проведено аналіз якості програмного забезпечення оцінкою коду за допомогою різних метрик та видів тестування.

КЛЮЧОВІ СЛОВА: ВЕБ-ЗАСТОСУНОК, ОБ'ЄКТНЕ СХОВИЩЕ ДАНИХ, DJANGO, ВІРТУАЛЬНІ ОБ'ЄКТИ.

ABSTRACT

The explanatory note of the diploma project consists of four sections, contains 52 tables, 18 figures, and 22 sources – a total of 72 pages.

The diploma project is dedicated to the development of a web application to support the processes of selling virtual objects.

The aim of the work is to simplify the processes of selling and buying virtual objects and to provide the user with the ability to confirm their ownership rights to virtual objects.

Research object: a web application to support the processes of selling and buying virtual objects.

Research subject: the processes of selling, buying, and storing virtual objects.

The first section investigated the subject area and fields of application of the software. An analysis of the key requirements for the software was also carried out.

The second chapter, dedicated to software modeling, a detailed description of the general structure and main components of the application, and data security analysis.

In the third section, a software quality analysis was carried out by evaluating the code using various metrics and types of testing.

KEYWORDS: WEB APPLICATION, OBJECT STORAGE, DJANGO, VIRTUAL OBJECTS.

Пояснювальна записка до дипломного проєкту

на тему: веб-застосунок підтримки процесів продажу віртуальних об'єктів.

КПІ.ІТ-8304.045440.01.81

Київ – 2022

ЗМІСТ

ВСТУП	4
1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ.....	5
1.1 Аналіз існуючих технологій та успішних ІТ-проектів	5
1.2 Аналіз вимог до програмного забезпечення	17
1.3 Постановка задачі	31
Висновки до розділу	32
2 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	33
2.1 Моделювання та аналіз програмного забезпечення.....	33
2.2 Архітектура програмного забезпечення.....	35
2.3 Конструювання програмного забезпечення.....	37
2.4 Аналіз безпеки даних	54
Висновки до розділу	55
3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	57
3.1 Аналіз якості ПЗ.....	57
3.2 Опис процесів тестування.....	59
Висновки до розділу	67
ВИСНОВКИ.....	68
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	69
ДОДАТКИ.....	72

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ

AWS	– Amazon Web Services – хмарна платформа яка надає можливості використовувати свої сервіси та технології.
IT	– Інформаційні технології.
P2P	– Peer-to-peer – варіант архітектури системи, в основі якої стоїть мережа рівноправних вузлів.
S3	– Сервіс-сховище даних, один з Amazon Web Services призначений для зберігання й отримання даних.
JPEG, PNG, BMP, GIF, TIFF	– Joint Photographic Experts Group – Формат зображень, розроблений групою експертів з фотографії.
MVT	– Model View Template
HTTP	– Hypertext Transfer Protocol – Протокол передачі гіпертексту, для передачі даних в Інтернеті.
UI	– UI – User Interface – Інтерфейс користувача, система взаємодії між користувачем і програмою.
JSON	– JavaScript Object Notation – Нотація об'єктів JavaScript, формат даних для передачі та зберігання даних.
YAML	– YAML Ain't Markup Language – Це не мова розмітки, формат для зручного запису даних, призначений для спілкування між людьми та машинами.
CSS	– Cascading Style Sheets – Каскадні таблиці стилів, мова опису зовнішнього вигляду веб-документів.
SQL	– Structured Query Language – Мова структурованих запитів, для роботи з реляційними базами даних.
БД	– База даних – структуроване сховище інформації, яке дозволяє зберігати, змінювати і отримувати дані.
ID	– ID – Identifier – Ідентифікатор, унікальний номер або назва, що призначений для ідентифікації об'єкта чи сутності.

ВСТУП

Початково можливість здійснювати торгові операції в Інтернеті була не звична для широких мас населення, але з часом завоювала довіру більшості користувачів і довела, що за певних обставин вона може бути набагато надійнішою, ніж традиційний підхід до фінансових операцій. Надійність стала не єдиною перевагою використання глобальної мережі для торговельних відносин.

Логістичне питання взаємодії споживача та постачальника дуже часто негативно впливало на можливість проведення багатьох операцій та їх фінансову рентабельність через витрати часу та коштів на подолання великих відстаней. Введення всесвітньої мережі у сферу торгівлі дозволило значно спростити такі процеси, а інколи і зовсім прибрати необхідність у їх проведенні.

Багато сфер сучасного світу вимагають можливості здійснювати торгову взаємодію з іншими людьми. Забезпечення надійності та безпеки таких операцій вимагає від організаторів великої кількості часу та зусиль, що часто не виправдовує отриманого результату. Часто такі процеси також вимагають знань у багатьох областях і значно ускладнює взаємодію користувачів з інструментами які допомагають спростити аспекти надійності. Великою перевагою розробленого програмного рішення стала саме простота та швидкість надання потрібних послуг бажаним провести угоду.

Користувачам сервісу не потрібно приділяти увагу питанням проведення та підготовки торгових операцій, а також перейматися за їх надійність. Сучасний стан об'єкту розробки дозволяє користувачам досягнути поставлених цілей заощаджуючи свій час та зусилля і відкриває перед ними можливість приділити більше уваги своїм справам.

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		4

1 АНАЛІЗ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Аналіз існуючих технологій та успішних ІТ-проектів

Проаналізуємо відоме на сьогодні алгоритмічне забезпечення у даній області та технічні рішення, що допоможуть у реалізації веб-застосунку для підтримки процесів продажу віртуальних об'єктів. Далі будуть розглянуті допоміжні програмні засоби, засоби розробки та готові програмні рішення.

1.1.1 Аналіз відомих алгоритмічних та технічних рішень

Існує різноманіття архітектурних рішень, які можуть бути використані для веб-сервісів, серед яких клієнт-серверна архітектура, Р2Р (однорангова мережа), мікросервісна архітектура та інші. Для веб-сервісу, який спрямований на продаж віртуальних об'єктів, важливим є вибір правильного архітектурного рішення, що забезпечить належну продуктивність, надійність та здатність до масштабування.

Клієнт-серверна архітектура [1] - це один з найбільш традиційних та поширених підходів. Вона заснована на відносинах між двома сторонами: сервером, який надає ресурси або послуги, та клієнтом, який ці ресурси або послуги використовує. Головні переваги цієї архітектури полягають в її простоті, надійності та зручності управління.

Однак, порівняно з мікросервісною архітектурою[2], клієнт-серверна архітектура може бути менш гнучкою в плані масштабування та розподілу навантаження, оскільки вся обробка даних централізована на сервері. Водночас, мікросервісна архітектура може бути переважною для більш великих та складніших систем, але її впровадження та підтримка можуть виявитися більш витратними.

Для розробки веб-сервісу для продажу віртуальних об'єктів, клієнт-серверна архітектура, була визначена оптимальним технічним рішенням, оскільки вона забезпечує необхідний баланс між простотою, надійністю та вартістю.

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		5

Щодо використання об'єктного сховища для зберігання файлів, завантажених користувачами, це також може бути ефективним рішенням. Об'єктні сховища [3] є досить гнучкими та масштабованими, що дозволяє зберігати великі обсяги неструктурованих даних. Вони забезпечують швидкий доступ до даних та надійність зберігання завдяки використанню механізму редундантності та дублювання даних. Такі системи, як Amazon S3, Google Cloud Storage та Microsoft Azure Blob Storage, надають потужні об'єктні сховища, які можуть бути використані для веб-сервісів[3,4,5].

Враховуючи вищезазначене, для вирішення поставленого завдання найбільш доцільним було б використання клієнт-серверної архітектури у поєднанні з об'єктним сховищем для зберігання користувацьких файлів.

Вибір архітектурного паттерну сильно спростило рішення використати фреймворк Django[6] у процесі розробки застосунку. Це дозволило отримати переваги одного з класичних підходів для розробки веб-застосунків, а саме MVT паттерну, уникнувши усіх слабких сторін та недоліків цього вибору.

Як і відомий паттерн MVC[7], MVT [6] поділяють на три основні шари. Шар Моделі представляє структуру даних. Вона відповідає за дані та правила, які керують цими даними. В Django моделі визначають структуру бази даних, ініціалізують з'єднання з базою даних та містять основні операції CRUD (створення, читання, оновлення та видалення).

За контроль відображення, обробку запитів користувача та відповідь яка надається користувачу відповідає шар Перегляд. Він бере дані з моделі і передає їх до шаблону.

Найбільші переваги використано паттерну зосереджені в третьому його шарі – Шаблон. Це частина, яка відповідає за представлення даних, тобто вона відповідає за те, як дані відображаються користувачам. Шаблони Django використовують просту мову шаблонів Django (Django Template Language [6], DTL), яка дозволяє вставляти дані з Python і Django в HTML [8,9].

Описані раніше шари впливають на велику кількість технічних переваг. Одною з таких є можливість відокремити логіку додатку. Логіка додатку

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

розділена на три частини (Модель, Перегляд і Шаблон), що дозволяє розділити обов'язки між різними частинами додатку. Це полегшує процес розробки та обслуговування коду.

Ця особливість обраного паттерну також впливає на, ще одну важливу перевагу розробки додатку у Django. Дякуючи відокремленості, код можна легко повторно використовувати. Наприклад, одну і ту ж модель можна використовувати в різних виглядах або шаблонах.

Серед менших але, дуже зручних особливостей також можна назвати такі переваги:

- Підтримка багатьох інтерфейсів: За допомогою шаблонів можна створити багато різних інтерфейсів для однієї і тієї ж моделі або набору моделей.
- Легкість тестування: Кожен компонент (модель, перегляд, шаблон) можна тестувати окремо, що спрощує процес написання та виконання тестів.
- Гнучкість: Django дозволяє легко змінювати окремі компоненти системи без впливу на інші компоненти.

1.1.2 Аналіз допоміжних програмних засобів та засобів розробки

В результаті розробки було створено веб-сервіс, який забезпечує взаємодію з користувачами, а також програмне забезпечення, яке залишається невидимим для кінцевого користувача і взаємодіє з розробленим веб-сервісом. Для написання сервісу була використана мова програмування Python та фреймворк Django.

Python є об'єктно-орієнтованою мовою програмування високого рівня з динамічною строгою типізацією, розробленою Гвідо ван Россумом у 1990 році. Вона відзначається своєю універсальністю і широким спектром застосувань. Деякі з основних переваг Python включають можливість переносу програм між різними операційними системами, компактний синтаксис порівняно з іншими мовами програмування, швидкість розробки програм та

підтримку модульності і пакетів. Модульність дозволяє повторно використовувати код і сприяє швидкості розробки програм. Всі ці переваги Python є результатом філософії, на якій вона базується.

Після ретельного аналізу доступних рішень було вирішено використовувати Django як основну платформу для розробки веб-системи. Django є потужним фреймворком для створення веб-сайтів на мові програмування Python. Він відрізняється від інших аналогів своєю архітектурою, яка дозволяє розробляти сайти з поділеними на модулі компонентами. Ця архітектура подібна до шаблону MTV (Модель-Шаблон-Представлення), що відрізняється від стандартного шаблону MVC (Модель-Вид-Контролер).

Django надає розробникам багато можливостей, які спрощують процес розробки програм, спрощують роботу з базами даних, дозволяють легко модифікувати програму і додавати новий функціонал, а також забезпечують захист від поширених атак. Він має зрозумілу і зручну структуру проєкту, яку генерує автоматично, що спрощує написання програм. Однією з важливих переваг Django є вбудована панель адміністратора, яка полегшує роботу розробника, оскільки не потребує ручного створення. Проте, функціонал панелі адміністратора можна легко модифікувати, додавши або видаливши необхідні параметри.

Однією з ключових бібліотек, використаних в проєкті, є boto3[10], клієнтська бібліотека Amazon Web Services (AWS) для Python. Boto3 дає можливість працювати з об'єктним сховищем AWS S3, що використовується для зберігання файлів користувачів. Django Storages[11] - альтернатива boto3, яка також підтримує AWS S3, проте boto3 було вибрано завдяки його гнучкості та широким можливостям інтеграції з AWS сервісами.

Щодо клієнтської частини веб-сервісу, було вибрано jQuery[12] - швидко, маленьку та багатофункціональну бібліотеку JavaScript. jQuery дозволяє працювати з HTML документами, обробляти події, створювати анімації та додавати Аях-функціональність з мінімальними зусиллями.

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		8

Розроблене програмне рішення намагається зробити користування сервісом найбільш зручним для користувача тому було вирішено використати Javascript[13] файли для забезпечення вирішення простих задач.

Pillow[14] - бібліотека для обробки зображень в Python. Вона дозволяє виконувати різноманітні операції з зображеннями, такі як масштабування, обертання, зміна формату та інше. Оскільки робота сервісу передбачає обробку та відображення великої кількості зображень було використано бібліотеку яка буде задовольняти всі потреби пов'язані з обробкою зображень. Pillow підтримує широкий спектр форматів зображень, включаючи популярні формати як JPEG, PNG, BMP, GIF, TIFF та інші. Вона надає широкий спектр операцій для обробки зображень, таких як зміна розміру, обертання, перетворення зображень (наприклад, з кольорового в чорно-біле), налаштування кольорових параметрів (яскравість, контраст), а також багато інших операцій, що включають фільтри та ефекти.

Щодо зв'язку з сервісом для створення сертифікатів на основі технології блокчейн, було вибрано бібліотеку http.client. Цей модуль дає змогу створювати HTTP запити, що дуже важливо для взаємодії з сервісом. Бібліотека http.client [15] у Python дає можливість виконувати HTTP запити до серверів. Це стандартна бібліотека, що входить в дистрибутив Python, тому вона не вимагає додаткової установки. Вона підтримує різні типи HTTP запитів, включаючи GET, POST, PUT, DELETE, HEAD і т.д. Ця бібліотека також підтримує обробку відповідей HTTP, включаючи статуси відповідей, заголовки та тіло відповідей. Порівняно з іншими бібліотеками, http.client має ту перевагу, що вона є частиною стандартної бібліотеки Python, тому не потребує додаткового встановлення та інтеграції.

У якості основного середовища розробки було вибрано Visual Studio Code[16]. Це вільно поширюване середовище розробки від Microsoft, яке підтримує множину мов програмування та має велику кількість плагінів. Для взаємодії з блокчейн сервісом були використані додаткові сервіси Insomnia та Swagger.[17,18]

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		9

Insomnia є потужним інструментом для тестування API, що має багато переваг у порівнянні з аналогами. Він підтримує великий набір функціоналу: від простих HTTP запитів до розширеного тестування з використанням сценаріїв. Користувачі можуть створювати колекції запитів, що допомагає організувати роботу з API. Крім того, Insomnia надає засоби для автоматизації тестування API, що включає можливість використовувати змінні оточення, шаблони запитів та інше.

Swagger - це набір інструментів для проектування, побудови, документування та використання RESTful веб-сервісів. Він дозволяє швидко проектувати API за допомогою Swagger Editor, автоматично генерувати документацію для API з використанням Swagger UI, а також генерувати та використовувати клієнтські бібліотеки за допомогою Swagger Codegen. Це надає значні переваги в порівнянні з іншими інструментами для роботи з API.

В якості інструменту для контейнеризації застосунку було обрано Docker. Docker[19] - це платформа для розробки, розгортання та запуску застосунків за допомогою контейнерів. Він має багато переваг у порівнянні з традиційними методами розгортання. Docker гарантує, що застосунок буде працювати однаково в різних оточеннях, так як він включає всі залежності в контейнер. Також Docker дозволяє легко масштабувати застосунки, управляти ресурсами системи та ізолювати процеси. Docker дозволяє пакувати застосунок із всіма його залежностями в стандартизований контейнер, що можна легко розгорнути на будь-якому сервері.

1.1.3 Аналіз відомих програмних продуктів

Після проведення аналізу аналогічних рішень було помічено, що сайти для продажу NFT[20] мають схожий функціонал. З метою детального аналізу переваг та недоліків розробленого веб-сервісу, вирішено було порівняти його з відомим сайтом для продажу NFT - binance.com.

Не дивлячись на те що провідною функцією сайту Binance.com є торгівля криптовалютами, він також має спеціальний розділ, Binance NFT,

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		10

призначений саме для продажу віртуальних об'єктів. Тому ми проаналізуємо можливості цього розділу.

Головну сторінку додатку можна побачити на рисунку 1.1.

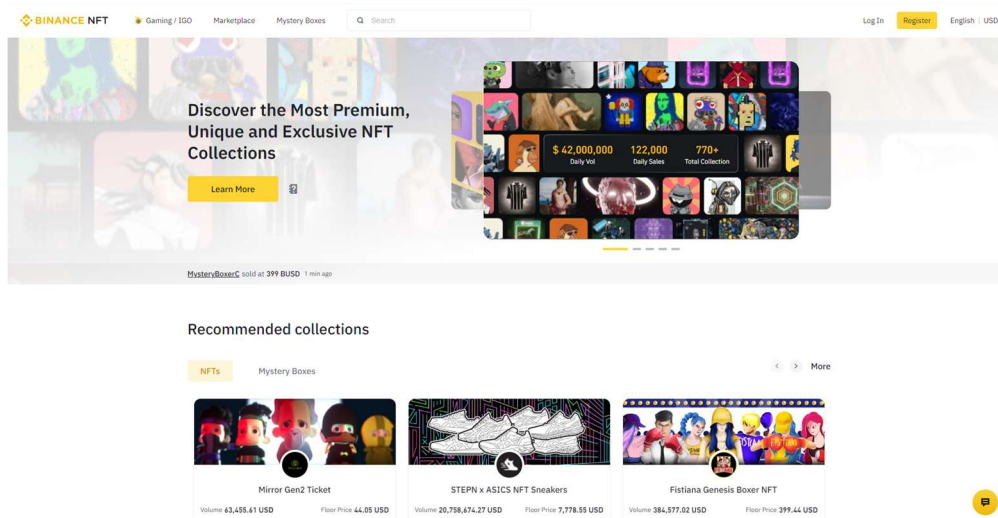


Рисунок 1.1 – Головна сторінка сайту Binance NFT

Binance NFT є підрозділом компанії Binance, що спеціалізується на торгівлі NFT - цифровими об'єктами, які існують унікально у форматі цифрових файлів. Ці файли можуть бути фотографіями, картинками, відео або аудіо файлами. На веб-сторінці www.binance.com/en/nft/marketplace доступний список товарів, які можна придбати. Перегляд цих товарів доступний навіть без реєстрації власного облікового запису користувача.

На рисунку 1.2 зображено інтерфейс для відображення каталогу товарів та короткої інформації про них.

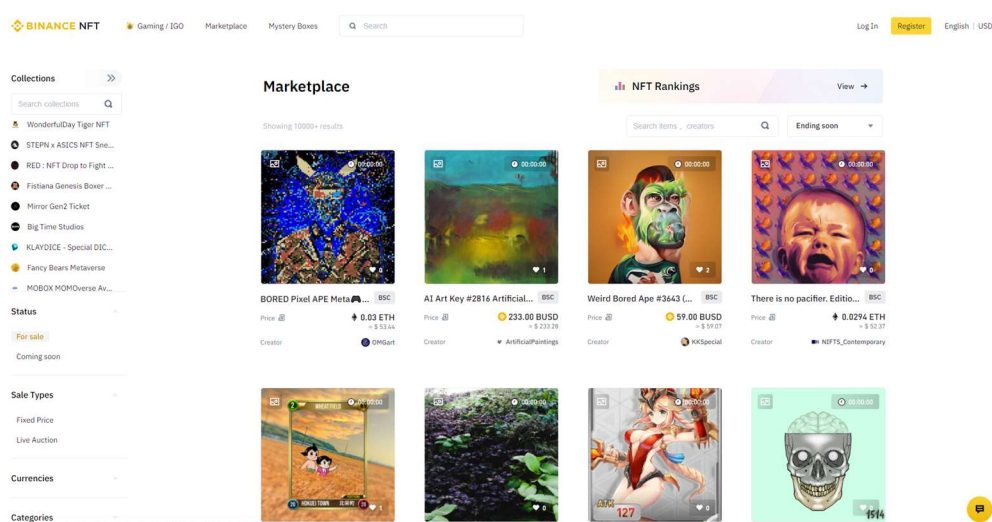


Рисунок 1.2 – Перелік продуктів на сайті Binance NFT

Порівняння існуючого аналогу з розробленим програмним рішенням варто почати з процедури реєстрації користувача на сайті.

На рисунку. 1.3 можна побачити вікно створення облікового запису користувача де перед користувачем одразу з'являється можливість обрати спосіб реєстрації.

Welcome to Binance!

By creating an account you agree to our [Terms and Conditions](#) and [Data Protection Guidelines](#).

Sign up with phone or email

Or

 Continue with Apple

 Continue with Google

Already registered? [Log In](#)



accounts-social-register-
buyCryptoNFT

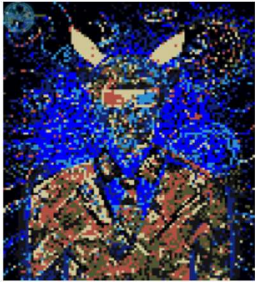
accounts-social-register-buyCryptoTipsNFT

Рисунок 1.3 – Сторінка реєстрації

Після обрання одного з варіантів реєстрації, процес включає в себе кілька окремих кроків.

Їх можна розділити на:

- процес вказування особистих даних користувача;
- підтвердження реєстрації за допомогою телефону або електронної пошти;
- ідентифікацію особи, де користувач повинен надати документ, що підтверджує його особу;
- надсилання фотографії для підтвердження автентичності документу.



BORED Pixel APE Meta  #141

Collectibles

Price

 0.03 ETH
≈ \$ 53.56

Make Offer

Provenance

Name	Action	Trade Price	Time
OMGart	mint		2022-05-03 04:45:14

Description

BORED Pixel APE Meta  #141 These Meta Apes, which came out of the old game consoles, are waiting for the old computer cable connection on their journey to the metaverse. Get it and connect it. 
500x500 OMGart @thisisomgart

Details

Creator	 OMGart
Owner	 OMGart
Network	BSC
Contract Address	0x1dDB...a9Ec
Token ID	100301173058

Рисунок 1.4 – Сторінка перегляду товару

Наступним кроком у процесі аналізу можливостей сайту була перевірка сторінки одного з товарів. Рисунок 1.4 показує, що для придбання цього товару необхідно використовувати криптовалюту.

Конкретно цей товар був виставлений на аукціон, де користувачі можуть робити ставки, а покупець з найвищою ставкою отримує цей товар.

Також існує можливість встановлення фіксованої ціни за одиницю товару. При покупці такого товару користувач отримує сам об'єкт і спеціальний токен, а операція реєструється у публічному блокчейні.

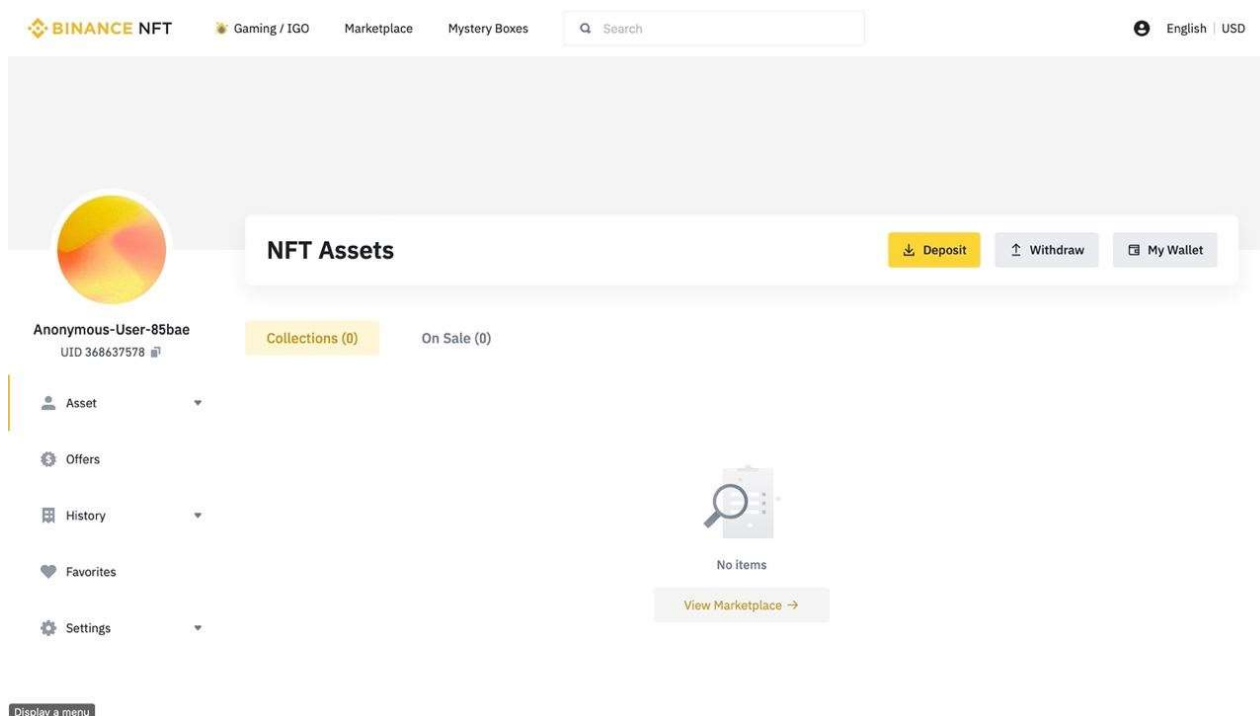


Рисунок 1.5 – Власний кабінет користувача

У особистому кабінеті користувача доступний перегляд придбаних товарів, а також товарів, які користувач виставив на продаж. Крім того, в кабінеті є історія дій, де можна переглянути попередні покупки та продажі, пов'язані з обліковим записом користувача.

Для здійснення покупок на сайті необхідно використовувати криптовалюту, встановлену продавцем. Користувач має можливість переказати певну суму зі свого гаманця на сайт або прив'язати свій гаманець до облікового запису для зручності оплати товарів.

Криптовалюти для оплати покупки вказуються власником, що вистави товару на продаж. На вибір надаються підтверджені та найбільш популярні і стабільні ринкові валюти. Можливість виставити декілька варіантів оплати відсутня

Для порівняння отриманого в результаті розробки дипломного проекту з аналогом була складена таблиця 1.1.

					КПІ.ІТ-8304.045440.01.81		Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.			14

Таблиця 1.1 – Порівняння з аналогом

Функціонал	FunnyBusiness	Binance NFT	Пояснення
Можливість проходити реєстрацію на сайті.	Наявний	Наявний	На обох сайтах наявна така можливість
Підтвердження особи користувача за допомогою документа, що може завіряти особу.	Відсутній	Наявний	Необхідність підтвердити свою особу негативно впливає на бажання використовувати такий сайт.
Можливість купувати товари на сайті.	Наявний	Наявний	Така можливість наявна в обох сервісах.
Можливість виставляти віртуальні об'єкти на продаж.	Наявний	Відсутній	Для цього сайт-аналог використовує сторонні сервіси. У дипломному проєкті це можна зробити на самому сайті.
Реєстрація проведеної торгової операції у блокчейні	Наявний	Наявний	Обидва сайти надають таку можливість.
Отримання сертифікату про право власності на віртуальний об'єкт.	Наявний	Відсутній	Сайт-аналог не надає такої можливості на відміну від розробленого сервісу.

Продовження таблиці 1.1

Функціонал	Розроблени й веб- сервіс	Аналог для порівняння	Пояснення
Пошук товарів за категоріями та назвами.	Наявний	Наявний	Обидва сайти мають можливість проводити пошук за такими параметрами.
Можливість використовувати нецифрові валюти при придбанні товарів.	Наявний	Відсутній	Сайт аналог використовує лише криптовалюту для продажу товарів.
Виставлення товару на аукціон.	Відсутній	Наявний	Розроблений проєкт передбачає фіксовану ціну на кожний товар.
Унікальність кожного придбаного товару	Наявний	Відсутній	У дипломній роботі кожен виставлений товар є унікальним на відміну від аналогу.
Можливість передивлятись придбані та виставлені на продаж товари у власному кабінеті.	Наявний	Наявний	Така можливість присутня на обох сайтах.
Можливість передивлятись історію операцій користувача у власному кабінеті.	Наявний	Наявний	Обидва сервіси надають користувачеві таку можливість

1.2 Аналіз вимог до програмного забезпечення

Основною функцією розробленого програмного рішення є надання користувачу можливості виставляти на продаж або купувати віртуальні об'єкти. Також одною з найважливіших функцій є можливість отримати підтвердження права власності на відповідний об'єкт при придбанні віртуального об'єкта у іншого користувача. Для реалізації цієї можливості і забезпечення захищеності цієї інформації було розроблено інтерфейс прикладного програмування який використовує технологію блокчейн.

Створена блокчейн мережа зберігає данні про право власності користувачів і за допомогою інтерфейсу передає їх до веб-сервісу. Для взаємодії з самим користувачем було розроблено окремий веб-сервіс який відповідає за функції пов'язані з віртуальними об'єктами і дозволяє виконувати з ними операції придбання, продажу та зберігання.

Після проходження реєстрації на сайті користувач може обрати з переліку доступних товарів об'єкт та придбати його після чого він отримає сертифікат який буде підтверджувати його право власності на цей об'єкт. Також він має можливість виставити на продаж власний товар і переглянути свої виставлені на продаж або придбані товари у власному кабінеті. Більш детально варіанти використання можна побачити на рисунку 1.6.

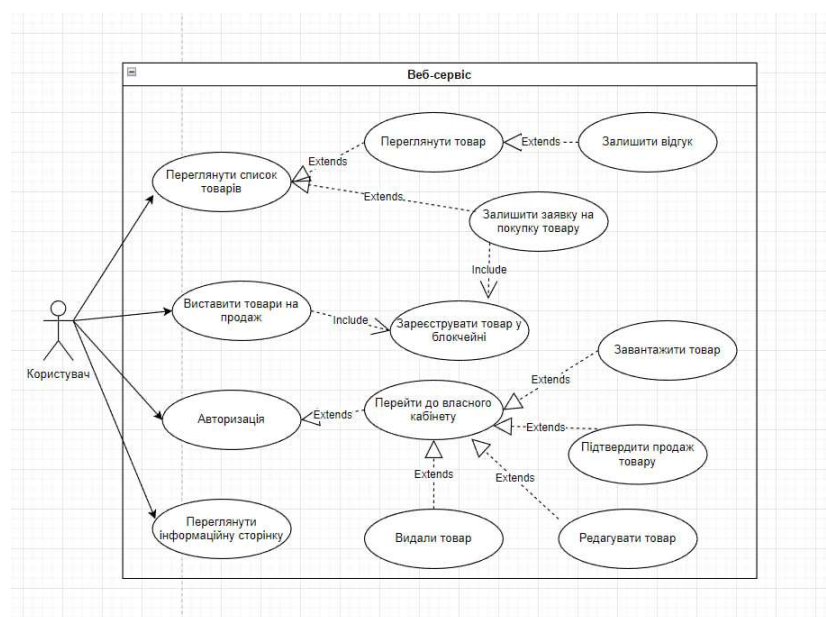


Рисунок 1.6 – Діаграма варіантів використання

В таблицях 1.2 – 1.10 наведені варіанти використання розробленого програмного рішення.

Таблиця 1.2 - Варіант використання UC-1

Назва	Реєстрація на сайті
Опис	Реєстрація профілю нового користувача
Учасники	Користувач який ще не проходив реєстрацію, або бажає пройти її повторно
Постумови	Запис інформації про користувача та збереження його профілю у веб сервісі.
Основний сценарій	Користувач здійснює перехід на сторінку створення облікового запису та заповнює поля форми. Після заповнення полей логіну користувача, електронної пошти, паролю, підтвердження паролю та натискання кнопки підтвердження профіль буде створено. Користувача буде переадресовано на сторінку з інформацією про приватний та публічний ключ, а також відбудеться вхід у систему під обліковим записом який було створено.
Розширений сценарій	Користувач також може заповнити необов'язкові поля імені та прізвища. У випадку введення некоректних даних або якщо користувач залишає обов'язкові поля пустими він отримає повідомлення з підказкою які поля потрібно заповнити.

Таблиця 1.3 - Варіант використання UC-2

Назва	Виставлення товару на продаж
Опис	Користувач може виставити власний товар на продаж
Учасники	Зареєстрований користувач
Постумови	Додання товару на головну сторінку, або до профілю користувача.
Основний сценарій	Користувач переходить на сторінку створення товару і заповнює такі поля: назва товару, опис товару, ціна товару та ввести свій ключ. Також користувач має обрати категорію з наданого переліку залишити чек бокс активованим або прибрати мітку та завантажити файл зображення товару і файл який бажає виставити на продаж. Після натискання на кнопку зберегти, товар з'явиться в власному кабінеті користувача та, за умови активованого чек боксу, буде виставлений до переліку доступних для придбання товарів
Розширений сценарій	У випадку введення некоректних даних або якщо користувач залишає обов'язкові поля пустими висвічується текст який вказує які поля потрібно заповнити.

Таблиця 1.4 - Варіант використання UC-3

Назва	Створення заявки на придбання товару
Опис	Користувач залишає заявку на придбання товару
Учасники	Зареєстрований користувач
Постумови	Користувач виконить перший крок до придбання товару, а його власник отримає заявку на покупку.

Продовження таблиці 1.4

Основний сценарій	Користувач має знайти потрібний товар на сторінці каталогу. Він може одразу натиснути кнопку придбати або перейти на сторінку з детальною інформацією про товар та після більш детального огляду товару вирішити придбати його. Після натискання кнопки придбання товару користувача буде направлено на сторінку с інструкціями до подальших дій, а у його власному кабінеті з'явиться інформація про те, що він очікує на підтвердження продажу від власника товару.
Розширений сценарій	Якщо цей товар вже було продано, або виникли певні помилки з обліковим записом користувача він отримає повідомлення про помилку.

Таблиця 1.5 - Варіант використання UC-4

Назва	Підтвердження заявки на придбання товару
Опис	Власник товару підтверджує передачу товару
Учасники	Зареєстрований користувач, що володіє певним товаром
Постумови	Товар буде передано замовнику
Основний сценарій	Власник товару який бажають придбати має перейти у власний кабінет де буде відображено заявку на передачу товару. У таблиці товарів користувача буде присутнє поле куди він повинен ввести свій приватний ключ для підтвердження володіння цим товаром. Після введення правильного приватного ключа та натискання на кнопку підтвердження операції товар буде передано замовнику.

Продовження таблиці 1.5

Розширений сценарій	Якщо перевірка ключа не пройшла успішно власник товару отримає повідомлення з причиною помилки у операції передачі та товар не буде передано.
---------------------	---

Таблиця 1.6 - Варіант використання UC-5

Назва	Редагування інформації про товар
Опис	Придбаний або виставлений на продаж товар можна редагувати у власному кабінеті
Учасники	Зареєстрований користувач
Постумови	Інформацію про товар буде відредаговано
Основний сценарій	Для редагування потрібної інформації про товар користувачу який виставив на продаж або придбав певний товар необхідно перейти до власного кабінету. У цьому кабінеті користувач може натиснути на кнопку редагувати після чого його буде перенаправлено на форму редагування товару. Після цього він може обрати поля для редагування та змінити інформацію в них.
Розширений сценарій	Якщо користувач залишить обов'язкові поля пустими, або спробує ввести не коректні дані у поля він отримає повідомлення з інструкціями які помилки необхідно виправити для збереження товару

Таблиця 1.7 - Варіант використання UC-6

Назва	Видаляти товар
Опис	Придбаний або виставлений на продаж товар можна видалити у власному кабінеті
Учасники	Зареєстрований користувач

Продовження таблиці 1.7

Постумови	Дані товару будуть видалені з застосунку та зникнуть з власного кабінету користувача.
Основний сценарій	Для видалення товару користувачу який виставив на продаж або придбав певний товар необхідно перейти до власного кабінету. У цьому кабінеті користувач повинен натиснути на кнопку видалити розташовану поряд з об'єктом.

Таблиця 1.8 - Варіант використання UC-7

Назва	Авторизація
Опис	Авторизація під обліковим записом який користувач створив раніше
Учасники	Зареєстрований користувач
Постумови	Користувач буде авторизований на сайті під певним обліковим записом
Основний сценарій	Для авторизації на сайті користувачеві необхідно перейти до сторінки авторизації і заповнити поля логіну та пароллю. Після правильного заповнення полів форми авторизації користувач повинен натиснути на кнопку увійти. Його буде перенаправлено на головну сторінку сайту під обліковим записом у який він здійснював вхід.
Розширений сценарій	У випадку внесення некоректних даних до полів користувач отримає повідомлення з інструкціями для правильного проходження авторизації. Також користувач може поставити мітку у чек бокс і його дані для авторизації запам'ятають для автоматичної авторизації при наступному відвіданні сайту.

Таблиця 1.9 - Варіант використання UC-8

Назва	Пошук товарів на сайті
Опис	Користувач може використати функцію пошуку та знайти потрібний товар
Учасники	Зареєстрований користувач
Постумови	Користувача буде перенаправлено на сторінку з переліком усіх товарів, що відповідають умовам запиту.
Основний сценарій	Користувач може скористатися панеллю пошуку товару з усіх сторінок сайту. Після введення інформації за якою користувач бажає здійснити пошук він повинен натиснути на кнопку пошуку, або на клавішу Enter. В результаті таких дій його буде переадресовано на сторінку для відображення всіх товарів які задовольняють умови пошуку.
Розширений сценарій	Якщо користувач залишить поле пошуку пустим або за результатами пошуку не буде знайдено товарів користувач отримає відповідне повідомлення.

Таблиця 1.10 - Варіант використання UC-9

Назва	Перехід на сторінку профілю користувача
Опис	Користувач може перейти на сторінку свого профілю для використання деяких функцій сайту.
Учасники	Зареєстрований користувач
Постумови	Користувач отримає можливість переглядати свій профіль та здійснювати операції з товарами якими він володіє.
Основний сценарій	Користувач може перейти на сторінку де буде відображено товари якими він володіє то товари, що очікують на підтвердження продажу від власника.

Продовження таблиці 1.10

Розширений сценарій	Якщо користувач не зареєстрований на сайті, або не увійшов у свій обліковий запис то він не зможе переглянути сторінку профілю. Також якщо користувач не завантажив та не придбав жодного товару то його профіль буде пустим.
---------------------	---

1.2.1 Розроблення функціональних вимог

Створений застосунок структуровано на окремі модулі, кожний з яких виконує конкретний набір функцій. Детальний опис функціональних вимог до розробленого веб-сервісу подається в таблицях 1.12 - 1.31, а загальну модель вимог можна побачити на таблиці 1.11 Для забезпечення зручності, веб-сервіс було розбито на різні модулі. Матрицю трасування вимог можна переглянути на рисунку 1.7.

Таблиця 1.11 – Загальна модель вимог

Номер	Опис вимоги	Код	Ризики
1	Управління обліковим записом користувача	FR-1	2
1.1	Реєстрація облікового запису користувача на сервісі	FR-2	3
1.1.1	Запис додаткової інформації про користувача	FR-3	1
1.2	Вхід у систему	FR-4	2
1.2.1	Запам'ятати користувача у системі	FR-5	0
1.3	Перегляд профілю користувача	FR-6	0

Продовження таблиці 1.11

Номер	Опис вимоги	Код	Ризики
2	Управління цифровими об'єктами	FR-7	2
2.1	Реєстрація нового товару на сервісі	FR-8	3
2.2	Додавання зображення до товару	FR-9	1
2.2.1	Обробка зображення	FR-10	0
2.3	Покупка товару	FR-11	2
2.2.1	Створення заявки на придбання товару	FR-12	0
2.2.2	Підтвердження заявки на придбання товару	FR-13	2
2.4	Видалення товару	FR-14	0
2.5	Редагування товару	FR-15	1
2.6	Завантаження цифрового об'єкту	FR-16	2
3	Операції пошуку товару	FR-17	0
3.1	Пошук товару за категорією	-	0
3.2	Пошук товару за назвою	-	0
4	Взаємодія з програмним інтерфейсом	FR-18	2

Продовження таблиці 1.11

Номер	Опис вимоги	Код	Ризики
4.1	Отримання ключів	FR-19	3
4.2	Перевірка права власності	FR-20	1

Проведено оцінку ризиків представлених вимог. Оцінки від 0 до 3 показують ступінь складності забезпечення безпеки для обробки певних процесів. Вимоги з найбільшою чутливістю даних були позначені третім рівнем ризику.

Таблиця 1.12 – Функціональна вимога FR-1

Назва	Управління обліковим записом користувача
Опис	Користувач повинен мати можливість здійснювати операції з своїм обліковим записом.

Таблиця 1.13 – Функціональна вимога FR-2

Назва	Реєстрація облікового запису користувача на сервісі
Опис	Користувачу має бути надана можливість створити новий обліковий запис.

Таблиця 1.14 – Функціональна вимога FR-3

Назва	Запис додаткової інформації про користувача
Опис	Розроблена система повинна надавати користувачеві можливість під час реєстрації вказати своє ім'я та прізвище.

Таблиця 1.15 – Функціональна вимога FR-4

Назва	Вхід у систему
Опис	Користувачу має бути надана змога увійти до системи, ввівши свій логін та пароль в призначені для цього поля.

Таблиця 1.16 – Функціональна вимога FR-5

Назва	Запам'ятати користувача у системі
Опис	Система повинна надавати користувачам опцію "запам'ятати мене". Ця функція дозволяє зберігати інформацію про логін та пароль користувача після того, як він пройшов процедуру входу в систему.

Таблиця 1.17 – Функціональна вимога FR-6

Назва	Перегляд профілю користувача
Опис	Користувачу має бути надана можливість перейти до свого профілю.

Таблиця 1.18 – Функціональна вимога FR-7

Назва	Управління цифровими об'єктами
Опис	Користувач повинен мати можливість управляти товарами .

Таблиця 1.19 – Функціональна вимога FR-8

Назва	Реєстрація нового товару на сервісі
Опис	Користувачу має бути надана можливість виставити товар на продаж, заповнивши відповідні поля: назву товару, опис, ціну, вибравши категорію товару, ввівши ключ а також завантаживши файл, який планує продати.

Таблиця 1.20 – Функціональна вимога FR-9

Назва	Додавання зображення до товару
Опис	Система повинна надавати можливість користувачу додати зображення до товару.

Таблиця 1.21 – Функціональна вимога FR-10

Назва	Обробка зображення
Опис	Веб-сервіс повинен надавати користувачеві можливість обробити зображення товару при його реєстрації.

Таблиця 1.22 – Функціональна вимога FR-11

Назва	Покупка товару
Опис	Користувачу має бути надана можливість придбати товару, що виставлено на продаж.

Таблиця 1.23 – Функціональна вимога FR-12

Назва	Створення заявки на придбання товару
Опис	Система повинна надавати можливість користувачу залишити заявку на придбання товару.

Таблиця 1.24 – Функціональна вимога FR-13

Назва	Підтвердження заявки на придбання товару
Опис	Користувач має мати можливість підтвердити заявку на покупку його товару.

Таблиця 1.25 – Функціональна вимога FR-14

Назва	Видалення товару
Опис	Повинна існувати можливість видалити зареєстрований або придбаний користувачем товар.

Таблиця 1.26 – Функціональна вимога FR-15

Назва	Редагування товару
Опис	Розроблений сервіс повинен надавати користувачеві можливість редагувати поля товару дозволяючи йому змінити назву, опис, ціну та категорію товару яким він володіє.

Таблиця 1.27 – Функціональна вимога FR-16

Назва	Завантаження цифрового об'єкту
Опис	Користувач має мати можливість завантажувати віртуальні об'єкти у якості товару.

Таблиця 1.28 – Функціональна вимога FR-17

Назва	Операції пошуку товару
Опис	Повинна існувати можливість проводити пошук товарів на сайті.

Таблиця 1.29 – Функціональна вимога FR-18

Назва	Взаємодія з програмним інтерфейсом
Опис	Повинна бути налагоджена системна взаємодія сервісу з програмним інтерфейсом.

Таблиця 1.30 – Функціональна вимога FR-19

Назва	Отримання ключів
Опис	Користувач має мати можливість отримати свій приватний та публічний ключ.

Таблиця 1.31 – Функціональна вимога FR-20

Назва	Отримання ключів
Опис	Веб-сервіс повинен виконувати перевірку права власності при здійсненні основних операцій з віртуальними об'єктами.

	FR-1	FR-2	FR-3	FR-4	FR-5	FR-6	FR-7	FR-8	FR-9	FR-10	FR-11	FR-12	FR-13	FR-14	FR-15	FR-16	FR-17	FR-18	FR-19	FR-20
UC-1	+	+	+															+	+	
UC-2							+	+	+	+										+
UC-3							+				+	+								
UC-4							+				+		+							+
UC-5							+								+					
UC-6							+							+						
UC-7	+			+	+															
UC-8																	+			
UC-9	+					+														

Рисунок 1.7 – Матриця трасування вимог

1.2.2 Розроблення нефункціональних вимог

Розроблений веб-сервіс повинен:

- мати інтуїтивно зрозумілий та зручний користувацький інтерфейс;
- бути зручним у використанні;
- працювати на усіх найпопулярніших браузерях;
- працювати без затримок і довгих завантажень;
- надавати додаткові інструменти для полегшення використання

функцій сайту.

Завдяки використанню різних видів інструментів фрон-енд розробки вдалося досягти значних успіхів у створенні ефективного інтерфейсу. Завдяки модулям на мові програмування Javascript було створено декілька додаткових інструментів для роботи з зображеннями і спрощення використання інших функцій сервісу.

Для забезпечення зрозумілості та зручності використання розробленого сервісу було проведено ряд тестів, що включало кейси тестування основних функцій користувачами, які були не знайомі з додатком. Враховуючи їх зауваження та побажання.

1.3 Постановка задачі

Розробка, над якою проводиться робота, призначена для користувачів всесвітньої мережі, що бажають придбати право власності на віртуальні об'єкти онлайн. Цей веб-сервіс спрямований на вирішення важливих задач, що стоять перед користувачами, прагнучими придбати або продати віртуальні об'єкти.

Основною метою розробки є спрощення процедури придбання віртуальних об'єктів. Це досягається шляхом створення інтуїтивно зрозумілого інтерфейсу та автоматизації основних процесів, пов'язаних з торгівлею віртуальними об'єктами.

Ключовою задачею є автоматизація процесу створення сертифікатів, що підтверджують право власності на віртуальний об'єкт. Це дозволить користувачам мати впевненість у валідності їхніх прав на придбаний віртуальний об'єкт та уникнути можливих юридичних конфліктів.

Іншою важливою задачею є забезпечення можливості безпечного зберігання віртуальних об'єктів в хмарному сховищі до їх продажу або після придбання. Це дасть користувачам зручний і надійний спосіб зберігання своїх віртуальних об'єктів, що зменшить ризик їх втрати або крадіжки.

Отже, основні задачі, що стоять перед розробкою, включають спрощення процесу придбання віртуальних об'єктів, автоматизацію створення сертифікатів власності та забезпечення безпечного зберігання віртуальних об'єктів.

Висновки до розділу

Перший розділ дипломної роботи зосереджується на детальному дослідженні предметної області та галузі застосування програмного забезпечення. Під час аналізу предметної області було виявлено ключові характеристики та виклики, які визначають потреби та вимоги до програмного забезпечення.

При розробці програмного забезпечення було забезпечено підтримку великої кількості типів файлів, що розширює можливості для користувачів при виборі товарів для продажу. Розроблений програмний продукт пропонує простий та зручний процес покупки, який допомагає користувачам легко здійснювати торговельні операції.

Функція отримання сертифікат для підтвердження права власності забезпечує додаткову впевненість користувачів в їх правах на придбані товари.

Додавання можливості торгувати віртуальними об'єктами надає нові можливості для користувачів і розширює діапазон товарів, доступних для продажу. Опис розробленого програмного забезпечення продемонстрував, що розроблений веб-сайт має інтуїтивно зрозумілий і простий процес реєстрації, що полегшує вступ нових користувачів до сервісу.

Забезпечення унікальності кожного об'єкту, що виставлено на продаж допомагає гарантувати високий рівень захисту прав власності користувачів і додатково підвищує привабливість продаваних об'єктів.

2 КОНСТРУЮВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Моделювання та аналіз програмного забезпечення

Для опису бізнес процесу програмного забезпечення використовується BPMN модель (рисунок 2.1).

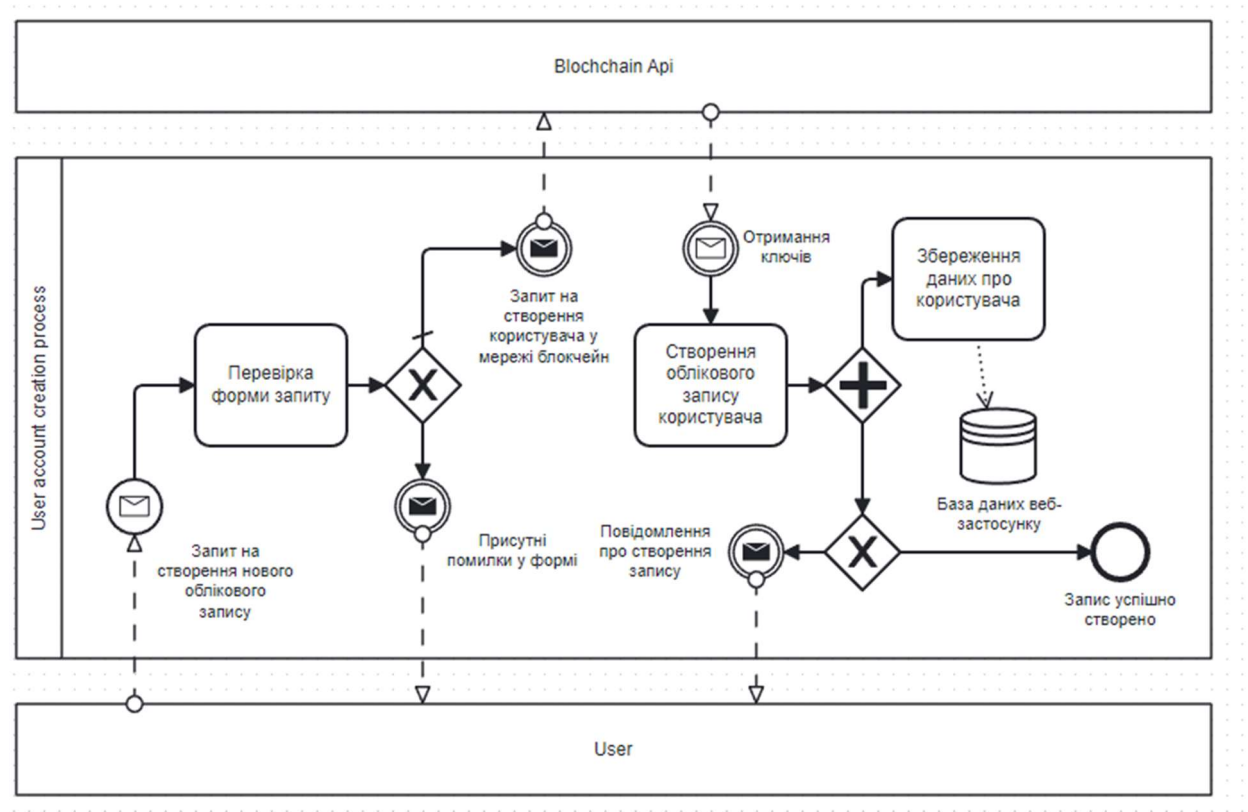


Рисунок 2.1 – BPMN процесу створення облікового запису користувача

Опис послідовності створення нового облікового запису користувача на сайті:

- Користувач заповнює та надсилає форму для створення нового облікового запису;
- Сервіс перевіряє всі обов’язкові поля форми на наявність, в потім усі поля на правильність;
- У випадку виявлення помилок у формі користувач отримає відповідне повідомлення;

- Якщо форма заповнена коректно додаток надсилає запит до зовнішнього сервісу на створення запису;
- Після отримання ключів користувача відбувається створення облікового запису на сервісі;
- Дані про користувача зберігаються, а він отримує повідомлення про успішну реєстрацію.

Для опису бізнес процесу додавання нового товару на сайт використовується BPMN модель, що зображена на рисунку 2.2.

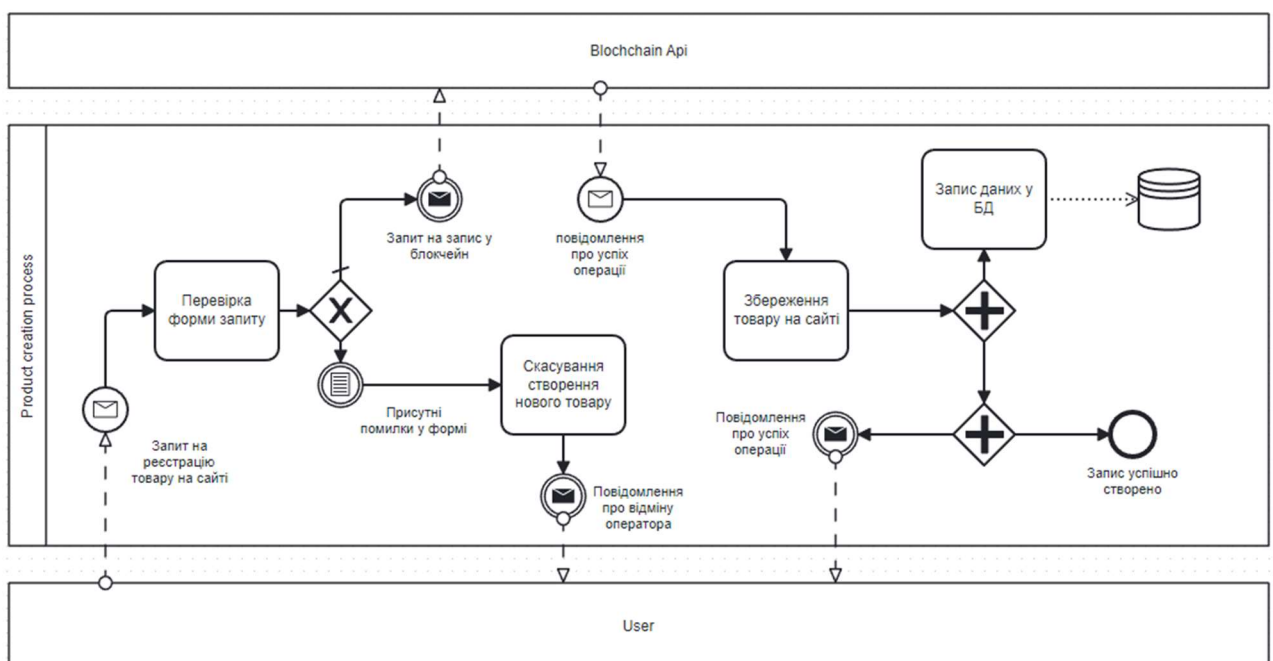


Рисунок 2.2 – BPMN процесу додавання нового товару на сайт

Опис послідовності створення нового товару на сайті:

- Користувач заповнює та надсилає форму для створення нового товару;
- Сервіс перевіряє всі обов’язкові поля форми на наявність, в потім усі поля на правильність;
- Якщо виявлено помилки у запит користувач отримає повідомлення про відміну операції;

- У випадку правильно оформленого запиту сервіс надсилає запит до зовнішнього програмного інтерфейсу;
- Після отримання відповіді з інформацією про успіх операції товар додається на сайт, а усі його поля зберігаються в БД;
- Користувач отримує повідомлення про успіх операції.

2.2 Архітектура програмного забезпечення

На рисунку 2.3 наведено схему взаємодії обраної архітектури у розробленому проєкті.

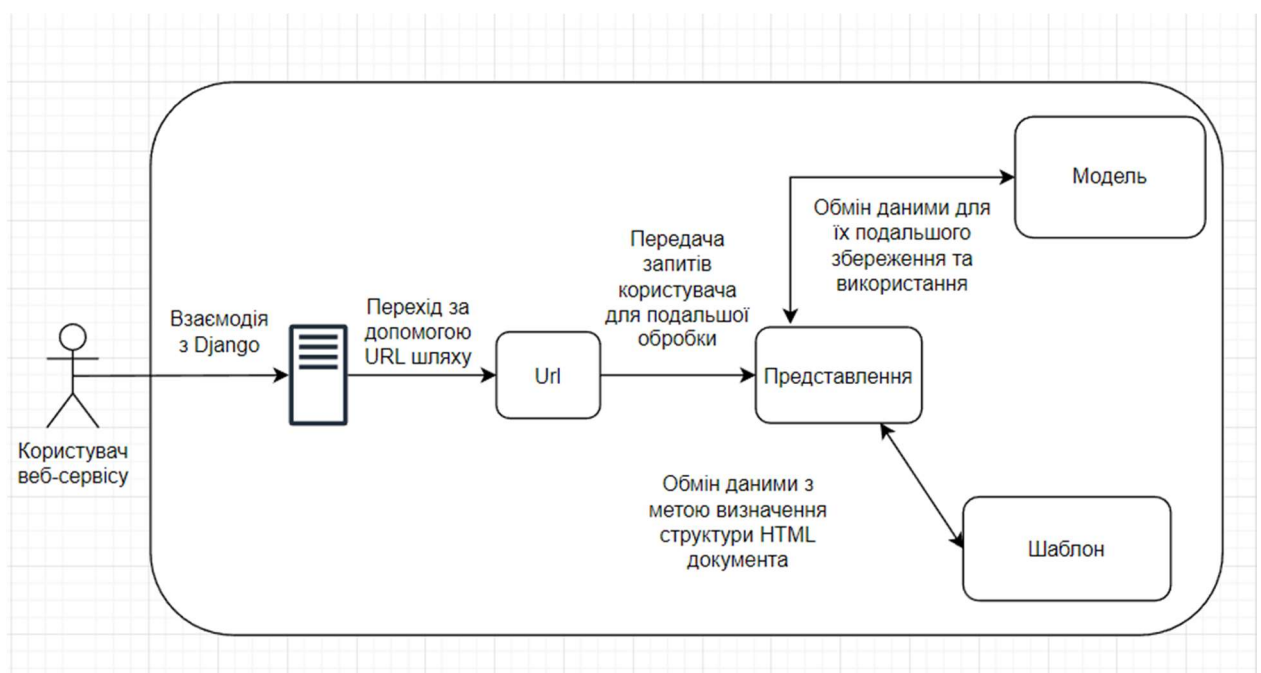


Рисунок 2.3 – Взаємодія компонентів веб-застосунку

Основні компоненти взаємодії, що представлені в розробленому проєкті можна поділити на такі шари:

2.2.1 Шар бізнес-логіки

Шар бізнес-логіки у шаблоні MVT схожий на класичний підхід до такого шару у подібних паттернах і відповідає за обробку даних та управління бізнес-правилами застосунку. В контексті MVT, цей шар часто відображається через моделі. Моделі у MVT представляють структуру даних і зазвичай є

відображенням таблиці бази даних. Моделі відповідають за збереження, отримання та маніпуляцію даними. Вони можуть містити методи для обробки даних та виконання бізнес-правил. Таким чином, шар бізнес-логіки є важливою частиною архітектури MVT, оскільки він містить основну логіку застосунку і визначає, як дані обробляються та представляються користувачеві.

2.2.2 Шар представлення

Цей шар є відповідальним за обробку запитів від користувачів і відправлення відповідей.

Відмінність шару View у Django від традиційних систем MVC (Model-View-Controller) полягає в тому, що в Django View виконує роль як контролера, так і представлення в звичайному розумінні MVC. View у Django обробляє вхідні HTTP запити, виконує всю необхідну логіку, взаємодіє з моделями для доступу до даних, а потім використовує шаблони для створення HTTP відповідей.

Цей шар у варіації шаблону MVT для Django може бути представлений у виді функцій або класів. При розробці було вирішено використовувати функції які приймають вхідний запит і віддають відповідь. Вони легкі для розуміння і використання, але можуть стати складними при обробці більш складних потреб.

2.2.3 Шар шаблону

Основною ціллю шару Template (шаблон) є визначення структури вихідного документа яка представлена у цьому проєкті у формі HTML сторінки. Це досягається за допомогою текстових файлів, що використовують простий синтаксис для дозволу динамічної генерації вмісту HTML.

Шаблон відіграє важливу роль у процесі відокремлення бізнес логіки додатку від представлення даних. Це дозволяє полегшити розробку, відладку та підтримку коду.

Ще одною особливістю цього шару в Django є можливість використання власного, простого та зрозумілого синтаксису шаблонів. Він містить теги, фільтри та плейсхолдери для змінних, що дозволяє легко створювати динамічні HTML документи. Окрім власного синтаксису, Django також надає можливість використати наслідування шаблонів під час розробки. Така функція дозволяє визначати базові шаблони з блоками, які можуть бути перевизначені в дочірніх шаблонах. Це дуже зручно для створення єдиної структури сторінки та дозволяє виділити спільні елементи для всіх сторінок застосунку, або для обраної розробником групи.

Завдяки використанню такого шару вдається також забезпечити такі важливі для розробленого програмного рішення параметри як безпека та гнучкість розробки. Автоматичне екранування вихідних даних в шаблонах атаки через вставку коду стають не можливими для використання шахраями. Можливості використовувати будь-які текстові формати у шаблонах дозволяє досягнути більшої гнучкості у відображенні даних для користувача.

2.3 Конструювання програмного забезпечення

2.3.1 Детальний опис використаних бібліотек

Для роботи різних компонентів програми використовуються окремі бібліотеки. Оскільки розроблений веб-сервіс використовує велику кількість таких бібліотек найбільш ефективним варіантом буде використання окремого віртуального простору для роботи з проектом. Імпортувати ці бібліотеки можна за допомогою згенерованого файлу – requirements.txt, збереженого у самому проєкті. У нього записано перелік самих бібліотек та версія яку необхідно використовувати. У таблиці 2.1 наведено список основних бібліотек та коротко описати їх призначення.

					КПІ.ІТ-8304.045440.01.81	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		37

Таблиця 2.1 - Основні бібліотеки

asgiref==3.5.0	ASGI (Asynchronous Server Gateway Interface) є стандартом для Python асинхронних веб-серверів і додатків. Asgiref - це бібліотека, що містить загальні утиліти для роботи з ASGI.
astroid==2.15.5	Astroid - це загального призначення, бібліотека для обробки Python абстрактних синтаксичних дерев. Часто використовується разом з pylint.
beautifulsoup4==4.10.0	BeautifulSoup - це бібліотека Python для парсингу HTML та XML документів. Вона створена для веб-скрапінгу, витягування даних з веб-сторінок.
boto3==1.23.2	Boto3 - це SDK Amazon AWS для Python, що дозволяє Python програмам взаємодіяти з Amazon веб-сервісами, такими як S3, DynamoDB та іншими.
botocore==1.26.2	Botocore - це низькорівневий, ядро інтерфейсу до AWS сервісів для Python. Boto3 будується на основі Botocore.
cement==2.8.2	Cement - це легкий, відкритий фреймворк для побудови надійних командних рядків (CLI) та демонів на Python.
certifi==2021.10.8	Certifi - це бібліотека Python, що містить офіційний набір корневих сертифікатів довіри Mozilla.
charset-normalizer==2.0.12	Ця бібліотека використовується для нормалізації кодування символів в текстових даних.
colorama==0.4.6	Colorama - це бібліотека, що робить ANSI екран контролю (колір та стилі) працюючи на Windows, як і на Unix/Linux.

Продовження таблиці 2.1

dill==0.3.6	Dill - це бібліотека для серіалізації даних в Python, що може серіалізувати та десеріалізувати більше типів Python, ніж вбудовані модулі pickle та cPickle .
Django==3.2.12	Django - це високорівневий Python веб-фреймворк, що заохочує швидку розробку та чистий, простий дизайн.
django-bootstrap-v5==1.0.11	Ця бібліотека дозволяє інтегрувати Bootstrap v5 з Django проектами.
django-storages==1.12.3	Набір вбудованих в Django класів для роботи зі сховищами даних, такими як Amazon S3, Azure Storage та іншими.
future==0.16.0	Future - це бібліотека для забезпечення сумісності Python 2 / Python 3.
idna==3.3	Бібліотека для підтримки Internationalized Domain Names in Applications (IDNA).
jmespath==0.10.0	JMESPath (JSON Matching Expressions) - це мова запитів для JSON. Використовується в AWS CLI.
lazy-object-proxy==1.9.0	Бібліотека, яка дозволяє створювати "ліниві" проксі об'єкти для відкладеного обчислення.
minio==7.1.15	Клієнт Python для Minio, високопродуктивного розподіленого об'єктового сховища.
numpy==1.21.5	NumPy - це основна бібліотека для наукових обчислень в Python.
Pillow==9.0.1	Pillow - це бібліотека для роботи з растровою графікою в Python, це активно розвивається форк оригінального PIL (Python Imaging Library).

Продовження таблиці 2.1

platformdirs==3.5.1	Бібліотека для визначення стандартних директорій на різних платформах.
pycryptodome==3.18.0	PyCryptodome - це самодостатня бібліотека Python для криптографії.
pywin32==223	Пакет, що надає доступ до деяких API Windows для Python.
python-dateutil==2.8.2	Dateutil - це бібліотека для роботи з датами та часом в Python, включаючи парсинг рядків, роботу з часовими зонами, арифметику дат та інше.
python-dotenv==1.0.0	Python-dotenv - це бібліотека, яка дозволяє Python програмам читати конфігураційні дані з файлів .env.
pytz==2022.1	Pytz - це бібліотека Python для роботи з часовими зонами.
pywin32==303	Pywin32 - це набір розширень Python для Windows, який надає доступ до багатьох API Windows.
PyYAML==5.4.1	PyYAML - це бібліотека Python для аналізу та генерації YAML даних.
requests==2.26.0	Requests - це бібліотека Python для виконання HTTP запитів. Вона використовується для взаємодії з веб-сервісами.
semantic-version==2.8.5	Бібліотека для роботи з версіями відповідно до специфікації Semantic Versioning.
six==1.14.0	Six - це бібліотека Python, яка пропонує прості утиліти для забезпечення сумісності коду Python 2/3.
sqlparse==0.4.2	Sqlparse - це бібліотека для аналізу SQL.

Продовження таблиці 2.1

toml==2.0.1	Toml - це бібліотека для аналізу та генерації даних TOML.
tomlkit==0.11.8	Tomlkit - це альтернативна бібліотека для аналізу та генерації даних TOML, зосереджена на збереженні відступів та коментарів.
tzdata==2022.1	Tzdata - це бібліотека, що надає дані про часові зони.
urllib3==1.26.9	Urllib3 - це бібліотека Python для виконання HTTP запитів. Вона надає багато особливостей, таких як пул-конекторів, автоматичне перенаправлення та обробка помилок.

2.3.2 Структура модулів веб-сервісу

Сучасні версії фреймворку Django надають можливість розбивати веб-застосунок на окремі модулі у процесі розробки. Кожен з таких модулів призначений для групування функцій та методів, що відносяться до обробки окремих структурних задач. У таблиці 2.2 буде наведено перелік існуючих у проєкті модулів та їх короткий опис.

Таблиця 2.2 – Модулі додатку

№ п/п	Назва модулю	Опис
1	FunnyBusiness	Основний модуль додатку який відповідає за налаштування всього проєкту. Не містить класів або функцій але регулює роботу усього сервісу.
2	Customers	Містить класи які пов'язані з користувачами сервісу та функції реєстрації, авторизації та взаємодії з власним кабінетом користувача.

Продовження таблиці 2.2

№ п/п	Назва модулю	Опис
3	Main	Модуль з основним функціоналом сайту. У ньому прописано такі класи як: відгук, категорія, товар та рейтинг. Містить у собі функції відображення товарів у каталозі, відображення додаткової інформації, механізму створення нових товарів, виведення інформації про конкретний продукт на сторінку продукту, механізму редагування товарів, механізму видалення товарів, механізму покупки товарів, механізму завантаження файлів, процесу створення відгуку про товар та механізму пошуку товару.
4	Services	Модуль, що відповідає за роботу внутрішніх сервісів додатку. Містить у собі компоненти для інтеграції інтерфейсу прикладного програмування який використовує технологію блокчейн

Завдяки специфіці розробки додатків на базі фреймворку Django модулі, що відповідають за відображення та роботу сайту поділені на класи які одночасно виступають макетами для таблиць у БД та методи, що відповідають за основні функції.

Розглянемо більш детально модуль Customers, що зображено на ристунку 2.4.

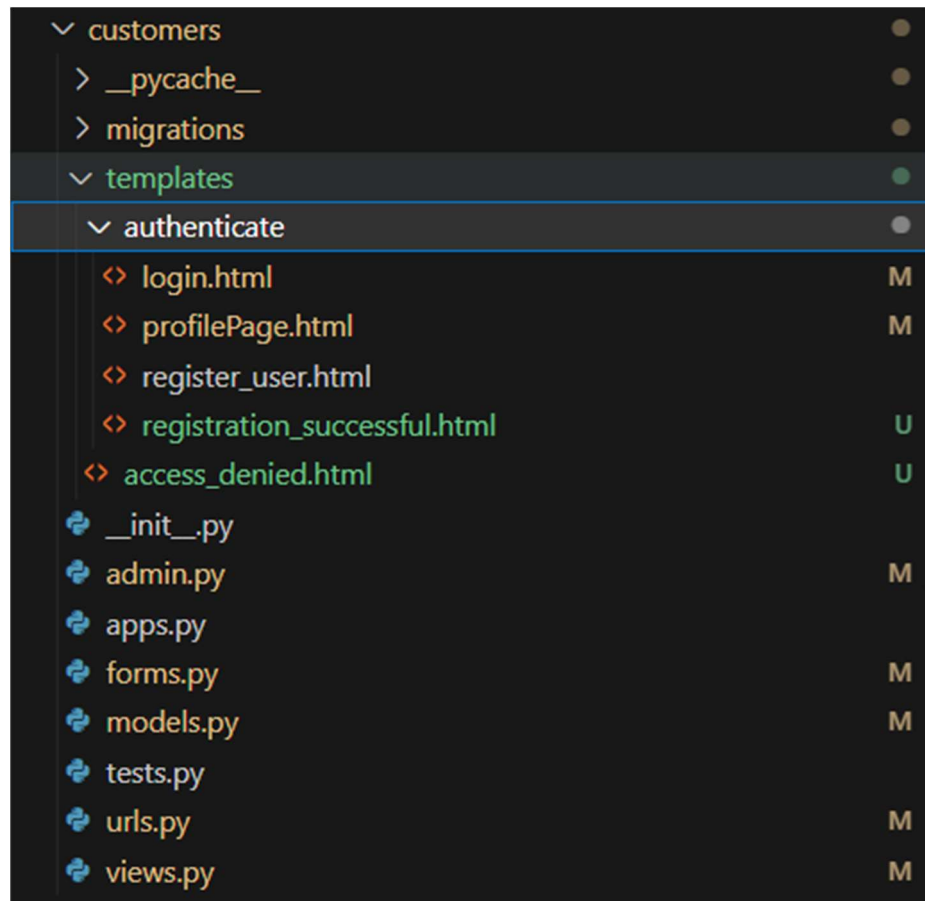


Рисунок 2.4 – Зміст модулю Customers

У файлі models.py реєструються моделі додатку у вигляді класів. Даний файл містить у собі допоміжний клас користувача.

Таблиця 2.3 – Опис класу CustomUser

Назва поля класу	Опис
ID	ID для зв'язку з класом користувача.
blockchainPublicKey	Зберігає публічний ключ користувача.

У файлі views.py створюються методи додатку. Даний файл містить у собі методи для роботи з авторизацією, реєстрацією та власним кабінетом користувача.

Таблиця 2.4 – Опис методів модулю Customers

Назва методу	Опис
profile_page	Відповідає за відображення товарів у власному кабінеті користувача
login_user	Надає користувачу можливість заходити в обліковий запис.
logout_user	Надає користувачу можливість виходити з облікового запису.
register_user	Відповідає за створення облікового запису користувачем.
registration_success	Надає користувачеві його ключі для збереження

Наступним варто розглянути найбільший модуль додатку Main. Його структуру та наповнення можна побачити на рисунку 2.5

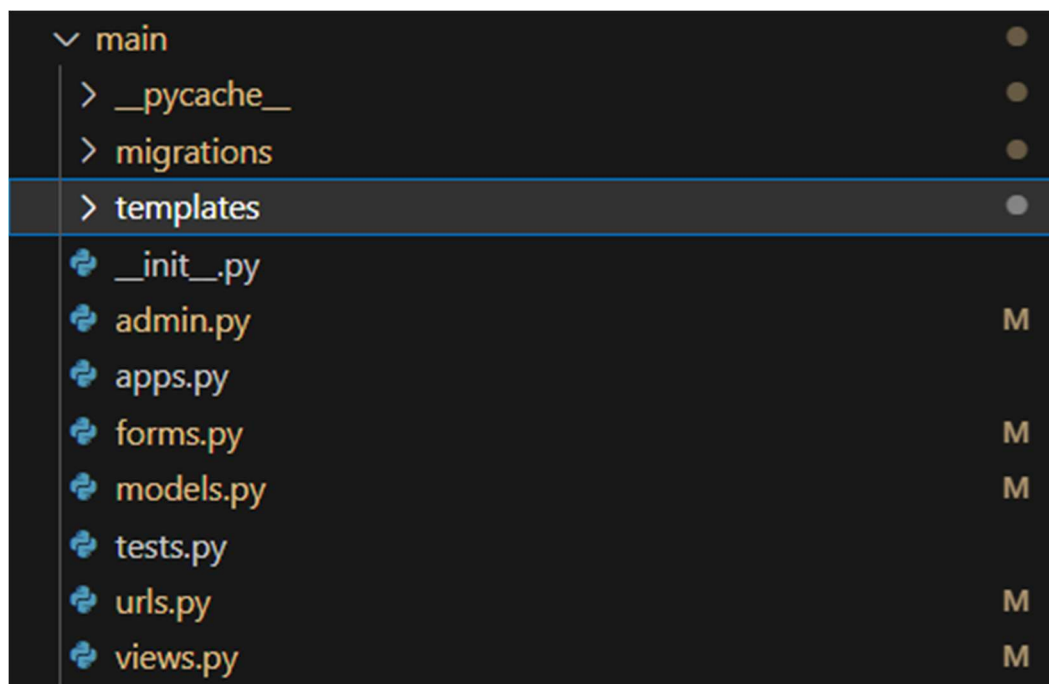


Рисунок 2.5 – Зміст модулю Main

У файлі models.py реєструються моделі додатку у вигляді класів. Даний файл містить у собі класи пов'язані з товарами та відгуками. У таблицях 2.5-2.8 наведено перелік та опис основних класів модулю Main.

Таблиця 2.5 – Опис класу Category

Назва поля класу	Опис
ID	ID категорії.
name	Назва категорії.

Таблиця 2.6 – Опис класу Raiting

Назва поля класу	Опис
ID	ID оцінки.
name	Назва оцінки.

Таблиця 2.7 – Опис класу Review

Назва поля класу	Опис
ID	ID відгуку.
comments	Коментарі користувача при створенні відгуку.
product_raiting	Вибір оцінки для товару.
autor_raiting	Вибір оцінки для продавця.
target	Вказівник на товар який оцінив користувач

Таблиця 2.8 – Опис класу Products

Назва поля класу	Опис
ID	ID товару.
name	Назва товару.
description	Опис товару.
price	Ціна товару.
isAvailable	Булевий вказівник на доступність товару.
requested	Булевий вказівник на те, чи існує запит на такий товар.
category	Категорія до якої належить товар.
img	Зображення товару.

Продовження таблиці 2.8

Назва поля класу	Опис
owner	Власник товару.
futureOwner	Інформація про майбутнього власника.
fileEntityHashSSH256	Хеш файлу збереженого у товарі.
fileEntity	Поле для завантаження файлу товару.

У файлі views.py створюються методи додатку. Даний файл містить у собі методи для роботи з продуктами. Для демонстрації їх ролі у проєкті було наведено таблицю 2.9 з описом та назвами основних функцій розглянутого модулю.

Таблиця 2.9 – Опис методів модулю Main

Назва методу	Опис
index	Відображає товари на сторінці каталогу
about	Відображає контент на сторінці опису функцій сервісу.
addProduct	Надає можливість виставити товар на продаж заповнивши всі необхідні поля та завантаживши файл. А також допомагає зареєструвати його в блокчейні.
viewProduct	Відображає власну сторінку товару з усіма його характеристиками.
deleteProduct	Надає можливість видаляти товар у власному кабінеті.
buyProduct	Відповідає за можливість здійснювати передачу товару.
downloadFile	Створює посилання для завантаження файлу з хмарного сховища.

Продовження таблиці 2.9

Назва методу	Опис
rateProduct	Надає користувачу можливість створити відгук на товар та оцінити продавця.
searchProduct	Допомагає використовувати пошук товарів у відповідному полі за назвою товару або категорії.
acceptRequest	Дозволяє власнику товару прийняти запит на його покупку.

Далі буде більш детально розглянуто модуль застосунку для здійснення операцій зв'язку з програмним інтерфейсом, обробки запитів та здійснення операцій. Його можна побачити на рисунку 2.6.



Рисунок 2.6 – Зміст модулю Services

Цей модуль сильно відрізняється від попередніх оскільки він призначений для роботи з надсиланням та обробленням запитів. Його функції та класи призначені для здійснення внутрішньої взаємодії і залишаються прихованими для користувачів додатку. Кожен сервіс має специфічне наповнення призначене для виконання допоміжних функцій у процесі роботи основних операцій застосунку тому у таблиці 2.10 буде описано призначення сервісів, а у таблиці 2.11 основні методи та класи які вони включають.

Таблиця 2.10 – Опис модулю Services

Назва сервісу	Призначення
blockChainRegisterProperty.py	Цей сервіс використовується при виконання операцій реєстрації нових товарів на сайті.
blockChainService.py	Призначений для завершення реєстрації облікового запису користувача.
blockChainTransferService.py	Використовується під час здійснення перевірки та передачі права власності на товар у момент його продажу.
formatChecker.py	Призначений для перевірки поля файлу у формі створення товару.
privateKeyValidator.py	Основною метою цього сервісу є перевірка ключа який надає користувач.

Таблиця 2.11 – Основні методи та класи модулю Services

Назва компоненту	Вид компоненту	Загальний опис роботи
registerProperty	Метод	Передає інформацію про публічний та приватний ключ користувача і хеш файлу для перевірки товару на унікальність і реєстрації його власника у блокчейні. Повертає інформацію про успішність операції.
create_new_user	Метод	Викликається для створення запиту на отримання ключів для нового користувача. Повертає отримані від програмного інтерфейсу приватний та публічний ключ користувача.

Продовження таблиці 2.11

Назва компоненту	Вид компоненту	Загальний опис роботи
transferProperty	Метод	Передає інформацію про приватний та публічний ключ власника товару, публічний ключ користувача, який бажає стати новим власником та хеш товару. Повертає інформацію про успішність операції здійснення передачі права власності.
ContentTypeRestrictedFileField	Клас	Містить у собі налаштування для обмеження розміру та типів файлів, які користувачі бажають завантажити у вигляді товару.
is_valid_rsa_key	Метод	Первинна перевірка ключів, що здійснюється до їх передачі для перевірки у блокчейні.

В таблицях 2.12 – 2.14 наведено структуру найважливіших таблиць які знаходяться у базі даних SQLite.

Таблиця 2.12 – Опис таблиці Products

Назва таблиці	Назва поля	Тип даних	Опис
products	ID	ID	ID товару
	name	varchar(100)	Назва товару
	description	varchar(400)	Опис товару
	category_id	integer	ID категорії до якої належить товар

Продовження таблиці 2.12

Назва таблиці	Назва поля	Тип даних	Опис
products	img	varchar(100)	Посилання на зображення товару
	isAvailable	bool	Вказівник на доступність товару для продажу
	price	integer	Ціна товару
	owner_id	integer	ID користувача якому належить цей товар.
	futureOwner	integer	ID користувача який залишив заявку на придбання товару.
	fileEntity	varchar(100)	Посилання для завантаження файлу товару.
	requested	bool	Вказівник на існування запиту для покупки
	fileEntityHashSSH256	varchar(400)	Зберігає хеш файлу

Таблиця 2.13 – Опис таблиці Review

Назва таблиці	Назва поля	Тип даних	Опис
review	ID	ID	ID відгуку
	comments	varchar(400)	Текст відгуку
	target	integer	Товар на який залишають відгук

Продовження таблиці 2.13

Назва таблиці	Назва поля	Тип даних	Опис
	product_raiting_id	integer	ID який вказує на оцінку товару
	autor_raiting_id	integer	ID який вказує на оцінку продавця

Таблиця 2.14 – Опис таблиці User

Назва таблиці	Назва поля	Тип даних	Опис
user	ID	ID	ID користувача
	password	varchar(128)	Текст відгуку
	last_login	datetime	Дата і час коли користувач заходив в обліковий запис
	username	varchar(150)	Користувацьке ім'я
	first_name	varchar(150)	Ім'я користувача
	last_name	varchar(150)	Прізвище користувача
	email	varchar(254)	Адреса електронної пошти користувача
	is_staff	bool	Вказівник на наявність розширених прав
	is_active	bool	Вказівник на те чи є цей запис активним
	date_joined	datetime	Дата і час коли користувач створив обліковий запис.

Для взаємодії з користувачем було створено окремі HTML сторінки з різним наповненням. У таблиці 2.15 буде наведено опис наповнення основних шаблонів користувацького інтерфейсу розробленого додатку.

Таблиця 2.15 – Шаблони інтерфейсів для взаємодії з користувачем

№ п/п	Назва сторінки	Опис
1	login.html	Сторінка де користувач може ввести логін та пароль у відповідні поля для авторизації на сайті.
2	profilePage.html	Сторінка профілю користувача. На ній відображено таблицю з товарами котрими володіє користувач. Тут відображено ті товари які він виставив на продаж та товари які він придбав. Поряд з кожним товаром є кнопки для редагування інформації про товар, завантаження файлу чи видалення товару. Також у випадку існування запиту на придбання товару буде вказано поле та кнопка для підтвердження цієї дії. Також тут відображено список товарів у статусі очікування на підтвердження операції.
3	register_user.html	Це сторінка де користувач проходить реєстрацію на сайті. Перед користувачем буде відображено поля які він має заповнити і інструкції до них. Знизу знаходиться кнопка для завершення реєстрації.
4	about.html	Сторінка де відображено текстову інформацію про сайт.

Продовження таблиці 2.15

№ п/п	Назва сторінки	Опис
5	addProduct.html	На цій сторінці користувач може виставити товар на продаж. Після заповнення полів і завантаження файлу користувач може натиснути кнопку в кінці форми і виставити товар на продаж або додати його до кабінету.
6	index.html	Сторінка каталогу товарів. На ній відображено доступні для придбання товари. Поряд з кожним товаром знаходяться кнопки для переходу на сторінку самого товару або для його придбання.
7	rateProduct.html	Ця сторінка використовується для залишення відгуку про товар та продавця. На ній є поля для коментаря користувача та два поля де можна обрати оцінку продавця та товару.
8	searchProduct.html	Сторінка для відображення контенту який шукає користувач. Після пошуку на екрані з'являється перелік товарі які задовольняють умови пошуку і кнопки для покупки або переходу на сторінку товару.
9	viewProduct.html	На цій сторінці відображено всю інформацію про товар та наявні кнопки для покупки товару та створення відгуку на товар.

Для розробки веб-сервісу також було використано набір програмних утиліт, що полегшили та прискорили процес створення програми. Для їх переліку та опису буде наведено таблицю 2.16

Таблиця 2.16 – Опис програмних утиліт

№ п/п	Назва утиліти	Опис застосування
1	Visual Studio Code	Редактор програмного коду який було використано для розробки основної частини веб-додатку.
2	DB Browser for SQLite	Програмне забезпечення для перегляду та редагування таблиць у базі даних SQLite.
3	Docker	Програмне забезпечення для автоматичного розгортання та управління додатками з підтримкою контейнеризації.
4	Swagger	Набір відкритих інструментів, призначених для проектування, побудови, документування та використання RESTful веб-сервісів.

2.4 Аналіз безпеки даних

Безпека даних відіграє критичну роль у розробці веб-сервісів, що здійснюють торгові операції. У розробленому веб-сервісі було використано

фреймворк Django, що має багато вбудованих інструментів для забезпечення безпеки даних.

Django є одним з найбезпечніших веб-фреймворків і має вбудовані механізми захисту від багатьох поширених атак. Наприклад, захист від Cross Site Scripting (XSS) перешкоджає виконанню небажаного скрипта на стороні клієнта, що може викрасти конфіденційну інформацію користувача. Cross Site Request Forgery (CSRF) захист перешкоджає атакам, що намагаються змусити клієнта виконати небажані дії на веб-сайті, поки він автентифікований [22]. Захист від SQL Injection допомагає відбивати атаки, спрямовані на маніпулювання SQL запитами, що можуть викривити, змінити або видалити дані.

Отримана в результаті розробки система зберігає дані в базі даних SQLite, відомій своєю надійністю та ефективністю управління даними. Ми використовуємо додаткові міри безпеки для обмеження доступу до критичних полів моделей. Наприклад, ми стремимосся мінімізувати кількість даних, які ми зберігаємо в базі даних, щоб знизити ризик витоку важливої інформації.

Ми не зберігаємо приватні ключі користувачів в нашому сервісі. Це дає нам можливість гарантувати, що навіть у випадку вдалої атаки на нашу систему, приватні дані користувачів залишатимуться в безпеці. Така політика дозволяє нам забезпечити безпеку персональних даних наших користувачів, незалежно від того, які транзакції вони проводять.

Висновки до розділу

В цьому розділі дипломного проєкту було проведено аналіз та висвітлено ключові аспекти розробки, виявлені під час процесу.

У цьому розділі було детально описано структуру програмного рішення, включаючи всі ключові модулі та компоненти, які сприяють його ефективності. Розроблене програмне рішення спрямоване на забезпечення максимальної продуктивності та надійності, використовуючи передові технології.

Основні модулі розробленого програмного рішення були аналізовані та описані для надання чіткого розуміння їх функцій та взаємодії. Найважливіші класи та методи, які використовуються в проекті, були ретельно проаналізовані, їх ролі та важливість у загальному контексті програмного рішення були розглянуті й описані у цьому розділі. Структура бази даних, яка використовується в цьому проекті, була детально аналізована для кращого розуміння її ролі та впливу на загальну продуктивність системи.

Також було проведено ретельний аналіз шаблону проектування, що використовується в розробленому веб-сервісі, щоб демонструвати його вплив на загальну архітектуру та зручність використання сервісу.

Ще одною важливою частиною проекту можна зазначити розроблені шаблони сторінок додатків, які були ретельно розглянуті та описані. Це допомагає визначити, як вони впливають на загальний користувацький досвід та взаємодію користувачів із системою.

3 АНАЛІЗ ЯКОСТІ ТА ТЕСТУВАННЯ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

3.1 Аналіз якості ПЗ

3.1.1 Статичний аналіз коду

Для аналізу структури коду веб-застосунку та його якості використовувався інструмент PyLint[21]. При цьому брались до уваги різні параметри, включаючи стиль написання коду, помилки і застереження, неоптимальні конструкції та наявність антипатернів.

Під час аналізу виявлені такі антипатерни:

- В деяких файлах відсутній останній рядок;
- Імпортування функцій і сервісів не відповідає рекомендованому порядку: імпорти власних модулів виконано перед імпортом модулів з поточного каталогу;
- В коді виявлено не використані імпорти;
- В деяких файлах наявний зайвий пробіл в кінці рядка.

Ці питання потребують вирішення для покращення якості коду.

Не зважаючи на наявність деяких шкідливих практик у коді, результат аналізу показав, що якість коду на високому рівні – 8.28 з 10 можливих балів. На рисунку 3.1 зображений результат статичного аналізу веб-застосунку з використанням PyLint.



Рисунок 3.1 – Результат статичного аналізу інструментом PyLint

3.1.2 Цикломатична складність коду

Одним з важливих показників якості програмного забезпечення є складність коду. Для проведення такого аналізу було вирішено провести тестування цикломатичної складності коду. Цей показник розраховується за

допомогою кількості лінійних незалежних шляхів через вихідний код. Проведення такого дослідження вдалося спростити за допомогою спеціального модулю для мови програмування Python – Radon.[21] Такий інструмент дозволяє проводити різні види тестування коду, але вирішено було провести з його допомогою перевірку складності. Результати цикломатичної складності мають шкалу від A до F, де A - це найпростіший код, а F - найскладніший. Результати роботи тестування можна побачити на рисунку 3.2

```
main\migrations\0004_alter_products_fileentity.py
  C 7:0 Migration - A
Services\blockChainRegisterProperty.py
  F 9:0 registerProperty - A
Services\blockChainService.py
  F 9:0 create_new_user - A
Services\blockChainTransferService.py
  F 8:0 transferProperty - A
Services\formatChecker.py
  C 6:0 ContentTypeRestrictedFileField - A
  M 26:4 ContentTypeRestrictedFileField.clean - A
  M 20:4 ContentTypeRestrictedFileField.__init__ - A
Services\privateKeyValidator.py
  F 7:0 is_valid_rsa_key - A

59976 blocks (classes, functions, methods) analyzed.
Average complexity: A (2.9230692276910766)
o (FunnyBusiness_venv) PS C:\Users\User\Documents\New Project\FunnyBusiness> |
```

Рисунок 3.2 – Результат цикломатичного аналізу інструментом Radon

Загальна оцінка складності показала, що розроблений код має високий результат простоти. Це означає, що він відповідає всім вимогам для забезпечення наступних параметрів:

Простота. Код з низькою цикломатичною складністю зазвичай легко читати та розуміти. Це сприяє більш швидкому засвоєнню коду новими членами команди та зменшує ймовірність непорозумінь або помилок.

Тестування: Низький показник цикломатичної складності коду показує, що такий додаток легко тестувати. Менше можливих шляхів через код означає, що ви маєте більше шансів покрити весь код юніт-тестами, що зменшує ймовірність помилок.

Рефакторинг і підтримка: Простий код зазвичай легше модифікувати та підтримувати. Коли код не містить складних умовних інструкцій або

вкладених циклів, зміни можна вносити з меншим ризиком непередбачених побічних ефектів.

Надійність: Код з низькою цикломатичною складністю зазвичай має менше потенційних місць для появи помилок, що сприяє створенню більш надійного програмного забезпечення.

Зазвичай оцінка А цикломатичної складності коду не може забезпечити підтвердження якості програмного забезпечення. Саме тому важливо проводити аналіз за допомогою різних метрик якості коду. Таким чином результат цієї перевірки підтвердив позитивну оцінку статичного аналізу коду.

3.2 Опис процесів тестування

Тестування є важливою складовою розробки будь-якого програмного забезпечення. Воно направлене на створення певних гарантій, що до правильності та надійності роботи усього застосунку. Для проведення такого тестування прийнято використовувати багато різних підходів та рівнів. Це допомагає провести повноцінну перевірку всіх аспектів розробленого рішення. Проте доцільність використання кожного з таких видів тестування залежить від принципів роботи, основних функцій та виду програмного забезпечення. Загалом вирізняють такі види тестування:

Юніт-тестування: Це найнижчий рівень тестування, який зосереджений на перевірці окремих структурних одиниць коду. Ними можуть бути функції або методи додатку. Мета юніт-тестування - виявити і виправити проблеми на рівні коду, перш ніж вони стануть проблемами на вищих рівнях.

Інтеграційне тестування: Цей тип тестування зосереджений на перевірці взаємодії між різними компонентами системи. Мета інтеграційного тестування - виявити проблеми, які виникають, коли окремі компоненти працюють разом.

Мануальне тестування: Для проведення такого виду тестування зазвичай використовуються люди, які активно використовують програмне забезпечення для знаходження помилок. Мануальне тестування може бути

неструктурованим або структурованим. Неструктурованим мануальним тестуванням називають звичайне користування функціями розробленого програмного забезпечення, в той час як для структурованого тестування створюють окремі сценарії.

Автоматичне тестування: Всі види автоматичного тестування виконуються програмами. Юніт-тести, інтеграційні тести та деякі системні тести можуть бути автоматизованими.

За результатом аналізу розробленого програмного рішення було вирішено використовувати юніт-тести та мануальне тестування. Через специфіку роботи застосунку та вимоги, що були висунуті до його роботи, не виявлено необхідності у застосуванні інших видів тестів.

3.2.1 Юніт-тестування

Юніт-тестування - це метод тестування програмного забезпечення, при якому окремі частини програми, зазвичай її функції або методи, перевіряються на правильність роботи. Основна мета юніт-тестування - ізолювати окремі частини програми та показати, що ці окремі частини працюють належним чином.

Через розбиття проєкту на окремі модулі тестування проводилося окремо для кожного модулю. Для написання таких тестів використовувалися внутрішні ресурси для тестування від Django та Python. Були використані декоратори та заміщувачі класів для повноцінної перевірки коректності роботи атомарних частин програми. Це допомогло ізолювати частину системи для тестування, та перевірити взаємодію між її компонентами.

На рисунку 3.3 можна побачити декілька юніт тестів для модулю Customers який відповідає за взаємодію з користувачем

```

class TestLoginUserView(TestCase):
    def setUp(self):
        self.client = Client()
        self.user = User.objects.create_user(username='testuser', password='12345')
        self.login_url = reverse('LogIn')

    def test_login_user_success(self):
        response = self.client.post(self.login_url, {'username': 'testuser', 'password': '12345'})
        self.assertRedirects(response, reverse('Hub'))
        self.assertEqual(int(self.client.session['_auth_user_id']), self.user.pk)

    def test_login_user_failure(self):
        response = self.client.post(self.login_url, {'username': 'wronguser', 'password': 'wrongpassword'})
        self.assertRedirects(response, reverse('LogIn'))
        self.assertNotIn('_auth_user_id', self.client.session)
        messages = list(get_messages(response.wsgi_request))
        self.assertEqual(str(messages[0]), "Not possible to log in")

    def test_login_user_get(self):
        response = self.client.get(self.login_url)
        self.assertEqual(response.status_code, 200)
        self.assertTemplateUsed(response, 'authenticate/login.html')

```

Рисунок 3.3 – Один з класів для тестування модулю Customers

На цьому рисунку було наведено клас який потрібен для тестування компонентів функції логування користувача на сервісі. Він включає в себе перевірки 3 тест-кейсів з певним очікуваним результатом та частину для підготовки необхідних компонентів які будуть використані у процесі тестування.

Далі буде наведено приклад тестування модулю Main де знаходяться основні функції для взаємодії з товарами та віртуальними об'єктами. Фрагмент його тестування можна побачити на рисунку 3.4

```

You, 1 hour ago | 1 author (You)
class AddProductTest(TestCase):
    def setUp(self):
        self.factory = RequestFactory()
        self.user = CustomUser.objects.create_user(username='testuser', password='12345')
        self.category = Category.objects.create(name='Test Category')

    @patch('main.views.hashlib.sha256')
    @patch('main.views.registerProperty')
    def test_add_product(self, mock_registerProperty, mock_hashlib):

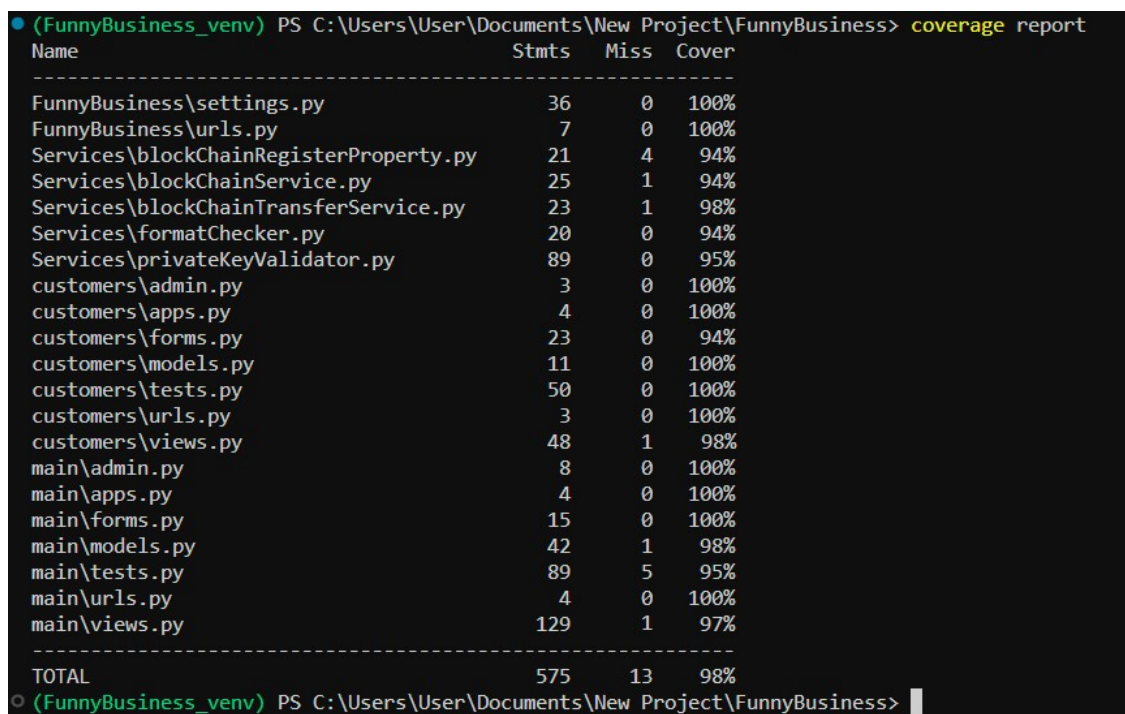
        file_data = BytesIO(b'some file data')
        file_data.name = 'myfile.txt'
        data = {
            'product_private_key': 'some private key',
            'name': 'Test product',
            'description': 'Test description',
            'price': 99,
            'isAvailable': True,
            'category': self.category.id,
            'img': file_data,
            'fileEntity': file_data
        }
        form = ProductForm(data, {'img': file_data, 'fileEntity': file_data})

```

Рисунок 3.4 – Фрагмент тестування модулю Main

На цьому рисунку можна побачити проведення тесту для перевірки додавання товару. Це одна з найважливіших функцій застосунку тому для її перевірки було створено декілька тестів, один з яких можна побачити на малюнку.

Також одним з важливих показників ефективності проведеного юніт тестування є покриття програмного коду тестами. Для перевірки такого показника було отримано процентне покриття основних програмних файлів з різних модулів та наведена загальне покриття у процентному співвідношенні. Результати такої оцінки тестування можна побачити на рисунку 3.5



```

(FunnyBusiness_venv) PS C:\Users\User\Documents\New Project\FunnyBusiness> coverage report
Name                               Stmts  Miss  Cover
-----
FunnyBusiness\settings.py          36     0   100%
FunnyBusiness\urls.py              7     0   100%
Services\blockChainRegisterProperty.py 21     4    94%
Services\blockChainService.py       25     1    94%
Services\blockChainTransferService.py 23     1    98%
Services\formatChecker.py           20     0    94%
Services\privateKeyValidator.py      89     0    95%
customers\admin.py                  3     0   100%
customers\apps.py                   4     0   100%
customers\forms.py                  23     0    94%
customers\models.py                 11     0   100%
customers\tests.py                  50     0   100%
customers\urls.py                   3     0   100%
customers\views.py                  48     1    98%
main\admin.py                       8     0   100%
main\apps.py                        4     0   100%
main\forms.py                       15     0   100%
main\models.py                      42     1    98%
main\tests.py                       89     5    95%
main\urls.py                        4     0   100%
main\views.py                       129    1    97%
-----
TOTAL                               575    13    98%
(FunnyBusiness_venv) PS C:\Users\User\Documents\New Project\FunnyBusiness>

```

Рисунок 3.5 – Проведення тестів та їх загальне покриття

Покриття додатку тестами знаходиться на високому рівні і забезпечує перевірку усіх важливих функцій застосунку. Це дозволить швидко та ефективно проводити локалізацію та усунення помилок при майбутній розробці або внесенні змін.

3.2.2 Мануальне тестування

Проведення структурованого мануального тестування відбувалося за складеними сценаріями та передбачало виконання користувачем дій для перевірки основного функціоналу застосунку. Результати виконання

мануального тестування веб-застосунку та опис відповідних тестів наведено у таблицях 3.1 – 3.5.

Таблиця 3.1 – Тест реєстрації користувача

Тест	Реєстрація користувача
Модуль	Реєстрація користувача
Номер тесту	1.1
Початковий стан системи	Користувач знаходиться на сторінці реєстрації
Вхідні данні	Електронна пошта, пароль, підтвердження паролю
Опис проведення тесту	У відповідні поля вводяться: коректна електронна пошта, яка до цього не була зареєстрована в системі, пароль від 10 до 64 символів, який містить хоча б з одну англійську літеру, одне число і один спеціальний символ, і який не входить у топ 10000 найпопулярніших паролей, підтвердження паролю, яке співпадає з раніше введеним паролем. Після цього натискається кнопка підтвердження реєстрації.
Очікуваний результат	Реєстрація проходить успішно, дані користувача записується у базу даних і відбувається перехід на сторінку де можна отримати приватний ключ.
Фактичний результат	Реєстрація проходить успішно, користувач додається у систему і перенаправляється на сторінку з ключами.

Таблиця 3.2 – Тест виставлення товару на продаж

Тест	Виставлення товару на продаж
Модуль	Операції з товаром
Номер тесту	2.1
Початковий стан системи	Користувач знаходиться на створення товару
Вхідні данні	Назва товару, його опис, ціна товару, зображення товару, файл товару та ключ користувача
Опис проведення тесту	У відповідні поля вводяться: Назва товару, його опис та ціна товару. Також потрібно обрати категорію товару з переліку, завантажити та обробити фото товару, додати у відповідне поле файл який буде відображати цей товар та вказати ключ користувача. Після цієї операцій натискається кнопка збереження товару.
Очікуваний результат	Додавання товару проходить успішно, дані товару записуються до бази даних і відбувається перехід до сторінки каталогу.
Фактичний результат	Операція проходить успішно, товар додається у систему та виставляється на сторінку каталогу і у профіль користувача. Користувач переадресовується на сторінку каталогу.

Таблиця 3.3 – Тест залишення заявки на придбання товару

Тест	Створення заявки на придбання товару
Модуль	Операції з товаром
Номер тесту	2.2
Початковий стан системи	Користувач знаходиться на сторінці каталогу
Вхідні данні	Відсутні
Опис проведення тесту	Користувач переходить на сторінку товару або натискає кнопку “Придбати” на сторінці каталогу.
Очікуваний результат	Заявка реєструється в системі. У профілі користувача відображається статус очікування підтвердження продажу. Користувач переходить на сторінку з подальшими інструкціями.
Фактичний результат	Заявка зареєструвалась та записалась у сервісі. Користувач може побачити статус товару у власному кабінеті. Відбувається переадресація на сторінку з інструкціями.

Таблиця 3.4 – Тест проведення підтвердження заявки на продаж

Тест	Підтвердження заявки на продаж
Модуль	Операції з товаром
Номер тесту	2.3
Початковий стан системи	Користувач знаходиться на сторінці каталогу

Продовження таблиці 3.4

Вхідні данні	Приватний ключ користувача
Опис проведення тесту	Користувач переходить на сторінку власного кабінету. Обирає товар який очікує підтвердження та вписує свій приватний ключ у відповідне поле. Після введення ключа користувач натискає кнопку підтвердження і товар зникає з списку його власності.
Очікуваний результат	Операція проходить успішно. Замовник отримує товар, а користувач, що підтверджує операцію бачить зникнення товару з списку його власностей.
Фактичний результат	Операція проходить успішно. Товар з'являється на сторінці профілю замовника і зникає на такій самій сторінці користувача.

Таблиця 3.5 – Тест пошуку товарів на сайті

Тест	Пошук товарів
Модуль	Пошук
Номер тесту	3.1
Початковий стан системи	Користувач знаходиться на будь-якій сторінці застосунку
Вхідні данні	Назва або категорія товару, повна або часткова.
Опис проведення тесту	Користувач вводить дані у поле пошуку яке відображається на усіх сторінках товару. Після натискання клавіші “Пошук” його переадресовує на сторінку з усіма результатами, що задовольняють результати пошуку.

Продовження таблиці 3.4

Очікуваний результат	Користувач успішно проводить пошук товарів та опиняється на сторінці з усіма потрібними результатами.
Фактичний результат	Пошук проводиться успішно. Користувача перенаправляє на сторінку де відображено усі товари за результатом пошуку.

Висновки до розділу

Розділ, присвячений аналізу якості програмного забезпечення, охоплює ретельний огляд структури коду і процесів тестування. Особливу увагу було приділено виявленню та усуненню антипатернів у коді, що сприяло покращенню якості кінцевого продукту.

Структурний аналіз коду допоміг у виявленні антипатернів та їх виправленні, що призвело до значного покращення якості коду та стабільності веб-сервісу. За допомогою цього аналізу, було можливо зробити код більш читабельним, ефективним, і, як наслідок, простішим для подальшої розробки та супроводу. Також було проведено аналіз цикломатичної складності коду та наведено підтвердження ефективності роботи та відсутності проблем пов'язаних з недостатньою простотою і швидкістю виконання важливих процесів.

У рамках цього розділу також було проведено тестування програмного забезпечення. Використовуючи мануальне тестування, було проведено глибокий аналіз основних функцій та модулів веб-сервісу, а юніт тестування дозволило забезпечити правильність роботи та стійкість до помилок. Цей процес дозволив виявити та виправити потенційні помилки та недоліки.

Загалом, ці міри забезпечили високу якість розробленого програмного забезпечення. Вони підтверджують, що код веб-сервісу відповідає стандартам якості, що в свою чергу гарантує стабільну та надійну роботу веб-сервісу.

ВИСНОВКИ

У результаті виконання обраної теми дипломної роботи було розроблено веб-сервіс для продажу віртуальних об'єктів, зокрема, різних типів файлів і віртуальних даних. Дослідження було здійснено згідно зі сучасними тенденціями в галузі наукових та технічних знань, що дозволило створити свій внесок велику кількість сучасних та актуальних областей життя людини.

Впровадження результатів цієї роботи може знайти застосування в різних сферах, включаючи торгівлю цифровими товарами, інтелектуальною власністю та навіть віртуальними товари в сфері ігор. Все це підтверджує високу науково-технічну та соціально-економічну значущість роботи.

Застосовуючи мову програмування Python сучасний веб-фреймворк Django і базу даних SQLite, було створено високоякісне програмне забезпечення. Розроблені компоненти веб-сервісу проявили стабільність і гнучкість в роботі. За допомогою використання інструментів розробки фронт-енд частини вдалося створити зручний у використанні та інтуїтивно зрозумілий інтерфейс користувача, що значно спростило процес використання сервісу.

У процесі розробки вдалося досягнути результатів у реалізації всіх обраних на стадії затвердження вимог та забезпечити багато покращень ідей яких виникли безпосередньо під час створення програмного рішення. Окрім забезпечення необхідного функціоналу вдалося впровадити багато інструментів, що спрощують взаємодію користувача з системою. Одним з найбільш значущих та складних прикладів таких інструментів можна назвати функцію обробки зображень через інтерфейс сайту.

В результаті роботи було отримано веб-застосунок для продажу права власності на віртуальні об'єкти, що здійснює взаємодію зі стороннім сервісом і задовольняє сучасні потреби користувачів мережі та детальний опис усіх рішень та технічних особливостей і підходів до його розробки.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Gallagher J. M., Ramanathan S. C. Choosing a Client/Server Architecture. *Information Systems Management*. 1996. Т. 13, № 2. С. 7–13. URL: <https://doi.org/10.1080/10580539608906981> (дата звернення: 11.05.2023).
2. Microservice Architecture using ASP.NET Core. *International Journal of Innovative Technology and Exploring Engineering*. 2020. Т. 9, № 8. С. 175–178. URL: <https://doi.org/10.35940/ijitee.h6247.069820> (дата звернення: 12.04.2023).
3. Penberthy W., Roberts S. S3 Object Storage. *Pro .NET on Amazon Web Services*. Berkeley, CA, 2022. С. 269–301. URL: https://doi.org/10.1007/978-1-4842-8907-5_7 (дата звернення: 31.05.2023).
4. Lichtendahl K. C., Andrasko J., Boatright B. Google Cloud Platform: Cloud Storage. *SSRN Electronic Journal*. 2022. URL: <https://doi.org/10.2139/ssrn.4133739> (дата звернення: 03.06.2023).
5. Azure Storage / J. Soh та ін. *Microsoft Azure*. Berkeley, CA, 2020. С. 271–291. URL: https://doi.org/10.1007/978-1-4842-5958-0_14 (дата звернення: 14.05.2023).
6. S G., Dalvi D., Tandel M. Bootstrap and Django Framework. *International Journal of Advanced Research in Science, Communication and Technology*. 2021. С. 130–133. URL: <https://doi.org/10.48175/ijarsct-2158> (дата звернення: 08.05.2023).
7. Kelly S. Model View Controller. *Python, PyGame and Raspberry Pi Game Development*. Berkeley, CA, 2016. С. 141–151. URL: https://doi.org/10.1007/978-1-4842-2517-2_17 (дата звернення: 04.03.2023).
8. Doherty M. HTML. *Computers and Composition*. 1996. Т. 13, № 3. С. 340–341. URL: [https://doi.org/10.1016/s8755-4615\(96\)90022-5](https://doi.org/10.1016/s8755-4615(96)90022-5) (дата звернення: 10.04.2023).

9. Python / R. Toal та ін. *Programming Language Explorations*. 2016. С. 59–76. URL: <https://doi.org/10.1201/9781315314334-4> (дата звернення: 01.01.2023).

10. Sharma V. Object Detection and Recognition using Amazon Rekognition with Boto3. 2022 *6th International Conference on Trends in Electronics and Informatics (ICOEI)*, м. Tirunelveli, India, 28–30 квіт. 2022 р. 2022. URL: <https://doi.org/10.1109/icoei53556.2022.9776884> (дата звернення: 07.05.2023).

11. Django-storages documentation. *django-storages* 1.13.2. URL: <https://django-storages.readthedocs.io/en/latest/> (дата звернення: 01.06.2023).

12. McCormick E., De Volder K. JQuery. *Companion to the 19th annual ACM SIGPLAN conference*, м. Vancouver, BC, CANADA, 24–28 жовт. 2004 р. New York, New York, USA, 2004. URL: <https://doi.org/10.1145/1028664.1028670> (дата звернення: 16.03.2023).

13. Queinnec C. Javascript. *Technologies logicielles Architectures des systèmes*. 2017. URL: <https://doi.org/10.51257/a-v2-h3120> (дата звернення: 09.02.2023).

14. Pillow. *Pillow* (PIL Fork). URL: <https://pillow.readthedocs.io/en/stable/> (дата звернення: 14.04.2023).

15. Http.client HTTP protocol client. *Python documentation*. URL: <https://docs.python.org/3/library/http.client.html> (дата звернення: 30.03.2023).

16. Clow M. Visual Studio Code. *Angular 5 Projects*. Berkeley, CA, 2018. С. 57–68. URL: https://doi.org/10.1007/978-1-4842-3279-8_5 (дата звернення: 18.02.2023).

17. Insomnia Documentation. *Insomnia Docs*. URL: <https://docs.insomnia.rest> (дата звернення: 26.01.2023).

18. Swagger Documentation. *API Documentation & Design Tools for Teams | Swagger*. URL: <https://swagger.io/docs/> (дата звернення: 27.05.2023).

19. Docker Docs: How to build, share, and run applications. *Docker Documentation*. URL: <https://docs.docker.com> (дата звернення: 10.04.2023).

20. NFT. *Encyclopedic Reference of Genomics and Proteomics in Molecular Medicine*. С. 1286. URL: https://doi.org/10.1007/3-540-29623-9_304 (дата звернення: 16.03.2023).

21. Django Web Framework Software Metrics Measurement Using Radon and Pylint / S. Liawatimena та ін. *2018 Indonesian Association for Pattern Recognition International Conference (INAPR)*, м. Jakarta, Indonesia, 7–8 верес. 2018 р. 2018. URL: <https://doi.org/10.1109/inapr.2018.8627009> (дата звернення: 11.03.2023).

22. 2. Jaeyalakshmi M., Supriya T., Sweta S. Streamlining Departmental Data Management-A Django based website. *2023 7th International Conference on Intelligent Computing and Control Systems (ICICCS)*, м. Madurai, India, 17–19 трав. 2023 р. URL: <https://doi.org/10.1109/iciccs56967.2023.10142666> (дата звернення: 02.06.2023).

ДОДАТОК А ЗВІТ ПОДІБНОСТІ



Ім'я користувача:
Лісовиченко Олег Іванович

Дата перевірки:
15.06.2023 07:40:17 EEST

Дата звіту:
15.06.2023 07:43:49 EEST

ID перевірки:
1015607592

Тип перевірки:
Doc vs Internet + Library

ID користувача:
76913

Назва документа: IT-83_Василець_ПЗ

Кількість сторінок: 64 Кількість слів: 10963 Кількість символів: 84252 Розмір файлу: 3.45 MB ID файлу: 1015255627

6.33%
Схожість

Найбільша схожість: 1.89% з джерелом з Бібліотеки (ID файлу: 1015069764)

2.49% Джерела з Інтернету 127 Сторінка 66

6.26% Джерела з Бібліотеки 307 Сторінка 68

0% Цитат

Вилучення цитат вимкнене

Вилучення списку бібліографічних посилань вимкнене

0%
Вилучень

Немає вилучених джерел

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

ВЕБ-ЗАСТОСУНОК ПІДТРИМКИ ПРОЦЕСІВ ПРОДАЖУ

ВІРТУАЛЬНИХ ОБ’ЄКТІВ

Текст програми

КПІ.IT-8304.045440.02.12

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Максим ГОЛОВЧЕНКО

Нормоконтроль:

_____ Максим ГОЛОВЧЕНКО

Виконавець:

_____ Данило ВАСИЛЕЦЬ

Київ – 2023

Файл manage.py

```
#!/usr/bin/env python
"""Django's command-line utility for administrative tasks."""
import os
import sys

def main():
    """Run administrative tasks."""
    os.environ.setdefault('DJANGO_SETTINGS_MODULE', 'FunnyBusiness.settings')
    try:
        from django.core.management import execute_from_command_line
    except ImportError as exc:
        raise ImportError(
            "Couldn't import Django. Are you sure it's installed and "
            "available on your PYTHONPATH environment variable? Did you "
            "forget to activate a virtual environment?"
        ) from exc
    execute_from_command_line(sys.argv)

if __name__ == '__main__':
    main()
```

Файл models.py

```
from unicodedata import category, name
from django.db import models
from django.contrib.auth.models import User
from Services.formatChecker import ContentTypeRestrictedFileField
from django.core.validators import MaxValueValidator
from django.core.validators import MinValueValidator
from django.utils.safestring import mark_safe

class Category(models.Model):
    name = models.CharField('name', max_length= 10)

    def __str__(self):
        return self.name

class Raiting(models.Model):
    name = models.CharField('name', max_length= 10)

    def __str__(self):
        return self.name

class Review(models.Model):

    comments = models.TextField('comments', blank=False, max_length= 400)
```

					КПІ.IT-8304.045440.02.12	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

        product_raiting = models.ForeignKey(Raiting, related_name="product_raiting" ,
on_delete= models.CASCADE)
        autor_raiting = models.ForeignKey(Raiting, related_name="autor_raiting" ,
on_delete= models.CASCADE)
        target = models.IntegerField("Review Target", blank=False)
        class Meta:
            verbose_name = 'Review'
            verbose_name_plural = 'Reviews'

class Products(models.Model):

    name = models.CharField('name', max_length= 100)
    description = models.TextField('description', max_length= 400)
    price = models.PositiveIntegerField('price',
validators=[MaxValueValidator(9999999999),MinValueValidator(1)])
    isAvailable = models.BooleanField(default=True)
    requested = models.BooleanField(default=False) #New
    category = models.ForeignKey(Category, on_delete= models.CASCADE)
    img = models.ImageField(null=True, blank=True, upload_to="images/")
    owner = models.IntegerField("Product Owner", blank=False)
    futureOwner = models.IntegerField("Future Product Owner", blank=True,
null=True) #New
    fileEntity = ContentTypeRestrictedFileField(
        null=False,
        blank=False,
        upload_to="files/",
        max_upload_size=1719664640,
#1719664640 = 2 gb#
        content_types=['*/*']
    )
    # fileEntity = models.FileField(null=False, blank=False, upload_to="files/")
    fileEntityHashSSH256 = models.CharField('fileHash',default='default_value',
max_length= 400, null=False, blank=False)

    def img_tag(self):
        return mark_safe(''
% (self.img))

    def __str__(self):
        return self.name

    class Meta:
        verbose_name = 'Product'
        verbose_name_plural = 'Products'

```

Файл urls.py

```
from django.urls import path
```

					КПІ.IT-8304.045440.02.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		3

```

from . import views
from Services import newFunction
from Services import blockChainService

urlpatterns = [
    path('', views.index, name='Hub'),
    path('addProduct', views.addProduct, name='AddProduct'),
    path('about-fb', views.about, name='About'),
    path('viewProduct/<product_id>', views.viewProduct, name='ViewProduct'),
    path('editProduct/<product_id>', views.editProduct, name='EditProduct'),
    path('deleteProduct/<product_id>', views.deleteProduct,
name='DeleteProduct'),
    path('buyProduct/<product_id>', views.buyProduct, name='BuyProduct'),
    path('acceptRequest/<product_id>', views.acceptRequest,
name='AcceptRequest'),
    path('downloadFile/<product_id>', views.downloadFile, name='DownloadFile'),
    path('rateProduct/<product_id>', views.rateProduct, name='RateProduct'),
    path('searchProduct', views.searchProduct, name='SearchProduct'),
    path('newTest', blockChainService.create_new_user, name='NewTest')
]

```

Файл views.py

```

from multiprocessing import context
import os
from unicodedata import category
from django.shortcuts import render, redirect
from numpy import product
from requests import post
from io import BytesIO
from Services.blockChainRegisterProperty import registerProperty
from Services.blockChainTransferService import transferProperty
from Services.privateKeyValidator import is_valid_rsa_key

from customers.models import CustomUser
from .models import Category, Products, Review
from .forms import ProductForm, ReviewForm
from PIL import Image
from django.http import HttpResponseRedirect, FileResponse
from django.contrib import messages
from django.contrib.auth.models import User
import hashlib
import base64
from django.core.files.uploadedfile import TemporaryUploadedFile

def index(request):

```

					КПІ.IT-8304.045440.02.12	Арк.
Вмін.	Арк.	№ докум.	Підп.	Дата.		4

```

    products = Products.objects.all()
    category = Category.objects.all()
    return render(request, 'main/index.html', {'title': 'Hub',
'products': products, 'category': category})

def about(request):
    return render(request, 'main/about.html')

def viewProduct(request, product_id):
    product = Products.objects.get(pk=product_id)
    review = Review.objects.all()
    product_owner = CustomUser.objects.get(pk=product.owner)
    return render(request, 'main/viewProduct.html',
{
    'product': product,
    'product_owner': product_owner,
    'review': review,
})

def editProduct(request, product_id):
    product = Products.objects.get(pk=product_id)
    form = ProductForm(request.POST or None, request.FILES or None,
instance=product)
    if form.is_valid():
        form.save()
        return redirect ('ProfilePage')
    return render(request, 'main/editProduct.html', {'product': product, 'form':
form})

def deleteProduct(request, product_id):
    product = Products.objects.get(pk=product_id)
    if request.user.id == product.owner:
        product.delete()
        messages.success(request, ("Product Deleted"))
        return redirect ('ProfilePage')
    else:
        messages.success(request, ("0000oppss"))
        return redirect ('ProfilePage')

def buyProduct(request, product_id):
    product = Products.objects.get(pk=product_id)
    product.futureOwner = request.user.id
    product.isAvailable = False
    product.requested = True

```

```

product.save()

context = {
    'product': product
}

return render(request, 'buyProduct.html', context)

def downloadFile(request, product_id):
    product = Products.objects.get(pk=product_id)
    prod = product.fileEntity
    name = product.name

    return FileResponse(prod, as_attachment=True)

def rateProduct(request, product_id):
    error = ''
    if request.method == 'POST':
        product = Products.objects.get(pk=product_id)
        form = ReviewForm(request.POST, request.FILES)
        if form.is_valid():
            Review = form.save(commit=False)
            Review.target = product.id
            Review.save()
            return redirect('/')
        else:
            error = 'Incorrect Form'

    form = ReviewForm()
    context = {
        'form': form,
        'error': error
    }

    return render(request, 'main/rateProduct.html', context)

def searchProduct(request):

    if request.method == "POST":
        searched = request.POST["searched"]

        products = Products.objects.filter(name__contains=searched)
        products =
products.union(Products.objects.filter(category__name__contains=searched))

        context = {

```

					КПІ.IT-8304.045440.02.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		6

```

        'searched': searched,
        'products': products
    }

    return render(request, 'main/searchProduct.html', context)

else:
    return render(request, 'main/searchProduct.html')

def autorisation(request):
    return render(request, 'main/autorisation.html')

def acceptRequest(request, product_id):
    if request.method == "POST":
        private_key = request.POST.get('private_key')
        product = Products.objects.get(pk=product_id)

        if product.owner == request.user.id :
            currentOwner = CustomUser.objects.get(pk=product.owner)
            futureOwner= CustomUser.objects.get(pk=product.futureOwner)
            if is_valid_rsa_key(private_key) == True:
                data = transferProperty(currentOwner.blockchainPublicKey,
private_key, futureOwner.blockchainPublicKey, product.fileEntityHashSSH256)
                print(data)
                if data['isSuccess'] == True:

                    product.owner = product.futureOwner
                    product.futureOwner = None
                    product.requested = False
                    product.save()
                    return redirect ('ProfilePage')

            else:
                # messages.error(request, data['Payload'])
                messages.error(request, data['error'])
        else:
            messages.error(request, 'Invalid private key')

    else:
        messages.error(request, 'Invalid private key')

    return redirect ('ProfilePage')

def autorisation(request):
    return render(request, 'main/autorisation.html')

```

```

def addProduct(request):
    error = ''
    if request.method == 'POST':
        form = ProductForm(request.POST, request.FILES)
        private_key = request.POST.get('product_private_key')

        if form.is_valid():
            product = form.save(commit=False)
            product.owner = request.user.id

            # Read the file data from InMemoryUploadedFile
            file_data = request.FILES['fileEntity'].read()

            # Calculate the hash
            readable_hash = hashlib.sha256(file_data).digest()
            Hash = base64.b64encode(readable_hash).decode()

            # Save the hash to the product
            product.fileEntityHashSSH256 = Hash

            # Process the private key and register the property
            User = CustomUser.objects.get(pk=request.user.id)
            if is_valid_rsa_key(private_key):
                data = registerProperty(User.blockchainPublicKey, private_key,
Hash)
                if data['isSuccess']:
                    product.save()
                    return redirect('/')
                else:
                    messages.error(request, data['error'])
            else:
                messages.error(request, 'Invalid private key')
        else:
            error = 'Incorrect Form'
    else:
        form = ProductForm()

    context = {
        'form': form,
        'error': error
    }
    return render(request, 'main/addProduct.html', context)

```

Файл forms.py

```

from dataclasses import field
from .models import Products, Review
from django.forms import ModelForm, CheckboxInput, ClearableFileInput
from django.forms import TextInput, NumberInput, RadioSelect
from django.forms import Textarea, Select

```

					КПІ.IT-8304.045440.02.12	Арк. 8
Змін.	Арк.	№ докум.	Підп.	Дата.		

```

class ProductForm(ModelForm):
    class Meta:
        model = Products
        fields = ["name",
"description", "price", "isAvailable", "category", "img", "fileEntity"]
        widgets = {
            "name": TextInput(attrs={
                'class': 'form-control',
                'placeholder': 'Enter Product Name'
            }),
            "description": Textarea(attrs={
                'class': 'form-control',
                'placeholder': 'Enter Product Description'
            }),
            "price": NumberInput(attrs={
                'class': 'form-control',
                'placeholder': 'Enter Product Price'
            }),
            "category": Select(attrs={
                'class': 'form-control',
                'placeholder': ''
            }),
            "img": ClearableFileInput(attrs={
                'class': 'form-control',
                'placeholder': 'Enter'
            }),
            "fileEntity": ClearableFileInput(attrs={
                'class': 'form-control',
                'placeholder': 'Enter'
            }),
        }
}

```

```

class ReviewForm(ModelForm):
    class Meta:
        model = Review
        fields = ["comments", "product_raiting", "autor_raiting"]
        widgets = {
            "comments": Textarea(attrs={
                'class': 'form-control',
                'placeholder': 'Enter Product Comments'
            }),
            "product_raiting": Select(attrs={
                'class': 'form-control',
                'placeholder': ''
            }),
            "autor_raiting": Select(attrs={

```

```

        'class': 'form-control',
        'placeholder': ''
    )),

    }

```

Файл forms.py

```

from dataclasses import fields
from django.contrib.auth.forms import UserCreationForm
from django.contrib.auth.models import User
from django import forms
from customers.models import CustomUser

class RegisterUserForm(UserCreationForm):
    email = forms.EmailField(widget=forms.EmailInput(attrs={'class' : 'form-
control'}))
    first_name = forms.CharField(max_length = 50,
widget=forms.TextInput(attrs={'class' : 'form-control'}))
    last_name = forms.CharField(max_length = 50,
widget=forms.TextInput(attrs={'class' : 'form-control'}))

    class Meta:
        model = CustomUser
        fields = ('username', 'first_name', 'last_name', 'email',
'password1', 'password2')

    def __init__(self, *args, **kwargs):
        super(RegisterUserForm, self).__init__(*args, **kwargs)

        self.fields['username'].widget.attrs['class'] = 'form-control'
        self.fields['password1'].widget.attrs['class'] = 'form-control'
        self.fields['password2'].widget.attrs['class'] = 'form-control'

    def save(self, commit=True):
        user = super(RegisterUserForm, self).save(commit=False)
        user.blockchainPublicKey = 'abc'
        if commit:
            user.save()
        return user

```

Файл views.py

```

from django.shortcuts import render, redirect
from django.contrib.auth import authenticate, login, logout
from django.contrib.auth.forms import UserCreationForm
from django.contrib import messages
from Services import blockChainService
from customers.models import CustomUser
from main.models import Products, Category
from .forms import RegisterUserForm

```

```

def profile_page(request):
    products = Products.objects.all()
    category = Category.objects.all()

    return render(request, 'authenticate/profilePage.html',
{'title': 'ProfilePage', 'products': products, 'category': category})

def login_user(request):
    if request.method == "POST":

        username = request.POST['username']
        password = request.POST['password']
        user = authenticate(request, username=username, password=password)
        if user is not None:

            login(request, user)
            return redirect('Hub')
        else:
            messages.success(request, ("Not possible to log in"))
            return redirect('LogIn')
    else:
        return render(request, 'authenticate/login.html', {})

def logout_user(request):

    logout(request)
    messages.success(request, ("You were logged Out"))
    return redirect('Hub')

def register_user(request):

    if request.method == "POST":
        form = RegisterUserForm(request.POST)
        if form.is_valid():
            form.save()
            username = form.cleaned_data['username']
            password1 = form.cleaned_data['password1']
            user = authenticate(username=username, password=password1)
            login(request, user)
            messages.success(request, ("You were successfully registered"))
            return redirect('registration_successful')
        else:
            form = RegisterUserForm()

    return render(request, 'authenticate/register_user.html', {'form': form})

def registration_success(request):
    if request.session.get('registration_success'):
        # User has already accessed the page, handle accordingly

```

```

        return render(request, 'access_denied.html')

    blockchainService.create_new_user(request)
    data = blockchainService.create_new_user(request)

    customUser = request.user

    # Set session variable to indicate successful registration
    request.session['registration_success'] = True

    return render(request, 'authenticate/registration_successful.html',
    {
        'customUser': customUser,
        'blockchainPrivateKey': data['private']
    })

```

Файл `urls.py`

```

from django.urls import path, include
from . import views

urlpatterns = [
    path('login_user', views.login_user, name='LogIn'),
    path('logout_user', views.logout_user, name='LogOut'),
    path('register_user', views.register_user, name='RegisterUser'),
    path('profile_page', views.profile_page, name='ProfilePage'),
    path('registration_successful', views.registration_success,
name='registration_successful'),

]

```

Файл `blockChainRegisterProperty.py`

```

import http
from django.conf import settings
import requests
from django.shortcuts import render
import json

from customers.models import CustomUser

def registerProperty(fromPublicKey, blockchainPrivateKey, fileHash):
    url = f'{settings.BLOCKCHAIN_ENDPOINT_URL}api/Block/registerProperty'

    # Construct the object
    payload = {
        "fromKeys": {
            "publicKey": "string",
            "privateKey": "string"
        },
    },

```

					КПІ.IT-8304.045440.02.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		12

```

        "property": "string"
    }

    # Update the payload with the actual values
    payload['fromKeys']['publicKey'] = fromPublicKey
    payload['fromKeys']['privateKey'] = blockchainPrivateKey
    payload['property'] = fileHash

    headers = {
        "Content-Type": "application/json"
    }

    try:
        response = requests.post(url, data=json.dumps(payload), headers=headers)
        # print(payload)
        # print(response.json())
        if response.status_code == 200:
            # Successful request
            response_data = response.json()
            # Access the response data
            # isSuccess = response_data['isSuccess']
            return response_data
            # ... process the response data
        else:
            # Request failed
            print('Failed to register product. Status code:',
response.status_code)
    except requests.exceptions.RequestException as e:
        # Request exception occurred
        print('An error occurred:', e)

```

Файл **blockChainService.py**

```

import http
import requests
from django.shortcuts import render
import json
from django.conf import settings

def create_new_user(request):
    url = f'{settings.BLOCKCHAIN_ENDPOINT_URL}api/user/new'
    data = {}
    try:
        response = requests.get(url)
        if response.status_code == 200:
            # Successful request
            print('User created successfully.')
            print(response.content)
        else:

```

```

        # Request failed
        print('Failed to create user. Status code:', response.status_code)
except requests.exceptions.RequestException as e:
    # Request exception occurred
    print('An error occurred:', e)

# response_content = '{"key": "value"}'

# Deserialize the JSON content into a Python object
json_data = json.loads(response.content)

# Access the data as an object

isSuccess = json_data['isSuccess']
# print(isSuccess)
# print(json_data['error'])
# print(json_data['payload'])
data = json_data['payload']
# print(data['public'])

customUser = request.user
customUser.blockchainPublicKey = data['public']
blockchainPrivateKey = data['private']
customUser.save()
conn = http.client.HTTPConnection("localhost", 5001)

# Call the function
return data

```

Файл **blockChainTransferService.py**

```

import http
import requests
from django.shortcuts import render
import json
from django.conf import settings

def transferProperty(fromPublicKey, blockchainPrivateKey, toPublicKey, fileHash):
    url = f'{settings.BLOCKCHAIN_ENDPOINT_URL}api/Block/transferProperty'

    # Construct the object
    payload = {
        "fromKeys": {
            "publicKey": "string",
            "privateKey": "string"
        },
        "to": "string",
        "property": "string"
    }

```

```

# Update the payload with the actual values
payload['fromKeys']['publicKey'] = fromPublicKey
payload['fromKeys']['privateKey'] = blockchainPrivateKey
payload['to'] = toPublicKey
payload['property'] = fileHash

headers = {
    "Content-Type": "application/json"
}

try:
    response = requests.post(url, data=json.dumps(payload), headers=headers)
    print(payload)
    print(response.json())
    if response.status_code == 200:
        # Successful request
        response_data = response.json()
        # Access the response data
        # isSuccess = response_data['isSuccess']
        return response_data
        # ... process the response data
    else:
        # Request failed
        print('Failed to create user. Status code:', response.status_code)
except requests.exceptions.RequestException as e:
    # Request exception occurred
    print('An error occurred:', e)

```

Файл **privateKeyValidator.py**

```

from Crypto.PublicKey import RSA
from base64 import b64decode, b64encode
from binascii import Error as BinasciiError
from Crypto.PublicKey import RSA
import base64

```

```

def is_valid_rsa_key(key_string):

    try:
        return base64.b64encode(base64.b64decode(key_string)) ==
key_string.encode()
    except Exception:
        return False

```

Файл **addProduct.js**

```

// const { data } = require("jquery")
const input = document.getElementById('id_img');
const confirmBtn = document.getElementById('confirm-btn');
const imageBox = document.getElementById('image-box');
const hiddenElement = document.getElementById('hidden-element');
let isCropped = false; // Track whether the image has been cropped

```

					КПІ.IT-8304.045440.02.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		15

```

input.addEventListener('change', () => {
  console.log('changed');
  hiddenElement.className = "text-center mb-2 visible"
  const img_data = input.files[0];
  const url = URL.createObjectURL(img_data);
  imageBox.src = url;

  const cropper = new Cropper(imageBox, {
    aspectRatio: 9 / 9,
    viewMode: 0,
  });

  confirmBtn.addEventListener('click', function (event) {
    event.preventDefault(); // Prevent form submission
    const croppedImageDataURL =
cropper.getCroppedCanvas().toDataURL("image/png");
    imageBox.src = croppedImageDataURL;
    isCropped = true; // Set the isCropped flag to true

    // Convert data URL to Blob
    const blob = dataURLtoBlob(croppedImageDataURL);

    // Create a new File object

    let now = new Date();
    let formattedDate = '' + now.getFullYear()
+ padZero(now.getMonth() + 1)
+ padZero(now.getDate())
+ padZero(now.getHours())
+ padZero(now.getMinutes())
+ padZero(now.getSeconds())
+ padZero(now.getMilliseconds(), 3);

    function padZero(num, size=2) {
      let s = String(num);
      while (s.length < (size || 2)) {s = "0" + s;}
      return s;
    }

    const croppedImageFile = new File([blob], formattedDate+'.png', { type:
'image/png' });

    // Create a new DataTransfer object
    const dataTransfer = new DataTransfer();

    // Add the cropped image file to the data transfer object
    dataTransfer.items.add(croppedImageFile);

    // Assign the files from the data transfer object to the file input
    input.files = dataTransfer.files;

```

```

    });
});

// Helper function to convert data URL to Blob object
function dataURLtoBlob(dataURL) {
    const arr = dataURL.split(',');
    const mime = arr[0].match(/:(.*?);/)[1];
    const bstr = atob(arr[1]);
    let n = bstr.length;
    const u8arr = new Uint8Array(n);
    while (n--) {
        u8arr[n] = bstr.charCodeAt(n);
    }
    return new Blob([u8arr], { type: mime });
}

```

Файл registration_successful.js

```

function copyBlockchainKey() {
    var copyText = document.getElementById("blockchainPrivateKey");

    if (navigator.clipboard && typeof navigator.clipboard.writeText === 'function')
    {
        // Use the Clipboard API for modern browsers
        navigator.clipboard.writeText(copyText.value)
            .then(function() {
                alert("Copied the private key: " + copyText.value);
            })
            .catch(function(error) {
                console.error("Failed to copy private key:", error);
            });
    } else {
        // Fallback for older browsers
        copyText.select();
        copyText.setSelectionRange(0, 99999); // For mobile devices

        alert("Copied the private key: " + copyText.value);
    }
}

```

Файл urls.py

```

from django.contrib import admin
from django.urls import path, include
from django.conf import settings
from django.conf.urls.static import static

urlpatterns = [
    path('admin/', admin.site.urls),
    path('', include('main.urls')),

```

```

    path('customers/', include('django.contrib.auth.urls')),
    path('customers/', include('customers.urls'))
]+ static(settings.MEDIA_URL, document_root=settings.MEDIA_ROOT)

```

Configure Admin Titles

```

admin.site.site_header = 'FunnyBusiness Administration Page'
admin.site.site_title = 'Admin Tab'

```

Файл urls.py

```

from django.urls import path
from . import views
from Services import newFunction
from Services import blockChainService

urlpatterns = [
    path('', views.index, name='Hub'),
    path('addProduct', views.addProduct, name='AddProduct'),
    path('about-fb', views.about, name='About'),
    path('viewProduct/<product_id>', views.viewProduct, name='ViewProduct'),
    path('editProduct/<product_id>', views.editProduct, name='EditProduct'),
    path('deleteProduct/<product_id>', views.deleteProduct,
name='DeleteProduct'),
    path('buyProduct/<product_id>', views.buyProduct, name='BuyProduct'),
    path('acceptRequest/<product_id>', views.acceptRequest,
name='AcceptRequest'),
    path('downloadFile/<product_id>', views.downloadFile, name='DownloadFile'),
    path('rateProduct/<product_id>', views.rateProduct, name='RateProduct'),
    path('searchProduct', views.searchProduct, name='SearchProduct'),
    path('newTest', blockChainService.create_new_user, name='NewTest')
]

```

					КПІ.IT-8304.045440.02.12	Арк.
Змін.	Арк.	№ докум.	Підп.	Дата.		18

Факультет інформатики та обчислювальної техніки
Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

**ВЕБ-ЗАСТОСУНОК ПІДТРИМКИ ПРОЦЕСІВ ПРОДАЖУ
ВІРТУАЛЬНИХ ОБ’ЄКТІВ**

Програма та методика тестування

КПІ.IT-8304.045440.03.51

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Максим ГОЛОВЧЕНКО

Нормоконтроль:

_____ Максим ГОЛОВЧЕНКО

Виконавець:

_____ Данило ВАСИЛЕЦЬ

Київ – 2023

ЗМІСТ

1	ОБ’ЄКТ ВИПРОБУВАНЬ.....	3
2	МЕТА ТЕСТУВАННЯ	4
3	МЕТОДИ ТЕСТУВАННЯ.....	5
4	ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ	6

					КПІ.ІТ-8304.045440.03.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		2

1 ОБ'ЄКТ ВИПРОБУВАНЬ

Об'єктом випробування є веб-застосунок для підтримки процесів продажу віртуальних об'єктів. Це програмне забезпечення розроблене для використання на всіх популярних веб браузерях, тому було вирішено провести тестування коректної роботи застосунку та правильного відображення всіх візуальних компонентів для різних браузерів. Було проведено тестування компонентів представлення, бізнес логіки та роботи зі сховищами даних.

					КПІ.ІТ-8304.045440.03.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		3

2 МЕТА ТЕСТУВАННЯ

Метою тестування є наступне:

- перевірка правильності роботи програмного забезпечення відповідно до функціональних вимог;
- перевірка збереження даних;
- перевірка сумісності веб-додатку з останніми версіями сучасних браузерів (Google Chrome, Mozilla Firefox, Microsoft Edge, Opera, Brave);
- знаходження проблем, помилок і недоліків з метою їх усунення;
- перевірка зручності графічного інтерфейсу.

					КПІ.ІТ-8304.045440.03.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		4

3 МЕТОДИ ТЕСТУВАННЯ

Для тестування програмного забезпечення використовуються такі методи:

- статичне тестування – перевіряється програма разом з усією документацією, яка аналізується на предмет дотримання стандартів програмування;
- динамічне тестування – застосовується в процесі виконання програми. Коректність програмного засобу перевіряється на певній кількості тестів. При прогоні кожного з них збираються та аналізуються дані про проблеми та помилки в роботі програми;
- функціональне тестування – полягає у перевірці відповідності реальної поведінки програмного забезпечення очікуваній;
- мануальне тестування – тестування без використання автоматизації, тест-кейси пише особа, що тестує програмне забезпечення;
- тестування «білої скриньки» – об'єктом тестування тут є внутрішня поведінка програми. Перевіряється коректність побудови всіх елементів програми та правильність їхньої взаємодії один з одним.

					КПІ.ІТ-8304.045440.03.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		5

4 ЗАСОБИ ТА ПОРЯДОК ТЕСТУВАННЯ

Тестування виконується мануально та написанням unit-тестів, з метою знаходження помилок та недоліків як у функціональній частині програмного забезпечення так і в зручності користування. Для того, щоб перевірити працездатність та відмовостійкість застосунку, необхідно провести наступні тестування:

- динамічне тестування на відповідність функціональним вимогам;
- тестування на мобільних пристроях з різною роздільною здатністю екрану;
- тестування на виведення повідомлень про помилку, коли це необхідно;
- тестування інтерфейсу користувача;
- тестування зручності використання;
- тестування компонентів, що взаємодіють з програмним інтерфейсом.
- юніт-тестування бізнес-логіки;
- юніт-тестування компонентів, що працюють зі сховищами даних.

					КПІ.ІТ-8304.045440.03.51	Арк.
Змін	Арк.	№ докум.	Підп.	Дата.		6

Факультет інформатики та обчислювальної техніки

Кафедра інформатики та програмної інженерії

“ЗАТВЕРДЖЕНО”

Завідувач кафедри

_____ Едуард ЖАРІКОВ

“ ____ ” _____ 2023 р.

ВЕБ-ЗАСТОСУНОК ПІДТРИМКИ ПРОЦЕСІВ ПРОДАЖУ

ВІРТУАЛЬНИХ ОБ’ЄКТІВ

Керівництво користувача

КПІ.IT-8304.045440.04.34

“ПОГОДЖЕНО”

Керівник проєкту:

_____ Максим ГОЛОВЧЕНКО

Нормоконтроль:

_____ Максим ГОЛОВЧЕНКО

Виконавець:

_____ Данило ВАСИЛЕЦЬ

Київ – 2023

ЗМІСТ

1	ПРИЗНАЧЕННЯ ПРОГРАМИ	3
2	ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ.....	4
2.1	Системні вимоги для коректної роботи.....	4
2.2	Завантаження застосунку.....	4
2.3	Перевірка коректної роботи.....	4
3	ВИКОНАННЯ ПРОГРАМИ	5

					КПІ.ІТ-8304.045440.04.34	Арк.
						2
Змін.	Арк.	№ докум.	Підп.	Дата.		

1 ПРИЗНАЧЕННЯ ПРОГРАМИ

Розроблене програмне рішення являє собою веб-застосунок, що можна переглядати та використовувати за допомогою браузера перейшовши на відповідну адресу.

Додаток FunnyBusiness призначений для здійснення торгових операцій з віртуальними об'єктами. Він забезпечує підтвердження права власності користувачів на унікальні об'єкти та надає їм можливість виставляти на продаж та купувати такі товари у інших користувачів.

					КПІ.ІТ-8304.045440.04.34	Арк.
						3
Змін.	Арк.	№ докум.	Підп.	Дата.		

2 ПІДГОТОВКА ДО РОБОТИ З ПРОГРАМНИМ ЗАБЕЗПЕЧЕННЯМ

2.1 Системні вимоги для коректної роботи

Для успішної роботи даного застосунку необхідне виконання наступних вимог:

- наявність пристрою, що підтримує можливість використання популярних браузерів;
- наявність доступу до Інтернету;
- наявність підтримуваної версії браузера;
- можливість завантажувати файли з пристрою.

2.2 Завантаження застосунку

Розроблене програмне рішення передбачає використання застосунку без необхідності у інсталяції та завантаження. Користувач має можливість перейти на сторінку сайту у браузері та проводити подальші операції.

2.3 Перевірка коректної роботи

У випадку коректного відображення застосунку у браузері користувача додаткові дії для його перевірки не потрібні. При виникненні помилок у процесі завантаженні сторінки варто перевірити налаштування браузера користувача.

					КПІ.ІТ-8304.045440.04.34	Арк.
						4
Змін.	Арк.	№ докум.	Підп.	Дата.		

3 ВИКОНАННЯ ПРОГРАМИ

Після переходу на адресу веб-застосунку користувача буде адресовано на головну сторінку. Незареєстрований користувач одразу побачить повідомлення про необхідність реєстрації на сайті з відповідним посиланням на сторінку реєстрації (рисунок 3.1).

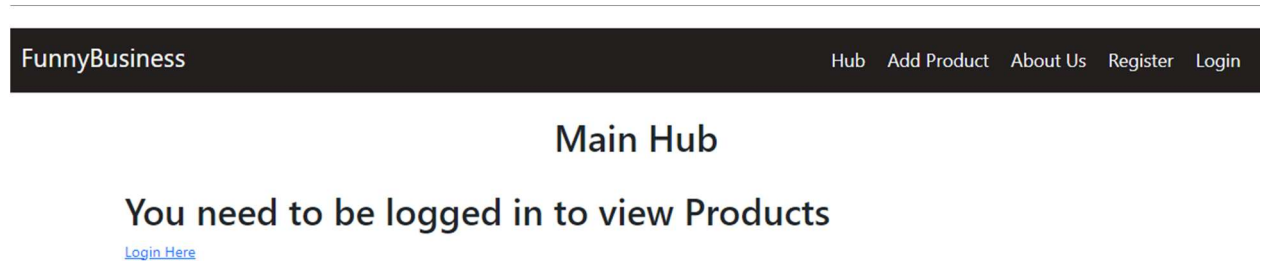


Рисунок 3.1 – Початкова сторінка веб-додатку

Для можливості використовувати функції сайту необхідно створити обліковий запис користувача. Він може зробити це переходом за посиланням на головній сторінці, або натиснувши на кнопку панелі навігації. Після цього його буде переадресовано на сторінку з формою після заповнення полів якої користувача буде зареєстровано у системі (рисунок 3.2).

					КПІ.ІТ-8304.045440.04.34	Арк.
						5
Змін.	Арк.	№ докум.	Підп.	Дата.		

Register

Username:

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

Last name:

Email:

Password:

- Your password can't be too similar to your other personal information.
- Your password must contain at least 8 characters.
- Your password can't be a commonly used password.
- Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Submit

Рисунок 3.2 – Сторінка реєстрації

Після успішного заповнення всіх полів відповідно вимогам зазначеним у інструкціях та надсилання форми користувача буде зареєстровано у системі та переадресовано на сторінку для збереження приватного ключа користувача (рисунок 3.3).

Your account was successfully registered!

Important information about your account

Don't close or refresh this page until you copy and save Your **PrivateKey**

PublicKey

Your PublicKey is open information that helps us to identify you in blockchain. We store it the same as other information about our users.

See my PublicKey

PrivateKey

PrivateKey is your secret key and you need to save it outside of any services. Even our site doesn't store this information. After you leave this page you won't be able to get it again.

Zh9bvjztE9Twx2PK8qHArAbZRPCb6FZX1WU4EgcRR3YAPloE6+EkFFA6zD

Copy my PrivateKey

Рисунок 3.3 – Сторінка для збереження ключів

					КПІ.ІТ-8304.045440.04.34	Арк.
						6
Змін.	Арк.	№ докум.	Підп.	Дата.		

Користувач повинен зберегти свій приватний ключ самостійно оскільки веб-сервіс не зберігає його у цілях забезпечення безпеки користувача. Він може виділити ключ самостійно, або скористатися функцією копіювання.

Після завершення процедури реєстрації користувач може приступити до використання основних функцій застосунку. Для виставлення товару на продаж користувачу необхідно перейти на вкладку додавання товару, що розташована на панелі навігації і доступна з усіх сторінок застосунку.

На сторінці створення товару користувачу буде надана форма після заповнення полів якої товар буде збережено у системі (рисунок 3.4).

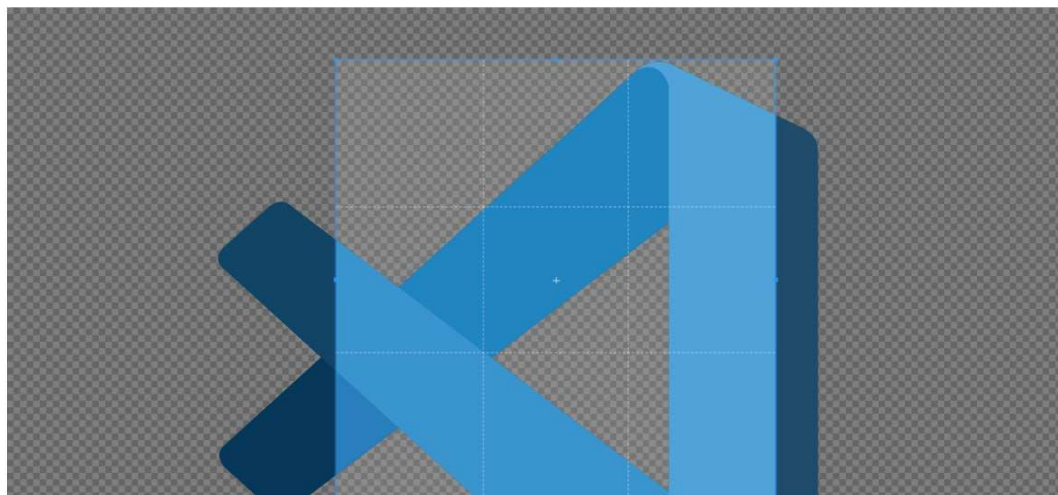
The screenshot shows the 'Create Product' page of the 'FunnyBusiness' application. The header contains the site name 'FunnyBusiness' and navigation links: 'Hub', 'Add Product', 'About Us', 'Profile Page', 'Logout', a search bar, and a 'Search' button. The main content area is titled 'Create Product' and contains the following form elements:

- Enter Product Name:** A single-line text input field.
- Enter Product Description:** A multi-line text area.
- Enter Product Price:** A single-line text input field.
- Availability:** A checkbox labeled 'Availability' with a note: 'Activating this option will make your product visible for other customers'.
- Chose category:** A dropdown menu showing '-----'.
- Edd Preview image:** A section with a 'Choose File' button and the text 'No file chosen'.
- Upload File:** Another section with a 'Choose File' button and the text 'No file chosen'.
- Enter your PrivateKey:** A single-line text input field followed by a green 'Save' button.

Рисунок 3.4 – Сторінка створення товару

Також у процесі створення нового товару користувачу необхідно обробити фото, що буде відображати товар на сайті. Після налаштування рамки обрізання фото користувач має натиснути кнопку підтвердити (рисунок 3.5).

Edd Preview image



confirm

Выберите файл 20230609164514074.png

Upload File

Выберите файл pngwing.com.png

vq/o/EashHZABFdnqif

Save

Рисунок 3.5 – Віджет редагування зображення

Після успішного створення товару користувача буде направлено на сторінку з каталогом доступних товарів (рисунок 3.6).

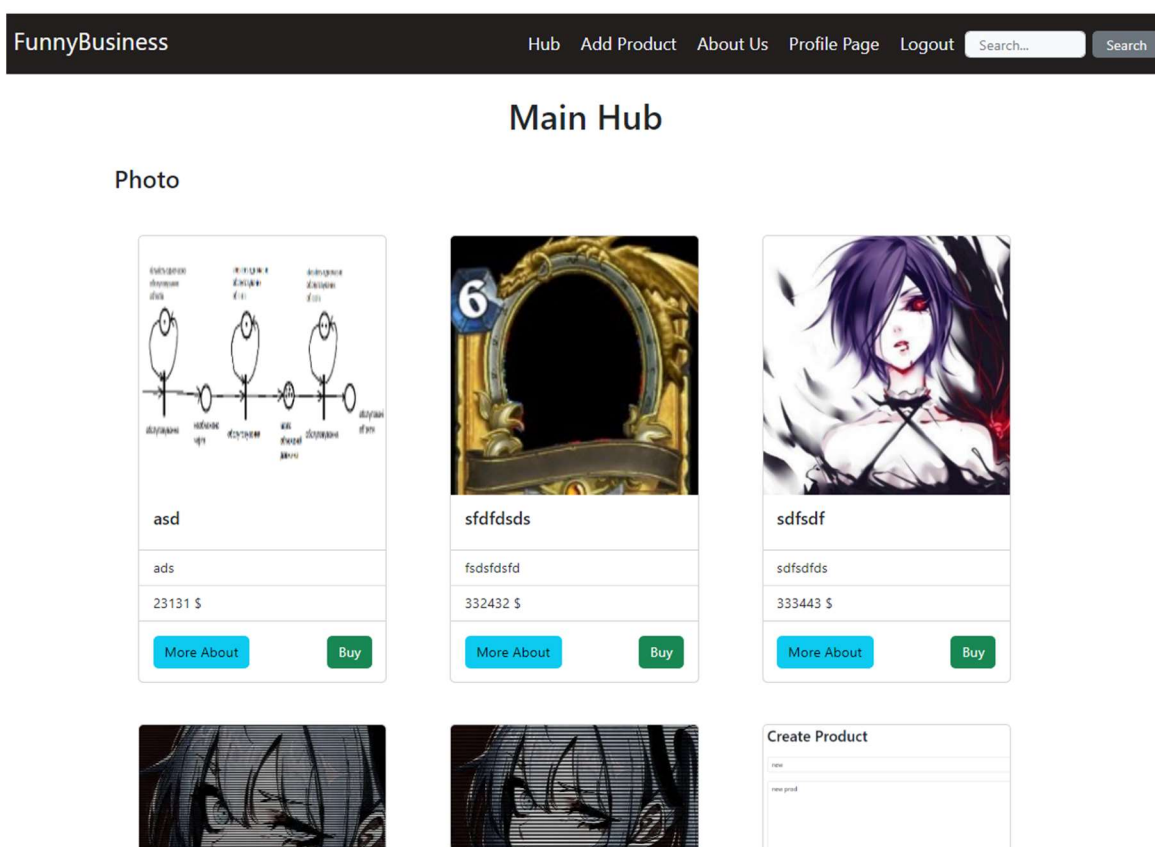


Рисунок 3.6 – Сторінка каталогу товарів

Змін.	Арк.	№ докум.	Підп.	Дата.

КПІ.IT-8304.045440.04.34

Арк.

8

Якщо користувач обрав відповідний чек-бокс у процесі створення товару його буде відображено на цій сторінці. Також його товар незалежно від вибору у чек-боксі буде додано до профілю користувача (рисунок 3.7). Для переходу у свій профіль користувач має скористатися панеллю навігації.

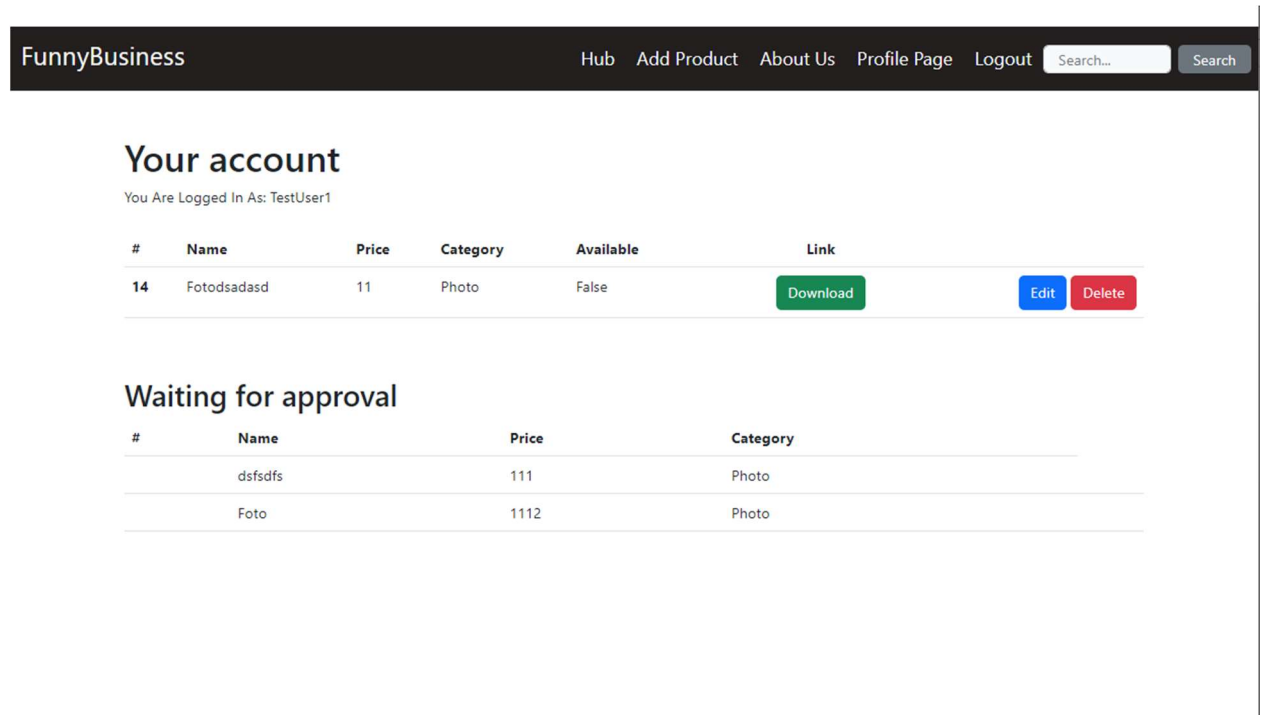


Рисунок 3.7 – Сторінка каталогу товарів

У власному профілі користувач може переглядати зареєстровані товари, вносити у них зміни, видаляти та завантажувати їх. Також тут відображено список товарів, що очікують підтвердження для покупки. У випадку існування запиту на придбання товару, на такій сторінці також буде відображено поле для введення приватного ключа користувача (рисунок 3.8). Після введення коректного ключа товар буде передано покупцю.

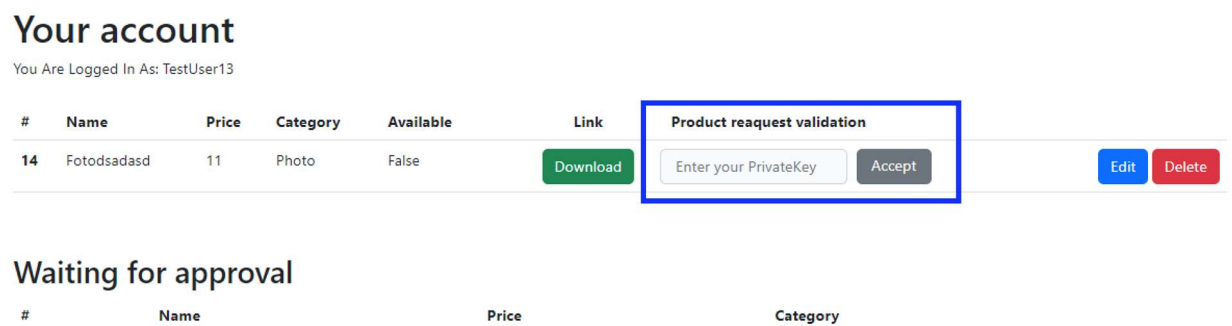


Рисунок 3.8 – Приклад товару який бажають придбати

Для початку процедури придбання товару користувачу необхідно обрати один з товарів на сторінці каталогу та натиснути кнопку придбати образу (рисунок 3.9), або зробити це на сторінці перегляду детальної інформації про товар (рисунок 3.10).

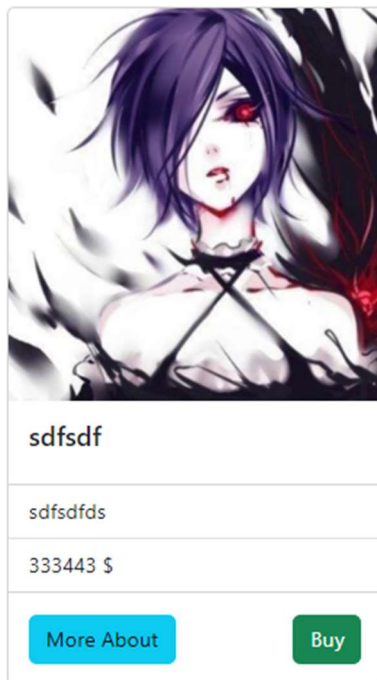


Рисунок 3.9 – Картка товару

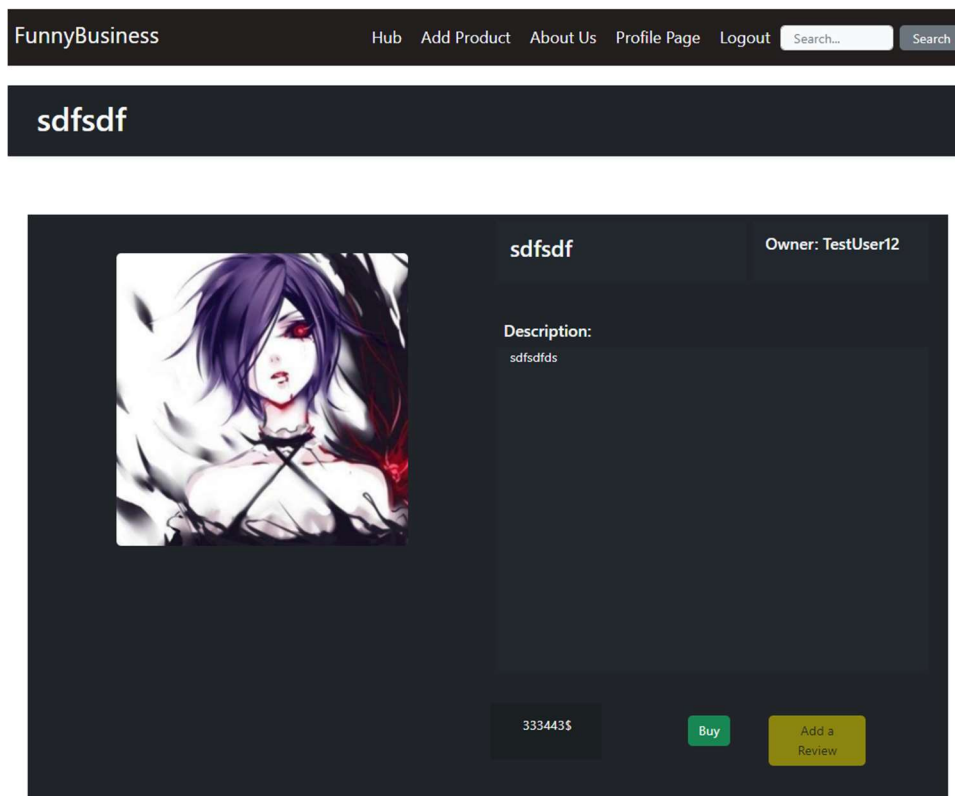


Рисунок 3.10 – Детальна інформація про товар

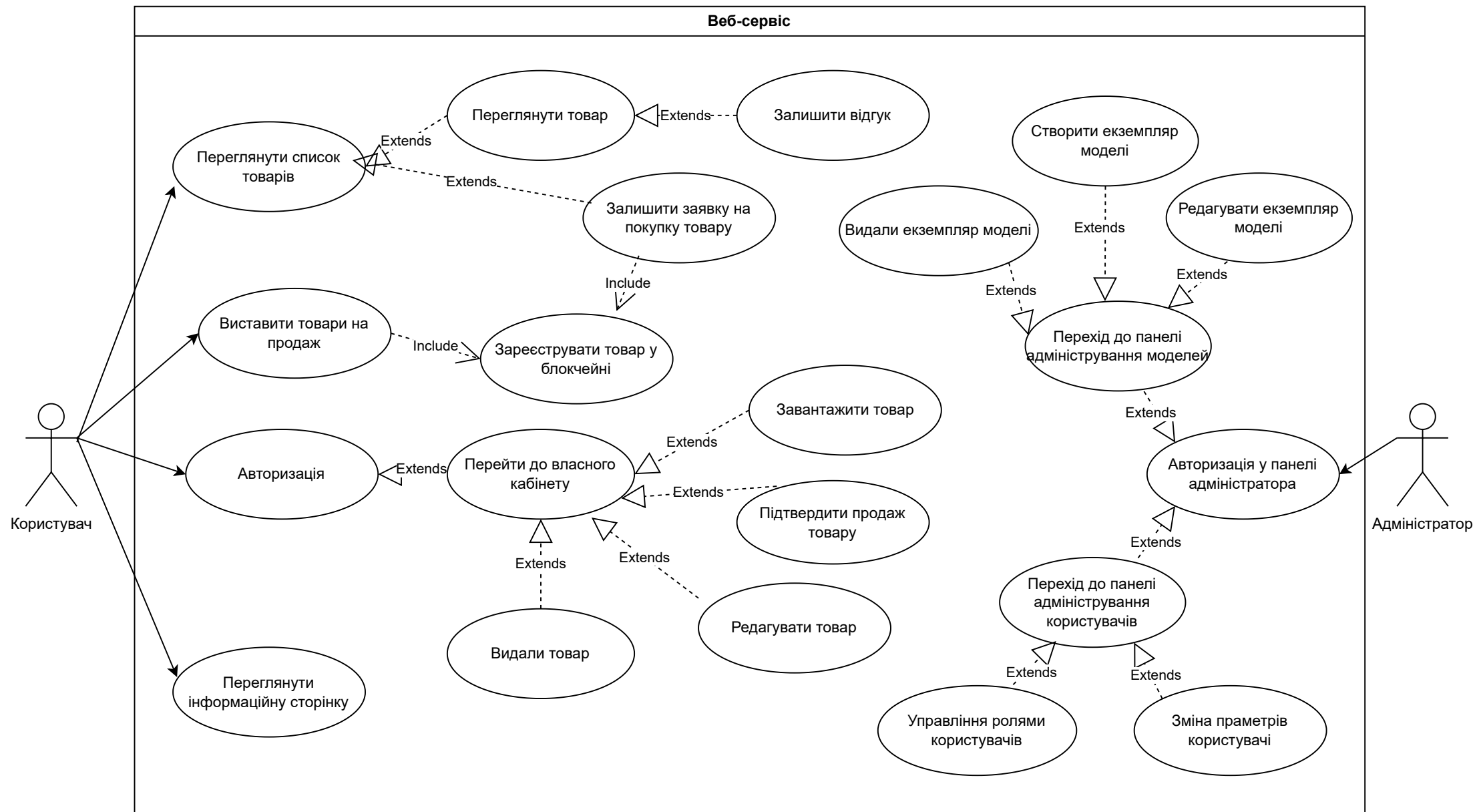
Після натискання на кнопку придбати користувача буде перенесено на сторінку з інструкціями (рисунок 3.11).



Your request was send to Product Owner. Please, wait until he will accept your offer.

Рисунок 3.11 – Детальна інформація про товар

Для завершення роботи з застосунком користувач не потрібно робити додаткових дій. Він може просто закрити сторінку застосунку та, за бажанням, може попередньо здійснити вихід з облікового запису скориставшись кнопкою вийти з запису на панелі навігації.



					КПІ.ІТ-8304.045440.05.99.CCB						
					Схема структурна варіантів використань	Літера			Маса	Масштаб	
Зм.	Арк.	№ документа	Підпис	Дата							
Розробив		Василець Д.Б.									
Перевірів		Головченко М.М									
Т. кон.											
					Веб-застосунок підтримки процесів продажу віртуальних об'єктів.	Аркуш			Аркушів		
Н. кон.		Головченко М.М				КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-83					
Затвердив		Жаріков Е.В									

Enter Product Price

Availability

Activating this option will make your product visible to other customers

Chose category

Photo

Editor

Edit Your Products Here

Name:

Fotodsasad

Description:

asdsadasdasda

Price:

11

IsAvailable:

Category:

Photo

Img: Currently: images/20230609165600762.png

Clear

Choose File

No file chosen

Update

Your account

You Are Logged In As: TestUser13

#	Name	Price	Category	Available	Link	Product request validation		
14	Fotodsasad	11	Photo	False	Download	Enter your PrivateKey Accept	Edit	Delete

Waiting for approval

#	Name	Price	Category
---	------	-------	----------

Error occurred

Invalid private key

#

Name

Price

Category

Available

Link

Product request validation

14

Fotodsasad

11

Photo

False

Download

Enter your PrivateKey

Accept

Your account

You Are Logged In As: TestUser13

Error occurred

You are trying to transfer the work of art that you are not owning.

#

Name

Price

Category

Available

Link

Product request validation

14

Fotodsasad

11

Photo

False

Download

Enter your PrivateKey

Accept

Edd Preview image

confirm

Выберите файл

20230609164514074.png

Upload File

Выберите файл

pngwing.com.png

vq/q/EashHZABFdngi

Save

Create Product

new Product

new product for tests

100

Availability

Activating this option will make your product visible to other customers

Chose category

Data

Error occurred

You are trying to register the work of art that is already registered.

Your account was successfully registered!

Your account

You Are Logged In As: TestUser13

#	Name	Price	Category	Available	Link
14	Fotodsasad	11	Photo	False	Download Edit Delete

Waiting for approval

#

Name

Price

Category

Description:

asdsadasdasda

Price:

11

IsAvailable:

Category:

Photo

Img: Currently: images/20230609165600762.png

Clear

Choose File

No file chosen

Register

Username:

NewUser

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

First name:

New

Last name:

User

Email:

newuser1@gmail.com

Password:

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

Your password can't be entirely numeric.

Password confirmation:

Enter the same password as before, for verification.

Submit

Password:

Your password can't be too similar to your other personal information.

Your password must contain at least 8 characters.

Your password can't be a commonly used password.

Your password can't be entirely numeric.

The password is too similar to the username.

This password is too short. It must contain at least 8 characters.

This password is too common.

Password confirmation:

Enter the same password as before, for verification.

Submit

A user with that username already exists.

Username:

admin

Required. 150 characters or fewer. Letters, digits and @/./+/-/_ only.

Your account was successfully registered!

Important information about your account

Dont close or refresh this page until you copy and save Your PrivateKey

PublicKey

Your PublicKey is open information that helps us to identify you in blockchain. We store it the same as other information about our users.

See my PublicKey

PrivateKey

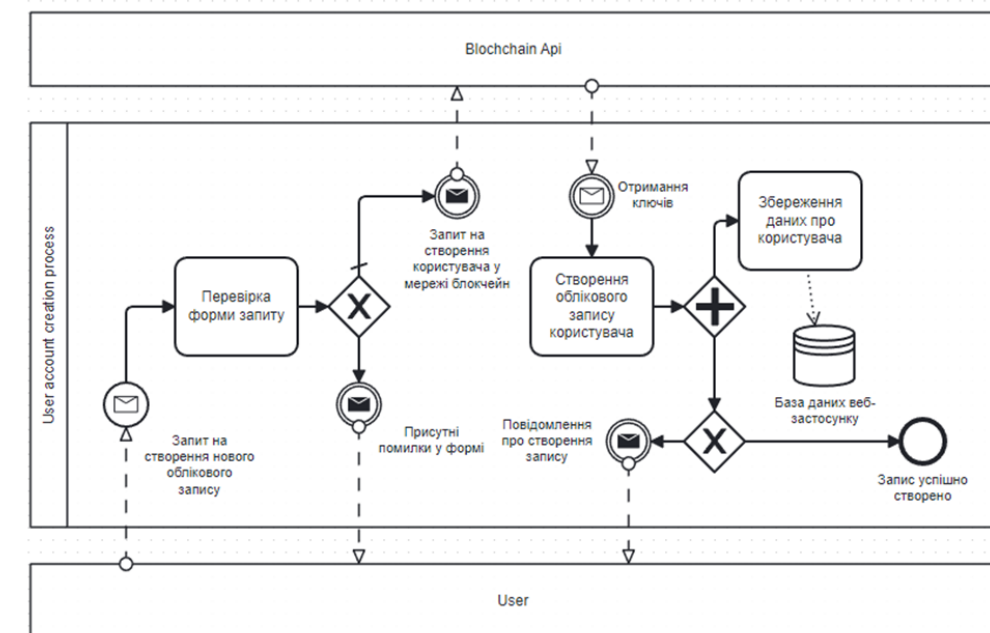
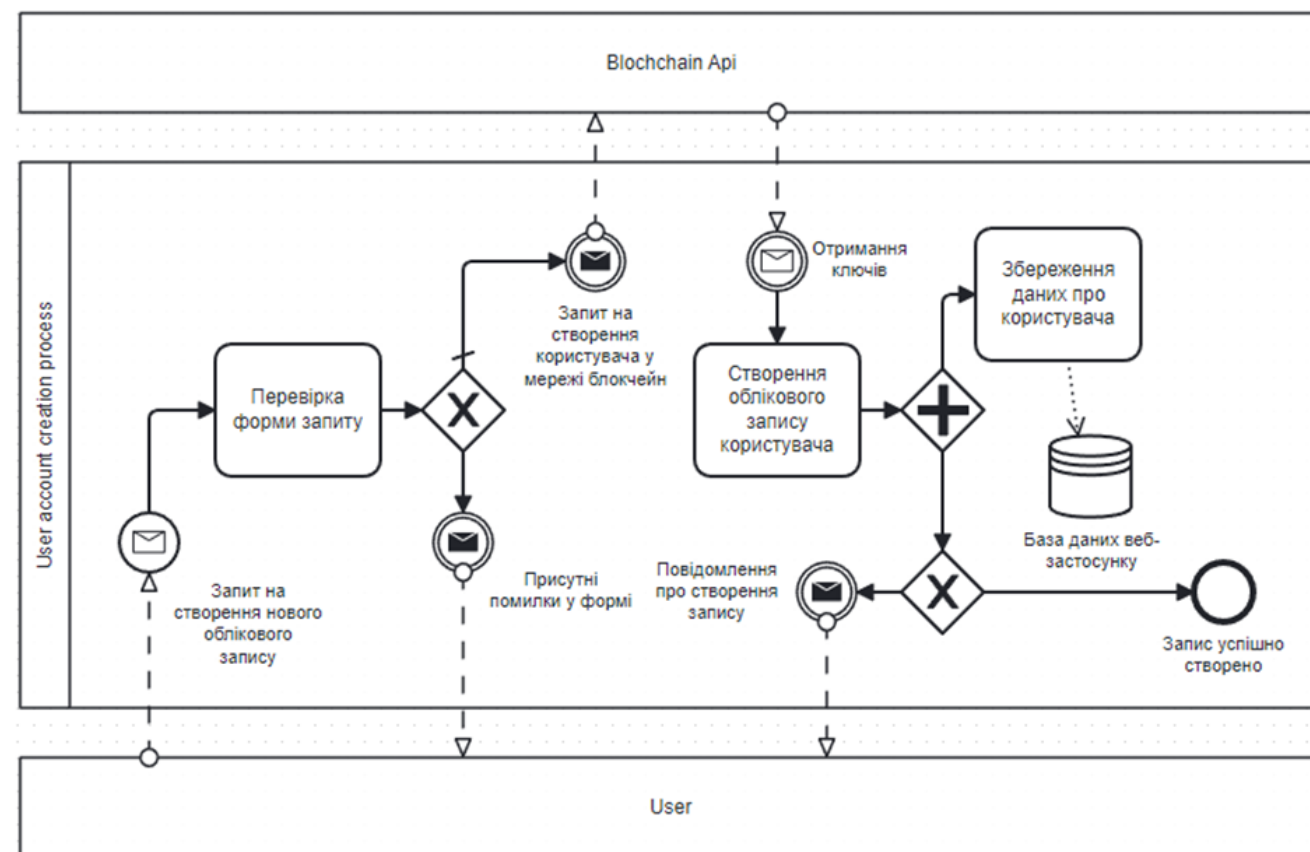
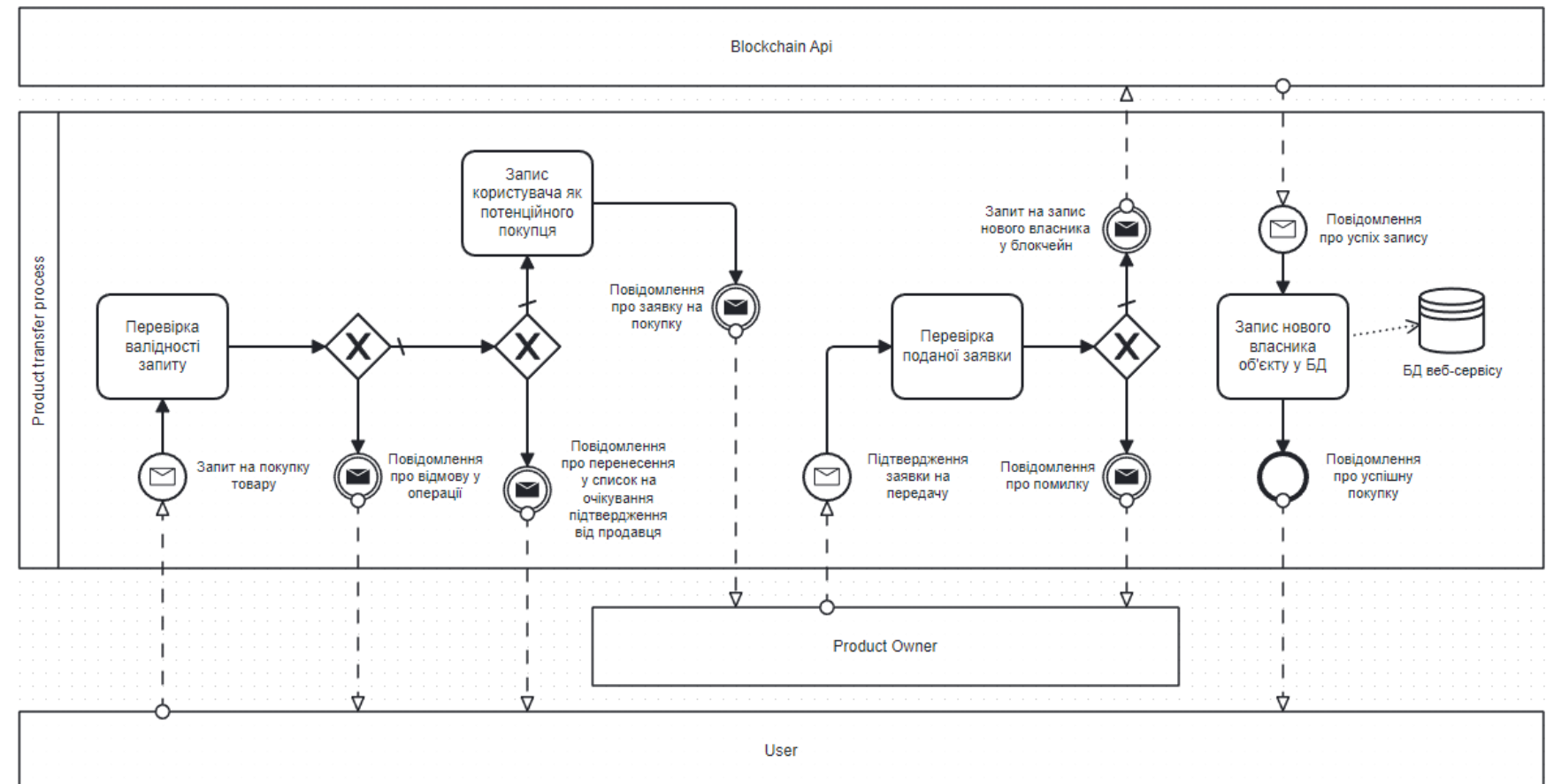
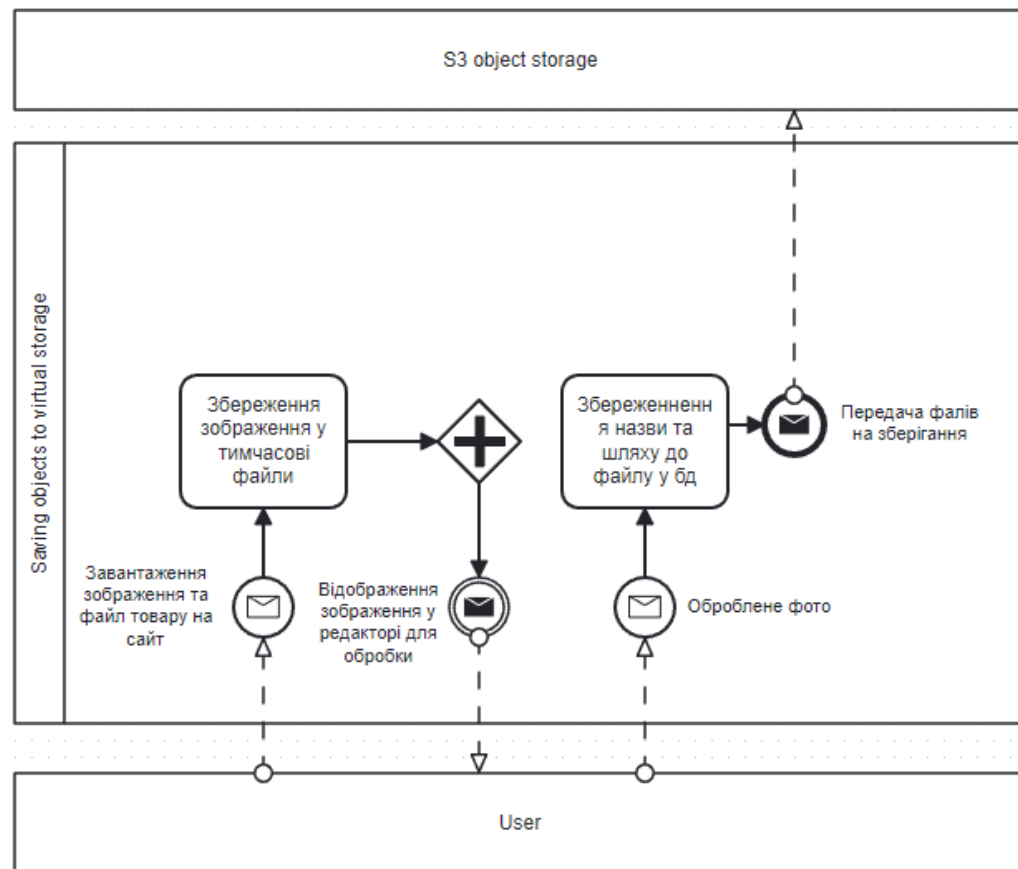
PrivateKey is your secret key and you need to save it outside of any services. Even our site doesn't store this information. After you leave this page you won't be able to get it again.

2n9byjzt9Twx2PK8qHAAAbZSPCo6fZX1WU4EgcR83YAPtoE6+EkFFA6zD

Copy my PrivateKey

					КПІ.ІТ-8304.045440.06.99.KE								
					Креслення вигляду екранних форм	Літера			Маса	Масштаб			
						Аркуш						Аркушів	
Зм.	Арк.	№ документа	Підпис	Дата		Веб-застосунок підтримки процесів продажу віртуальних об'єктів.							
Розробив		Василець Д.Б.			КПІ ім.Ігоря Сікорського Кафедра ІПІ гр. ІТ-83								
Перевірів		Головченко М.М											
Т. кон.													
Н. кон.		Головченко М.М											
Затвердив		Жаріков Е.В											

ПЛАКАТ СТРУКТУРИ БІЗНЕС-ПРОЦЕСІВ



Демонстраційний плакат до дипломного проекту

Веб-застосунок підтримки процесів продажу віртуальних об'єктів.

Виконав студент гр. ІТ-83 Василюк Д.Б.

Керівник Головченко М.М.