

**МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА
Факультет інформаційних технологій
Кафедра інтелектуальних технологій**

**ВИПУСКНА КВАЛІФІКАЦІЙНА РОБОТА
БАКАЛАВРА**


на тему:

Web застосунок для CPA мережі


Галузь знань **12 «Інформаційні технології»**
Спеціальність **122 «Комп'ютерні науки»**
Освітня програма **«Комп'ютерні науки»**
Освітній рівень: бакалавр

Виконав: студент 4 курсу, групи КН- 42

Коваленко О. В.


(прізвище та ініціали)

Керівник Федусенко О.В.


(прізвище та ініціали)

Кандидат технічних наук, доцент

(науковий ступінь, звання)

Випускна кваліфікаційна робота бакалавра допущена до захисту
рішенням кафедри *інтелектуальних технологій*
Протокол № 11 від 06.06.2022 р.
зав. кафедри _____ доц. Іларіонов О.Є.

Київ - 2022

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ ІМЕНІ ТАРАСА
ШЕВЧЕНКА**

Факультет інформаційних технологій
Кафедра інтелектуальних технологій
Спеціальність 122 «Комп'ютерні науки»

ЗАТВЕРДЖУЮ
Завідувач кафедри
інтелектуальних технологій
Іларіонов О.Є.

“ ___ ” _____ 2022 р.

**ЗАВДАННЯ
НА ВИПУСКНУ КВАЛІФІКАЦІЙНУ РОБОТУ СТУДЕНТОВІ**

Коваленку Олександр Володимировичу

(прізвище, ім'я, по батькові)

1. Тема проекту (роботи)
Web застосунок для CPA мережі

затверджена протоколом засідання кафедри від « 23 » грудня 2021 р. № 4

2. Термін здачі студентом закінченого проекту (роботи) 29 травня 2022 року
3. Вихідні дані до проекту (роботи)

Користувачькі умови, закон України про «Обробку і зберігання персональних даних».

4. Зміст розрахунково-пояснювальної записки (перелік питань, що їх належить розробити)

1. Аналіз процесу роботи CPA мережі
2. Розробка архітектури web застосунку для CPA мережі
3. Програмне забезпечення CPA мережі

5. Перелік презентаційного матеріалу (з точним зазначенням обов'язкових презентацій)

Актуальність дослідження (1 слайд), постановка задачі (1 слайд), термінологія програмного середовища (1 слайд), контекстна діаграма (1 слайд), декомпозиція діаграми IDEF0 (2 слайди), дерево функцій системи (1 слайд), інформаційне забезпечення (1 слайд), структура програмного забезпечення (1 слайд), огляд процесу тестування (1 слайд), висновок (1 слайд)

6. Консультанти з випускної кваліфікаційної роботи із зазначенням розділів випускної кваліфікаційної роботи, що їх стосуються

| Розділ | Консультант | Підпис, дата | |
|--------|-------------|----------------|------------------|
| | | Завдання видав | Завдання прийняв |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

7. Дата видачі завдання 15 лютого 2022 року

Керівник _____
(підпис)

/ Федусенко О.В. /
(ПІБ)

Завдання прийняв до виконання _____
(підпис)

/ Коваленко .О.В. /
(ПІБ)

КАЛЕНДАРНИЙ ПЛАН

| Пор. № | Назва етапів випускної кваліфікаційної роботи | Термін виконання етапів випускної кваліфікаційної роботи | Примітка |
|--------|---|--|----------|
| 1. | Аналіз літературних джерел, аналіз існуючих методів, аналіз основних процесів предметного середовища, постановка задачі | 15.02.2022 – 10.03.2022 | Виконано |
| 2. | Проектування web застосунку для СРА мережі | 10.03.2022 – 12.04.2022 | Виконано |
| 3. | Розробка та тестування web застосунку для СРА мережі | 12.04.2022 – 14.05.2022 | Виконано |
| 4. | Підготовка презентаційних матеріалів, оформлення пояснювальної записки | 14.05.2022 – 21.05.2022 | Виконано |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |
| | | | |

Студент-дипломник _____
(підпис)

/ Коваленко О.В. /
(ПІБ)

Керівник випускної кваліфікаційної роботи _____
(підпис)

/ Федусенко О.В. /
(ПІБ)

АНОТАЦІЯ

Студент Коваленко Олександр Володимирович виконав випускню кваліфікаційну роботу на тему «Web застосунок для CPA мережі» за спеціальністю 122 – «Комп'ютерні науки».

У випускній кваліфікаційній роботі проведено аналіз існуючих рішень для інформаційних систем подібного типу, розглянуто недоліки та переваги рішень, проведено UX/UI дослідження для розробки ефективної та зручної системи. Розроблено архітектуру системи, математичне, інформаційне та програмне забезпечення, що в результаті закрили потреби користувачів.

Ключові слова: офер, CPA мережа, web застосунок.

ANNOTATION

Student Kovalenko Oleksandr Volodymyrovych has completed the degree project on the topic "Web application for CPA network" in specialty 122 - "Computer Science".

The degree project analyzes the existing solutions for information systems of this type, considers the disadvantages and advantages of solutions, conducted UX / UI research to develop an efficient and convenient system. The system architecture, mathematical, informational and software were developed, which as a result closed the needs of users.

Keywords: offer, CPA network, web application.

ЗМІСТ

| | |
|---|----|
| ВСТУП | 7 |
| РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ РОБОТИ СРА МЕРЕЖІ | 9 |
| 1.1 Аналітичний огляд літератури за темою досліджень СРА мереж | 9 |
| 1.2 Аналіз існуючих СРА мереж | 14 |
| 1.3 Аналіз основних процесів предметного середовища | 18 |
| 1.4 Постановка задачі на розробку веб застосунку для СРА мережі | 24 |
| 1.5 Висновки до першого розділу | 26 |
| РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ WEB ЗАСТОСУНКУ ДЛЯ СРА МЕРЕЖІ | 27 |
| 2.1 Розробка архітектури | 27 |
| 2.1.1 Функціональний аналіз | 27 |
| 2.1.2 IDEF0 процесу відкриття оферу | 32 |
| 2.1.3 Архітектура інформаційної системи | 33 |
| 2.2 Математичне забезпечення web застосунку для СРА мережі | 35 |
| 2.3 Інформаційне забезпечення web застосунку для СРА мережі | 36 |
| 2.3.1 Розробка інформаційного забезпечення | 36 |
| 2.4 Висновки до другого розділу | 40 |
| РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ WEB ЗАСТОСУНКУ ДЛЯ СРА МЕРЕЖІ | 42 |
| 3.1 Обґрунтування вибору програмних засобів | 42 |
| 3.2 Структура програмного забезпечення | 43 |
| 3.3 Керівництво користувача | 46 |
| 3.4 Огляд процесу тестування | 56 |
| Висновки до третього розділу | 58 |
| Висновок | 59 |
| Список використаних джерел | 60 |
| ДОДАТОК А | 61 |
| ДОДАТОК Б | 71 |

ВСТУП

CPA мережа поєднує інтереси рекламодавця (продавця послуг/продуктів) та веб-майстрів (майданчики/вітрини/сайти, де рекламуються послуги/продукти для кінцевих клієнтів). Рекламодавець через посередників (веб-майстрів) отримує клієнтів, веб-майстер заробляє комісію за залучених клієнтів.

CPA мережа контролює якість та кількість трафіку для рекламодавців, дає асортимент продуктів/послуг, статистику веб-майстрам, а також веде взаєморозрахунки. Веб-майстер у CPA мережі бачить усі пропозиції від усіх рекламодавців зі статистикою, вибирає, з ким працювати, а гроші отримує агреговано від CPA мережі. Аналогічно рекламодавець платить одній CPA-мережі за продаж всіх веб-майстрів і не супроводжує технічні та фінансові процеси.

Зручність таких мереж у тому, що web майстрам не доводиться самотійно шукати рекламодавців, укладати договори з усіма компаніями та стежити за статистикою. Працюючи через CPA мережі, веб-майстер скорочує свої операційні витрати та розуміє за статистикою, які пропозиції вибрати.

Моє завдання – забезпечити коректний облік статистики, безперебійну систему комунікації, виплат та зобов'язань обох сторін.

CPA мережа — це унікальний B2B продукт, який забирає на себе найбільш моторні, з'їдаючі час та нерви процеси, такі як підбір продукту для свого майданчика, виплати, врегулювання робочих та іноді конфліктних ситуацій як на стороні рекламодавця, так і на стороні веб-майстра. Сучасні CPA мережі здатні позбавити всі сторони процесів від непотрібних проблем, що призводить не лише до зростання ринку, а й покращує його якість. [1]

Об'єктом дослідження даної роботи є процеси роботи у сфері партнерського маркетингу на прикладі відкриття оферів у CPA мережі.

Предметом дослідження роботи є алгоритми роботи з CPA мережею.

Мета дипломного проекту полягає у підвищенні ефективності співпраці рекламодавців та партнерської мережі веб-майстрів за рахунок створення

сучасного web застосунку для CPA мережі, який має містити необхідні у роботі web майстрів статистичні дані, зручний та зрозумілий інтерфейс, функціонал для відкриття, налаштування та запуску оферів в одному застосунку.

В даний час, як за технологіями, так і за змістом, це потужний інструмент, за допомогою якого можна реалізувати програму лояльності для клієнтів, залучити трафік з нових джерел, підвищити середній чек, знизити кількість невикуплених замовлень та багато іншого. Я би відніс його до обов'язкових рішень безлічі завдань маркетингу для бізнесу в інтернеті.

Для досягнення даної мети потрібно виконати наступні завдання:

- провести аналіз існуючих рішень для CPA мереж
- провести аналіз основних потреб користувачів
- провести аналіз алгоритмів та пайплайнів розробки систем подібного типу
- провести User Experience / User Interaction (далі UX/UI) дослідження
- сформулювати функціональні та нефункціональні вимоги для розроблюваного застосунку
- розглянути процеси даної предметної області
- розробити архітектуру додатку
- розробити та протестувати програмний продукт

РОЗДІЛ 1. АНАЛІЗ ПРОЦЕСУ РОБОТИ CPA МЕРЕЖІ

1.1 Аналітичний огляд літератури за темою досліджень CPA мереж

Темою дипломної роботи є дослідження роботи CPA мереж, розбір методик та метрик, а також розробка власної CPA мережі.

Об'єктом дослідження даної роботи є процеси роботи у сфері партнерського маркетингу на прикладі відкриття оферів у CPA мережі.

CPA (Cost Per Action) - модель партнерської реклами, при якій рекламодавець залучає так званих помічників для пошуку клієнтів та оплачує лише вчинені користувачами дії, такі як покупка товару, надсилання контактних даних (для подальшої роботи контакт центрів, так звана оплата за "лід", або встановлення додатку).

З сучасними темпами розвитку цифрових технологій, ведення бізнесу вже не є можливим без застосування інструментів інтернет маркетингу. Одним із таких інструментів є CPA маркетинг, який з'явився не так давно, але вже достатньо розповсюджений і приносить бізнесу прибутки.

У даній системі партнерами виступають Web-майстри зі своїми власними сайтами, які перенаправляють трафік на цільові сторінки.

Головним плюсом CPA маркетинга є оплата саме за результат, що є найбільш прозорою формою (в порівнянні з оплатою за покази в Бізнес менеджері Facebook).

Так як дана тематика достатньо специфічна, я наведу найчастіше вживані у даній сфері та в моїй роботі ключові терміни:

- Офер (англ. Offer) — рекламна пропозиція від компанії, тобто короткий тезисний опис того що потрібно зробити для компанії, мета пропозиції, сума виплати за результат. Web майстер отримавши офер може підключати його до рекламних компаній, які він налаштовує на цільову дію для замовника. Якісна CPA мережа має достатньо високооплачуваних оферів.

- Web майстер — спеціаліст, який вміє налаштовувати, запускати рекламні кампанії та досягати цільової дії.
- Лити (трафік на офер) — комплекс дій web майстра для досягнення мети оферу.
- Конверсія (CR) — відсоток користувачів сайту, що зробили цільову дію, характеризує ефективність лиття на офер.
- Трафік — потік користувачів, що виникає як наслідок роботи рекламної кампанії, відслідковується через сервіси Google Analytics та Facebook Ads.
- Tracking link — унікальне посилання офера, яке потрібно для відслідковування статистики рекламних кампаній.
- Пуш — повідомлення, яке отримує користувач при встановленні додатку, частина вирви продажу.
- Вирва продажу — комплекс взаємодії з клієнтом/користувачем для того, щоб він виконав цільову дію, як правило купив або вніс депозит.
- Депозит — інвестиція, внесення коштів в цільовий додаток або сайт, зроблений користувачем, з нього зазвичай web майстер отримує прибуток.
- Гео — країна(-и), на аудиторію яких розрахований офер, дана характеристика важлива тим, що потенційні клієнти різних країн мають різні поведінкові моделі, а отже і рекламні кампанії відрізняються.
- Хіти — переходи на цільовий сайт/додаток через рекламне оголошення.
- Ліди — користувачі, що пройшли реєстрацію на цільовому сайті/додатку.
- Хости — кількість унікальних переходів на домен.
- Visit to reg — математична модель підрахунку ефективності хітів, отже це $(\text{хіти}/100) * \text{ліди}$

- Affiliate менеджер — менеджер, що управляє оферами, розподіляє між web майстрами, аналізує статистику, також виступає як спеціаліст технічної підтримки.

CPA у якості основної моделі оплати можна задіяти:

- Facebook: виділення цільової аудиторії для початку
- В “Я.Дірект”: модель “Середня ціна конверсії”
- Google Ads: Цільова ціна за конверсію
- У CPA мережах:
- SEO просування сайтів
- Існуючі моделі розрахунку CPA

Основні моделі розрахунку які існують на даний момент:

- Ліди CPL (Cost Per Lead) – модель оплати тільки за приведений лід. Йдеться вже про клієнта з підтвердженими контактними даними.
- Кліки (CPC) — рекламодавець платить за кожен клік по оголошенню. Вартість кліка розраховується за аукціонною системою.
- Покази (CPM) — рекламодавець сплачує 1000 показів оголошення.
- Продаж CPS (Cost Per Sale) — використовується для позначення вартості одного оплаченого замовлення. Коли ми говоримо про CPS, то завжди маємо на увазі вже здійснений продаж.
- Налаштування програм (CPI) — рекламодавець платить за здійснену установку програми.
- інші, менш поширені цільові дії.[1]



Рисунок 1.1 — Канали залучення трафіка, характерні для CPA

Таким чином загальна формула: $CPA = \text{сума затрат на рекламу} / \text{кількість цільових дій}$. Під сумою ми розуміємо весь бюджет, витрачений на рекламну кампанію, а під кількістю цільових дій - сумарну кількість виконаних цільових дій.

Наприклад, ми запускаємо кампанію на покупки в онлайн-магазині прикрас з бюджетом 100 000 гривень. Рекламний трафік нам приніс 800 продажів прикрас різних видів. Отже рахуємо $100\,000 / 800 = 125$ гривень, тобто ціна однієї цільової дії (конверсії), для таких розрахунків в рекламних кабінетах зазвичай вбудовують мікросервіс, який дозволяє не тільки рахувати конкретну компанію, а і брати загалом декілька.

Цільові дії можуть бути різними на всіх етапах конверсії, загалом можна поділити на такі:

- Здійснення продажу (в CPS моделі)
- Заповнення і відправка форми з контактними даними
- Перехід на визначену Web-сторінку
- Конвертація користувача в ліда (CPL модель)
- Та інші менш розповсюджені в маркетингу дії, навіть перегляд відео н на протязі певного часу можна вважати такою дією, все залежить від цілей бізнеса та рекламної кампанії.

Отже можна виділити найрозповсюджені дії для CPA моделі:

- Заовлення товару (покупка)
- Додавання товару в корзину на сайті
- Заовлення Call back (дзвінок менеджера або консультанта)
- Підписка на розсилку
- Перегляд відео
- Встановлення додатків, програм, або скачування файлів
- Заовнення анкетних даних [2]

Тепер, коли ми розуміємо основний принцип роботи CPA моделі маркетингу, можна перейти до аналізу процесів роботи CPA мереж.

Крупні бренди і невеликі бізнеси просувають свої продукти через різні канали, але роблять це дедалі частіше не напряду, а через CPA мережі, тобто таку платформу, яка поєднує коло Web-майстрів з колом брендів, що надають так звані офери (запит на цільову дію, бюджет і готовність оплатити цільовий результат), так як вирішувати дані питання вручну займало би більше часу, та скоріш за все було б менш ефективно для бренду.

Отже CPA мережа це посередник між рекламодавцем і веб майстром, який буде розміщувати рекламу на своїх платформах. Детальніше взаємозв'язки можна розглянути на наведеній нижче схемі:

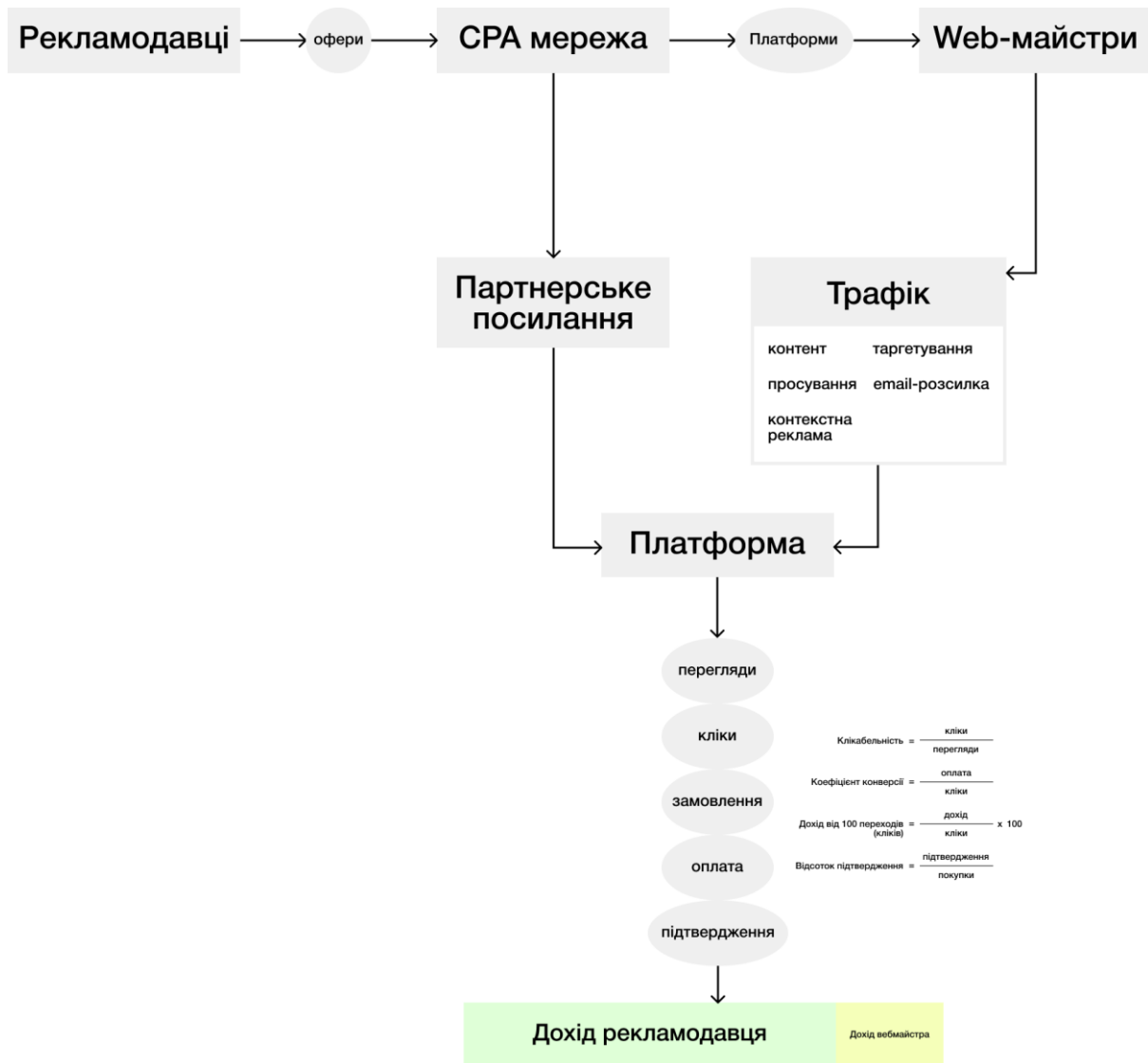


Рисунок 1.2 — схема роботи CPA мережі та важливі показники[3]

1.2 Аналіз існуючих CPA мереж

CPA мереж зараз стає дедалі більше з розвитком CPA маркетингу, найбільш важливими критеріями для вибору є:

1. Кількість і якість ексклюзивних та вигідних оферів.
2. Період заморозки (холда), що кошти можна виводити через деякий час після підтвердження.
3. Кількість партнерів.
4. Кількість “Гео”, тобто місць/країн проживання цільової аудиторії.
5. Наявність захисних від шахрайства сервісів.

6. Наявність технічної підтримки, доступність менеджерів (так званих affiliate manager)

Наведемо приклади найбільших, та найрозвинутіших CPA мереж:

- “Le Elephant” — Платформа, яка спеціалізується в основному на електронній комерції, продаж товарів різного призначення через онлайн сторінки, надає обширну кількість оферів від крупних компаній та малого бізнесу.

Одразу приковує увагу незручність фільтрації та сортування оферів, а вірніше їх відсутність. Платформа має обширний вибір мікросервісів, що можуть допомогти вебмайстрам ефективізувати роботу. Одним із недостатків платформи можна вважати бюрократизовану систему становлення партнером платформи, потрібно пройти всі перевірки згідно законодавства, а також підтвердити якість і реальність послуги, що надається. Цей процес займає достатньо великий проміжок часу і, на мою думку, його можна було б оптимізувати.

- “Everad” — CPA мережа у вертикалі “нутра”, мають достатньо широкий список оферів і гео, але достатньо закриту їх базу, щоб отримати інформацію про них, потрібно стати партнером, а отже реєструючись ми тільки можемо зробити висновки по продаючій лендинг сторінці.
- “ADVGame” — CPA-мережа з ігровими оферами, просувають мобільні застосунки і мобільні ігри, в основному реклама в соцмережах і в інших додатках і іграх, обширна кількість оферів від найбільших виробників, таких як Super Cell, Gameloft та ін. Має порівняно невеликий вибір інструментів в особистому кабінеті.
- “Huge Offers” — партнерська мережа в сфері електронної комерції, надають офери фінансових та крипто брендів Має прозору систему перегляду оферів та статистики, але натомість не зручний застосунок псує досвід користування платформою.

- “Admitad” — одна з найдовж діючих на ринку, мають офери по різних вертикалям, в тому числі комп’ютерні ігри і електронну комерцію, мають обширну систему фільтрації оферів, але незручну навігацію між сервісами і особистим кабінетом
- “Get4Click” — партнерська мережа в сфері електронної комерції, надають офери брендів Samsung, Skyeng, Philips, Skillbox та інших. Має зручну систему кастомізації особистого кабінета, але натомість непрозору систему відбору в партнери з боку Web-майстрів і обмежену кількість сервісів.

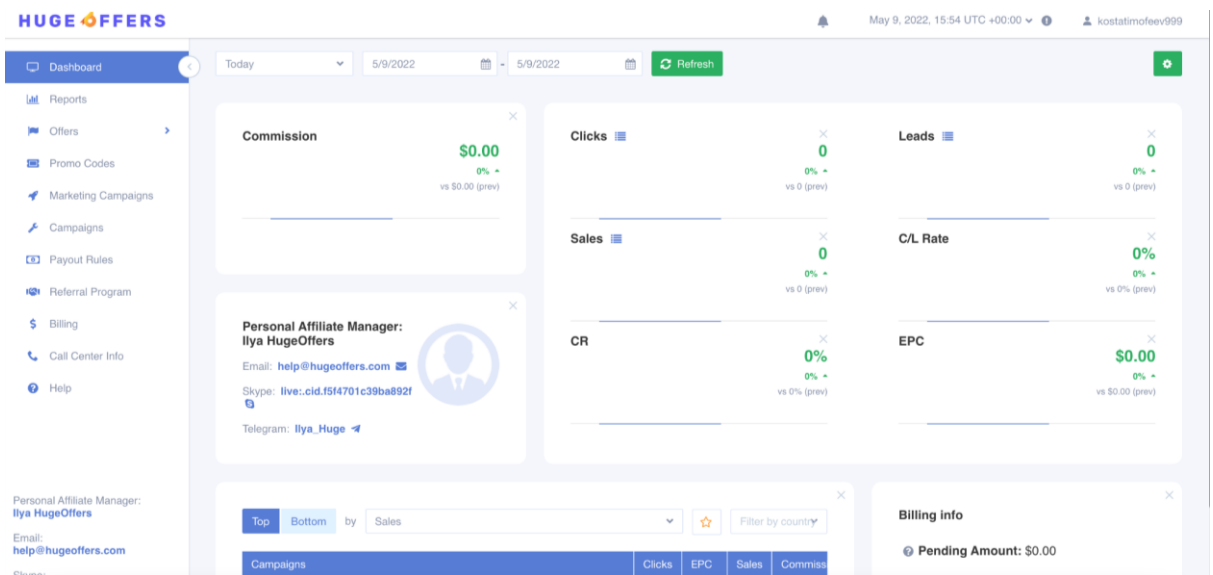


Рисунок 1.3 — сервіси CPA мережі “Huge Offers”

Висновок: більшість проаналізованих CPA мереж, як наведені вище, так і інші мають невелику кількість сервісів, обмежену систему фільтрації і офери в основному в одній або двох суміжних вертикалях, що може викликати дискомфорт у Web-майстрів, які працюють на декілька вертикалів.

Представлені вище партнерські мережі в основному не конкурують між собою і були обрані як лідери в своїх нішах, але як користувачу мені було достатньо складно орієнтуватись в інтерфейсі, всі представлені платформи мають приблизно однакову структуру блоків, що часто не дозволяє їх відрізнити між собою, особливо при наявності однакового кольорового рішення дизайну. Але загалом контент виглядає чисто і просто для

сприйняття інформації, якщо не брати складні системи підтверджень заявок і незручні системи навігації та фільтрації. Більшість мереж мають свої вбудовані мікросервіси, і я зіткнувся з проблемою: або сервісів надто багато і мало корисного, або просто недостатньо для комфортної роботи в рамках мережі, а отже я маю використовувати та підключати сторонні сервіси, які також на даний час вже достатньо широко представлені в інтернеті.

Отже найбільшою потребою ринку наразі є розробка зручної CPA мережі з потенційно великою кількістю оферів з різних вертикалей, зі зручною системою фільтрації, відслідковуванням власної статистики виходячи з CPA моделі обчислень, а також представленими необхідними для роботи сервісами, в тому числі з використанням API технологій.

Я обрав дослідження цієї теми на прикладі крипто ніші, яка зараз є найновішою та ще не має розвинутих CPA мереж та сервісів, як наприклад мережі в ніші e-commerce та gaming.

Факторами порівняння за існуючими рішеннями стали: зручність інтерфейсу, розвинена система фільтрації, наявні необхідні вбудовані сервіси, швидка процедура реєстрації та підтвердження реєстрації, адаптивність сайту, сучасний дизайн, відслідковування своєї статистики, та можливість завантажити звіти.

Таблиця 1.1. Порівняння існуючих рішень за факторами

| | Le Elephant | Eve rad | ADV Game | Admitad | Get4 Click | Huge Offers |
|---------------------------------|----------------|------------|-------------|---------|---------------|----------------|
| Зручність інтерфейсу | + | - | - | - | + | - |
| Розвинена система фільтрації | - | - | - | + | + | +- |
| Наявні необхідні | + | - | - | + | - | +- |

| | | | | | | |
|---|---|---|---|---|----|---|
| вбудовані сервіси | | | | | | |
| Швидка процедура реєстрації та її підтвердження | - | + | - | - | + | + |
| Адаптивність сайту | + | + | + | + | + | + |
| Сучасний дизайн | + | + | - | + | +- | + |
| Відслідковування своєї статистики | + | + | + | + | - | + |
| Можливість завантажити звіти | - | - | + | - | - | - |

Актуальність розробки функціональної та зручної платформи для партнерського маркетингу наразі гостра, кожна CPA мережа має свої недоліки і позитивні сторони, але перед клієнтами і партнерами постає складне питання вибору через недосконалість та обмеженість всіх CPA мереж, що існують на даний момент. Це може бути викликано порівняно невеликим віком галузі, яка розвивається швидкими темпами, компанії отримують більші прибутки і починають перелаштовуватись, але роблять це повільно. Альтернативних платформ з високим рівнем комфорту на даний момент не існує, або існують на рівні представлених. При розробці моєї платформи будуть враховані всі фактори, позитивні і негативні сторони існуючих для створення принципово нової та сучасної CPA мережі.

1.3 Аналіз основних процесів предметного середовища

Моя задача створити інформаційну систему для CPA мережі, використовуючи знання про CPA маркетинг, процеси та технології створення

подібних систем з метою розробити найбільш ефективне рішення в умовах сучасного ринку. Адміністратор може переглядати дії та підтверджувати офери для користувачів, користувач, у свою чергу може управляти своїми транзакціями, оферами, формувати звіти по конкретним оферам, фільтрувати їх по категоріям, а також переглядати статистику в реальному часі, використовуючи математичні моделі системи, у зручному інтерфейсі.

Задачами є:

1. Провести аналіз літератури
2. Провести порівняльний аналіз існуючих ІС
3. Проаналізувати процес роботи Web-застосунку для CPA-мережі
4. Розробити ефективну CRM систему для користувача платформи

Складові елементи:

1. База даних оферів
2. Підсистема авторизації
3. База даних користувачів
4. Модуль менеджменту оферами
5. Математичний модуль відстежування статистики
6. Модуль фільтрації оферів
7. Модуль поповнення балансу

Опис профілів зацікавлених сторін:

Внутрішні зацікавлені сторони:

- Розробник сайту
- Адміністратор сайту

Зовнішні зацікавлені сторони:

- Клієнти/Вебмайстри
- Рекламодавці

Модель оплати CPA розглянута в першому пункті – це вершина еволюції digital advertising. Головна її цінність у тому, що мережа дає рекламодавцям можливість використовувати майданчики та формати,

недоступні у стандартних каналах. Ідея в тому, що самостійно CPA мережа не генерує трафік. Для цього організація співпрацює з тисячами веб-майстрів, які є власниками сайтів, груп у соцмережах, Youtube-каналів або вміють добре налаштовувати різні види інтернет реклами для рекламодавців, які готові платити гроші за цільову дію на своєму web-ресурсі.

Ринок affiliate в Україні якраз перебуває на стадії становлення та формування. Є відчутна необхідність стимулювання зростання веб-майстрів, появи, створенні нових майданчиків. Якщо раніше, щоб заробити в партнерській мережі, потрібно було робити якийсь сайт, розбиратися з безліччю нюансів, то зараз є варіанти, як це можна зробити швидше та в рази простіше. Наприклад, звернутися до актуальних трендів генерації трафіку: Telegram-канали, YouTube та Instagram.

Можу виділити такі тренди:

1. Збільшення мобільного трафіку та обсягів продажів через mobile завдяки запуску в країні технологій 3G та 4G.
2. Зростання популярності відеоконтенту, який активно використовується для залучення трафіку рекламодавцями та веб-майстрами.
3. Розвиток нових каналів трафіку: месенджери, онлайн-трансляції, модель CPA-offline, SMS, YouTube, Instagram, сарафанне радіо, термінали та голосовий пошук.

Також можна зробити висновок щодо актуальних методів та модулів при проектуванні CPA мережі:

1. Наскрізна аналітика – трафік – продаж онлайн. Інтеграція за API з банками та мікро фінансовими компаніями для отримання онлайн статистики від них.
2. Технологічність - нові сервіси, процеси для клієнтів та веб майстрів. Наприклад, ми даємо веб майстрам шаблон, який легко інтегрувати в сайт і він автоматично підбирає кредиторів під профіль клієнта в онлайн.

3. Використання SMS, IVR, push-повідомлень, чат-ботів для максимальної монетизації трафіку.

Основною задачею є створення адаптивної, функціональної та зручної системи (веб-додатку) для CPA-мережі, використовуючи математичні CPA-моделі і сучасні технології та інструменти для розробки систем. Система має містити функціонал для управління, редагування оферів та сервісів платформи.

На першій діаграмі відображено процес «відкриття оферу» (рис. 1.4). На вході ми отримуємо персональні дані веб майстра, авторизаційні дані та дані про вертикаль роботи.

На виході, після виконаного процесу, ми отримуємо відкритий офер, його статистичні дані, що будуть відображатись списком та через математичний модуль застосунку.

Варто зазначити важливість дотримання закону України щодо “збору та обробки персональних даних” користувача, якими ми керуємось та гарантуємо йому конфіденційність і безпеку, на що також посилаються користувацькі умови і положення.

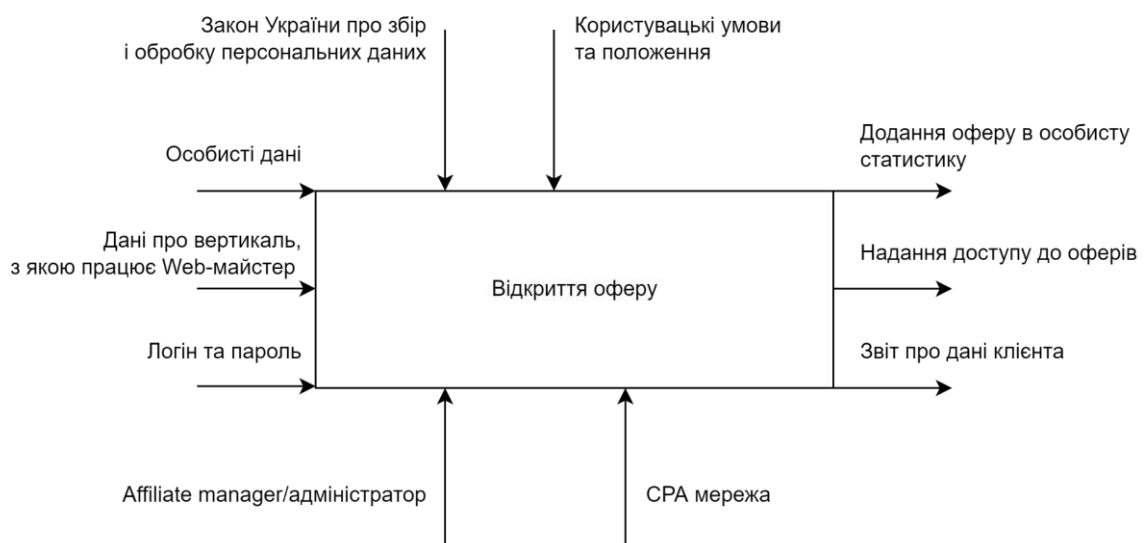


Рисунок 1.4 — контекстна діаграма ЯК Є

Розглянемо детальніше описаний процес, побудувавши декомпозицію його основних функцій (рис. 1.5). Для початку роботи веб майстер має пройти реєстрацію на платформі, після чого його дані обробляються системою і за ним закріплюється affiliate manager, який аналізує статистику веб майстра під час відкриття оферу.

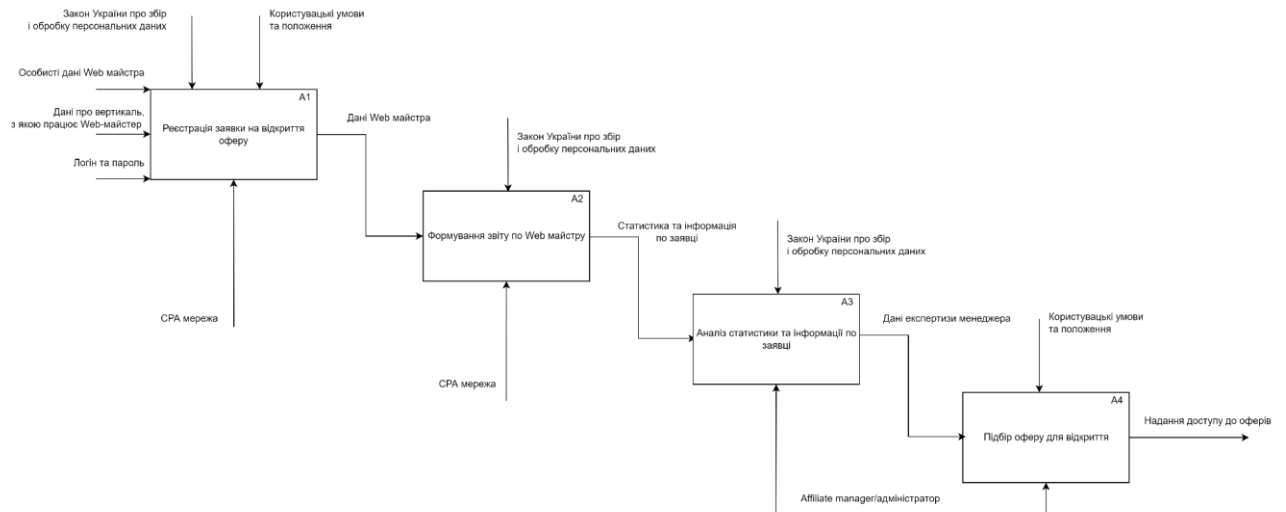


Рисунок 1.5 — Деталізована IDEF0 діаграма процесу відкриття оферу ЯК Є

Проведемо більш детальний аналіз процесу реєстрації на платформі (рис. 1.6).

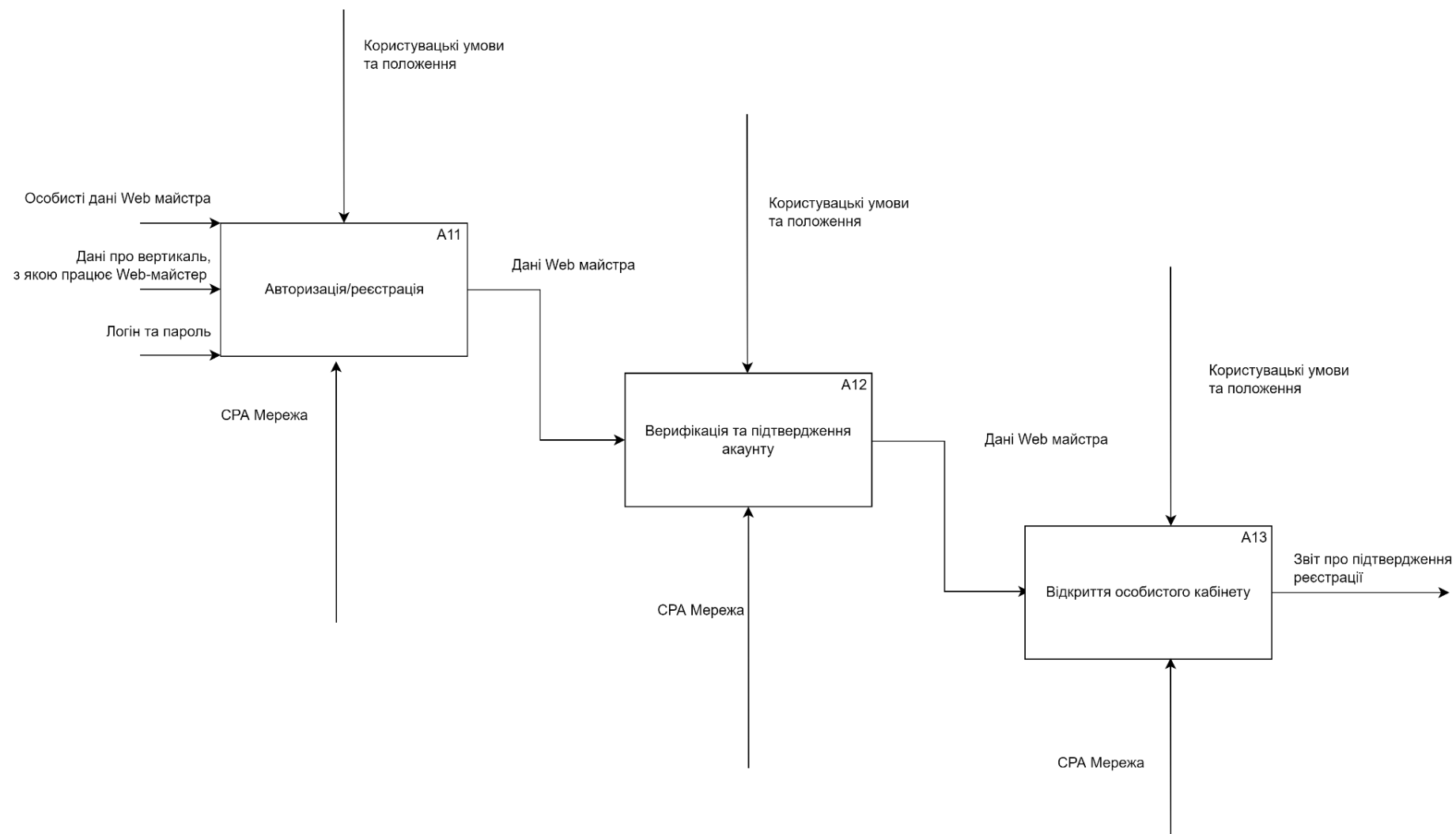


Рисунок 1.6 — Деталізація блоку А1 ЯК Є

У представлених вище діаграмах описується процес становлення учасником і партнером CPA мережі та отримання доступу до відкритого оферу.

Така складна схема реєстрації та підтвердження облікового запису потрібна, щоб менеджер, який супроводжує партнера провів особисту листування з ним та впевнився в достовірності наданої інформації, та підтвердив експертизу партнера щодо трафіку та його майстерності лити.

1.4 Постановка задачі на розробку веб застосунку для CPA мережі

Основною задачею є створення адаптивної, функціональної та зручної системи (веб-додатку) для CPA-мережі, використовуючи математичні CPA-моделі і сучасні технології та інструменти для розробки систем. Система має містити функціонал для управління, редагування оферів та сервісів платформи.

Основні функціональні вимоги до web застосунку:

1. Реєстрація та авторизація користувачів, підтвердження акаунту.

Стандартне рішення для всіх систем, додатків, платформ і сервісів задля безпеки особистих даних та цілісності системи. Користувач повинен мати свій особистий логін та пароль для користування системою, доступу до особистої статистики, а також задля безпеки особистих даних і фінансів.

2. Привабливість, зручність та ефективність інтерфейсу у взаємодії з користувачем.

Я задію свій досвід у розробці інтерфейсів, щоб створити максимально зручний та функціональний, згідно з правилами та принципами UX/UI дизайну, проведу дослідження, виявлю проблематику та запроваджу найбільш ефективне рішення

3. Модуль відкриття оферу та присвоєння користувачеві

Особливість розроблюваної платформи, необхідний функціонал для користувача і для адміністратора, відкриття офера одному користувачеві унеможлиблює відкриття того ж офера іншому користувачеві.

4. Система підрахунку показників у реальному часі

В особистому кабінеті на головній сторінці реалізована можливість у реальному часі відслідковувати показники по своїй статистиці, використовуючи математичні СРА-моделі. Користувач повинен мати змогу швидко візуально оцінити динаміку показників.

5. Індивідуальна система адміністратора з можливістю управління та редагування оферів та інформацією користувачів.

Необхідна для ефективного управління системою в бізнес проектах

6. Сервіс для виділення працюючих телефонних номерів на визначений час по заданій геолокації.

Додатковий модуль для більшої функціональності платформи, який закриває необхідність користувачів в сторонніх сервісах.

7. Система збереження та менеджменту статистикою та оферами.

Користувач повинен мати можливість завантажувати звіти, по його оферам.

8. Зручна система фільтрації.

Необхідна для швидкого знаходження потрібної інформації в системі.

9. Модуль поповнення балансу та виведення виплат на банківський рахунок

Користувач повинен мати можливість виводити зароблені кошти зручним для нього методом оплати, також для більшої зручності поповнювати баланс, щоб потім кошти розподіляти між рекламними кампаніями.

10.Додатковий сервіс паркування доменів що дозволяє реєструвати доменні імена на DNS серверах, дозволяє зберегти за собою право використовувати доменне ім'я для своїх сервісів або сайтів.

Нефункціональні вимоги:

1. Підтримка різних мов.

Стандартне рішення для всіх систем, додатків, платформ і сервісів задля зручності користування та комфорту користувачі з різних країн. Найпопулярніше рішення це використання англійської та української мови

2. Кастомізація теми.

Можливість змінювати колір інтерфейсу на власний смак, тобто користувач зможе налаштувати світлу/темну тему застосунку.

1.5 Висновки до першого розділу

На даний момент на ринку мало детально спроектованих CPA мереж, так як це найсучасніший вид маркетингу, який достатньо швидко набирає популярність, особливо в новій ніші криптовалют, які вже завоювали прихильність аудиторії та її потребу і ній, а з цього випливає необхідність в створенні системи, що підтримує різні ніші партнерського маркетингу, є ефективним інструментом та зручною системою для користувачів. Задіявши мій досвід та набуті навички, я створюю бізнес-рішення, продукт, що забезпечить користувача рішенням його потреб. Такий проект має високу цінність та конкурентоспроможність, так як всі подібні не мають всіх представлених систем і сервісів, а також не зручні у використанні. Моя система є гнучкою до змін з можливістю адаптовувати її під різні потреби завдяки математичним модулям та гнучкій системі редагування інформації. Тож система являє собою мультизадачну платформу для виконання затребуваних задач бізнесу і є вкрай необхідною в сучасності.

РОЗДІЛ 2. РОЗРОБКА АРХІТЕКТУРИ WEB ЗАСТОСУНКУ ДЛЯ СРА МЕРЕЖІ

2.1 Розробка архітектури

2.1.1 Функціональний аналіз

Побудова діаграм є чи не найголовнішим етапом розробки системи. Зокрема, якщо ми розглядаємо кінцеву ціль як бізнес продукт, ми маємо уявити картину функціонування, отримання вигоди та дослідити поведження користувача. Отже в цілому представлення про продукт ми сформували в першому розділі, але тепер потрібно ще структурувати в тому числі бізнес-процеси. В основі будь-якої платформи такого роду лежить дві версії, а саме користувацька і версія для адміністратора, в моєму випадку менеджера.

Affiliate менеджер виконує функцію адміністратора, додає, видаляє, редагує, а найголовніше - відкриває доступ до оферів користувачу (Web майстру).

Web майстер зі свого боку має доступ до математичного модуля, що вираховує статистику та дохід з його трафіку, звичайно що система має модуль реєстрації та модуль відкриття оферу. Web майстер має змогу обирати та надсилати запити на відкриття оферу менеджеру і в подальшому, після відкриття, користуватись платформою.

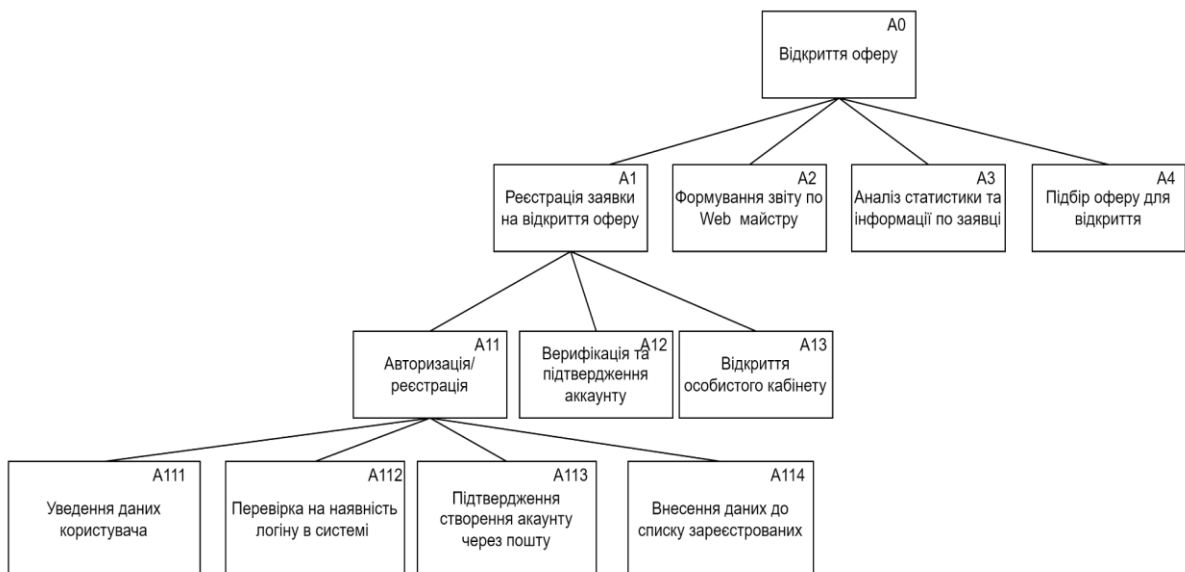


Рисунок 2.1 — Дерево функцій

Отже опишемо як функціонує система, спочатку користувач має зайти на сайт і, натиснувши кнопку “увійти”, ввести свій логін та пароль та авторизуватись, або зареєструватись, якщо ще немає акаунта в системі і потім проробити описані вище кроки. За умови реєстрації, дані клієнта записуються до списку web майстрів. Якщо він вже був зареєстрований, то система перевіряє наявність авторизаційних даних у системі. На головній сторінці користувач має можливість у зручному форматі користуватися автоматичним математичним модулем розрахунку статистики, переглянути динаміку та короткі контекстні меню переходу на сторінку оферів. Офери спочатку знаходяться не закріплені ні за ким і для відкриття користувачеві потрібно надіслати запит на потрібний офер, який система та affiliate-менеджер оброблюють. Після обробки запиту оффер додається на сторінку “Мої офери” та закріплюються в базі даних за ним, після цього майстер може переходити до налаштувань і працювати з офером, підключивши його до Google або Facebook Ads. Дані процеси відображено на діаграмі Event-Driven Process Chain (рис. 2.2).

Отже до бізнес процесів можна додати діаграму ланцюжка процесів, де основний процес це робота з платформою, що поділяється на підпроцеси, в тому числі реєстрацією, відкриттям оферів, відслідковуванням статистики, переглядом та редагуванням особистих даних та налаштуванням особистих оферів, знаходженням потрібного, відслідковування статистики по кожному окремому оферу, процеси поєднані між собою, мають декілька рівнів ієрархії та розташовані послідовно (рис. 2.3).

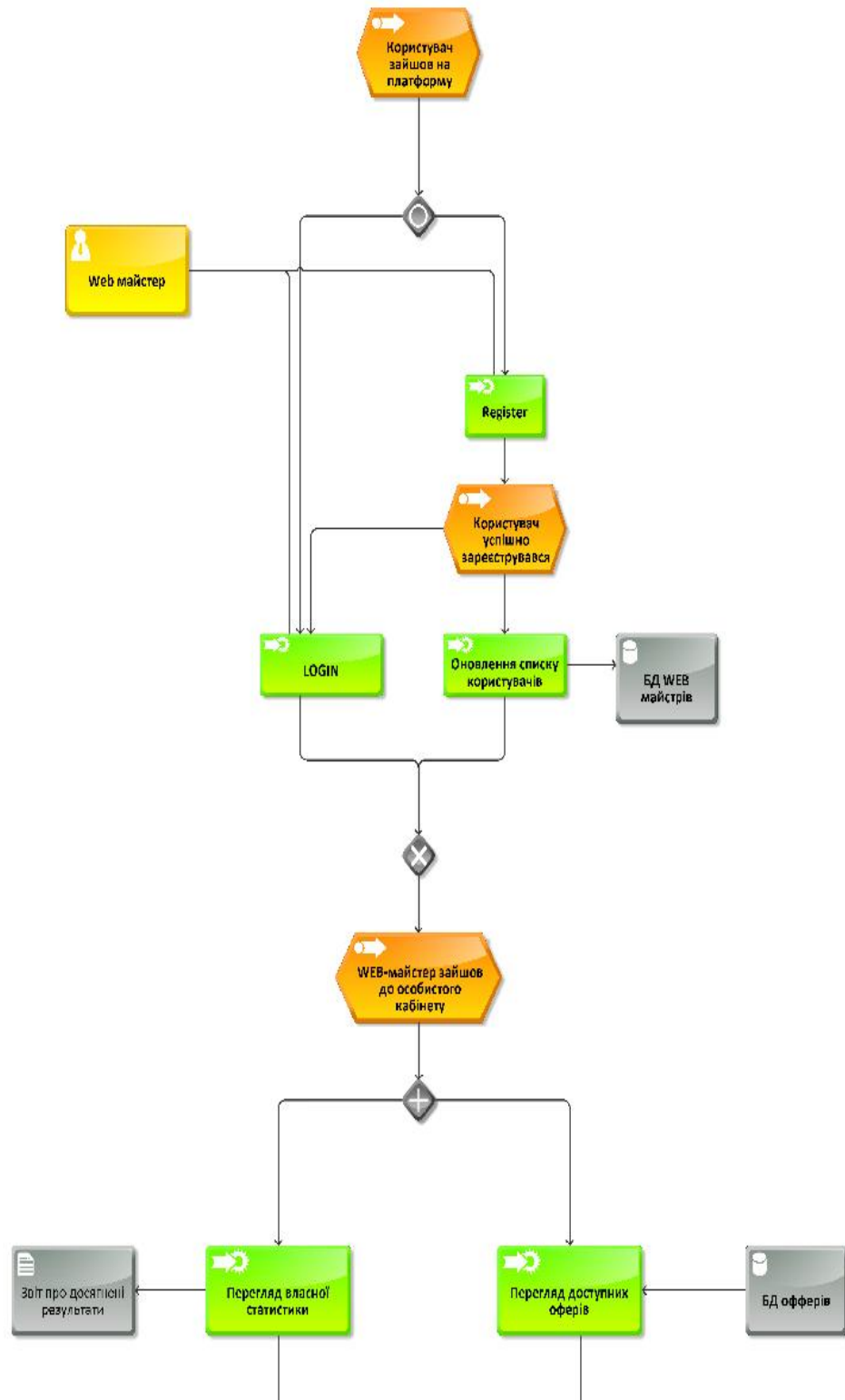


Рисунок 2.2 — Діаграма Event-Driven Process Chain

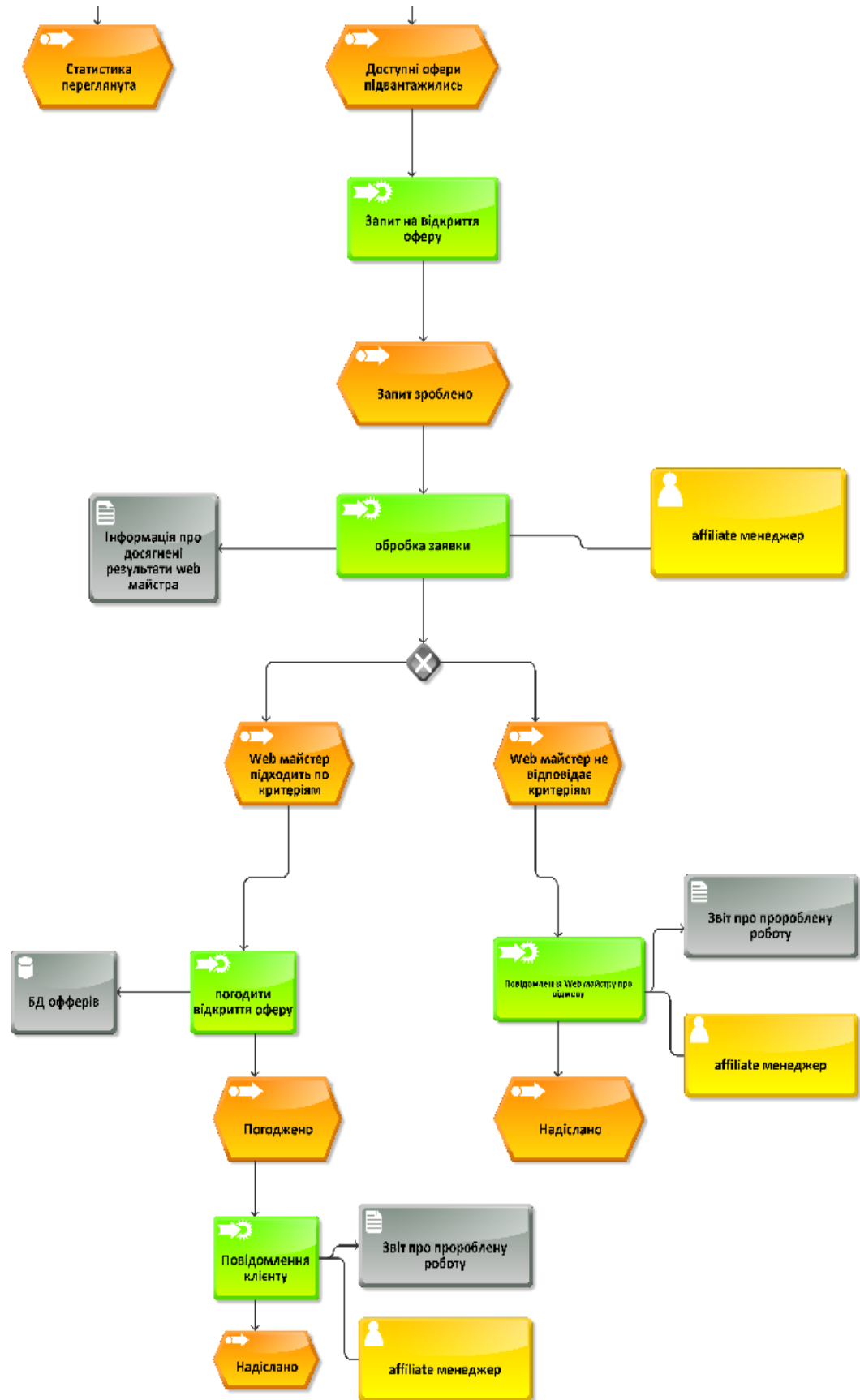


Рисунок 2.2 — продовження діаграми Event-Driven Process Chain

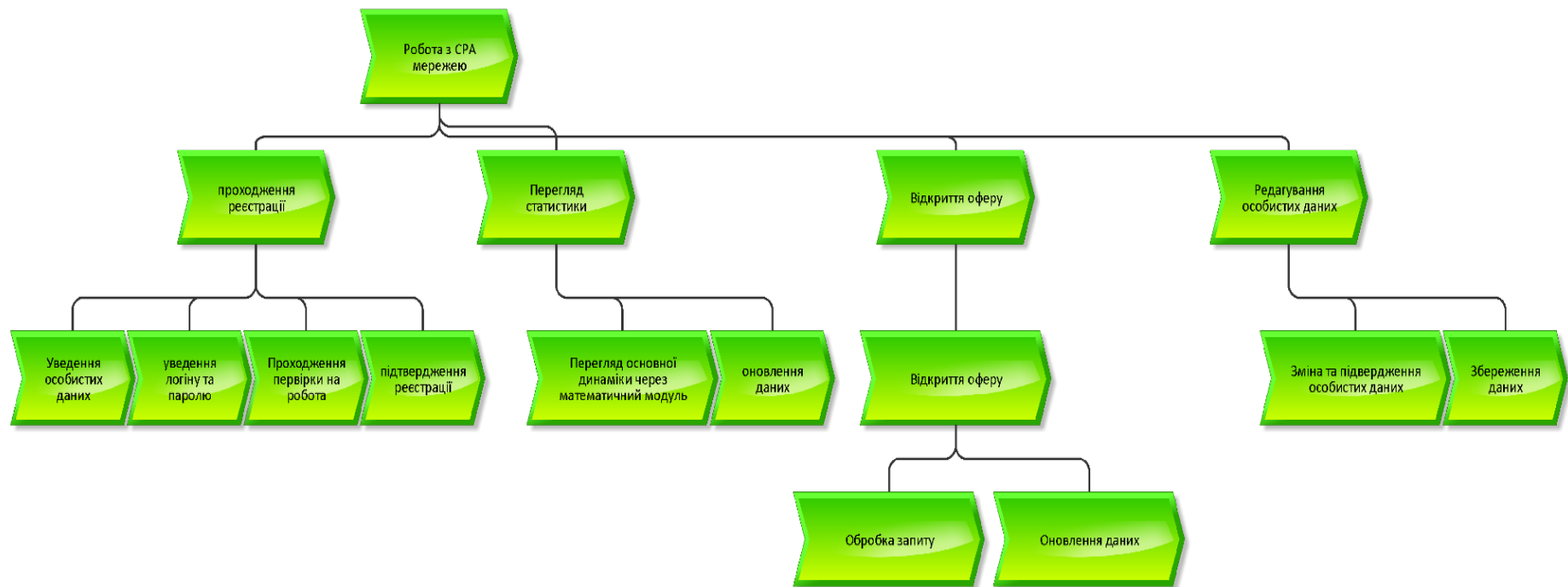


Рисунок 2.3 — Діаграма ланцюжка процесів Value Added Chain

2.1.2 IDEF0 процесу відкриття оферу

Представимо виконувані процеси у предметному середовищі за допомогою діаграм «ЯК БУДЕ» у нотації IDEF0, тобто вже з використанням web застосунку.

Спочатку будуємо контекстну діаграму, де вихідні дані це особисті та авторизаційні дані web майстра, а на виході отриманий звіт по статистиці оферу і дані про офер веб майстра.

Елементом керування є закон України про «Обробку і зберігання персональних даних», а також користувацькі умови та положення, так як користувач дає нам згоду на обробку його персональних даних, також збір статистики його веб переглядів і ми гарантуємо їх безпеку та захист.

Механізмами у даному випадку є CPA мережа та affiliate менеджер.

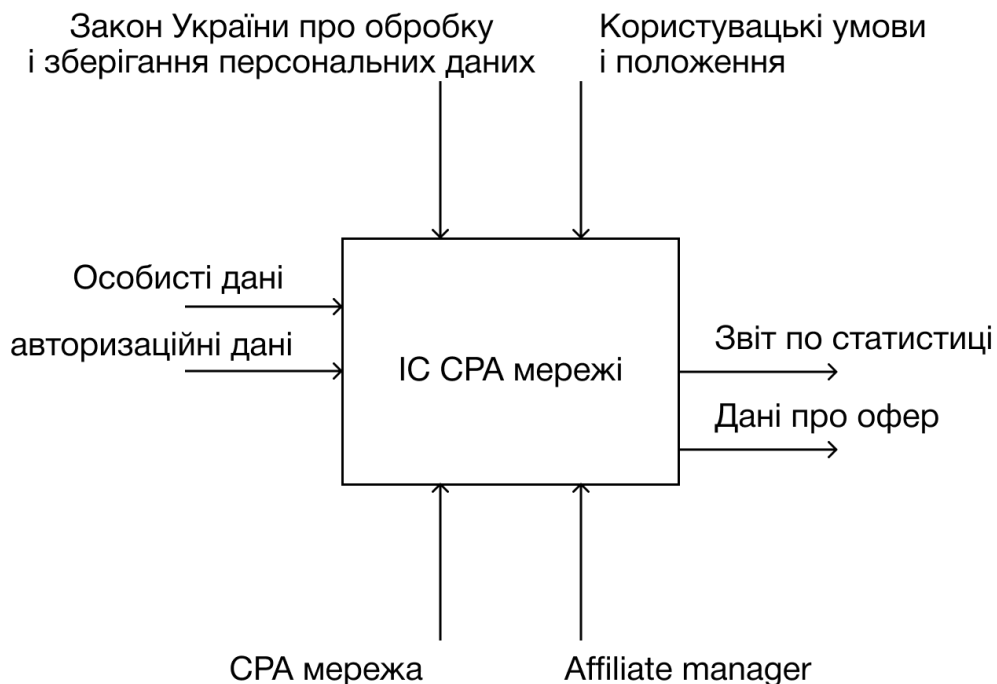


Рисунок 2.4 — Контекстна діаграма ЯК БУДЕ

На наступному етапі проведемо декомпозицію основних процесів діаграми ЯК БУДЕ (рис. 2.5).

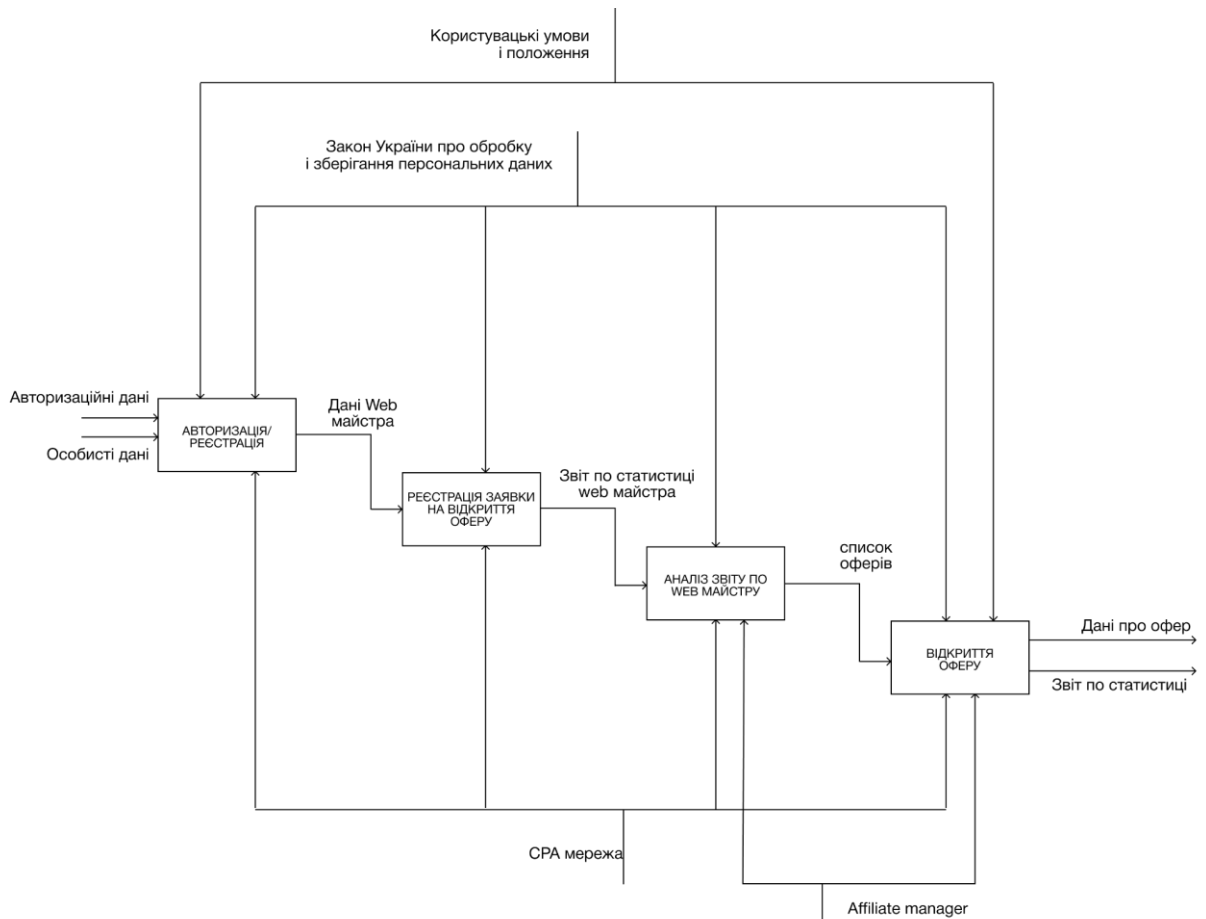


Рисунок 2.5 — Деталізована IDEF0 діаграма ЯК БУДЕ

Спочатку проходить авторизація користувача в системі, після чого дані користувача передаються в системі далі на реєстрацію заявки на відкриття оферу, система і менеджер аналізують статистичні показники веб майстра, на основі чого можуть приймати рішення щодо відкриття оферу для нього. На виході отримуємо офер, його дані і звіти про статистику.

2.1.3 Архітектура інформаційної системи

Наступний крок це побудова архітектури системи, що включає в себе модулі відповідно до функцій системи. Загалом присутні 3 модулі: клієнтський, математичний, модуль affiliate менеджера (рис. 2.6).

Клієнтський модуль включає модуль обробки персональних даних, а також пошуку та фільтрування оферів відповідно до потреб користувача

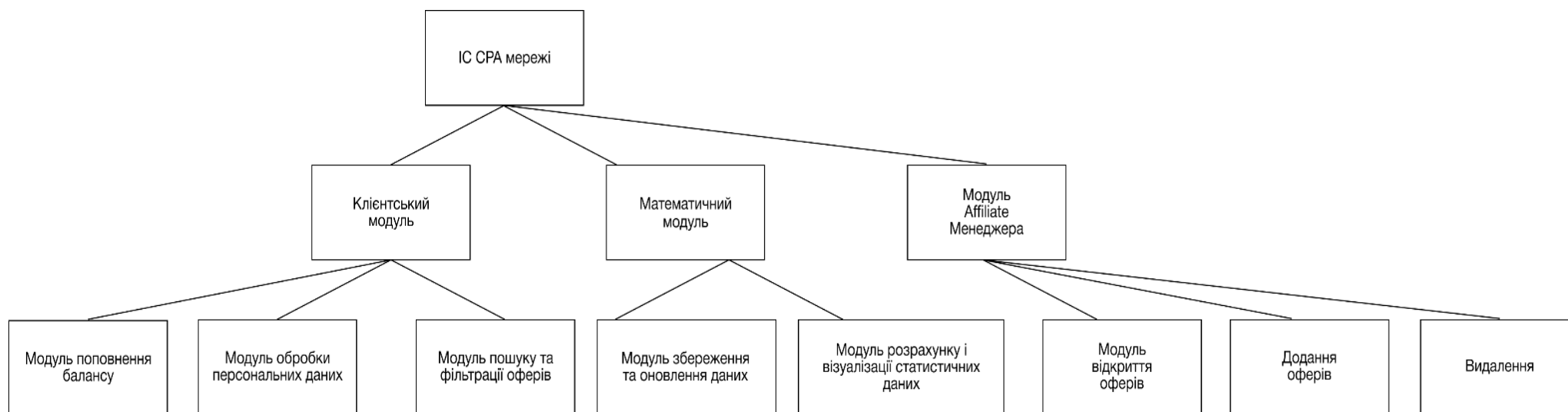


Рисунок 2.6 — Архітектура Інформаційної системи

2.2 Математичне забезпечення web застосунку для CPA мережі

Для математичного забезпечення використовуються метрики із підключених особистих кабінетів Google та Facebook Ads, тобто дані про переходи за рекламним оголошенням, відслідковування дій користувача напрямку пов'язані з API цих сервісів. Для підключення потрібна tracking link з налаштувань оферу (рис. 2.7).

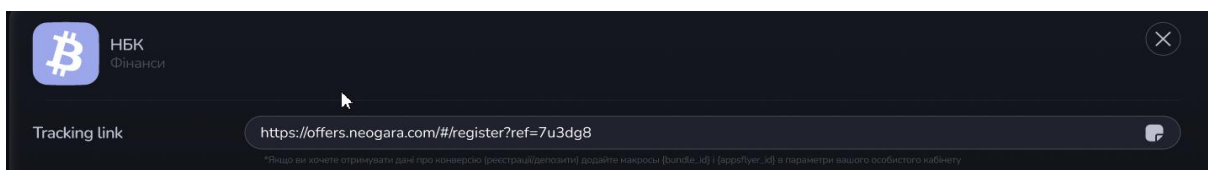


Рисунок 2.7 — Tracking link оферу

Далі потрібно підключити сам кабінет Google або Facebook, для цього потрібно скопіювати ID кабінету або рекламної кампанії та вставити у відповідне поле (рис. 2.8, 2.9).

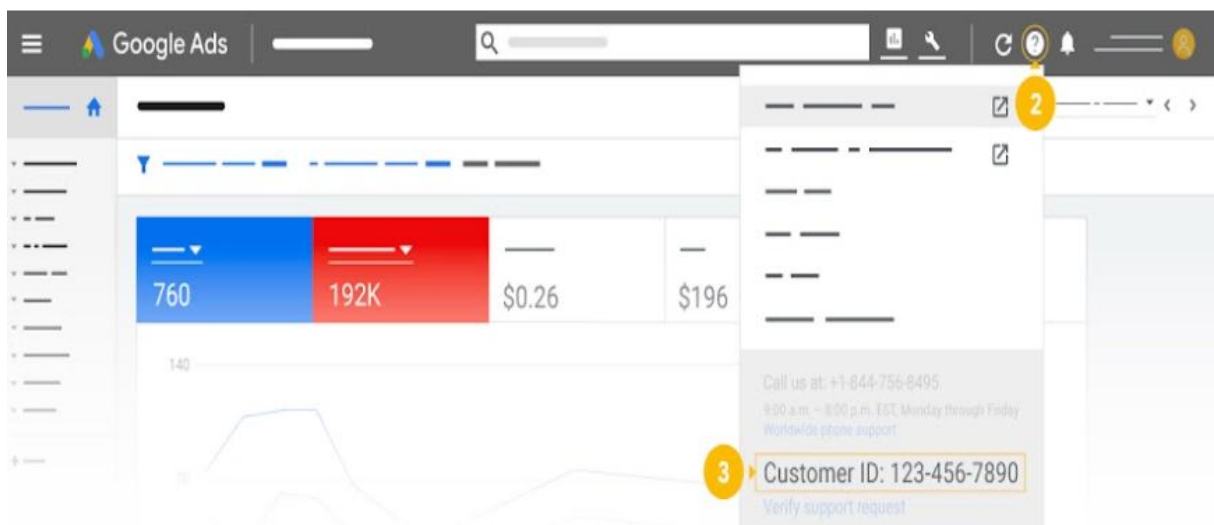


Рисунок 2.8 — Google Ads ID



Рисунок 2.9 — Підключення в застосунку

Після створення оферу і підключення його метрик, система застосовує математичні формули розрахунку ефективності роботи рекламної кампанії, а

саме CR (коефіцієнт конверсії), а також Visit to reg (коефіцієнт ефективності реєстрацій щодо прибутків).

$$CR = Leads/Hits * 100$$

$$Visit\ to\ reg = Deposits/Leads * 100$$

2.3 Інформаційне забезпечення web застосунку для CPA мережі

2.3.1 Розробка інформаційного забезпечення

Так як система оперує великими масивами даних, що стосуються оферів, користувача та налаштувань, потрібно описати і створити базу даних.

Спочатку розробимо концептуальну модель бази даних нотації Чена (рис. 2.10), де будуть відображені сутності системи:

1. Web майстер
2. Affiliate manager
3. Офер
4. Мій офер
5. Конверсія

Кожна з них має свої атрибути, та всі сутності пов'язані між собою. Кожен web майстер має свого менеджера, має доступ до списку оферів, а після відкриття оферу, може переглядати свої офери і статистику, яка в свою чергу і є конверсією.

Таблиця 2.1 Опис зв'язків між сутностями

| № | Сутності, що утворюють зв'язок | Тип зв'язку | Пояснення |
|---|-----------------------------------|------------------|--|
| 1 | Affiliate manager- Web майстер | Один до багатьох | Один менеджер може бути закріплений за багатьма користувачами, але кожен користувач має тільки одного менеджера. |
| 2 | Web майстер – Мої офери | Один до багатьох | Кожен користувач може мати багато оферів, але один офер може бути закріплений тільки за одним користувачем |

| | | | |
|---|-----------------------|------------------|---|
| 3 | Мої офери – Конверсії | Один до багатьох | Один мій офер має загальну статистику, що вираховується від багатьох конверсій. |
| 4 | Мої офери – Офери | Один до одного | Кожен офер спочатку не закріплений ні за ким, після відкриття цей ще офер передає свої параметри і переходить до таблиці моїх оферів. |

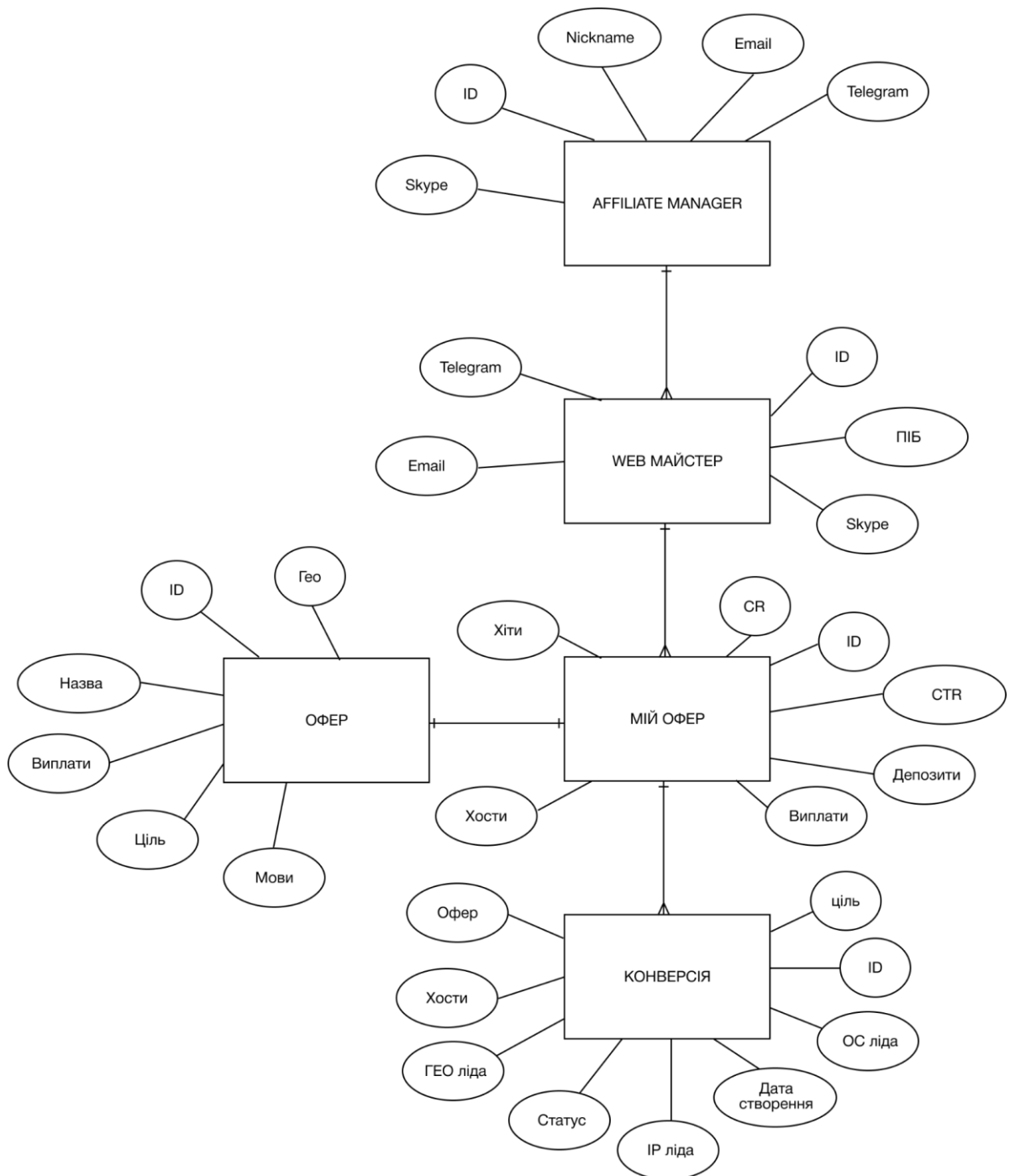


Рисунок 2.10 — Концептуальна модель бази даних

Наступною є даталогічна модель (рис. 2.11), яка більш детально ілюструє зв'язки між сутностями бази даних із вказанням ключових полів таблиць.

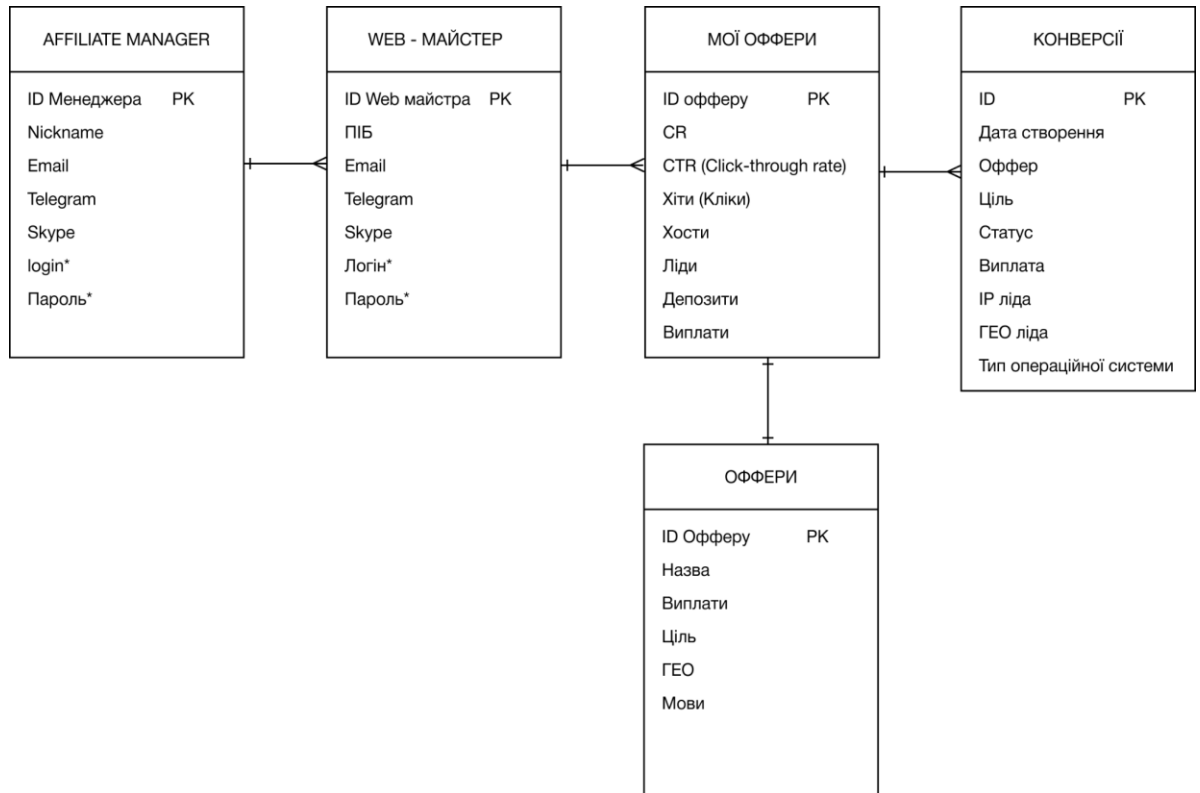


Рисунок 2.11 – Даталогічна модель бази даних

Також необхідно сформулювати обмеження для атрибутів, при розробці фізичної моделі бази даних, що робиться на основі даталогічної моделі. Обмеження описані у таблиці 2.2.

Таблиця 2.2 Опис атрибутів таблиць

| № п/п | Назва елемента даних | Тип елемента даних | Обов'язкове значення | Обмеження | Ключ |
|--|----------------------|--------------------|----------------------|-----------|------|
| Таблиця Web майстер «веб майстер» | | | | | |
| 1 | ID | Ціле число | так | 4 | ПК |
| 2 | password | Символьний | так | 20 | |
| 3 | login | Символьний | так | 20 | |
| 4 | Email | Символьний | так | 20 | |

| | | | | | |
|---|----------|----------------|-----|-----|-----------|
| | | й | | | |
| 5 | Skype | Символьни й | ні | 20 | |
| 6 | Telegram | Символьни й | ні | 20 | |
| Таблиця Affiliate manager «Affiliate менеджер» | | | | | |
| 1 | ID | Ціле число | так | 4 | ПК |
| 2 | password | Символьни й | так | 20 | |
| 3 | login | Символьни й | так | 20 | |
| 4 | Email | Символьни й | так | 20 | |
| 5 | Skype | Символьни й | ні | 20 | |
| 6 | Telegram | Символьни й | ні | 20 | |
| Таблиця Offers «Офери» | | | | | |
| 1 | ID | Ціле число | так | | ПК |
| 2 | name | Символьни й | так | 20 | ЗК |
| 3 | cost | Ціле число | так | | ЗК |
| 4 | goal | Символьни й | так | 20 | ЗК |
| 5 | country | Символьни й | так | | ЗК |
| 6 | language | Символьни й | так | | ЗК |
| 7 | info | Символьни й | ні | 150 | ЗК |
| Таблиця MyOffers «Мої Офери» | | | | | |
| 1 | ID | Ціле число | так | | ПК, ЗК |
| 2 | name | Символьни й | так | 20 | ЗК |
| 3 | cost | Ціле число | так | | ЗК |
| 4 | goal | Символьни й | так | 20 | ЗК |
| 5 | country | Символьни й | так | | ЗК |
| 6 | language | Символьни й | так | | ЗК |
| 7 | info | Символьни | ні | 150 | ЗК |

| | | | | | |
|--|-------------|----------------|-----|-----|----|
| | | й | | | |
| 1 | CR | Числовий | так | 3 | |
| 2 | hits | Ціле число | так | 10 | |
| 3 | leads | Ціле число | так | 10 | |
| 4 | deposits | Ціле число | так | 10 | |
| 5 | host | Ціле число | так | 10 | |
| 6 | proc | Ціле число | так | 20 | |
| Таблиця Conversions «Конверсії» | | | | | |
| 1 | ID | Ціле число | так | 150 | ПК |
| 2 | Date create | Дата | так | 9 | |
| 3 | offer | Ціле число | так | 10 | ЗК |
| 4 | goal | Символьни й | так | 10 | |
| 5 | status | Символьни й | так | 10 | |
| 6 | money | Ціле число | так | 10 | |
| 7 | lead IP | Символьни й | так | 20 | |
| 8 | lead GEO | Символьни й | так | 50 | |
| 9 | lead OS | Символьни й | так | 10 | |

2.4 Висновки до другого розділу

Було проведено функціональний аналіз та розроблено архітектуру системи за допомогою побудови дерева функцій, було змодельовано бізнес-процеси інформаційної системи діаграмами Event-Driven Process Chain та Value Added Chain Diagram, що ілюструють логіку роботи програмного застосунку. Також було представлено діаграми IDEF0 "ЯК БУДЕ", які показують процеси відкриття оферів для веб майстрів за допомогою інформаційної системи.

На наступному етапі була розроблена архітектура системи, що відображає внутрішні модулі та підсистеми, а також взаємозв'язок між ними.

Було досліджено процес застосування математичних розрахунків і отримання даних від API сервісів Google та Facebook.

Було розроблене інформаційне забезпечення, а саме виявлені та описані сутності інформаційної системи за допомогою концептуальної логічної моделі бази даних, а також детально описані атрибути полів таблиць.

РОЗДІЛ 3. ПРОГРАМНЕ ЗАБЕЗПЕЧЕННЯ WEB ЗАСТОСУНКУ ДЛЯ CPA МЕРЕЖІ

3.1 Обґрунтування вибору програмних засобів

Для розробки були використані наступні інструменти :

- Live Server
- Figma
- Google Analytics
- Facebook Ads
- JavaScript
- CSS
- HTML
- Bootstrap
- Visual Studio Code

Visual Studio Code — це один з найбільш популярних редакторів коду, розроблений корпорацією Microsoft, це швидка, зручне і доволі адаптивне програмне середовище, яке широко використовується в сучасній розробці веб сайтів. Тож я вирішив використувувати саме його для розробки web застосунка.

Figma — Найсучасніший засіб для створення макетів інтерфейсів сайтів, веб або мобільних додатків. Figma виділяється своїм порівняно широким функціоналом, можливістю використовувати плагіни для створення ілюстрацій, іконок, макетів, та в проектуванні складних дизайн систем, прототипуванню інтерфейсів та дослідження користувацького досвіду. Figma Дозволяє розробити проект одночасно з колегою в реальному часі, без витрати часу на перекидання файлів. Як дизайнер з досвідом роботи, можу рекомендувати використовувати саме її.

Bootstrap — вільний набір інструментів для сайтів і додатків. Використовувався для створення динамічної адаптивності сайтів і Web-застосунків.

Live Server — додаткове програмне забезпечення для Visual Studio Code, що надає можливість емулювати сервер на своєму комп'ютері.

Google Analytics — сервіс, що надається Google, для створення детальної статистики відвідувачів веб-сайтів. Статистика збирається на сервері Google, користувач лише розміщує JS-код на сторінках свого сайту. Код відстеження спрацьовує, коли користувач відкриває сторінку у своєму веб-браузері (за умови дозволеного виконання Javascript у браузері).[6]

Facebook Ads — це інструмент Facebook, який дає змогу створювати рекламу у Facebook і керувати нею. Тут ви можете переглядати всі свої кампанії, набори реклами й окремі оголошення у Facebook, вносити до них зміни й дивитися результати. [7]

JavaScript — мультипарадигменна мова програмування. Підтримує об'єктно-орієнтований, імперативний та функціональний стилі. Є реалізацією специфікації ECMAScript. [8]

CSS — спеціальна мова, що використовується для запису оформлення сторінок, написаних мовами розмітки даних.

HTML — мова розмітки даних.

Проаналізувавши технічне завдання та спланувавши етапи розробки, я почав виконувати завдання.

3.2 Структура програмного забезпечення

Розглянемо структуру розробленого програмного забезпечення web застосунку для CPA мережі (рис. 3.1).

файли розподілені за розширеннями:

- .html
- .js
- .css

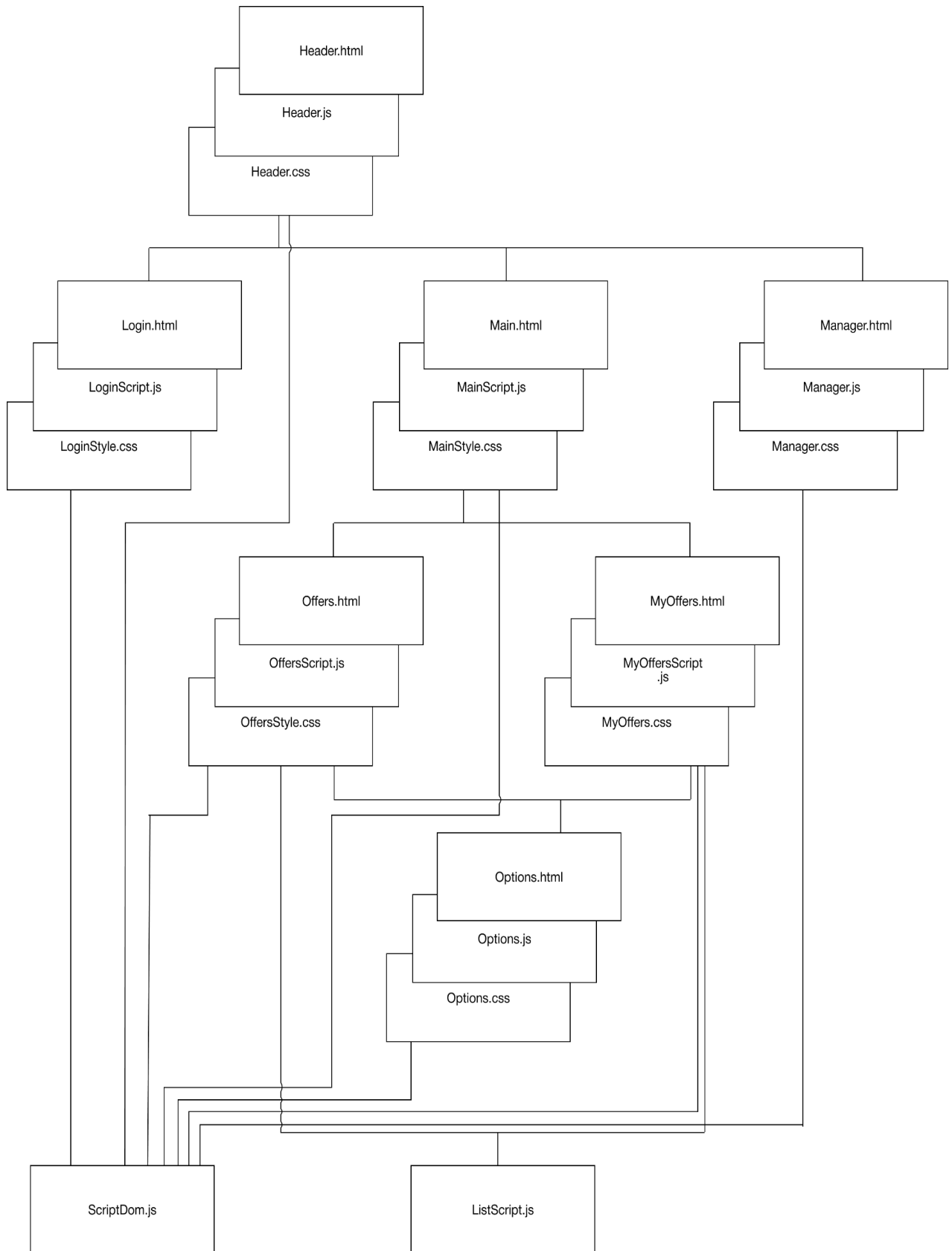


Рисунок 3.1 – Структурна схема програмних модулів

Опишемо специфікацію програмних модулів, вихідні і вхідні дані, пояснення щодо функціоналу (табл. 3.1).

Таблиця 3.1 Специфікація програмних модулів

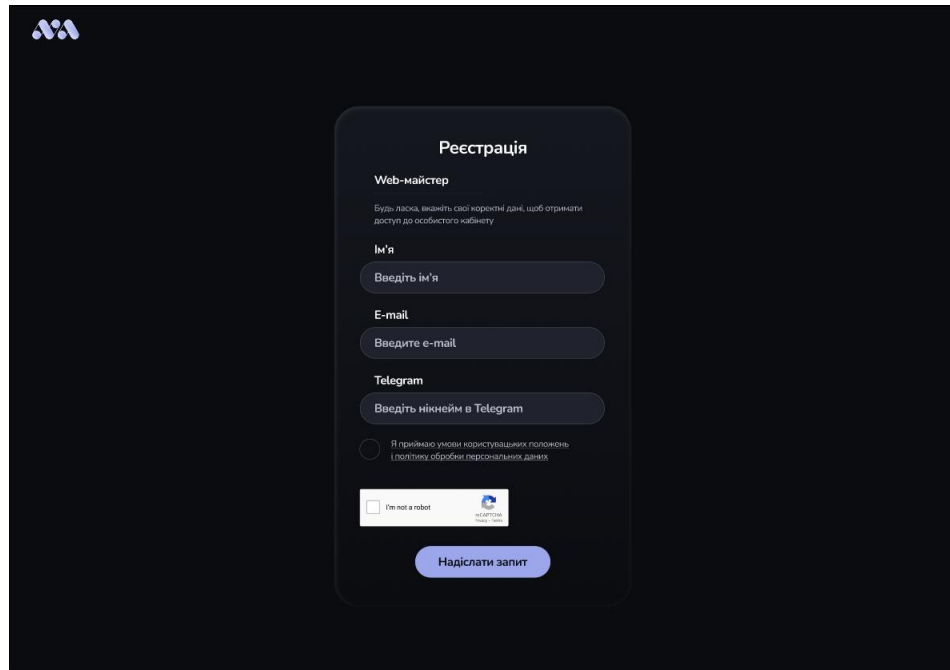
| Модуль | Опис |
|---------|---|
| Header | <p>Меню системи, що потрібне для зручної навігації, завжди супроводжує користувача, тобто модуль переходу та навігації між сторінками системи.</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Бажана сторінка <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Інтерфейс відповідної сторінки |
| Login | <p>Модуль входу до системи, включає реєстрацію та вхід.</p> <p>Вхідна інформація:</p> <p>Дані користувача:</p> <ul style="list-style-type: none"> - при вході: логін та пароль - при реєстрації: ім'я, логін, пароль, email, telegram нікнейм <p>Вихідна інформація:</p> <p>Зареєстрований та залогінений користувач, персональні дані про нього.</p> |
| Main | <p>Модуль головної сторінки, де відображається збірна статистика, візуалізація даних та особиста інформація користувача.</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Вид доступу - персональні дані користувача <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Статистика по рекламним кампаніям щодо відкритих оферів веб майстра - Графічна візуалізація основних показників - Дані про наявність відкритих оферів - Дані профілю та особистого менеджера |
| Manager | <p>Модуль сторінки з часто запитуваними проблемами, а також контактами особистого менеджера..</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Вид доступу - дані про менеджера <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - інформація для зв'язку з affiliate менеджером - відповіді на Frequently Asked Questions |

| | |
|---------------|---|
| Offers | <p>Модуль сторінки оферів доступних для відкриття, відображаються всі офери.</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Вид доступу, таблиця оферів <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Відображення усіх оферів з БД, з можливою фільтрацією за критерієм - можливість запросити офер на відкриття |
| MyOffers | <p>Модуль сторінки моїх оферів доступних для відкриття, відображаються всі офери.</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Вид доступу, таблиця оферів <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Відображення усіх оферів веб майстра, з можливою фільтрації - Статистика рекламних кампаній по відкритим оферам |
| Options | <p>Модуль сторінки налаштування відкритого оферу.</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Вид доступу, вибраний офер <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Унікальне посилання Tracking link - Відкритий офер |
| ListScript.js | <p>Модуль сервісу, що відповідає за відображення полів з бази даних у веб застосунку</p> <p>Вхідна інформація:</p> <ul style="list-style-type: none"> - Дані про користувача, тип запити <p>Вихідна інформація:</p> <ul style="list-style-type: none"> - Результати запитів |
| ScriptDom.js | <p>Модуль сервісу для побудови архітектури системи у вигляді дерева, що спрощує орієнтацію в коді, оптимізує деякі процеси розробки, такі як заповнення бракуючих елементів, редагування відповідних полів за потреби.</p> |

3.3 Керівництво користувача

Для того щоб почати користуватись системою web майстер має пройти процес реєстрації або ж аутентифікації за допомогою логіну та паролю, якщо він вже має створений акаунт.

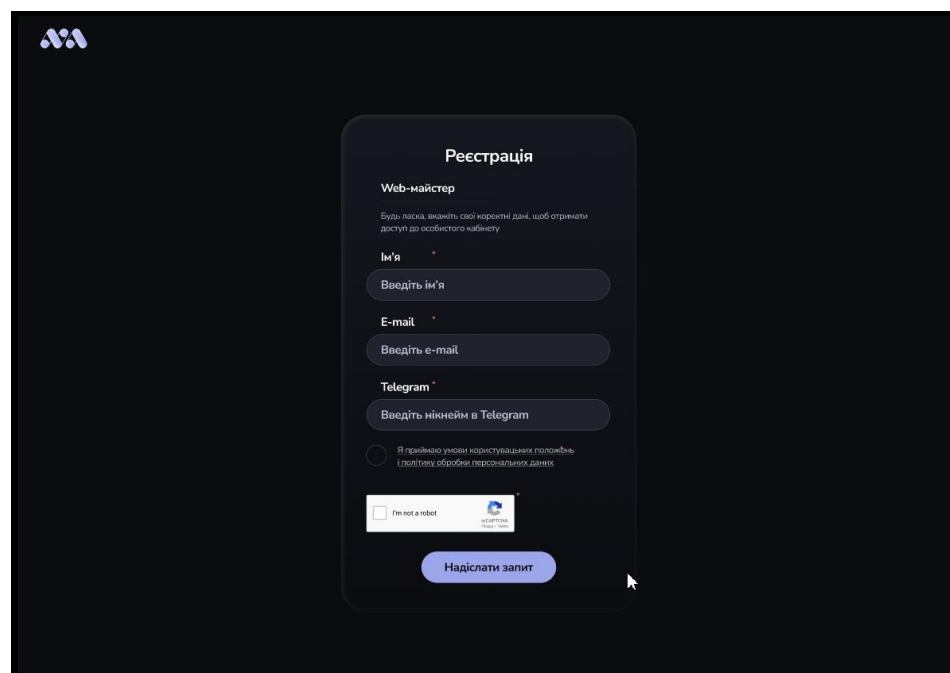
Під час реєстрації він має заповнити всі необхідні поля, пройти перевірку на бота за допомогою ReCaptcha та натиснути кнопку “зареєструватись”, після чого він зможе перейти на сторінку входу, де є обов’язкові для заповнення поля логіну та паролю. Після успішного заповнення та натиснення кнопки “увійти” система перевірить наявність акаунту в базі даних і користувач потрапить на головну сторінку системи.



The screenshot shows a registration form titled "Реєстрація" (Registration) under the heading "Web-майстер" (Web-master). The form includes the following elements:

- Instruction: "Будь ласка, введіть свої короткі дані, щоб отримати доступ до особистого кабінету" (Please enter your short details to get access to your personal cabinet).
- Field "Ім'я" (Name) with a placeholder "Введіть ім'я" (Enter name).
- Field "E-mail" with a placeholder "Введіть e-mail" (Enter e-mail).
- Field "Telegram" with a placeholder "Введіть нікнейм в Telegram" (Enter nickname in Telegram).
- A checkbox for "Я приймаю умови користування цими послугами і політику обробки персональних даних" (I accept the terms of use of these services and the personal data processing policy).
- A reCAPTCHA checkbox labeled "I'm not a robot".
- A blue button labeled "Надіслати запит" (Send request).

Рисунок 3.2 – Сторінка реєстрації



This screenshot is identical to Figure 3.2, but it highlights the required fields with red asterisks. The asterisks are placed above the "Ім'я" (Name), "E-mail", and "Telegram" labels, indicating that these fields are mandatory for registration.

Рисунок 3.3 – Повідомлення системи про обов’язкове заповнення полів

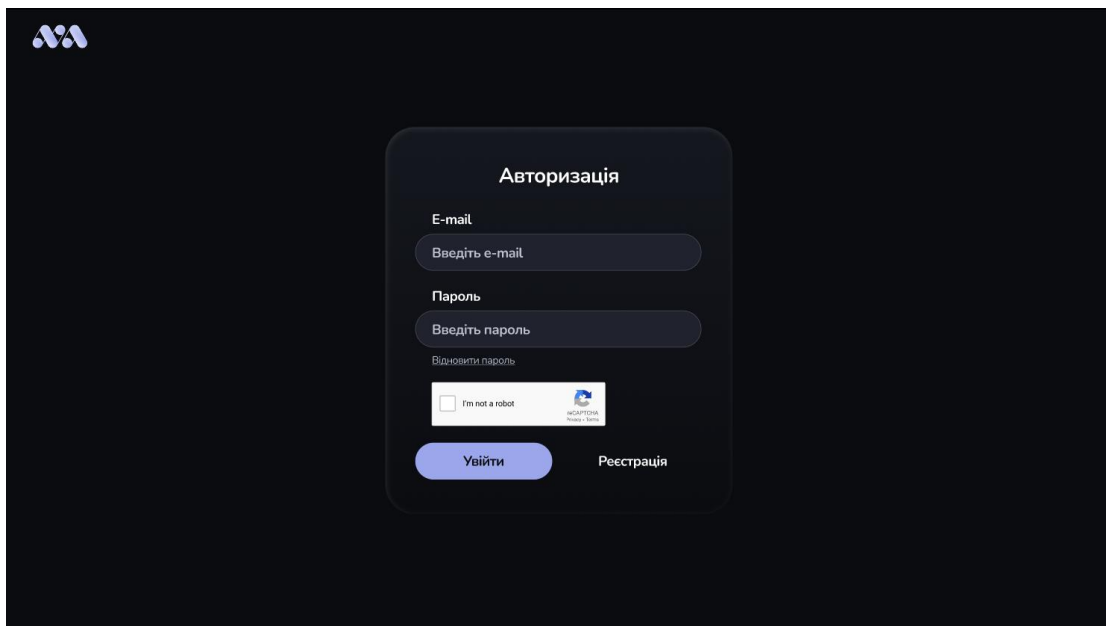


Рисунок 3.4 – Сторінка авторизації

На головній сторінці веб майстер має можливість відслідкувати актуальну статистику по його відкритим оферам, переглянути динаміку через візуальне представлення математичних формул і метрик рекламних кампаній (рис. 3.5).



Рисунок 3.5 – Головна сторінка

При наведенні на зони графіку, можна переглянути статистику за вибраний день.



Рисунок 3.6 – Відображення статистики за вибрану дату

У правій частині екрану веб майстер має можливість переглянути короткий список його відкритих оферів, а також офери, що оброблюються системою та affiliate-менеджером.

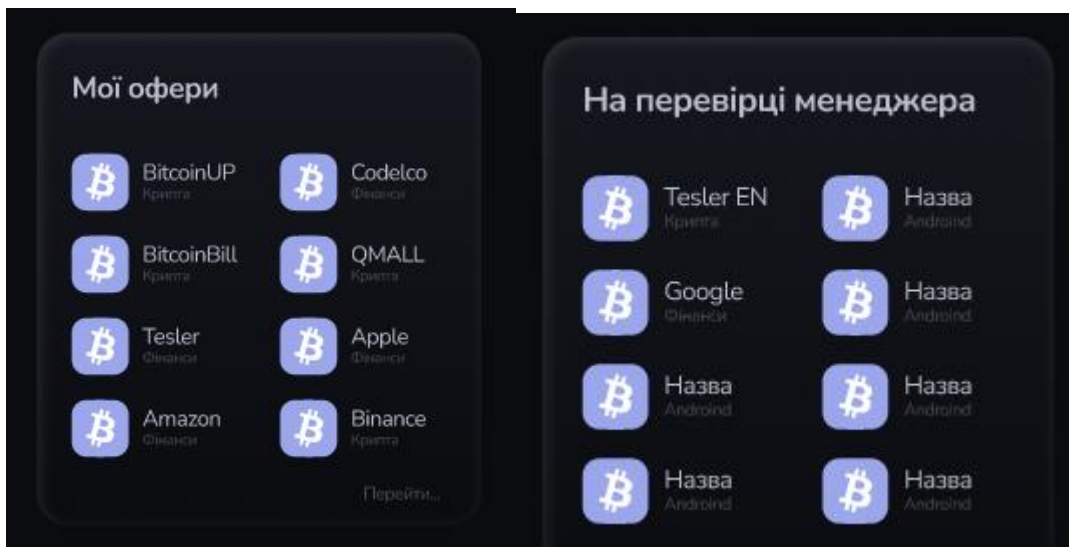


Рисунок 3.7 – Відображення коротких списків оферів

Натиснувши на кнопку “Поповнити” на фіолетовій картці web майстер може поповнити свій баланс зручним для себе способом.

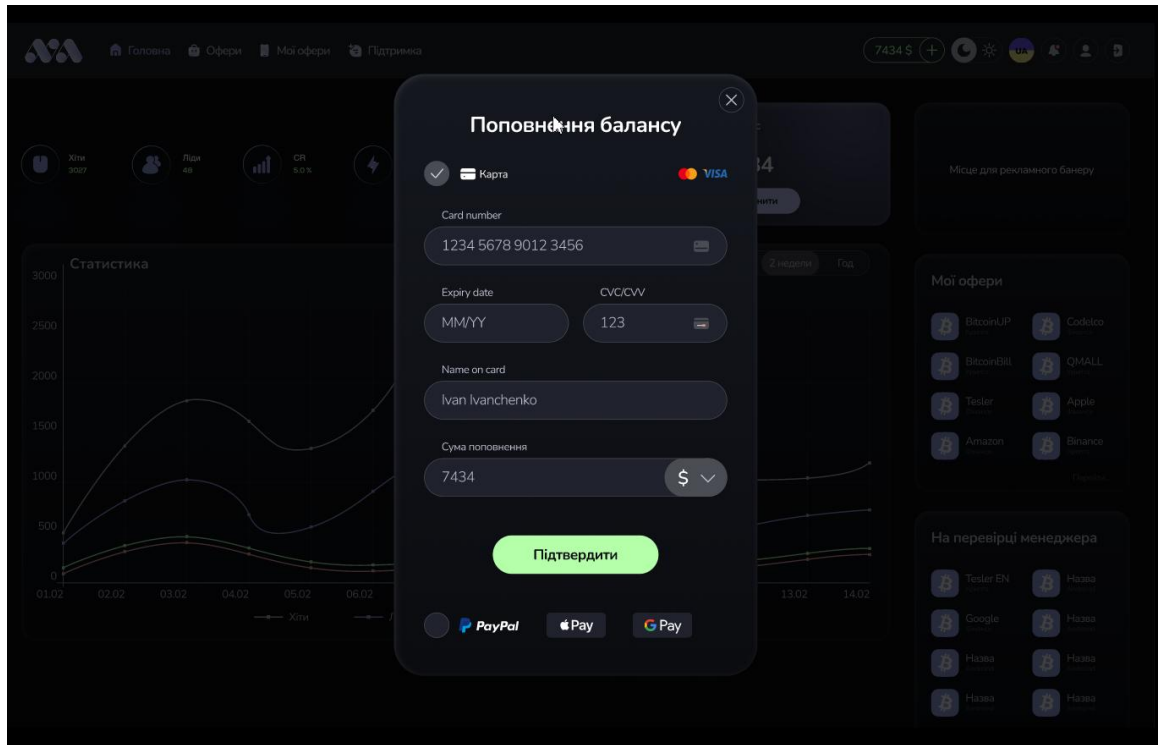


Рисунок 3.8 – Заповнення даних картки для поповнення

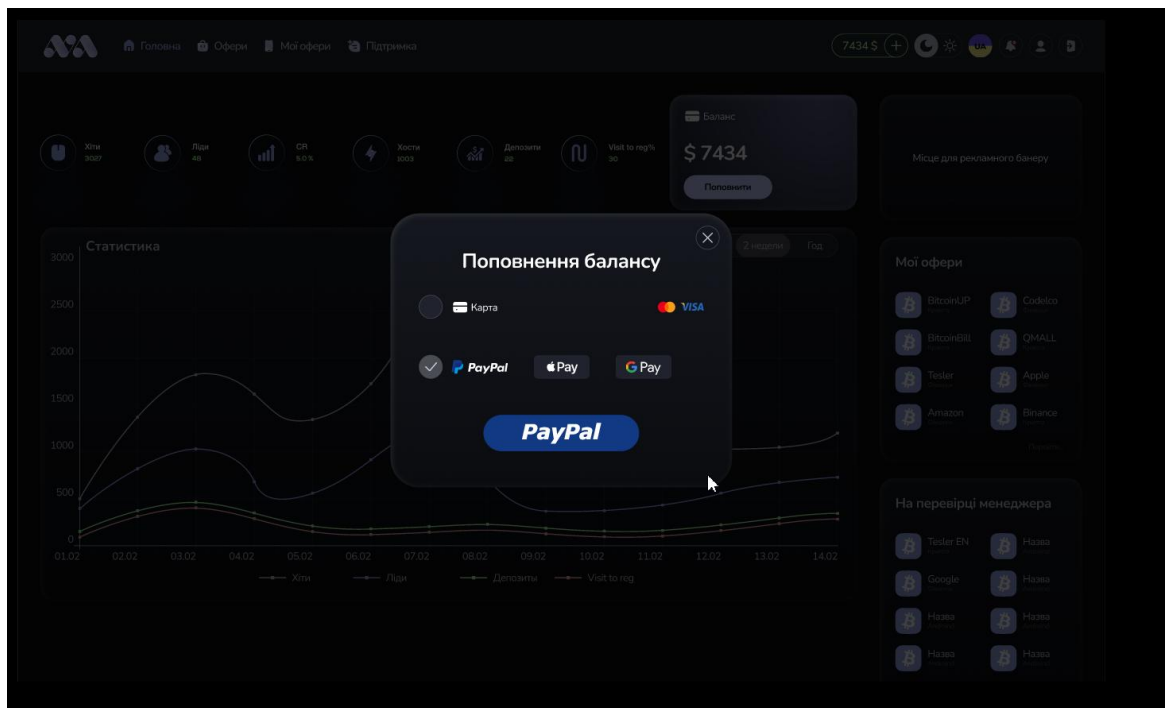


Рисунок 3.8 – Можливість скористатись іншим зручним способом поповнення

Натиснувши на кнопку “сонце і місяць”, розташовану в хедері, веб майстер може змінити кольорову тему застосунку.



Рисунок 3.9 – Світла кольорова тема застосунку

Також поряд з кнопкою зміни кольорової теми можна переглянути актуальні повідомлення системи, натиснувши кнопку дзвіночка.

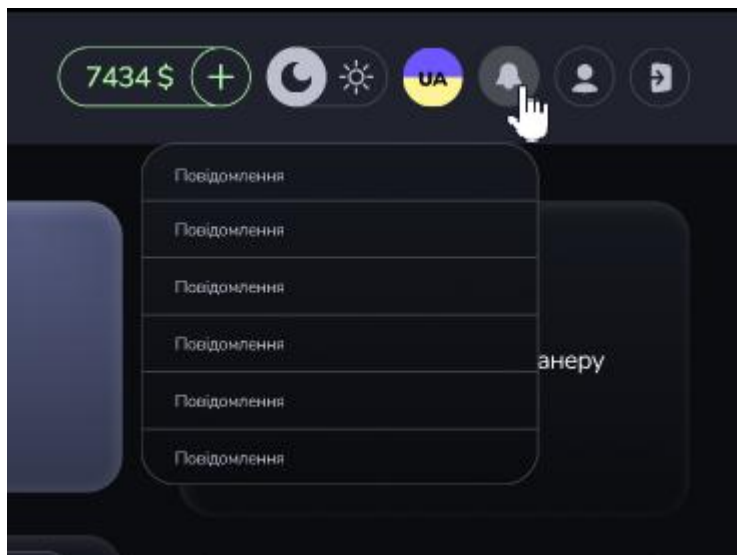


Рисунок 3.10 – Перегляд повідомлень

Поряд знаходиться кнопка переходу на сторінку з особистими даними, що вводились під час реєстрації з можливістю редагувати дані, а також доповнювати їх.

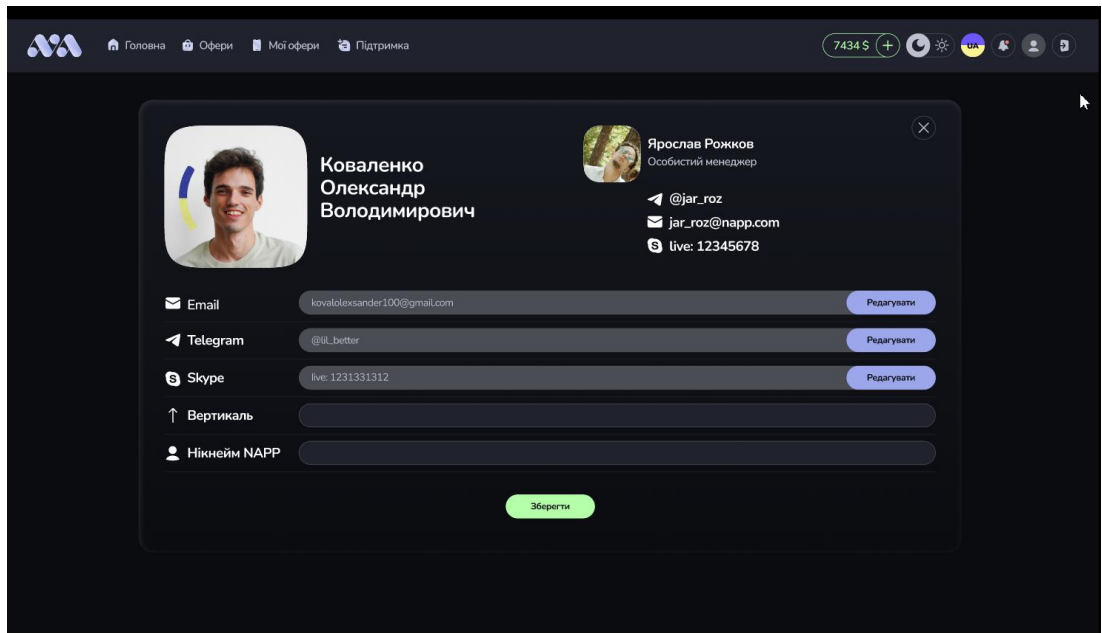


Рисунок 3.11 – Особистий кабінет

Також є можливість вийти із акаунту, натиснувши кнопку в правому верхньому куточку, в такому випадку ми потрапимо на сторінку авторизації.

Розглянемо більш детально основні функціональні сторінки застосунку. Між ними можна переключатись, використовуючи хедер застосунку.



Рисунок 3.12 – Хедер

Перейдемо на сторінку оферів, тут відображається список доступних до відкриття оферів, що мають свої відповідні атрибути. На даній сторінці реалізовано зручну систему фільтрації для швидкого пошуку оферу за потрібним критерієм або характеристикою (рис. 3.13). Кожному оферу відповідає кнопка відкриття оферу “запросити”, при натисненні якої відбувається підтвердження оферу зі списанням з балансу відповідної суми коштів, що буде використана для запуску рекламної кампанії, про що інформує відповідний поп-ап (рис 3.14)

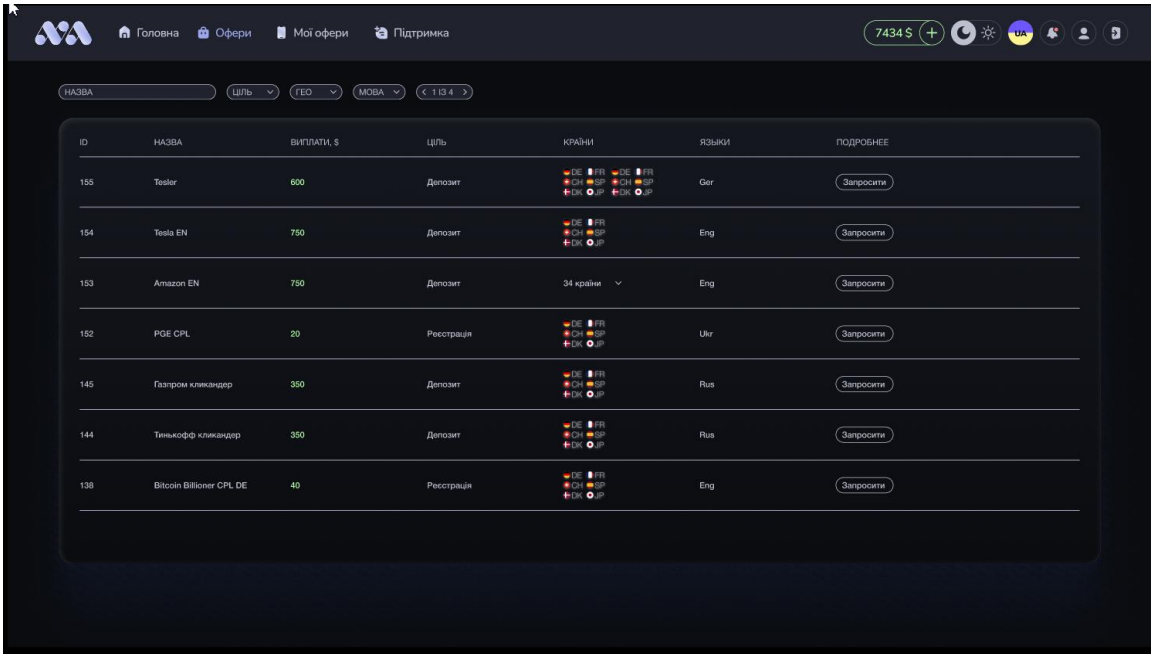


Рисунок 3.13 – Сторінка оферів

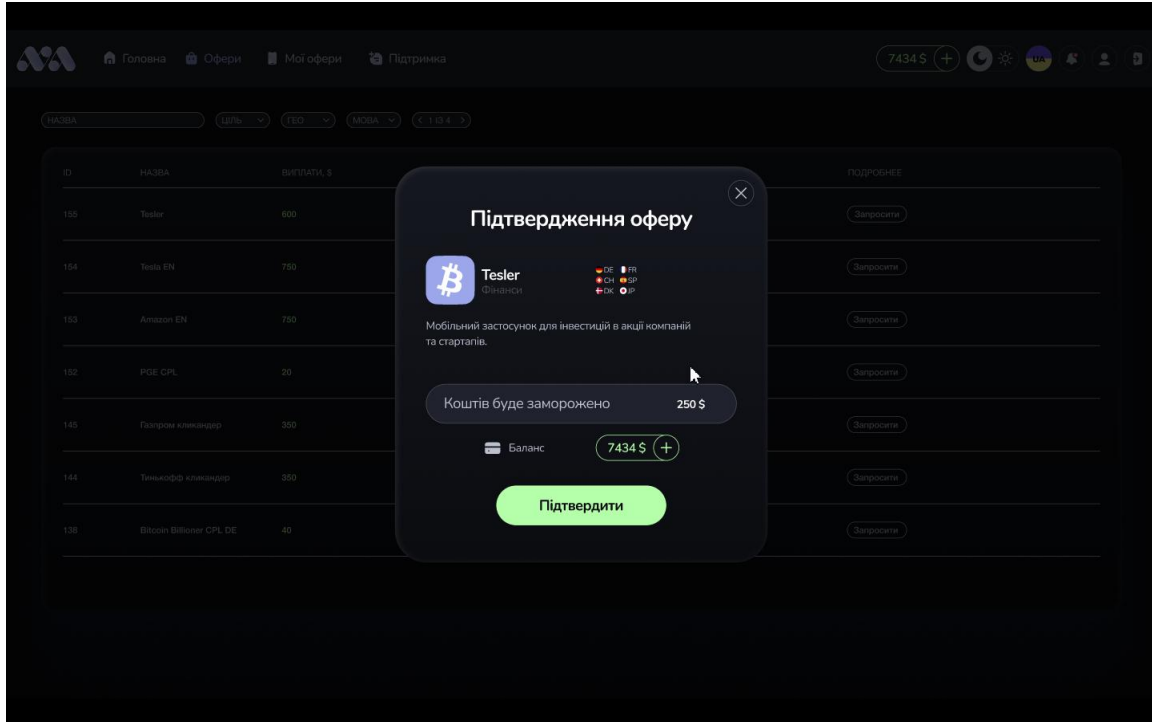


Рисунок 3.14 – Поп-ап підтвердження відкриття оферу

Після підтвердження відкриття оферу веб майстер має можливість відразу налаштувати рекламну кампанію(-і) по ньому з отриманням необхідних скриптів та Tracking Link від системи, а також підключити рекламну кампанію з Google Analytic або Facebook Ads для відслідковування статистики, налаштувати пуш повідомлення потенційним клієнтам. Пуш повідомлення бувають трьох типів:

- По події
- За розкладом
- Моментальні

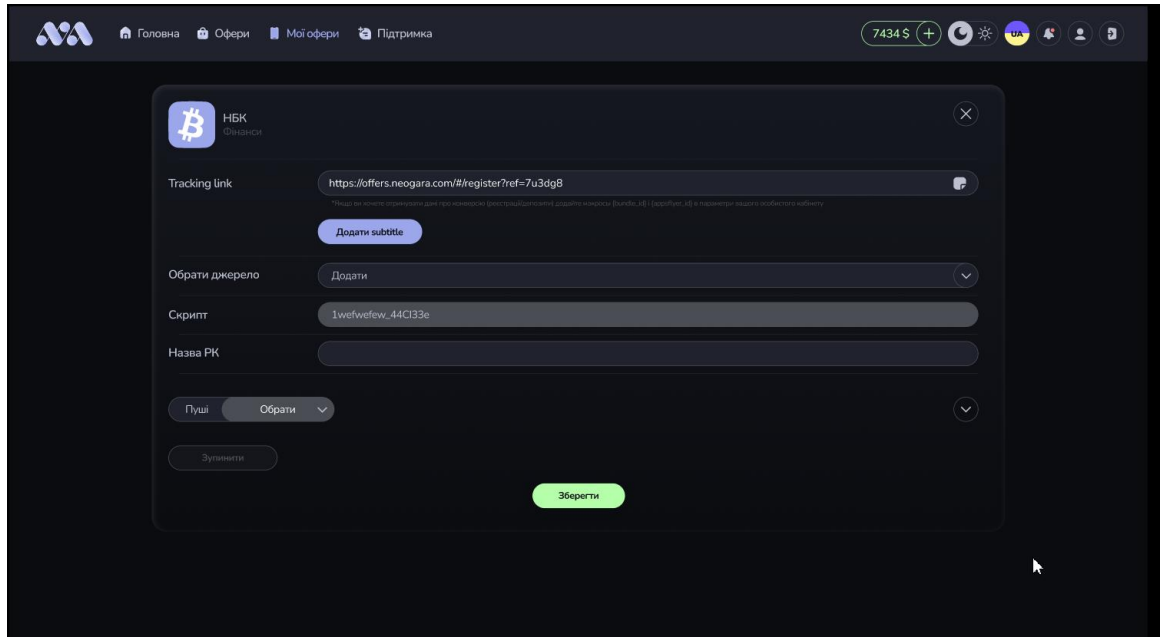


Рисунок 3.15 – Налаштування офери

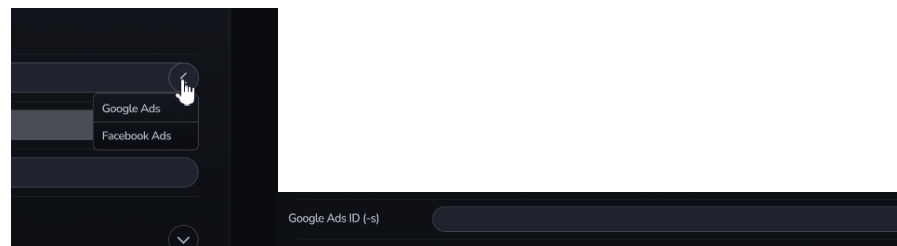


Рисунок 3.16 – Підключення Google Analytic

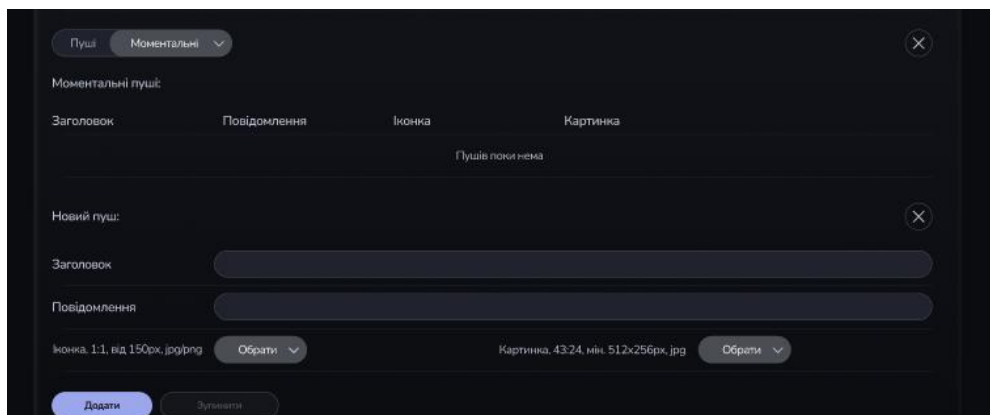


Рисунок 3.17 – Налаштування моментальних пушів

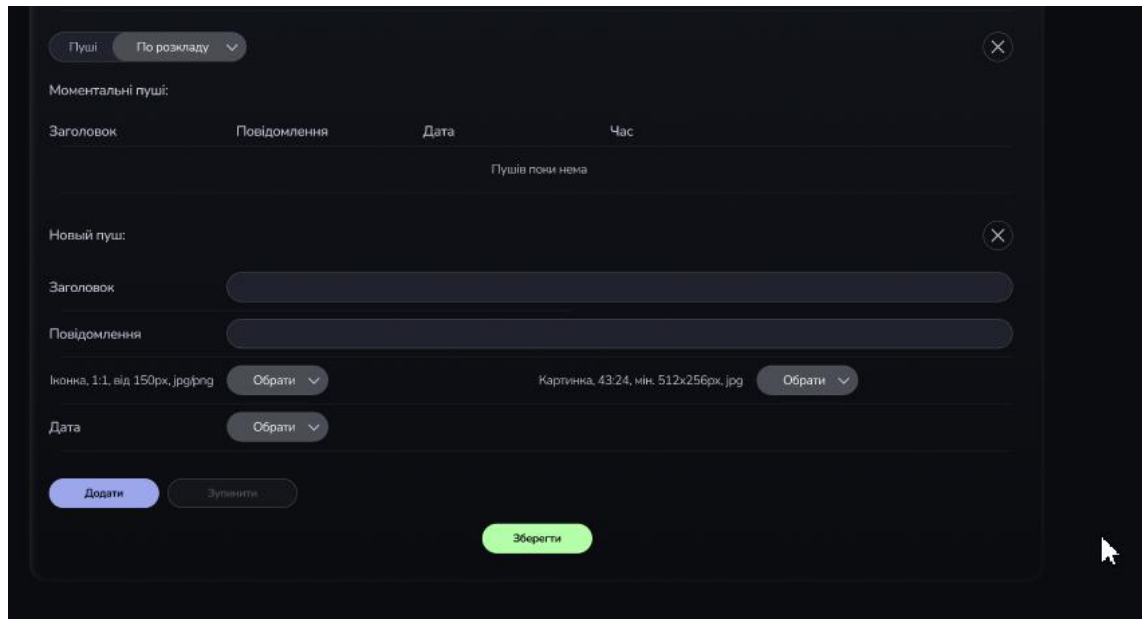


Рисунок 3.18 – Налаштування пушів за розкладом

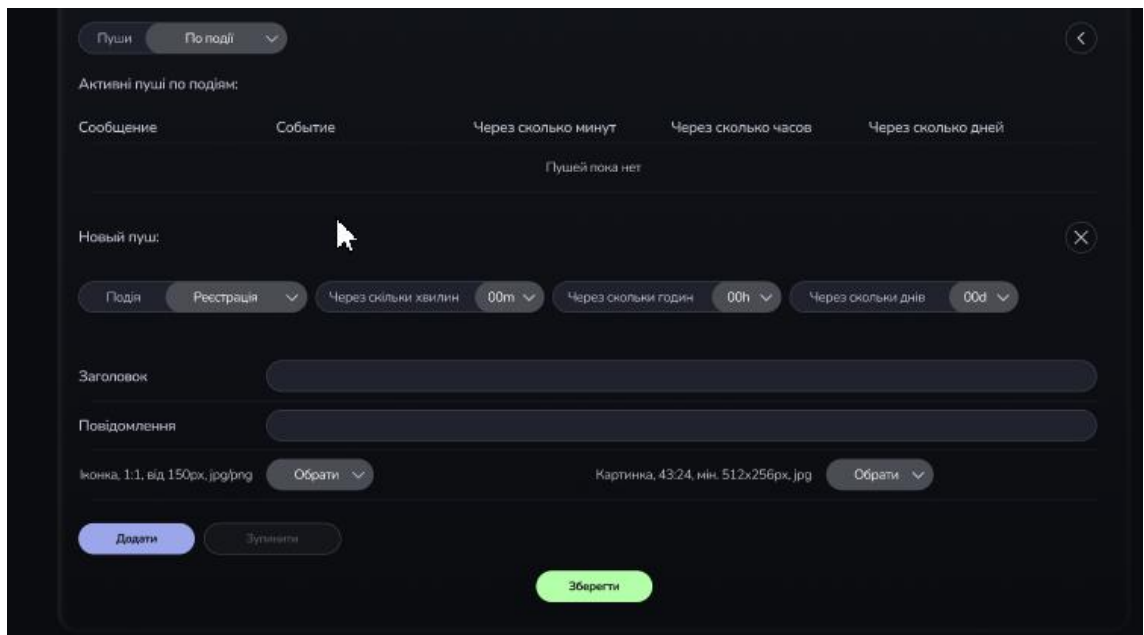



Рисунок 3.19 – Налаштування пушів по події

Після завершення налаштувань веб майстер натискає кнопку Зберегти, дані зберігаються в системі і вона починає відслідковувати статистику по відкритим оферам, також він одразу потрапляє на сторінку “Мої офери”, де він зможе переглянути конверсію кожного окремого офера. Зелені поля означають успішне виконання клієнтом цільових дій, сині поля це цільові дії що ще обробляються і червоні це цільові дії, які були виконані з помилкою, тобто веб майстер не отримує прибутку з них.

| ОФЕР | CR | ХІТИ | ХОСТИ | ЛІДИ | ДЕПОЗИТИ | ФІНАНСИ |
|-----------|--------|------|-------|-------------------------|---------------------------|-----------------------------|
| (0) TESLA | 14.64% | 2474 | 2000 | 902.5 449.25 1811 | 902.5 223.625 903.5 | 11732.5 8535.75 38031 |

Рисунок 3.20 – відображення статистики по оферам веб майстра

Також у веб майстра є можливість переглянути сторінку “Підтримка”, де є інформація про його особистого менеджера, та відповідний функціонал для перегляду відповідей на питання, що найчастіше виникають.


Рожков Ярослав
 Особистий менеджер

✉ jar_roz@gmail.com
 📞 @jar_roz
 🗣 live: 12345678

FAQ

- Питання
- Питання
- Питання
- Питання
- Питання
- Питання

Рисунок 3.21 – сторінка Підтримки

3.4 Огляд процесу тестування

Фінальною стадією розробки будь-якого програмного продукту є процес тестування застосунку, опишемо Test case у таблиці 3.3.

Таблиця 3.3 Опис Test case

| ID | Модуль | Подія | Очікувані результати |
|----|------------|---|--|
| 1 | Login.js | Реєстрація нового користувача: <ul style="list-style-type: none"> - Заповнити всі обов'язкові поля - Натиснути кнопку "Погоджуюсь з..." - Натиснути кнопку "ReCaptcha" - Натиснути кнопку "Відправити заявку" | <ol style="list-style-type: none"> 1. Користувач переходить на сторінку авторизації 2. Всі обов'язкові поля заповнені |
| 2 | Login.js | Вхід до особистого кабінету: <ul style="list-style-type: none"> - введення правильного логіну в поле логін - введення правильного паролю в поле пароль - натиснення кнопки "увійти" | <ol style="list-style-type: none"> 1. Успішна авторизація 2. Пароль відображено як крапки у полі паролю 3. Веб майстер переходить на головну сторінку |
| 3 | Login.js | Неправильні дані для авторизації: <ul style="list-style-type: none"> - Введення неправильного логіну - Ввести неправильного паролю - Ненатиснення кнопки "ReCaptcha" | <ol style="list-style-type: none"> 1. Користувач залишається на сторінці авторизації, незаповнені поля підсвічуються червоною зірочкою. |
| 4 | MyOffer.js | Перегляд нового відкритого офери: <ul style="list-style-type: none"> - Натиснути кнопку "запросити" - Підключити Google або Facebook Ads - Натиснути кнопку "Зберегти" | <ol style="list-style-type: none"> 1. Офер з'явився на сторінці "Мої офери" 2. Веб майстер потрапив на сторінку "Мої офери" 3. Веб майстер переглядає статистику по офери |

| | | | |
|---|------------|---|---|
| 4 | MyOffer.js | Новий відкритий офер неправильно підключено: <ul style="list-style-type: none"> - Натиснути кнопку “запросити” - Не підключити Google або Facebook Ads - Натиснути кнопку “Зберегти” | <ol style="list-style-type: none"> 1. Офер з’явився на сторінці “Мої офери” 2. Веб майстер потрапив на сторінку “Мої офери” 3. Статистика по оферу не відображається |
|---|------------|---|---|

3.5 Висновки до третього розділу

Для розробки програмного продукту було проведено ряд досліджень, у тому числі UX/UI експертиза. Були досліджені основні потреби користувача і знайдено ефективні рішення для покращення користувацького досвіду, а отже і ефективності його взаємодії з інформаційною системою.

Для розробки були обрані сучасні інструменти, вибір було обґрунтовано.

Під час розробки була побудована структура програмного веб застосунку і описані специфікації кожного модуля.

На завершальній стадії розробки було описано керівництво користувача з використанням знімків екрану з відкритим застосунком.

Окрім цього, на фінальному етапі був проведений огляд випадків тестування для перевірки функціонування, які були описані в таблиці.

Висновок

CPA маркетинг кожного дня дедалі більше набирає популярність, тисячі людей приймають рішення розвиватись і працювати у цій сфері, більшість крупних компаній вже користуються послугами партнерських програм. А це все означає, що наразі є актуальною тема розробки веб застосунку для CPA мережі, так як існуючі рішення не задовольняють потреб сучасного користувача. Новий програмний продукт, що закриває “болі” веб майстрів і при цьому є прозорим для рекламодавцям здатен змінити цінність компанії на ринку і залучити більше веб майстрів та рекламодавців.

Під час розробки веб застосунку були побудовані контекстні діаграми, діаграми декомпозиції процесів, бізнес моделі, архітектура системи, була спроектована база даних, що зберігає необхідні для роботи дані. Був досліджений процес роботи API сервісів Google і Facebook, що є основними в роботі веб майстрів.

Для розробки також було проведено ряд UX/UI досліджень для побудови правильної логіки, User flow, ефективного розміщення функціоналу системи у форматі зручного інтерфейсу з модулями візуалізації даних, а також було проведено тестування готового продукту.

В результаті виконання роботи було розроблено повноцінний web застосунок для CPA мережі, що має цінність на сучасному ринку, та і перспективі може стати прибутковим проектом.

Список використаних джерел

1. Стаття про загальні моделі та особливості CPA маркетингу:
[Електронний ресурс] // Режим доступу:
<https://www.unisender.com/ru/support/about/glossary/cpa-marketing-model-i-formula/> [1]
2. «Арбітраж Трафіка від А до Я». Автор: Стеценко Віталій сс. 36-54[2]
3. «Юридичні аспекти в CPA-діяльності» . Автор Максим та Єгор Пашковські, сс. 67-68 [3]
4. “Бази даних. Проектування, реалізація і супроводження. Теорія і практика”, Томас Конноллі, Каролін Бегг, сс. 103-110
5. “Алан Купер про інтерфейс. Основи проектування взаємодії”
6. Стаття опис роботи з Google Analytic:
[Електронний ресурс] // Режим доступу:
https://www.2checkout.com/how-to/interview/avinash-kaushik_7.htm [6]
7. Стаття опис роботи з Facebook Ads:
[Електронний ресурс] // Режим доступу:
<https://www.facebook.com/business/help/200000840044554> [7]
8. Create entire apps using JavaScript and Ajax (англ.). Aptana Inc (2009)., pp 34-55 [8]

ДОДАТОК А

Лістинг Front-end частини

Main.html

```
<!DOCTYPE html>

<html lang="en">

<head>

  <meta charset="UTF-8">

  <meta http-equiv="X-UA-Compatible" content="IE=edge">

  <meta name="viewport" content="width=device-width, initial-scale=1.0">

  <link href="style/main.css" rel="stylesheet" type="text/css">

  <title>Document</title>

</head>

<body>

  <header class="header__main">

    <div class="header__logo">

    </div>

    <div class="header__link">

      <div class="link">

        <a id="link1" onclick="nextPage(1)">

          <span>Головна</span></a>

        </div>

      <div class="link">

        <a id="link2" onclick="nextPage(2)">
```

Офери

</div>

<div class="link">

Мої офери

</div>

</div>

<div class="header__rifht-info">

<div class="money">

<p class="balanceUser">7434 \$</p><button>+</button>

</div>

<div class="day-night">

</div>

<div class="circle">

</div>

<div class="circle">

</div>

<div class="circle">

</div>

<div class="circle">

</div>

</div>

```

</header>
<header class="header">
  <div class="header__logo">
    
  </div>
</header>
<div class="container">
  <div class="main">
    <div class="container__left-main">
      <div class="left-main__upper">
        <div class="info">
          <div class="img-info">
            
          </div>
          <div class="text">
            <p>Хіти</p>
            <h3 class="info-hit">3027</h3>
          </div>
        </div>
      </div>
      <div class="info">
        <div class="img-info">
          
        </div>
        <div class="text">
          <p>Ліди</p>
          <h3 class="info-lid">48</h3>
        </div>
      </div>
    </div>
    <div class="info">
      <div class="img-info">

```

```

        
    </div>
    <div class="text">
        <p>CR</p>
        <h3 class="info-cr">5.0 %</h3>
    </div>
</div>
<div class="info">
    <div class="img-info">
        
    </div>
    <div class="text">
        <p>Хости</p>
        <h3 class="info-host">1003</h3>
    </div>
</div>
<div class="info">
    <div class="img-info">
        
    </div>
    <div class="text">
        <p>Депозити</p>
        <h3 class="info-dep">22</h3>
    </div>
</div>
<div class="info">
    <div class="img-info">
        
    </div>

```



```

        <div class="text">
            <p>Visit to reg%</p>
            <h3 class="info-reg">30</h3>
        </div>
    </div>
    <div class="balance">
        <div class="balance__title">
            
            Баланс
        </div>
        <h1 class="balanceUser">$ 7434</h1>
        <button
class="purple"><b>Поповнити</b></button>
        </div>
    </div>
    <div class="left-main__bottom">
        <canvas id="myChart"></canvas>
    </div>

</div>
<div class="container__right-main">
    <div class="advertising">

        <div class="user-img">
            <img class="user-avatar" src="" alt="фотка
юзера">
        </div>
        <div class="user-name__maneg-name">
            <div class="user-name">

        </div>
    </div>

```

```
<div class="meneg">
    <p>Менеджер: Рожков Ярослав</p>
</div>

</div>
</div>
<div class="advertising-info">
    <p>Мої офери</p>

</div>

</div>

</div>

</div>
<div id="my" class="container__list-ofer">

    <div id="my-ofer-list" class="ofer-list">
        <div class="ofer-list__header">
            <p onclick="myId()" class="ofer">Офер</p>
            <p onclick="myCr()" class="cr">CR</p>
            <p onclick="myhits()" class="hits">Хіти</p>
            <p onclick="myhosts()" class="hosts">Хости</p>
            <p>Ліди</p>
            <p>ДЕпозити</p>
            <p>Фінанси</p>
        </div>

    </div>

</div>
```

```

</div>
<div id="all" class="container__list-ofer">

    <div id="ofer-list-all" class="ofer-list">
        <div class="ofer-list__header">
            <p onclick="allId()">ID</p>
            <p onclick="allName()">Назва</p>
            <p onclick="allCost()">виплати, $</p>
            <p onclick="allGoal()">ціль</p>
            <p onclick="allCountry()">країни</p>
            <p onclick="allLanguage()">мова</p>
            <p>Докладніше</p>
        </div>

    </div>

</div>

<div class="window-big-list">

    <div class="window-big-list__title">
        
        <div class="name-offer-title">
            <h3 id="name-title">Название</h3>
            <p>Androind</p>
        </div>

    </div>

</div>

<div class="block-add">

```

```

        <h3>Tracking link</h3><input
value="https://offers.neogara.com/#/register?ref=7u3dg8">
    </div>
    <div class="block-add">
        <h3>Обрати джерело</h3><input>
    </div>
    <div class="block-add">
        <h3>Скрипт</h3><input value="1wefwefew_44CI33e">
    </div>
    <div class="block-add">
        <h3>Назва РК</h3><input>
    </div>
    <div class="block-add">
        <h3>Facebook ID (-s)</h3><input>
    </div>
    <div class="block-add">
        <h3>Заголовок</h3><input>
    </div>
    <div class="block-add">
        <h3>Повідомлення</h3><input>
    </div>
    <button onclick="closeBigWindow()" class="button-add-big-
window">Зберегти</button>
    <button onclick="closeBigWindow()" class="buttonclose-
window">Закрити</button>
    </div>
</div>
<div class="login__container">
    <div class="login">
        <h1>Авторизація</h1>
        <div class="container__input">
            <h2>E-mail</h2>

```

```

        <input class="email" placeholder="Введіть e-mail">
    </div>
    <div class="container__input">
        <h2>Пароль</h2>
        <input id="password" class="email" placeholder="Введіть
пароль">
    </div>
    <button onclick="login()" class="purple"><b>Увійти</b></button>
</div>
</div>

</body>
<script src="https://cdn.jsdelivr.net/npm/chart.js"></script>
<script src="script/script.js"></script>
<script>

```

```

const labels = [
    '10.05',
    '14.05',
    '18.05',
    '22.05',
    '26.05',
    '30.05',
    '03.06',
];

```

```

const data = {
    labels: labels,
    datasets: [
        {
            label: 'Хіти',

```

```

        backgroundColor: 'rgb(186, 191, 206)',
        borderColor: 'rgb(186, 191, 206)',
        data: [2000, 1500, 1400, 1700, 2000, 1300, 2000],
    },
    {
        label: 'Ліди',
        backgroundColor: 'rgb(155, 166, 235)',
        borderColor: 'rgb(155, 166, 235)',
        data: [1200, 700, 400, 450, 540, 400, 700],
    },
    {
        label: 'Депозити',
        backgroundColor: 'rgb(182, 255, 170)',
        borderColor: 'rgb(182, 255, 170)',
        data: [1100, 600, 300, 350, 240, 300, 600],
    },
    {
        label: 'Visit to reg',
        backgroundColor: 'rgb(255, 170, 170)',
        borderColor: 'rgb(255, 170, 170)',
        data: [1050, 550, 250, 300, 200, 250, 550],
    },
]
};

```

```

const config = {
    type: 'line',
    data: data,
    options: {}
}

```

```

    };
</script>
<script>
    const myChart = new Chart(
        document.getElementById('myChart'),
        config
    );
</script>

</html>

```

ДОДАТОК Б

Лістинг Back-end частини

Script.js

```

//перевірка логіну та паролю
function login() {
    const email = document.querySelector(".email").value
    const password = document.querySelector("#password").value

    for (let i = 0; i < user.length; i++) {
        if (user[i].email == email && user[i].password == password) {
            idUser = i

            userImgName(idUser)
            randomOffer(idUser)
            balanceUser()

            const headerLogin = document.querySelector(".header")
            headerLogin.remove()

            const divLogin = document.querySelector(".login__container")
            divLogin.remove()

            const mainDiv = document.querySelector(".main")
            mainDiv.style.display = "flex"

            const headerMain = document.querySelector(".header__main")
            headerMain.style.display = "flex"

        }
    }
}

```

```

}
}
login()

//рендер аватарки юзера і імені
function userImgName(id) {
const userAvatar = document.querySelector(".user-avatar")
userAvatar.setAttribute("src", user[id].img)

const userName = document.querySelector(".user-name")
userName.innerHTML = user[id].name

}

function nextPage(idPage) {

const mainPage = document.querySelector(".main")
const oferList = document.querySelector("#all")
const myOferList = document.querySelector("#my")

if (idPage == 1) {
    mainPage.style.display = "flex"
    oferList.style.display = "none"
    myOferList.style.display = "none"

    const link1 = document.querySelector("#link1")
    const link2 = document.querySelector("#link2")
    const link3 = document.querySelector("#link3")

    link1.innerHTML = "<span>Головна</span>"
    link2.innerHTML = "Офери"
    link3.innerHTML = "Мої офери"

}
else if (idPage == 2) {
    mainPage.style.display = "none"
    oferList.style.display = "flex"
    myOferList.style.display = "none"

    const link1 = document.querySelector("#link1")
    const link2 = document.querySelector("#link2")
    const link3 = document.querySelector("#link3")

    link1.innerHTML = "Головна"

```



```

    link2.innerHTML = "<span>Офери</span>"
    link3.innerHTML = "Мої офери"

    offerlistAll(offers)

}
else if (idPage == 3) {
    mainPage.style.display = "none"
    oferList.style.display = "none"
    myOferList.style.display = "flex"

    const link1 = document.querySelector("#link1")
    const link2 = document.querySelector("#link2")
    const link3 = document.querySelector("#link3")

    link1.innerHTML = "Головна"
    link2.innerHTML = "Офери"
    link3.innerHTML = "<span>Мої офери</span>"

    offerlistMy(myOffers)

}
}

function offerlistAll(offers) {

var oferList = document.querySelector("#ofer-list-all")

const blockRemuve = document.querySelectorAll("#ofer-list__block-ALL")
if (blockRemuve) {
    for (let i = 0; i < blockRemuve.length; i++) {
        blockRemuve[i].remove()
    }
}

for (let i = 0; i < offers.length; i++) {
    const offerBlock = document.createElement("div")
    offerBlock.className = "ofer-list__block"
    offerBlock.setAttribute("id", "ofer-list__block-ALL")

    const p1 = document.createElement("p")
    p1.className = "idlist"
    p1.innerHTML = offers[i].id

    const p2 = document.createElement("p")

```

```

    p2.innerHTML = offers[i].name

    const p3 = document.createElement("p")
    p3.className = "cost"
    p3.innerHTML = offers[i].cost

    const p4 = document.createElement("p")
    p4.innerHTML = offers[i].goal

    const p5 = document.createElement("p")
    p5.innerHTML = offers[i].country

    const p6 = document.createElement("p")
    p6.innerHTML = offers[i].language

    const button = document.createElement("button")
    button.setAttribute("onclick", `ask(${offers[i].id})`)
    button.className = "ofer-button"
    button.innerHTML = "Запросити"

    offerBlock.append(p1)
    offerBlock.append(p2)
    offerBlock.append(p3)
    offerBlock.append(p4)
    offerBlock.append(p5)
    offerBlock.append(p6)
    offerBlock.append(button)

    oferList.append(offerBlock)
}

}

function offerlistMy(offers) {
var oferList = document.querySelector("#my-ofer-list")

const blockRemuve = document.querySelectorAll("#ofer-list__block-MY")
if (blockRemuve) {
    for (let i = 0; i < blockRemuve.length; i++) {
        blockRemuve[i].remove()
    }
}

for (let i = 0; i < offers.length; i++) {
    const offerBlock = document.createElement("div")
    offerBlock.className = "ofer-list__block"
    offerBlock.setAttribute("id", "ofer-list__block-MY")

```

```
const p1 = document.createElement("p")
p1.className = "ofer"
p1.innerHTML = `(${offers[i].id}) ${offers[i].name} `

const p2 = document.createElement("p")
p2.className = "cr"
p2.innerHTML = offers[i].cr + "%"

const p3 = document.createElement("p")
p3.className = "hits"
p3.innerHTML = offers[i].hit

const p4 = document.createElement("p")
p4.className = "hosts"
p4.innerHTML = offers[i].host

const div1 = document.createElement("div")
div1.className = "color"
const h11 = document.createElement("h3")
h11.innerHTML = offers[i].lid[0]
const h12 = document.createElement("h3")
h12.innerHTML = offers[i].lid[1]
const h13 = document.createElement("h3")
h13.innerHTML = offers[i].lid[2]
div1.append(h11)
div1.append(h12)
div1.append(h13)

const div2 = document.createElement("div")
div2.className = "color"
const h21 = document.createElement("h3")
h21.innerHTML = offers[i].lid[0]
const h22 = document.createElement("h3")
h22.innerHTML = offers[i].dep[1]
const h23 = document.createElement("h3")
h23.innerHTML = offers[i].dep[2]
div2.append(h21)
div2.append(h22)
div2.append(h23)

const div3 = document.createElement("div")
div3.className = "color"
const h31 = document.createElement("h3")
h31.innerHTML = offers[i].finans[0]
const h32 = document.createElement("h3")
h32.innerHTML = offers[i].finans[1]
const h33 = document.createElement("h3")
h33.innerHTML = offers[i].finans[2]
div3.append(h31)
div3.append(h32)
```

```

    div3.append(h33)

    offerBlock.append(p1)
    offerBlock.append(p2)
    offerBlock.append(p3)
    offerBlock.append(p4)
    offerBlock.append(div1)
    offerBlock.append(div2)
    offerBlock.append(div3)

    offerList.append(offerBlock)
}

}

var aId = 1
function allId() {
    if (aId == 0) {
        offers.sort(function (a, b) {
            return a.id - b.id
        })
        aId++
        offerlistAll(offers)
    } else {
        offers.sort(function (a, b) {
            return b.id - a.id
        })
        aId--
        offerlistAll(offers)
    }

}

var aNa = 0
function allName() {
    if (aNa == 0) {
        offers.sort((a, b) => a.name > b.name ? 1 : -1)
        aNa++
        offerlistAll(offers)
    } else {
        offers.sort((a, b) => a.name < b.name ? 1 : -1)
        aNa--
        offerlistAll(offers)
    }
}

```

```

}

var aCo = 1
function allCost() {

  if (aCo == 0) {

    offers.sort(function (a, b) {
      return a.cost - b.cost
    })
    aCo++
    offerlistAll(offers)
  } else {
    offers.sort(function (a, b) {
      return b.cost - a.cost
    })
    aCo--
    offerlistAll(offers)
  }
}

var aGo = 0
function allGoal() {
  if (aGo == 0) {

    offers.sort((a, b) => a.goal > b.goal ? 1 : -1)
    aGo++
    offerlistAll(offers)
  } else {
    offers.sort((a, b) => a.goal < b.goal ? 1 : -1)
    aGo--
    offerlistAll(offers)
  }
}

var aCountry = 0
function allCountry() {
  if (aCountry == 0) {

    offers.sort((a, b) => a.country > b.country ? 1 : -1)
    aCountry++
    offerlistAll(offers)
  } else {
    offers.sort((a, b) => a.country < b.country ? 1 : -1)
    aCountry--
    offerlistAll(offers)
  }
}

var aLa = 0

```

```

function allLanguage() {
  if (aLa == 0) {

    offers.sort((a, b) => a.language > b.language ? 1 : -1)
    aLa++
    offerlistAll(offers)
  } else {
    offers.sort((a, b) => a.language < b.language ? 1 : -1)
    aLa--
    offerlistAll(offers)
  }
}

```

```

function ask(idOffer) {

```

```

  const windowSearch = document.querySelector(".window")
  if (windowSearch) windowSearch.remove()

```

```

  var selectedOffer

```

```

  for (let i = 0; i < offers.length; i++) {
    if (offers[i].id == idOffer) selectedOffer = offers[i]
  }

```

```

  const container = document.querySelector(".container")

```

```

  const window = document.createElement("div")
  window.className = "window"

```

```

  const title = document.createElement("h1")
  title.innerHTML = "Підтвердження оферу"

```

```

  const divImg = document.createElement("div")
  divImg.className = "divImg"

```

```

  const imgName = document.createElement("div")
  imgName.className = "imgName"

```

```

  const img = document.createElement("img")
  img.setAttribute("src", "img/ofer/img.png")

```

```

  const name = document.createElement("h2")
  name.innerHTML = selectedOffer.name

```

```

  const country = document.createElement("p")
  country.innerHTML = selectedOffer.country

```

```

  imgName.append(img)
  imgName.append(name)
  divImg.append(imgName)

```

```

divImg.append(country)

const info = document.createElement("p")
info.className = "info"
info.innerHTML = selectedOffer.info

const divCost = document.createElement("div")
divCost.className = "divCost"
const infoCost = document.createElement("h3")
infoCost.innerHTML = `Коштів буде заморожено:`
const cost = document.createElement("h3")
cost.innerHTML = `${selectedOffer.cost}$`

divCost.append(infoCost)
divCost.append(cost)

const balance = document.createElement("h3")
balance.className = "balanceDom"
balance.innerHTML = `Баланс: ${user[idUser].balance}$`

const buttonAdd = document.createElement("button")
buttonAdd.setAttribute("onclick", `openBigInfo(${idOffer})`)
buttonAdd.className = "buttonAdd"
buttonAdd.innerHTML = `Підтвердити`

const buttonclose = document.createElement("button")
buttonclose.setAttribute("onclick", "closeWindow()")
buttonclose.className = "buttonclose"
buttonclose.innerHTML = `Закрити`

window.append(title)
window.append(divImg)
window.append(info)
window.append(divCost)
window.append(balance)
window.append(buttonAdd)
window.append(buttonclose)

container.append(window)
}
//Закрити вікно
function closeWindow() {
const windowSearch = document.querySelector(".window")
if (windowSearch) windowSearch.remove()
}

//кнопка відправки
function buttonDone(idOffer) {

const buttonAddSearch = document.querySelector(".buttonAdd")
buttonAddSearch.innerHTML = "Відправлено  "
```

```

buttonAddSearch.setAttribute("class", "done")
buttonAddSearch.setAttribute("onclick", "")

const buttonclose = document.querySelector(".buttonclose")
buttonclose.remove()

setTimeout(() => {
    closeWindow()
}, 4000);

for (let i = 0; i < offers.length; i++) {
    if (offers[i].id == idOffer) user[idUser].balance -= offers[i].cost
    balanceUser()
}

setTimeout(() => {
    randomOffer(idOffer)
    balanceUser()
}, 2000);

}

//Рандом наших офферов
function randomOffer(idOffer) {

var newOffer

for (let i = 0; i < offers.length; i++) {
    if (offers[i].id == idOffer) newOffer = offers[i]
}

const crNewRandom = 600 + getRandomInt(2000)
const hitRandom = crNewRandom - getRandomInt(500)

const lidMath = []

lidMath[2] = hitRandom - getRandomInt(200)
lidMath[0] = (lidMath[2] / 2) - getRandomInt(10)
lidMath[1] = (lidMath[0] / 2) - getRandomInt(20)

const depMath = []

depMath[2] = lidMath[2] / 2 - getRandomInt(10)

```



```

depMath[1] = lidMath[1] / 2 - getRandomInt(10)
depMath[0] = lidMath[0] / 2 - getRandomInt(10)

const finansMath = []

finansMath[2] = lidMath[2] * (20 + getRandomInt(50))
finansMath[1] = lidMath[1] * (5 + getRandomInt(30))
finansMath[0] = lidMath[0] * (5 + getRandomInt(30))

const crMath = rounded(((lidMath[2] / crNewRandom) * 100) / (3 + getRandomInt(6)))

myOffers[myOffers.length] = {
  id: newOffer.id,
  name: newOffer.name,
  cr: crMath,
  hit: crNewRandom,
  host: hitRandom,
  lid: lidMath,
  dep: depMath,
  finans: finansMath,
}

myOffers = myOffers.reduce((o, i) => {
  if (!o.find(v => v.id == i.id)) {
    o.push(i);
  }
  return o;
}, []);

user[idUser].balance += finansMath[2]

myOffersMainMenu()

var hit = 0, lid = 0, cr = 0, host = 0, dep = 0, proc = 0

for (let i = 0; i < myOffers.length; i++) {

  const divhit = document.querySelector(".info-hit")
  const divlid = document.querySelector(".info-lid")
  const divcr = document.querySelector(".info-cr")
  const divhost = document.querySelector(".info-host")

```

```

const divdep = document.querySelector(".info-dep")
const divproc = document.querySelector(".info-reg")

hit += myOffers[i].hit
lid += myOffers[i].lid[2]
cr += myOffers[i].cr
host += myOffers[i].host
dep += myOffers[i].dep[2]
proc = 10 + getRandomInt(30)

divhit.innerHTML = hit
divlid.innerHTML = lid
divcr.innerHTML = cr
divhost.innerHTML = host
divdep.innerHTML = dep
divproc.innerHTML = proc
}

}

function getRandomInt(max) {
return Math.floor(Math.random() * max);
}

function rounded(number) {
return +number.toFixed(2);
}
//Перевірка балансу
function returnBalanceUser() {
balanceUser()
}
//Оновлення балансу
function balanceUser() {
const balanceUser = document.querySelectorAll(".balanceUser")
for (let i = 0; i < balanceUser.length; i++) {
    balanceUser[i].innerHTML = `${user[idUser].balance}`$`
}
setTimeout(() => {
    returnBalanceUser()

}, 10000);
}

//Сортування по ID для мого списку
var mId = 1
function myId() {

```

```

if (mId == 0) {

    myOffers.sort(function (a, b) {
        return a.id - b.id
    })
    mId++
    offerlistMy(myOffers)
} else {
    myOffers.sort(function (a, b) {
        return b.id - a.id
    })
    mId--
    offerlistMy(myOffers)
}

}

//Сортування по сг для мого списка
var mCr = 1
function myCr() {

if (mCr == 0) {

    myOffers.sort(function (a, b) {
        return a.cr - b.cr
    })
    mCr++
    offerlistMy(myOffers)
} else {
    myOffers.sort(function (a, b) {
        return b.cr - a.cr
    })
    mCr--
    offerlistMy(myOffers)
}

}

//Сортування по хостам для мого списка
var mHO = 1
function myhosts() {

if (mHO == 0) {

    myOffers.sort(function (a, b) {
        return a.host - b.host
    })
    mHO++
    offerlistMy(myOffers)
} else {
    myOffers.sort(function (a, b) {

```

```

        return b.host - a.host
    })
    mHi--
    offerlistMy(myOffers)
}

}

//Сортування по хітам для мого списку
var mHi = 1
function myhits() {

    if (mHi == 0) {

        myOffers.sort(function (a, b) {
            return a.hit - b.hit
        })
        mHi++
        offerlistMy(myOffers)
    } else {
        myOffers.sort(function (a, b) {
            return b.hit - a.hit
        })
        mHi--
        offerlistMy(myOffers)
    }

}

//оновлення списку моїх оферів на головній
function myOffersMainMenu() {
    const advertisingInfo = document.querySelector(".advertising-info")

    const listInfoSearch = document.querySelectorAll(".list-info")

    if (listInfoSearch) {
        for (let i = 0; i < listInfoSearch.length; i++)listInfoSearch[i].remove()
    }

    if (myOffers) {
        for (let i = 0; i < myOffers.length; i++) {
            const listInfocreate = document.createElement("div")
            listInfocreate.className = "list-info"
            if (i % 2 == 0) {
                const block1 = document.createElement("div")
                block1.className = "block"
            }
        }
    }
}

```

```
const blockImg1 = document.createElement("div")
blockImg1.className = "block-img"

const img1 = document.createElement("img")
img1.setAttribute("src", "img/ofere/ofere.png")
img1.setAttribute("alt", "иконка")

blockImg1.append(img1)

const text1 = document.createElement("div")
text1.className = "text"

const text1h4 = document.createElement("h4")
text1h4.innerHTML = myOffers[i].name

const text1p = document.createElement("p")
text1p.className = "text"
text1p.innerHTML = "Крипта"

text1.append(text1h4)
text1.append(text1p)

block1.append(blockImg1)
block1.append(text1)

listInfo.create.append(block1)

if (myOffers[i + 1]) {
  const block2 = document.createElement("div")
  block2.className = "block"

  const blockImg2 = document.createElement("div")
  blockImg2.className = "block-img"

  const img2 = document.createElement("img")
  img2.setAttribute("src", "img/ofere/ofere.png")
  img2.setAttribute("alt", "иконка")

  blockImg2.append(img2)

  const text2 = document.createElement("div")
  text2.className = "text"

  const text2h4 = document.createElement("h4")
  text2h4.innerHTML = myOffers[i + 1].name

  const text2p = document.createElement("p")
  text2p.className = "text"
  text2p.innerHTML = "Крипта"
}
```

```

        text2.append(text2h4)
        text2.append(text2p)

        block2.append(blockImg2)
        block2.append(text2)

        listInfocreate.append(block2)
    }

    advertisingInfo.append(listInfocreate)
}
}
}

function openBigInfo(id) {
    buttonDone(id)

    setTimeout(() => {

        const bigWindow = document.querySelector(".window-big-list")
        bigWindow.style.display = "flex"

        const title = document.querySelector("#name-title")

        for (let i = 0; i < offers.length; i++) {
            if (offers[i].id == id) {
                title.innerHTML = offers[i].name
            }
        }

    }, 4000);

}

function closeBigWindow() {
    const bigWindow = document.querySelector(".window-big-list")

    bigWindow.style.display = "none"
}

```