

**КИЇВСЬКИЙ НАЦІОНАЛЬНИЙ УНІВЕРСИТЕТ
ІМЕНІ ТАРАСА ШЕВЧЕНКА**

Факультет комп'ютерних наук та кібернетики
Кафедра теорії та технології програмування

**Кваліфікаційна робота
на здобуття ступеня бакалавра
за спеціальністю 122 Комп'ютерні науки**

на тему:

**РОЗРОБКА UX/UI ДИЗАЙНУ МОБІЛЬНОГО ЗАСТОСУНКУ
ПОШУКУ ТВАРИН**

Виконала студентка 4-го курсу
групи ТТП-42
Катерина МОВЧАНЮК



Науковий керівник:
доцент, канд. пед. наук
Наталія РУСІНА

Засвідчую, що в цій роботі немає запозичень
з праць інших авторів без відповідних посилань

Студент



(підпис)

Роботу розглянуто й допущено до захисту
на засіданні кафедри теорії та технології
програмування
«05» червня 2023 р., Протокол №18

Завідувач кафедри
Микола НІКІТЧЕНКО

(підпис)

Київ – 2023

РЕФЕРАТ

Обсяг роботи: 46 сторінки, 33 ілюстрації, 34 використаних джерел.

ІНТЕРФЕЙС, МОБІЛЬНИЙ ЗАСТОСУНОК, UX ДИЗАЙН, UI ДИЗАЙН, FIGMA, ПЛАГІН, USER GLOW, WIREFRAME.

Об'єкт дослідження – UX/UI дизайн.

Метою кваліфікаційної роботи є створення мобільного застосунку «PetFriend» для людей, які бажають знайти домашнього улюбленця.

Інструменти розроблення: векторний онлайн-сервіс розробки інтерфейсів та прототипування з можливістю організації спільної роботи Figma, Adobe Illustrator для мобільного застосунку «PetFriend».

Результат роботи: проведено дослідження основних етапів розробки мобільного застосунку та технології для його реалізації.

Результатом є створення застосунку, а саме UX/UI частини застосунку «PetFriend» для пошуку та купівля домашніх улюбленців. Створено прототипування та проведено аналіз з користувачами, оцінка важливості застосунку та його зручність.

ЗМІСТ

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ	4
ВСТУП	5
РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	7
РОЗДІЛ 2. ЗАСТОСУВАННЯ UX/UI ДИЗАЙНУ В РОЗРОБЦІ МОБІЛЬНИХ ЗАСТОСУНКІВ	14
2.1 Застосування UX/UI дизайну в розробці	14
2.2 Класифікація інтерфейсів	16
2.3 Принципи проектування інтерфейсів	17
2.4 Аналіз функціональних можливостей аналогів застосунку	18
РОЗДІЛ 3. РОЗРОБКА ДИЗАЙНУ	21
3.1 Постанова задачі застосунку	21
3.2 Принципи побудови UserFlow	22
3.3 Створення Wireframes	23
3.4 Логотип, іконки та компоненти	25
РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА	27
4.1 Початок роботи мобільного застосунку	27
4.2 Головне меню	30
4.3 Сторінки всіх оголошень, особистих та збережених	33
4.4 Особиста сторінка	46
4.5 Створення інших сторінок	38
ВИСНОВКИ	41
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ	42
ДОДАТКИ	45
Додаток А – SiteMap	45
Додаток Б - WIREFRAME	46

СКОРОЧЕННЯ ТА УМОВНІ ПОЗНАЧЕННЯ

- JPEG – Joint Photographic Experts Group, об'єднана група експертів із фотографії
- iOS – iPhone Operating system, iPhone Операційна система
- PX – PICTure'S Element, елемент зображення
- PNG – Portable Network Graphics, портативна графіка мережі
- UF – UserFlow, потік користувача
- UI – User Interface, користувацький інтерфейс
- UX – User Experience, користувацький досвід

ВСТУП

Оцінка сучасного стану об'єкта розробки. На сьогоднішній день домашні улюбленці є важливою складовою кожної сім'ї. Існують веб та мобільні застосунки або соціальні мережі, які пропонують послуги знайти тварину для бажаючих. У різних країнах поширені такі застосунки: PetMatch [1], WannaPet [2], PuppiesApp [3]. У нас в країні найпоширенішими платформами та соціальними мережами є: OLX [4], Instagram [5], Prom.ua [6]. На початку данні платформи створювались для заміни ринків, але пізніше розширили свій функціонал. Вищенаведені застосунки орієнтуються в більшості випадків саме на продажі тварин. Через повномасштабне вторгнення росії в Україну страждають не тільки люди але й тварини. Так, кількість бездомних тваринок збільшилась в два й більше разів, і на жаль, велика частка з них саме домашні коти чи собаки, яких люди лишали під час евакуації за різних обставин.

Актуальність роботи та підстави для її виконання. Тема застосунку є актуальною, адже через повномасштабне вторгнення в Україну кількість бездомних тварин збільшилася на 60% від початку 2022 року [7]. Деякі знаходяться в притулках, а інші просто гуляють містами у пошуках їжі. Досить часто, через голод, тварини стають злими та проявляють свою агресію до людей.

Проблема в тому, що на українському ринку досі не існує повноцінного застосунку для продажу, купівлі, прилаштуванні тварин, і люди не знають що робити з хатніми тваринами, яких беруть на перетримку.

Існують лише сайти на який можна купити, продати або віддати будь-що, але основною проблемою цих сайтів – шахраї. Часто вони створюють фейкові оголошення задля своєї наживи, або видають маленьких улюбленців за іншу породу.

Наш застосунок може стати ефективним інструментом для знаходження нової домівки для будь-яких тварин і зменшити їх кількість на вулицях чи притулках.

Крім того, застосунок може стати в нагоді господарям, які загубили свого домашнього улюбленця, так як дуже часто тваринки втікають від господарів.

Мета та завдання роботи. Метою кваліфікаційної роботи є створення мобільного застосунку «PetFriend» для людей, які бажають знайти домашнього улюбленця. Для досягнення мети поставлені такі завдання:

- проаналізувати існуючі аналоги на ринку та цільову аудиторію;
- провести аналіз цілей та задач застосунку;
- провести аналіз UX та UI складових;
- розробити мобільний застосунок.

Об'єкт, методи і засоби розроблення. Об'єктом є процес опису UX/UI дизайну застосунку, який допомагає в пошуку домашніх улюбленців. В якості інструменту розробки було обрано векторний професійний графічний редактор для створення та редагування векторної графіки Adobe Illustrator [8] та векторний онлайн-сервіс розробки інтерфейсів та прототипування Figma [9].

Можливі сфери застосування. Розроблений дизайн застосунку можна використати у створенні реального проекту на різних платформах, а застосунок буде допомагати українцям у пошуку домашніх тварин.

РОЗДІЛ 1. АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

Загальні відомості про мобільні застосунки

Мобільні застосунки стали невід'ємною частиною нашого повсякденного життя. Вони дозволяють нам здійснювати різноманітні дії на наших смартфонах, такі як спілкування, розваги, робота, покупки та інше. Розглянемо загальні відомості про мобільні застосунки, їхню історію, класифікацію, особливості розробки та використання.

Історія мобільних застосунків сягає кінця 90-х років, коли з'явилися перші мобільні телефони з підтримкою Java-платформи [\[10\]](#). Java-технології дозволили розробникам створювати застосунки для мобільних пристроїв, що було кроком вперед у порівнянні з попередніми технологіями. Згодом з'явилися інші платформи, такі як Symbian [\[11\]](#), Windows Mobile [\[12\]](#) та Palm OS [\[13\]](#), які також забезпечували можливість розробки застосунків.

Але наступний прорив відбувся в 2007 році, коли Apple представила перший iPhone, який став першим мобільним пристроєм з застосунками в App Store [\[14\]](#). За кілька місяців від появи App Store з'явилося більше 500 застосунків, а кількість завантажень зростала з кожним днем. Інші виробники також стали додавати можливості для створення застосунків в своїх мобільних пристроях, що привело до зростання ринку мобільних застосунків. Для того, щоб користувачі стали задоволеними і залишилися вірними, необхідно розробити мобільний додаток, який буде максимально простим і зручним для використання. Для цього необхідно враховувати специфіку мобільних пристроїв та вимоги користувачів.

Основні переваги мобільних застосунків в порівнянні з мобільними сайтами:

- висока швидкість роботи й більш висока продуктивність;
- зручність використання, завдяки оптимізації інтерфейсу під розмір екрану та інші особливості мобільних пристроїв;

- наявність додаткових можливостей, таких як використання камери, GPS, сенсорів та інших;
- наявність можливості працювати без доступу до Інтернету;
- більш високий рівень безпеки, завдяки використанню технологій шифрування та ідентифікації користувачів;
- зручність користування для покупців, які можуть з легкістю використовувати додатки для покупки товарів та послуг, а також здійснювати оплату за допомогою мобільних платіжних систем.

Типи мобільних застосунків

Нативні та гібридні мобільні застосунки - це два різних підходи до розробки програмного забезпечення для мобільних пристроїв, таких як смартфони або планшети. Опишемо детальніше про кожен з цих підходів.

Нативні мобільні застосунки.

Нативні застосунки розробляються для певної платформи, такої як iOS (для iPhone та iPad) або Android (для пристроїв на базі операційної системи Android). Для створення нативних застосунків використовуються мови програмування, які специфічні для кожної платформи. Наприклад, для iOS це Swift [\[15\]](#) або Objective-C [\[16\]](#), а для Android - Java або Kotlin [\[17\]](#).

Основні переваги нативних застосунків включають високу продуктивність, доступ до повного функціоналу пристрою (такого як камера, геолокація, сповіщення тощо) і найкращу інтеграцію з операційною системою. Нативні застосунки зазвичай мають більш гладкий і швидкий інтерфейс користувача і можуть надати більший контроль над функціональністю.

Однак, розробка нативних застосунків може бути ресурсоємною та займати багато часу, оскільки потрібно розробляти та підтримувати окремий код для кожної платформи.

Гібридні мобільні застосунки.

Гібридні застосунки розробляються з використанням веб-технологій, таких як HTML [\[18\]](#), CSS [\[18\]](#) і JavaScript [\[19\]](#), і виконуються всередині веб-

контейнера, вбудованого в нативний додаток. Вони можуть бути запущені як на iOS, так і на Android без потреби в окремому коді для кожної платформи.

Гібридні застосунки дозволяють зменшити зусилля для розробки, оскільки один код можна використовувати на обох платформах. В гібридних застосунках використовується фреймворк для мобільної розробки, такий як React Native [19], Flutter [20] або Ionic [21]. Ці фреймворки дозволяють розробникам використовувати веб-технології для створення інтерфейсу користувача та логіки застосунка, а потім обгорнути його в нативний контейнер, що надає доступ до функціональності пристрою.

Основні переваги гібридних застосунків полягають у швидкості розробки, одноразовому написанні коду для обох платформ, а також зручності у впровадженні оновлень і виправлення помилок. Крім того, гібридні застосунки можуть бути більш доступні з точки зору бюджету, оскільки не потребують окремого розроблення для кожної платформи.

Однак, гібридні застосунки можуть мати обмежену продуктивність порівняно з нативними застосунками й не забезпечувати таку ж глибоку інтеграцію з операційною системою та функціоналом пристрою. Також, деякі складні функції можуть вимагати використання нативного коду або плагінів, що може збільшити складність розробки.

Вибір між нативними та гібридними застосунками залежить від конкретних потреб і обмежень проекту, таких як бюджет, строк виконання, необхідні функціональність та платформи, на яких потрібно запускати додаток.

Важливо враховувати те, що мобільні застосунки необхідно розробляти з урахуванням потреб та очікувань користувачів. Для цього потрібно проводити дослідження ринку та аналізувати поведінку користувачів. Також варто звернути увагу на дизайн та інтерфейс користувача, щоб забезпечити зручну та інтуїтивно зрозумілу взаємодію з застосунком, якщо коротко, то провести повне UX/UI дослідження.

Програмні засоби для проєктування інтерфейсу

Для реалізації застосунку нам знадобляться програмні продукти, на описі яких зупинимось більш детально.

Платформа Figma

Figma [9] - це інтуїтивно зрозуміла дизайнерська платформа, яка дає вам змогу створювати будь-що - веб-сайти, програми, логотипи та багато іншого. Використання Figma дозволяє зробити перші кроки в дизайні користувацького інтерфейсу та дизайні взаємодії з користувачем — ключових навичках для розробки вражаючого портфоліо як індивідуально, так і потенційно в організації.

Переваги Figma:

- Високий рівень співпраці та співпрацівників: Figma є веб-застосунком, що дозволяє дизайнерам та іншим учасникам проєкту працювати в одному місці, спільно редагуючи макети та спілкуючись.
- Кросплатформенність: Figma доступний як для Mac, так і для Windows, а також має мобільні застосунки для iOS та Android, що дозволяє працювати з проєктами з будь-якого пристрою.
- Простота використання: Figma має простий та інтуїтивний інтерфейс, що дозволяє швидко створювати макети та прототипи.
- Безкоштовний план: Figma має безкоштовний план, що дозволяє користуватися застосунком безкоштовно, хоча обмеження використання функцій можуть бути встановлені.
- Контроль версій: Figma дозволяє зберігати копії макетів та стежити за їхніми змінами в часі, що дозволяє легко повертатися до попередніх версій.

Недоліки Figma:

- Відсутність локального застосунку: Figma є хмарним застосунком, тому його можна використовувати лише в тому випадку, якщо є підключення до Інтернету. Це може створити проблеми для тих, хто працює в обмеженому Інтернет-з'єднанні або в тих випадках, коли з'єднання з Інтернетом відсутнє.

- Недостатній інструментарій для роботи з векторною графікою: Figma не має такої рівної функціональності для роботи з векторною графікою, як Adobe Illustrator, що може стати перешкодою для тих, хто більш прив'язаний до цих інструментів.
- Відсутність інструментів для 3D-моделювання: Figma не підтримує 3D-моделювання, що може бути важливим для деяких дизайнерів і архітекторів.
- Обмеження в налаштуванні поведінки об'єктів: Figma має обмежену функціональність налаштування поведінки об'єктів, що може бути незручним для деяких дизайнерів.
- Відсутність підтримки для інших програм: Figma не підтримує інтеграцію з іншими програмами, що може бути проблемою для деяких дизайнерів, які використовують багато різних інструментів.

Редактор Adobe Illustrator

Adobe Illustrator [8] - це професійний векторний графічний редактор, розроблений компанією Adobe Systems [22]. Він широко використовується в дизайні логотипів, ілюстрацій, веб-дизайні та інших сферах дизайну.

Переваги Adobe Illustrator:

- Векторна графіка дозволяє масштабувати зображення без втрати якості;
- Висока точність при створенні ілюстрацій, малюнків та інших векторних зображень;
- Можливість працювати з різноманітними кольорами, градієнтами та іншими ефектами;
- Наявність багатьох інструментів для малювання та редагування векторів;
- Широкі можливості для роботи з текстом;
- Інтеграція з іншими програмами Adobe, такими як Photoshop та InDesign.

Недоліки Adobe Illustrator:

- Складність в освоєнні програми для новачків;
- Висока вартість ліцензії;
- Для роботи з фотографіями та іншими растровими зображеннями

краще використовувати Adobe Photoshop.

Загалом, Adobe Illustrator [\[8\]](#) є потужним та універсальним інструментом для створення векторної графіки, що дозволяє дизайнерам створювати складні та деталізовані ілюстрації та малюнки. Однак, для новачків може знадобитися трохи часу на освоєння програми та її функцій.

Онлайн-сервіс Canva

Canva [\[23\]](#) - це онлайн-сервіс для дизайну, що дозволяє створювати професійно виглядачі зображення та документи без необхідності мати досвід у графічному дизайні. Canva надає користувачам доступ до безлічі шаблонів та інструментів, які дозволяють створювати зображення для соціальних мереж, презентацій, брошур, постерів та багато іншого.

Переваги Canva:

- Простота використання та доступність для користувачів з будь-яким рівнем досвіду;
- Велика кількість шаблонів та готових елементів дизайну, що дозволяє швидко створювати виглядачі зображення без необхідності створювати все з нуля;
- Можливість працювати в команді та ділитися проектами з іншими користувачами;
- Підтримка багатьох мов, що дозволяє користувачам з різних країн використовувати сервіс на своїй рідній мові;
- Інтеграція з іншими програмами та сервісами, такими як Dropbox, Google Drive, Instagram та інші.

Недоліки Canva:

- Обмежена кількість варіантів кастомізації шаблонів та елементів дизайну;
- Висока вартість підписки на платформу для доступу до всіх функцій;
- Немає можливості редагування графічних файлів, створених у Canva, в інших програмах.

-

Плагіни Figma

Unsplash - це плагін, який дозволяє вставляти безкоштовні зображення з Unsplash безпосередньо в макети [\[24\]](#) .

Content Reel - це плагін, який дозволяє створювати та додавати в макети текстовий контент у вигляді липучих наклейок [\[25\]](#) .

Spellcheckers - це плагін, який перевіряє правопис в текстових елементах в макеті та надає корекційні пропозиції [\[26\]](#) .

РОЗДІЛ 2. ЗАСТОСУВАННЯ UX/UI ДИЗАЙНУ В РОЗРОБЦІ МОБІЛЬНИХ ЗАСТОСУНКІВ

2.1 Застосування UX/UI дизайну в розробці

Мобільні застосунки стали невід’ємною частиною нашого життя. Ми використовуємо їх для розваг, покупок, спілкування, виконання різноманітних робочих завдань. Однак величезна кількість застосунків на ринку означає, що розробникам необхідно виділитися серед конкурентів, щоб привернути увагу користувачів і зробити їх щасливими. Саме тут доцільно зупинитись на дизайні UX/UI.

Окремо розглянемо UX та UI.

Користувальницький досвід або UX. Взаємодія з користувачем (UX) стосується того, що користувачі відчувають під час використання ваших послуг або програм. Чи вдається йому досягти мети та наскільки просто чи складно це зробити. UX створює базовий вигляд для будь-якої програми, що допомагає залучити більше користувачів, перетворюючи потенційних клієнтів на лояльних [\[27\]](#).

Інтерфейс користувача або UI. Це процес покращення презентації та інтерактивності веб-або мобільної програми. Він добре фокусується на створенні інтерактивних віджетів, якими користуються користувачі. Це може бути кнопка, значок або візуальні елементи, з якими взаємодіють користувачі. Також ключову роль грає знання типографіки та колористики [\[28\]](#).

Важливо зрозуміти важливість дизайну інтерфейсу користувача (UI) та дизайну взаємодії з користувачем (UX). Хоча функціональність, безперечно, є перевагою, дизайн відіграє важливу роль у охопленні клієнтів і збільшенні бази користувачів будь-якої програми. UX/UI дизайн програми покращує задоволеність клієнтів, що, у свою чергу, стимулює зростання популярності програми.

Оскільки в наші дні конкуренція стає дедалі гострішою, ви повинні виділятися в натовпі, початок або розвиток невеликого підприємства вимагає акценту на користувацький інтерфейс і дизайн взаємодії з користувачем, особливо тому, що перше враження залишається назавжди; успішний дизайн UX і UI може зробити або порушити впізнаваність бренду. Ефективний користувацький інтерфейс/структура UX дозволяє користувачам повертатися, водночас створюючи лояльних клієнтів для бізнесу, а також швидкість завантаження застосунків забезпечує користувачам найвищий рівень користувацького досвіду з точки зору навігації, зовнішнього вигляду та відчуття для розширення клієнтської бази – зрештою це все зводиться до створення довіри між компаніями та їх аудиторією (рисунок 1.1).



Рисунок 1.1 – Приклад порівняння дизайну

2.2 Класифікація інтерфейсів

Дизайн можна класифікувати за кількома критеріями. Наведемо кілька загальних класифікацій.

- Функціональний дизайн.

Дизайн може бути класифікований на основі функціонального призначення, включаючи продукт, інтерфейс і веб-дизайн, а також графічний дизайн.

- Естетичний дизайн.

Дизайн класифікують відповідно до естетичної привабливості, креативності та художньої цінності, наприклад дизайн одягу, дизайн інтер'єру, художній дизайн тощо.

- Дизайн, орієнтований на користувача.

Даний критерій залежить від того, наскільки добре дизайн відповідає потребам і вподобанням користувачів; приклади включають дизайн досвіду користувача, дизайн інтерфейсу користувача та дизайн, орієнтований на людину.

- Індустріальний дизайн.

Конструкції використовують відповідно до передбачуваної галузі застосування; наприклад, архітектурні, автомобільні чи промислові зразки можуть підпадати під цю категорію.

- Процесно-орієнтований дизайн.

Даний вид класифікують на основі відповідних процесів проектування, наприклад гнучкий, економічний або ітераційний дизайн.

- Технологічні проекти.

Поділяють відповідно до технології, яку вони використовують, включаючи цифровий дизайн, 3D-дизайн і автоматизований дизайн.

- Дизайн на основі часу.

Поділяють на тимчасові, постійні та перехідні категорії залежно від тривалості їх існування.

Це лише деякі приклади того, як різні категорії дизайнерських рішень можна класифікувати, причому точна класифікація залежить від контексту та призначення.

2.3. Принципи проєктування інтерфейсів

За основу вирішили розглядати популярну евристику Якоба Нільсена під назвою “10 Usability Principles To Improve UI Design” [\[29\]](#) .

Дана теорія була запропонована в 1990 році Якобом Нільсеном і Рольфом Молічем і включала десять рекомендацій, які допоможуть розробити UI.

Евристики Нільсена є загальними принципами, тобто вони не визначають конкретних правил зручності використання. Натомість евристика це правила яких ви можете дотримуватися, щоб допомогти створити більш доступні, зручні та інтуїтивно зрозумілі цифрові продукти [\[29\]](#).

Вони створили ці евристики за допомогою спостережень і досвіду, набутого протягом років роботи. Нижче наведено короткий опис кожного.

1. Видимість стану системи - користувач повинен завжди знати, що відбувається в системі, щоб уникнути плутанини та помилок.
2. Відповідність системи та реального світу - інтерфейс повинен бути зрозумілим та знайомим для користувачів, щоб вони могли легко зрозуміти, як виконувати різні дії.
3. Контроль та свобода користувача - користувач повинен мати можливість відмінити свої дії або повернутися до попереднього кроку, якщо він десь помилився.
4. Послідовність та стандартизація - інтерфейс повинен мати стандартний та послідовний дизайн для різних елементів, щоб користувачі могли легко зрозуміти, як взаємодіяти з системою.

5. Запобігання помилкам - дизайн інтерфейсу повинен запобігати можливості виникнення помилок, навіть якщо користувач зробив щось неправильно.

6. Розпізнавання, а не відтворення - інформація та функції повинні бути легко зрозумілі та доступні для користувачів, щоб вони не забували, що можна робити.

7. Гнучкість та ефективність використання - інтерфейс повинен давати можливість швидко та легко виконувати різні дії, як для новачків, так і для досвідчених користувачів.

8. Естетика та мінімалізм дизайну - дизайн інтерфейсу повинен бути привабливим та естетичним, але не переповненим зайвою інформацією.

9. Допомога користувачам розпізнавати, діагностувати та виправляти помилки - система повинна давати зрозумілі та корисні повідомлення про помилки та проблеми, щоб користувачі могли легко їх розпізнавати та виправляти.

10. Доступність для всіх користувачів - інтерфейс повинен бути доступним та корисним для різних типів користувачів, включаючи тих, у кого є обмеження фізичних можливостей, іншомовних користувачів та тих, хто використовує адаптивні технології.

2.4 Аналіз функціональних можливостей аналогів застосунку

Важливим етапом в створенні дизайну інтерфейсу є аналіз існуючих застосунків. За допомогою аналізу можна можна з'ясувати, які елементи та рішення дизайну працюють найкраще. Вивчення їх функціональності, взаємодії та структури може надати ідеї для вдосконалення власного інтерфейсу та можна виявити часті помилки, незручності чи проблеми з взаємодією, які можна уникнути у власному дизайні. Грамотний дизайн допоможе стати найкращим в галузі, яку обрав клієнт. Для того, щоб почати роботи над нашим проектом, проаналізуємо деякі застосунки, які існують на ринку, а саме: PuppiesApp [3], WannaPet [2], PetMatch [1].

Для більш зручного аналізу створили таблицю 11), в якій виокремлено за кольорами деякі позиції:

- відсутність необхідного — червоний колір;
- погана реалізація — жовтий колір;
- ідеальна реалізація - зелений колір.

Таблиця 1. Аналіз існуючих застосунків

Фунціонал	PuppiesApp	WannaPet	PetMatch	Нотатки
Авторизація /Реєстрація	Зелений	Зелений	Зелений	
Авторизація через соціальні мережі	Зелений	Червоний	Зелений	
Фільтрація	Зелений	Жовтий	Зелений	Порода, рік, стать, розмір, ціна, локація,
Головна сторінка	Жовтий	Жовтий	Жовтий	
Пошук на головній сторінці	Зелений	Жовтий	Червоний	
Запити	Зелений	Зелений	Зелений	З фотографіями, інформація про продавця, рейтинг, контактні данні продавці, локація
Особиста сторінка	Зелений	Жовтий	Жовтий	
Додати особисту фотографію до сторінки	Зелений	Зелений	Зелений	
Режим редагування особистої сторінки	Зелений	Зелений	Жовтий	
Чат	Зелений	Жовтий	Зелений	

Збережені запити	Жовтий	Зелений	Жовтий	
Особисті запити	Жовтий	Жовтий	Зелений	
Написати відгук про продавця	Зелений	Червоний	Червоний	
Довідка про додаток	Червоний	Зелений	Зелений	Для чого створений додаток, як користуватися застосунком
Спосіб зв'язку з підтримкою	Червоний	Зелений	Зелений	

Проаналізувавши (таблиця 1), в нашому мобільному застосунку будуть реалізовані деякі пункти, які виділені червоним та жовтим кольорами, для того, щоб наш застосунок був конкурентоспроможним. Також хочеться уточнити що дизайн створений даними застосункам є вже застарілий, тому ще одною метою є реалізація сучасного та зручного застосунку.

РОЗДІЛ 3. РОЗРОБКА ДИЗАЙНУ

3.1 Постанова задачі застосунку

Мета: Розробити для ринку України дизайн застосунку «PetFriend» та в подальшому розвивати його. Застосунок має бути ефективним інструментом для продажу/притулку тварин, а також для пошуку загублених домашніх улюбленців.

Задачі:

1. Провести аналіз цілей та задач застосунку, чітко сформулювати, для чого він призначений, які проблеми вирішує та який інструментарій використовує.
2. Визначити цільову аудиторію застосунку, провести аналіз ринку.
3. Розробити дизайн інтерфейсу застосунку, що відповідає потребам цільової аудиторії.
4. Розробити архітектуру та функціональні можливості застосунку, включаючи такі функції, як:
 - можливість розміщувати оголошення про продаж/притулок тварин;
 - можливість шукати тварин за різними критеріями (порода, вік, стать тощо);
 - можливість перегляду профілів користувачів та їх оголошень;
 - можливість зв'язатися з користувачем за допомогою внутрішнього чату;
 - можливість написання відгуків про користувачів та їх тварин;
 - можливість додавання фотографій до оголошень;
 - можливість пошуку загублених домашніх улюбленців.
5. Розробити механізм забезпечення безпеки користувачів застосунку, включаючи перевірку та підтвердження акаунтів, а також систему відгуків про користувачів та їх тварин.

6. Розробка системи зворотного зв'язку, що дозволяє користувачам зв'язуватися з розробниками застосунку та надавати їм зворотний зв'язок.
7. Проведення тестування застосунку.

3.2 Принципи побудови User Flow

User Flow - це зображення послідовності дій, які користувач повинен виконати, щоб досягти своєї мети в проєкті [\[30\]](#). Це є способом дивитись на проєкт з точки зору користувача. UF використовує блоки для створення алгоритмів та аналізу взаємодії користувача з проєктом. Це допомагає зрозуміти, чи всі функції та процеси мають логічне завершення, або чи є якісь "слабкі місця". Від початку проєктування важливо зрозуміти, чи вирішує проєкт поставлені завдання, чи є всі необхідні етапи для досягнення мети, чи є зайві дії, які можуть заплутати користувача. UF допомагає зобразити процес взаємодії з продуктом.

Для створення User Flow необхідно зрозуміти, яка інформація потрібна. Існують два варіанти: створення User Flow на основі потенційного користувача, який буде вирішувати свої завдання за допомогою сайту, або уявити себе в ролі користувача та виконувати дії на сайті, записувати їх та створювати User Flow. User Flow може мати різні форми, наприклад, лінійну, розгалужену, з додатковими шляхами або петлями. Графічна схема дає можливість проілюструвати можливі сценарії взаємодії з продуктом. Це допомагає зрозуміти, як користувач буде взаємодіяти з продуктом, що дозволяє виправити можливі недоліки та покращити його функціональність.

Отже, для створення User Flow потрібно врахувати різні етапи взаємодії користувача з продуктом, такі як вхід на сайт, реєстрація, вибір товару, додавання до кошика, оплата та інші. Також важливо враховувати можливі варіації дій користувача та різні варіанти взаємодії з продуктом. При створенні UF важливо не забувати про простоту та зрозумілість для користувача. Всі блоки та стрілки повинні бути лаконічними та зрозумілими, щоб користувач міг швидко зорієнтуватись у процесі взаємодії з продуктом.

Окрім того, важливо враховувати контекст взаємодії з продуктом. Наприклад, якщо це мобільний додаток, то варто враховувати можливі обмеження на екрані та обмеження в роботі з застосунком на маленьких екранах. При створенні User Flow також важливо враховувати вимоги та очікування користувачів. Наприклад, якщо користувач очікує, що додавання товару до кошика буде здійснюватись шляхом простого натискання кнопки, то це повинно бути відображено в User Flow.

Узагальнюючи, створення User Flow є важливим етапом в процесі розробки продукту, який допомагає зрозуміти взаємодію користувача з продуктом та забезпечує логічне та зрозуміле завершення всіх етапів взаємодії. Користуючись принципами побудови User Flow, розробники можуть створити продукт, який буде зручним та зрозумілим для користувачів.

Приклад діаграми нашого застосунку наведено в додатку А.

3.3 Створення Wireframes

Wireframing — ефективний метод проектування веб-сайтів на структурному рівні. Каркаси зазвичай використовуються для розміщення вмісту та функціональних можливостей на сторінках, які враховують потреби та шляхи користувачів, закладаючи таким чином основу проєктів розробки до того, як на сторінку буде додано візуальний дизайн і вміст [31].

Каркаси служать для надання візуального огляду сторінок на початку проєкту, щоб отримати схвалення зацікавлених сторін і команди проєкту перед початком творчої роботи над дизайном. Вони також можуть використовуватися для встановлення глобальної та вторинної навігації, щоб відповідати очікуванням користувачів щодо структур навігації сайту. Каркаси, приклад наведено на рисунку 3.1, дозволяють швидко переглядати та змінювати структуру ключових сторінок, даючи як клієнтам, так і командам дизайнерів впевненість, що сторінка відповідає потребам користувачів і одночасно досягає цілей проєкту.

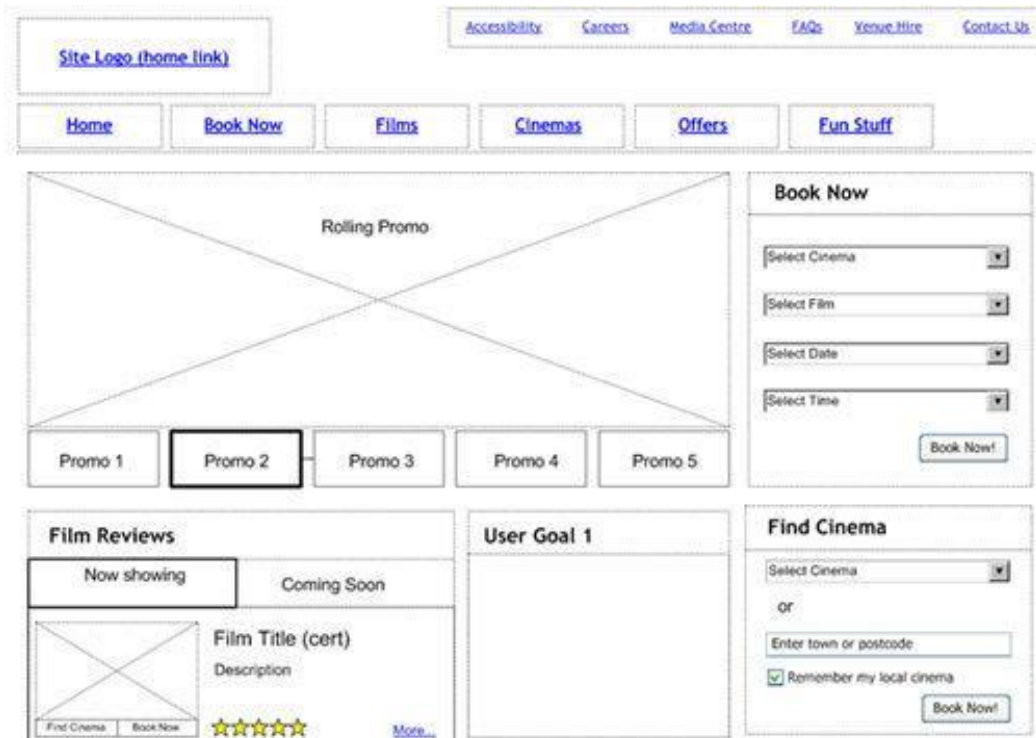


Рисунок 3.1– Приклад каркасу

Каркаси часто використовуються як частина процесу дизайну, орієнтованого на користувача, і на ранньому етапі проєктування, де вони служать для надання відгуків користувачам перед початком творчої роботи. Тест прототипу на зручність використання також може проводитися з використанням каркасних сторінок як відгук користувача перед початком творчих процесів.

Каркаси можна намалювати вручну або зібрати за допомогою такого програмного забезпечення, як Microsoft Visio [32], щоб забезпечити доставку на екрані, в нашому випадку будемо використовувати програму Figma [9].

Переваги каркасу: wireframing пропонує кілька ключових переваг перед концептуальними проєктами: він надає раннє візуальне зображення для перегляду клієнтами та користувачами; забезпечує ранній зворотний зв'язок під час тестування прототипу на зручність використання, легше вносити зміни та повертає впевненість у процеси проєктування після схвалення обома групами. З практичної точки зору каркаси допомагають забезпечити, щоб вміст і функціональність сторінки відповідали потребам користувачів і

бізнесу. Крім того, у міру просування проєкту вони надають безцінний спосіб для членів команди проєкту обговорити його бачення та масштаби на відкритому форумі.

Недоліки каркасу: через відсутність візуального дизайну або врахування технічних наслідків каркаси можуть бути складними для розуміння клієнтами. Дизайнери також повинні перетворити ці каркаси на проєкти - може знадобитися спілкування між обома сторонами щодо того, чому елементи сторінки розміщено саме так. Крім того, коли додається вміст, він спочатку може перевищувати те, що може вмістити каркасний макет – це вимагає тісної співпраці як дизайнера, так і копірайтера, щоб зробити цю роботу.

Приклад нашого каркасу представлено в додатку Б.

Логотип, іконки та компоненти

Логотип був створений за допомогою шрифту Inter, розмір 20 px, Semi Bold, а також ілюстрації за допомогою пера.



Рисунок 3.2 – Логотип

Іконки були створенні в професійному графічному редакторі для створення та редагування векторної графіки – Ілюстратор (рисунок 3.3-3.4).

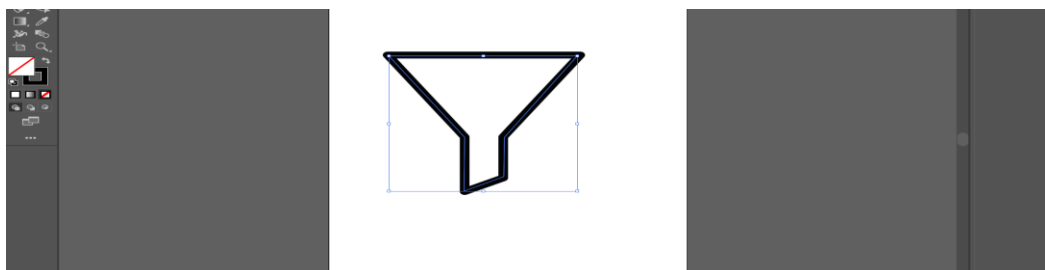


Рисунок 3.3 – Створення іконок в Ілюстратор

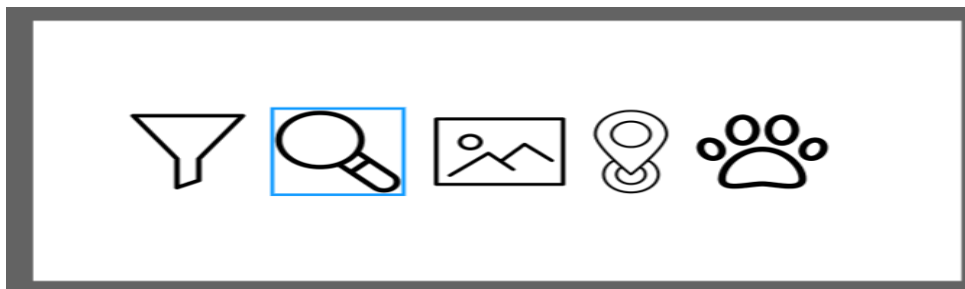


Рисунок 3.4 – Створені іконки

Ще однією важливою частиною створення дизайну – компоненти (рисунок 3.5).

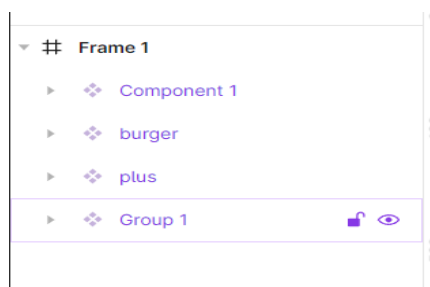


Рисунок 3.5 – Компоненти нашого застосунку

Компоненти – це елементи, які можна повторно використовувати у своїх дизайнах. Вони допомагають створювати та керувати послідовними дизайнами в різних проєктах. Можна створювати компоненти з будь-яких шарів або об'єктів, які спроектували. Це може бути цілий ряд речей, як-от кнопки, значки, макети тощо.

Компонент має два аспекти:

- 1) головний компонент визначає властивості компонента;
- 2) екземпляр — це копія компонента, яку можна повторно використовувати у своїх проєктах. Екземпляри пов'язані з основним компонентом і отримують будь-які оновлення, зроблені для компонента (рисунок 3.6).

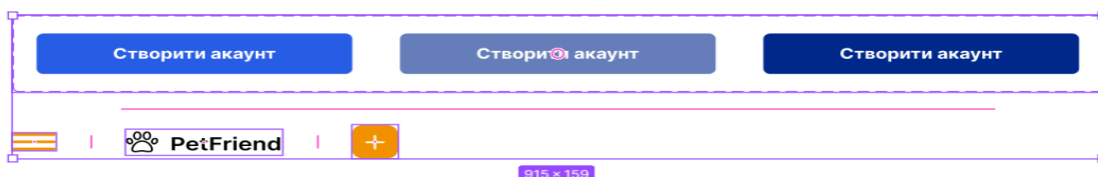


Рисунок 3.6 – Компоненти нашого проєкту

За допомогою цього розділу ми розглянули компоненти, які використовувалися в розробці застосунку.

РОЗДІЛ 4. ІНСТРУКЦІЯ КОРИСТУВАЧА

Для палітри взяті 2 акцентних кольорів: помаранчевий та синій.

Перший колір обраний, тому що асоціюється з радістю, задоволенням, прагнення до досягнення й самоствердження, колір енергії й сили. Він робить добродійну дію на психіку людини, знімаючи напругу.

Синій нагадує про відчуття спокою та розслаблення. Його часто описують як мирне, спокійне, безпечне та впорядковане. Синій колір сприймається як знак стабільності та надійності. Компанії, які хочуть створювати імідж безпеки, часто використовують синій колір у своїй рекламі та маркетингу.

Для тексту був обраний шрифт Inter у різних варіаціях: light, regular, semi bold, bold, в залежності від значимості тексту. Розміри від 13 px до 36 px.

Для застосунку були використані фотографії – тобто растрова графіка . Растрова графіка — це цифрові зображення, які складаються з крихітних прямокутних пікселів або елементів зображення, розташованих у сітці або растрі за координатами x і y (включаючи координату z у випадку 3D) таким чином, що це формує імідж [33]. Його також називають бітовим зображенням, оскільки воно містить інформацію, яка відображається безпосередньо в сітці дисплея.

4.1 Початок роботи мобільного застосунку

При вході в застосунок користувач має 2 шляхи : авторизуватися та зареєструватися.

Щоб авторизуватися користувач має 2 шляхи: ввести дані про себе, а саме емейл та пароль, або скористатися можливістю увійти за допомогою інших соціальних мереж.

Щоб зареєструватися користувач має ввести особисті дані, і придумати пароль, або так само скористатися можливістю інших соціальних мереж (рисунок 4.1).

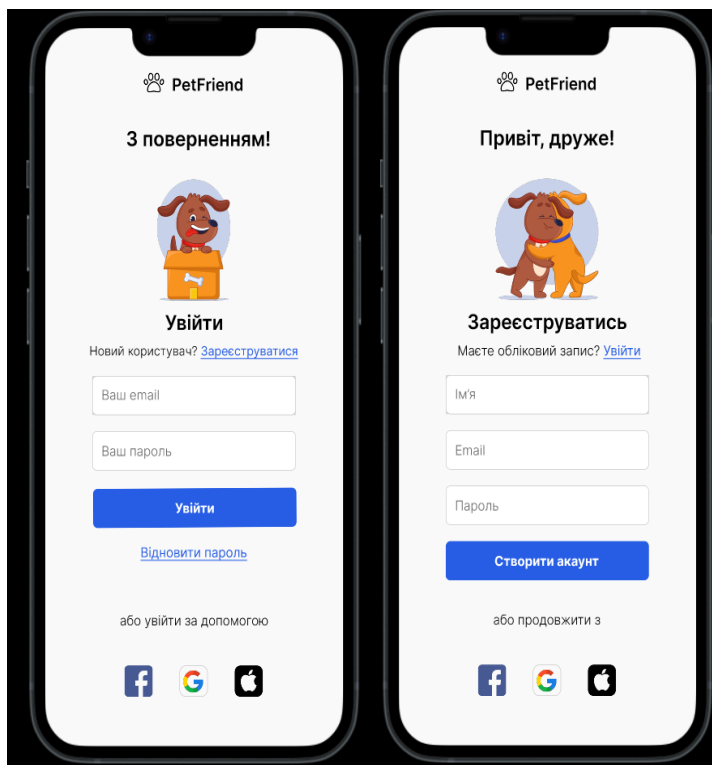


Рисунок 4.1 – Сторінка Авторизація/Сторінка Реєстрація

У випадку якщо користувач не згадав свій пароль у момент авторизації – є можливість його відновити, нажав на підкреслений текст “Відновити пароль” (рисунок 4.2).

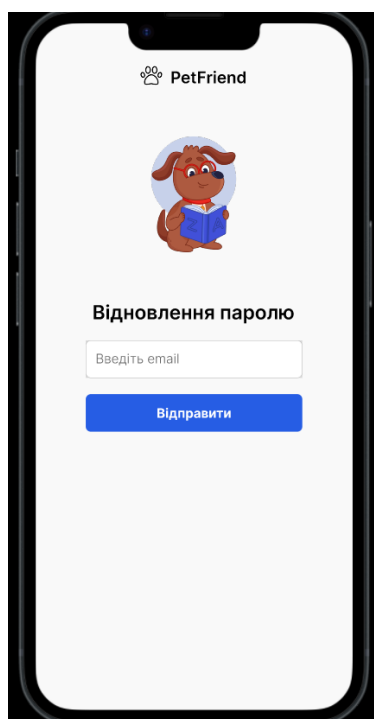


Рисунок 4.2 – Відновлення паролю

Відновлення паролю потребує пошти користувача, саме туди прийде код, який дозволить перейти до сторінки скидання старого паролю і створенню нового (рисунок 4.3 і рисунок 4.4)

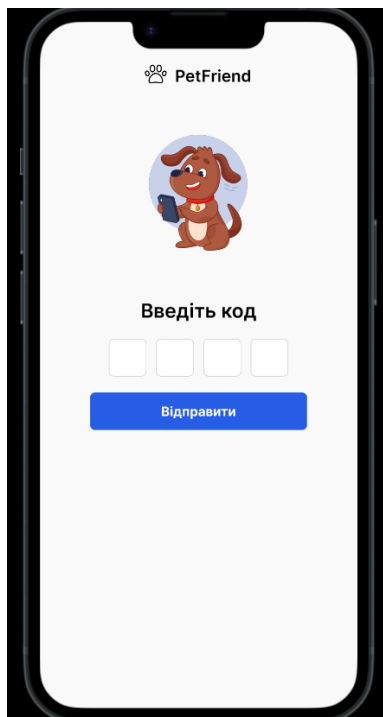


Рисунок 4.3 – Введення паролю

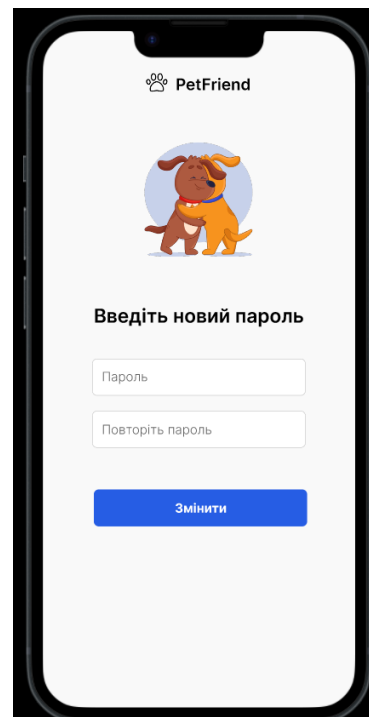


Рисунок 4.4 – Створення нового паролю

Після успішного створення паролю застосунок попереджає що новий пароль зберігся і ми можемо перейти в мобільний застосунок (рисунок 4.5).

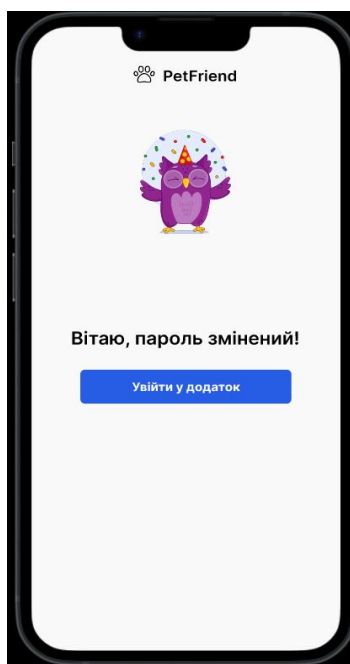


Рисунок 4.5 – Пароль змінено

4.2 Головне меню

Коли користувач успішно авторизувався або зареєструвався – він потрапляє на головну сторінку. У шапці застосунку бачимо – бургерне меню, логотип та іконку “Додати оголошення”. Нижче описано для чого застосунок та кнопка “Додати оголошення” для тих користувачів – які не побачили, або не зрозуміли іконку в шапці профілю. Нижче можемо перейти на сторінку “Всі оголошення”. Та у футері – іконки соціальних мереж розробників застосунку (рисунок 4.6).

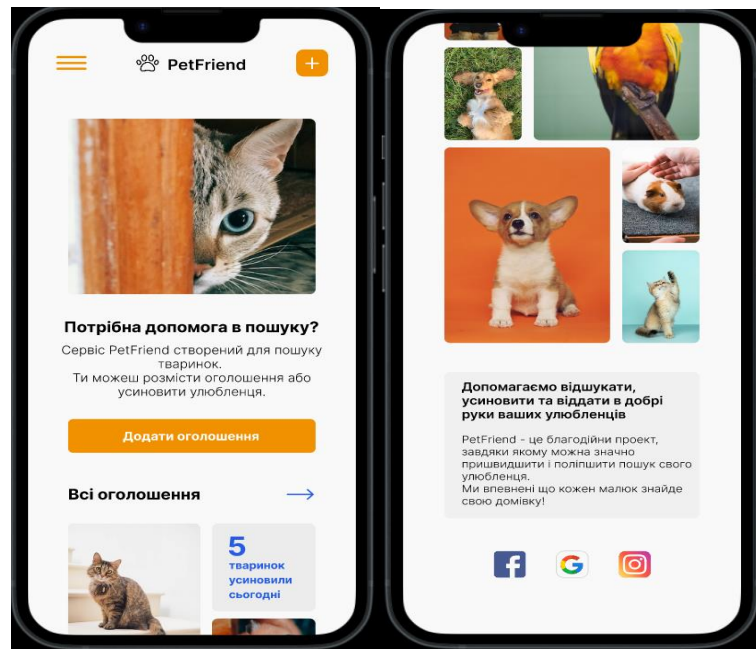


Рисунок 4.6 – Головна сторінка

Користувач може перейти в бургерне меню та передивитися весь список сторінок, на які може переміститися, а це: всі оголошення, мої оголошення, збережені оголошення, мій акаунт, мій чат, про нас, політика конфіденційності та контакти (рисунок 4.7)



Рисунок 4.7 – Бургерне меню

Також з Головного меню через кнопку “Додати оголошення” або іконку – ми можемо створити наше оголошення, в якому маємо такі можливості; додати фотографії, ввести ім’я улюбленця та породу, обрати вік, розмір, стать тваринки, додати коментар про особливості улюбленця, обрати населений пункт, дату і час, і особисті контакти. Після того, як всі дані введенні, користувач може натиснути кнопку “Опублікувати” і отримати текст про успішне виконання дії (рисунки 4.8-4.9).

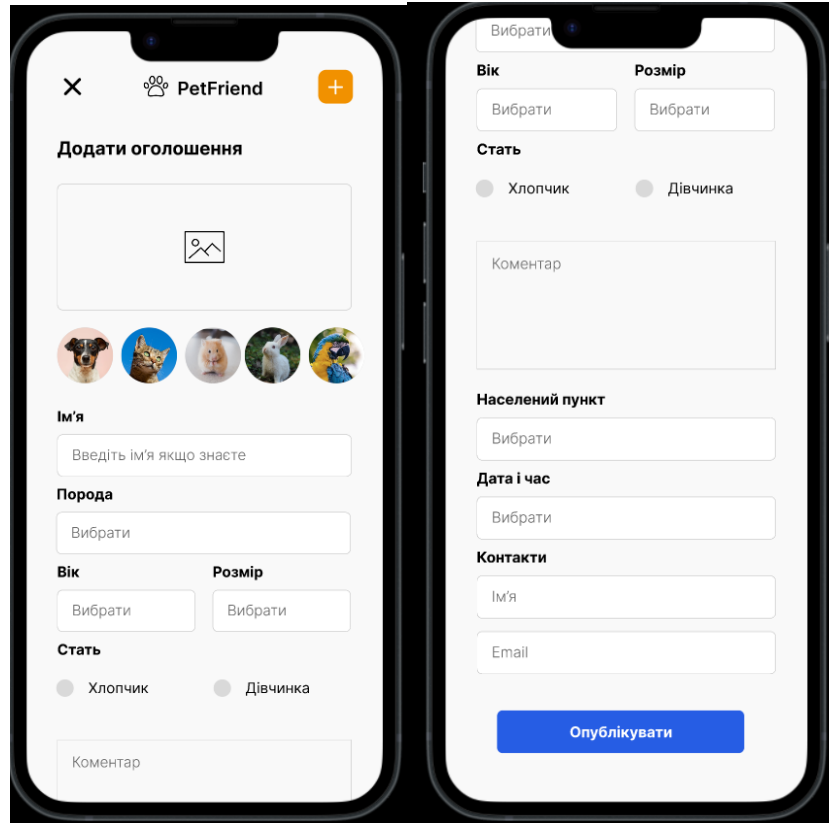


Рисунок 4.8– Створення оголошення

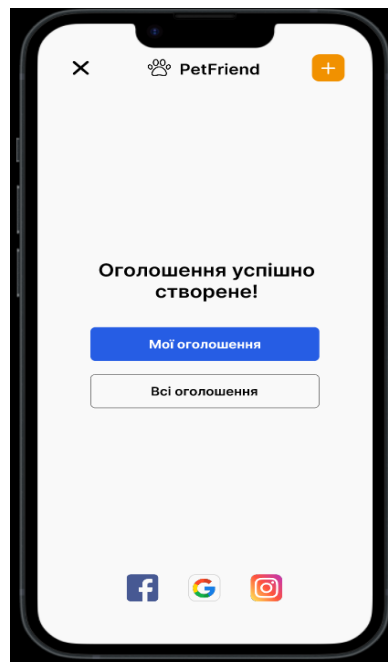


Рисунок 4.9 – Успішна публікація оголошення

4.3 Сторінки всіх оголошень, особистих та збережених

Для того щоб користувач натрапив на сторінку всіх оголошень є два варіанти: через бургерне меню або через головну сторінку. Бачимо вибір: шукаю або знайшов, пошук по слову, фільтр та всі актуальні оголошення, нижче можемо переходити на інші сторінки(якщо вони наявні) та футер наших соціальних мереж (рисунок 4.10).

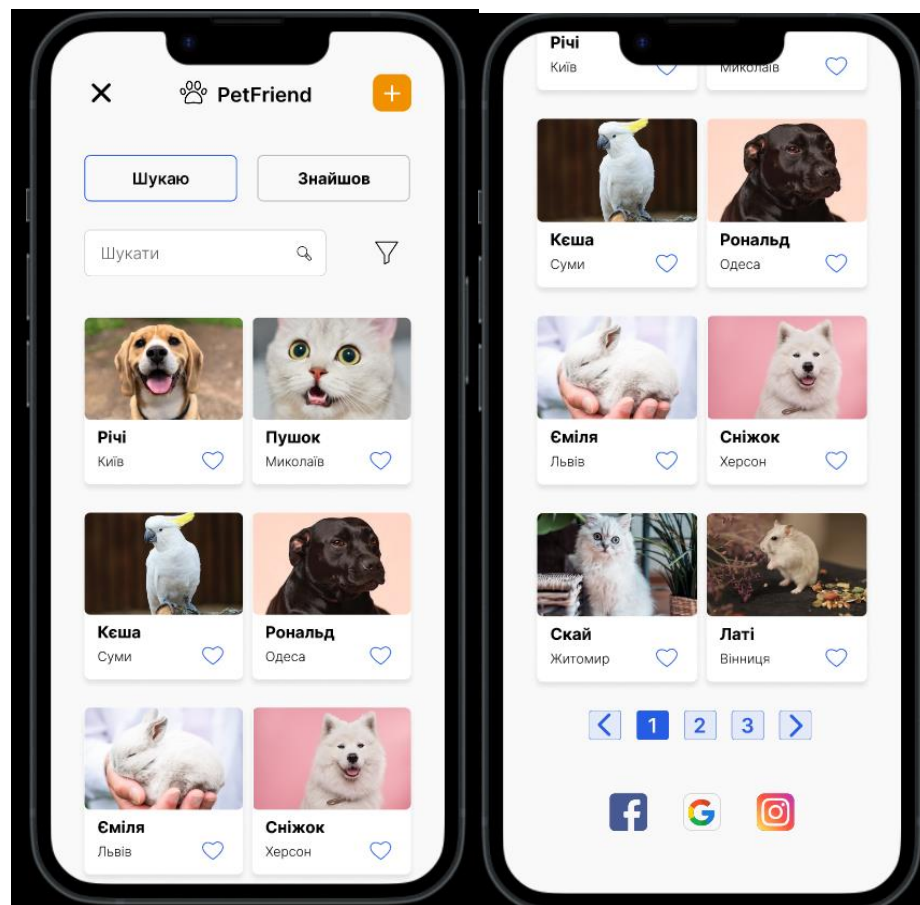


Рисунок 4.10 – Сторінка всіх оголошень

Якщо користувач обирає фільтр, він має декілька пунктів, з яких пошук вийде набагато швидше, а саме: порода, вік, стать, розмір, ціна і населений пункт. При натисненні на кнопку “Пошук” – всі пункти фільтру які були обрані застосуються (рисунок 4.11).

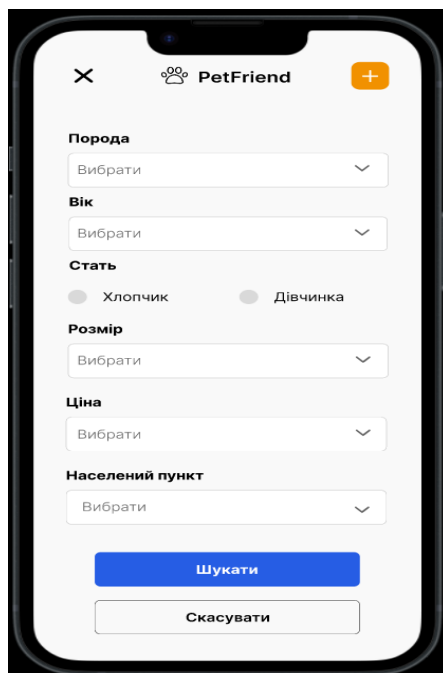


Рисунок 4.11 – Фільтр оголошень

Якщо користувач перейде до пункту “Знайшов” – можна побачити всіх тваринок, які за сьогоднішній день знайшли домівку, таким чином, якщо ви обрали цю ж тваринку – ви маєте можливість побачити чи ще вона вільна (рисунок 4.12).

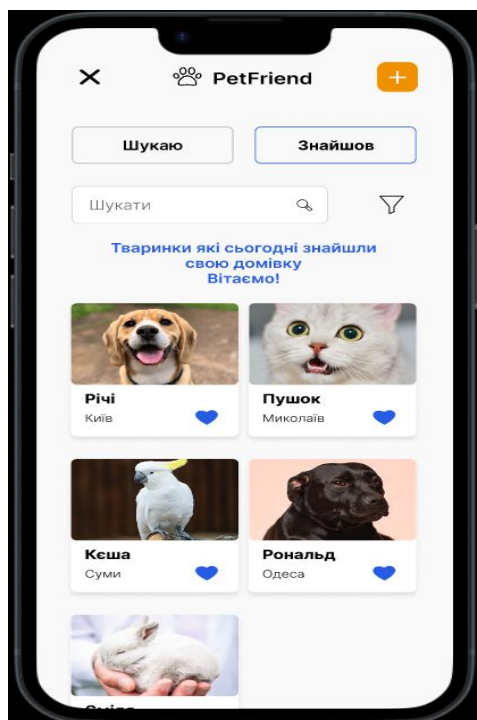


Рисунок 4.12 – Сторінка тварин яких вже забрали

Далі на черзі сторінка особистих оголошень, до неї можна натрапити такими шляхами: через особистий акаунт або бургерне меню.

На даній сторінці бачимо всі оголошення які ми опубліковували за весь час, а також маємо можливість передивитися оголошення які ми зберігли (рисунки 4.13 – 4.14).

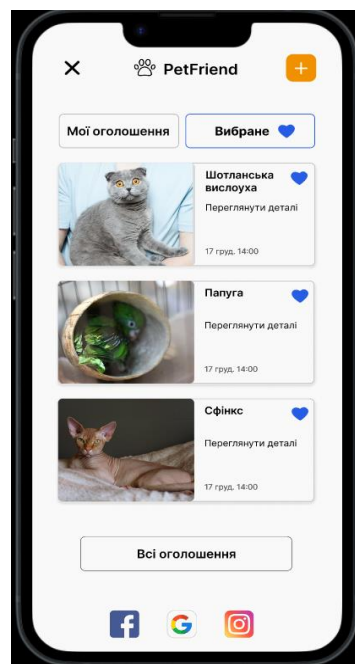
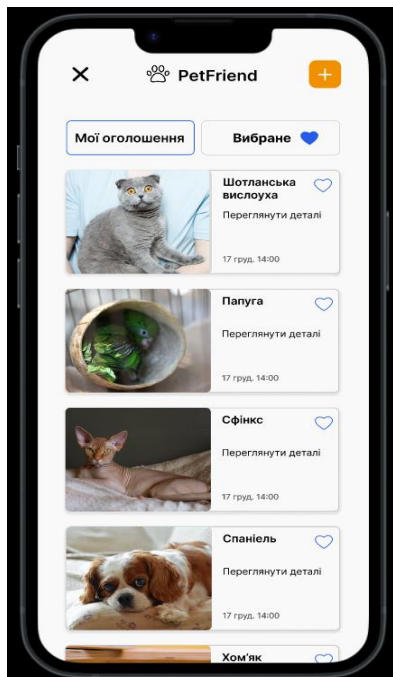


Рисунок 4.13 – Особисті оголошення Рисунок 4.14 – Вибрані оголошення

Подивимося як виглядає сторінка самого оголошення. Маємо можливість передивитися фотографії тваринки, ім'я, вік, стать, ціну, дату публікації, коментарі, контакти власника. Також ми натисненні кнопки “Написати особисто” користувач має змогу вже зараз розпочати розмову зі власником улюбленця. Користувач може передивитися приблизне розташування тварини, і перейти до всіх, або своїх, оголошень (рисунок 4.15).

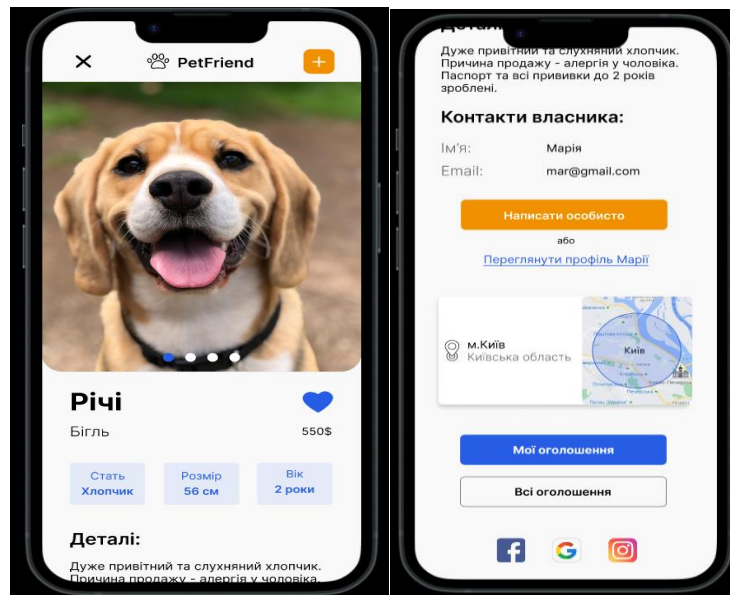


Рисунок 4.15 – Оголошення

4.4 Особиста сторінка

Особиста сторінка може бути використана для збору аналітики та інформації про користувачів вашого мобільного застосунку. Ви можете встановити аналітичні інструменти, які дозволять вам відстежувати показники використання. Особиста сторінка користувача (профіль користувача) також має важливе значення у розробці мобільного застосунку з наступних причин:

1. Персоналізація та налаштування: Особиста сторінка дозволяє користувачам налаштовувати налаштування та параметри застосунку згідно з їхніми вподобаннями і потребами. Вона надає можливість змінювати особисті дані, профіль, фотографії, налаштування конфіденційності тощо. Це допомагає створити індивідуальний досвід користувача та забезпечити його задоволення від використання застосунку.

2. Збереження даних та синхронізація: Особиста сторінка користувача є місцем для збереження та синхронізації даних. Користувачі можуть зберігати свої особисті дані, які вони бажають, такі як контакти тощо. Це дозволяє їм зручно користуватися застосунком на кількох пристроях і не втрачати дані при переході з одного пристрою на інший.

1. **Управління акаунтом та безпека:** Особиста сторінка користувача забезпечує можливість керування акаунтом, таку як реєстрація, вхід, відновлення пароля, зміна електронної пошти тощо. Вона також дозволяє встановлювати механізми безпеки, такі як двофакторна аутентифікація або управління доступом до особистих даних. Це допомагає забезпечити конфіденційність та безпеку даних користувачів.

Сторінка користувача в нашому застосунку має: фотографію, пошту, номер телефону(який бачить лише користувач), свій рейтинг та оголошення які були створені даним користувачем.

Також користувач має можливість перейти на сторінку всіх оголошень (рисунки 4.16 - 4.18).

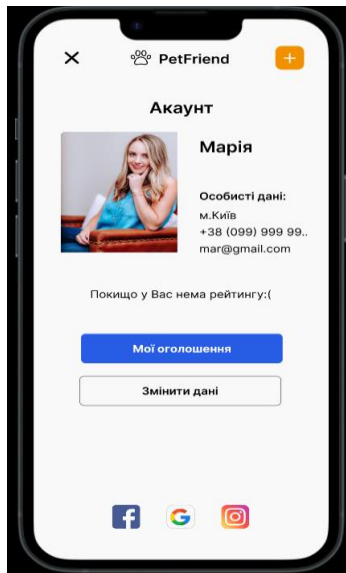


Рисунок 4.16 – Особистий акаунт

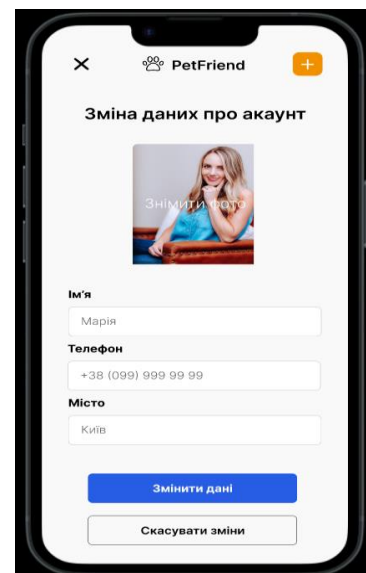


Рисунок 4.17 – Зміна даних

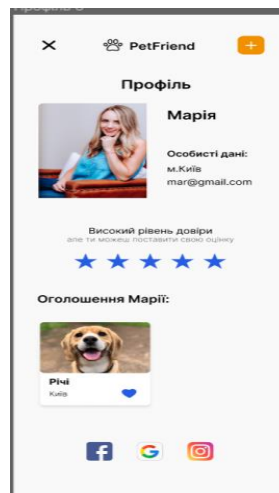


Рисунок 4.18 – Як інші користувачі бачать інші профілі

4.5 Створення інших сторінок

Чат користувачів (користувацький чат) також має важливе значення в мобільних застосунках, бо:

1) Спілкування та соціальна взаємодія: Чат користувачів створює можливість для користувачів спілкуватися між собою, обмінюватися повідомленнями та інформацією. Це стимулює соціальну взаємодію в межах застосунку і сприяє встановленню зв'язків між користувачами, обміну досвідом та знаннями.

2) Обмін контентом: Чат дозволяє користувачам обмінюватися різними видами контенту, такими як фотографії, відео, документи, посилання тощо. Це допомагає забезпечити більш багатогранний та цікавий досвід використання застосунку (рисунок 4.19).



Рисунок 4.19 – Фрагмент чату застосунку

В хорошому застосунку завжди можна знайти контактні дані підтримку для вирішення будь-яких питань або проблем. На дану сторінку можна потрапити через бургерне меню (рисунок 4.20).

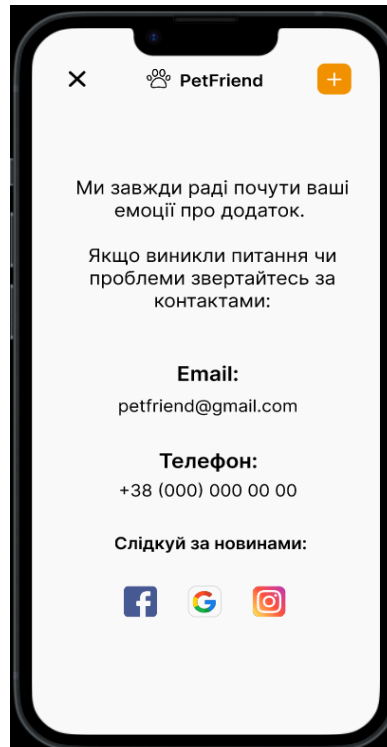


Рисунок 4.20 – Сторінка контактних даних підтримки

Обов'язковою частиною є політика конфіденційності.

Політика конфіденційності для мобільних застосунків є необхідною з кількох причин:

1. **Захист особистої інформації:** Користувачі мобільних застосунків надають свою особисту інформацію, таку як ім'я, адреса електронної пошти, номер телефону тощо. Політика конфіденційності встановлює зобов'язання розробників застосунку щодо захисту цих даних від несанкціонованого доступу, використання чи розкриття.

2. **Визначення збору та використання інформації:** Політика конфіденційності повинна чітко визначати, яку інформацію збирається в мобільному застосунку і як вона використовується. Це надає користувачам впевненість, що їхні дані не будуть використовуватися без їхньої згоди і в рамках визначених меж.

3. **Законодавча вимога:** Багато країн мають закони та нормативні акти, які вимагають від розробників мобільних застосунків встановлювати політику конфіденційності та дотримуватися її положень.

4. Довіра користувачів: Наявність політики конфіденційності сприяє підвищенню довіри користувачів до мобільного застосунку. Користувачі хочуть бути впевнені, що їхні дані будуть оброблятися з уважністю та відповідальністю, і політика конфіденційності допомагає встановити цю довіру.

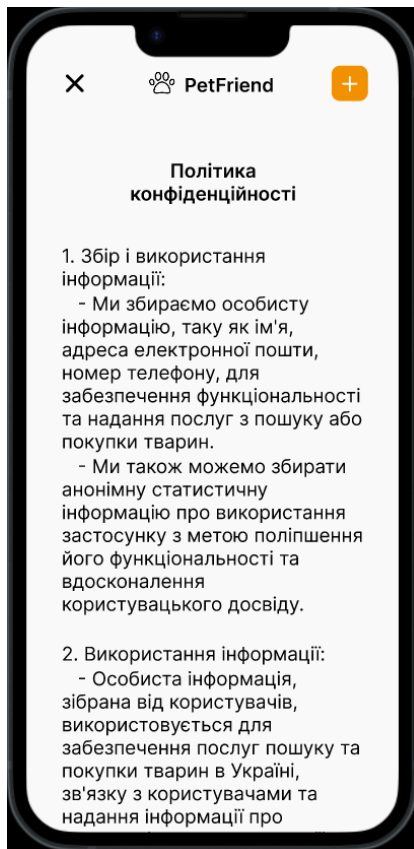


Рисунок 4.5.3 – Політика конфіденційності застосунку

Таким чином ми розглянули наявні екрани застосунку та його функціонал.

ВИСНОВКИ

В ході виконання кваліфікаційної роботи, для досягнення мети були виконані наступні завдання:

- проаналізовано існуючі аналоги на ринку та цільову аудиторію;
- проведено аналіз цілей та задач застосунку;
- проведено аналіз UX та UI складових;
- розроблено мобільний застосунок.

У результаті проведеного аналізу цілей та задач мобільного застосунку «PetFriend», було чітко сформульовано його призначення та функціонал. Він вирішує проблему зі знаходженням потенційних власників для тварин, що шукають новий дім, а також надає можливість розміщення оголошень про продаж або притулок тварин.

Проведений аналіз цільової аудиторії та конкурентів допоміг визначити основні особливості та потреби користувачів. На основі цього був розроблений інтерфейс застосунку, який відповідає потребам цільової аудиторії, забезпечуючи зручну навігацію та доступ до всіх функціональних можливостей.

Архітектура та функціональні можливості застосунку «PetFriend» були ретельно розроблені з урахуванням потреб користувачів. Застосунок надає можливість розміщення оголошень про продаж та притулок тварин, шукати тварин за різними критеріями, переглядати профілі користувачів та їх оголошень, використовувати внутрішній чат для зв'язку між користувачами, залишати відгуки про користувачів та їх тварин, додавати фотографії до оголошень та шукати загублених домашніх улюбленців.

Особлива увага була приділена системі безпеки користувачів. Застосунок передбачає перевірку та підтвердження акаунтів, а також систему відгуків про користувачів та їх тварин.

Застосунок «PetFriend» має потенціал стати популярним серед тих, хто шукає тварину або хоче знайти нового власника для своїх улюбленців.

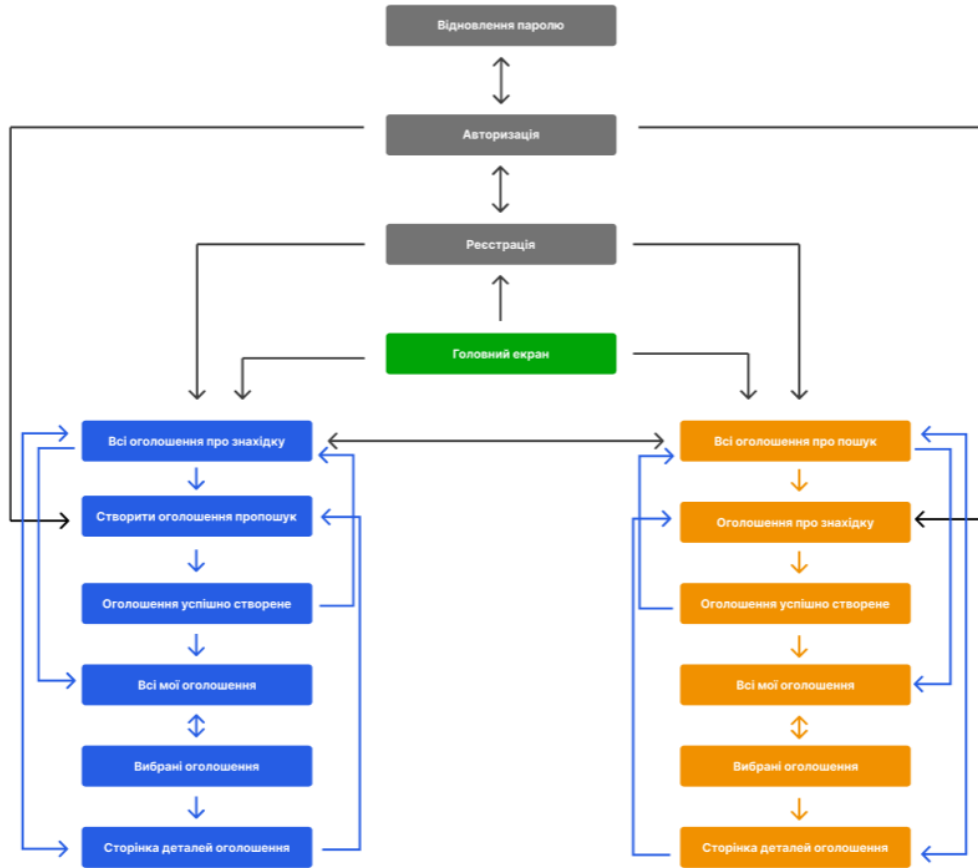
ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. PetMatch [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://petmatch.ie/>
2. The Puppies App [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://thepuppiesapp.com/>
3. WannaPet [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://wannapet.com/>
4. OLX [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://www.olx.ua/uk/>
5. Instagram [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://www.instagram.com/>
6. Prom.ua [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://prom.ua/>
7. Новина про безпритульних тварин [Електронний ресурс] – Режим доступу до ресурсу: <https://www.unian.ua/ecology/v-ukrajini-znachno-zbilshilasya-kilkist-bezdomnih-tvarin-zoozahisniki-sturbovani-12135804.html>
8. Adobe Illustrator [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://www.adobe.com/ua/products/illustrator.html>
9. Figma [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://www.freecodecamp.org/news/figma-crash-course/>
10. Richard F. Raposa, Sun Certified Programmer for Java Platform Study Guide: SE6 [Електронний ресурс]. – 2009. – Режим доступу до ресурсу: https://books.google.com.ua/books?id=1MvzKaWRQjoC&printsec=frontcover&redir_esc=y#v=onepage&q&f=false
11. Symbian OS [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: <https://hi-news.pp.ua/tehnka-tehnologyi/7467-operacyna-sistema-dlya-telefonv-symbian-os.html>
12. Windows Mobile [Електронний ресурс]:[Веб-сайт] – Режим доступу до ресурсу: https://microsoft.fandom.com/wiki/Windows_Mobile

13. Palm OS [Электронный ресурс] – Режим доступа до ресурсу: <https://www.britannica.com/technology/Palm-OS>
14. App Store [Электронный ресурс] – Режим доступа до ресурсу: <https://www.apple.com/ua/app-store/>
15. Swift [Электронный ресурс] – Режим доступа до ресурсу: <https://www.swift.org/>
16. Objective-C[Электронный ресурс] – Режим доступа до ресурсу: <https://developer.apple.com/library/archive/documentation/Cocoa/Conceptual/ProgrammingWithObjectiveC/Introduction/Introduction.html>
17. Kotlin [Электронный ресурс] – Режим доступа до ресурсу: <https://kotlinlang.org/>
18. HTML & CSS Design and Build Websites Jon Ducket [Электронный ресурс]. – 2011. – Режим доступа до ресурсу: <https://wtf.tw/ref/duckett.pdf>
19. JavaScript: The Definitive Guide: Master the World's Most-Used Programming Language, 2020. – 704 с. – (O'Reilly Media). – (7).
20. Flutter [Электронный ресурс] – Режим доступа до ресурсу: <https://flutter.dev/>
21. React Native [Электронный ресурс] – Режим доступа до ресурсу: <https://reactnative.dev/>
22. Adobe Systems [Электронный ресурс] – Режим доступа до ресурсу: http://xn--r1a3b.xn--b1amgblet.xn--j1amh/index.php/Adobe_Systems
23. Canva [Электронный ресурс] – Режим доступа до ресурсу: <https://www.canva.com/>
24. Unsplash [Электронный ресурс] – Режим доступа до ресурсу: <https://unsplash.com/license>
25. Content Reel [Электронный ресурс] – Режим доступа до ресурсу: <https://www.figma.com/community/plugin/731627216655469013/Content-Reel>
26. Spellcheckers[Электронный ресурс] – Режим доступа до ресурсу: <https://www.figma.com/community/plugin/738839069237725273/Spellchecker>

27. Що таке CRM-система та як вона працює? [Електронний ресурс] – Режим доступу до ресурсу: <https://www.creatio.com/page/uk/definition-crm>
28. Colors and Typography [Електронний ресурс] – Режим доступу до ресурсу: <https://medium.com/inkbot-design/colors-and-typography-f7de1b76bf9b>
29. Nielsen’s Heuristics: 10 Usability Principles To Improve UI Design [Електронний ресурс] – Режим доступу до ресурсу: <https://aelaschool.com/en/interactiondesign/10-usability-heuristics-ui-design/>
30. User Flow [Електронний ресурс] – Режим доступу до ресурсу: <https://dou.ua/lenta/columns/user-flows/>
31. Wireframing [Електронний ресурс] – Режим доступу до ресурсу: <https://www.experienceux.co.uk/faqs/what-is-wireframing/>
32. Microsoft Visio [Електронний ресурс] – Режим доступу до ресурсу: <https://www.microsoft.com/uk-ua/microsoft-365/visio/flowchart-software>
33. Data Structures for Raster Graphics, 2012. – 201 с. – (Marloes L.P.van Lierop Springer Science & Business Media).
34. Ліцензія ілюстрацій [Електронний ресурс] – Режим доступу до ресурсу: https://www.flaticon.com/free-sticker/dog_7648697?term=animal&page=3&position=9&origin=style&related_id=7648697

ДОДАТКИ ДОДАТОК А СИТЕМАР



ДОДАТОК Б WIREFRAME

