

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій

"Допущений до захисту"

Завідувач кафедри ІУСТ

к.т.н, доц. Михелєв І.Л.

" ____ " _____ 2022 р.

КВАЛІФІКАЦІЙНА РОБОТА

на здобуття ступеня вищої освіти "бакалавр"

на тему: «Розробка кросплатформного додатку «Соціальна мережа
мисливців та рибалок»

Виконав: студент групи

_____ *Парубченко В.С.*

(підпис)

Керівник роботи:

(посада, науковий ступінь вчене звання)

_____ *Казимиренко Ю.О.*

(підпис)

м. Миколаїв – 2022 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ

Національний університет кораблебудування імені адмірала Макарова
Навчально-науковий інститут комп'ютерних наук та управління проектами
Кафедра інформаційних управляючих систем та технологій
Спеціальність 122 "Комп'ютерні науки"
Освітня програма "Комп'ютерні науки"

"ЗАТВЕРДЖУЮ"

Гарант освітньої програми

к.т.н, доц. Гайда А.Ю.

" ____ " _____ 2022 р.

ЗАВДАННЯ

НА КВАЛІФІКАЦІЙНУ РОБОТУ

на здобуття ступеня вищої освіти "бакалавр"

Студенту Парубченко Владиславу Сергійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи: Розробка кросплатформного додатку «Соціальна мережа мисливців та рибалок»

Керівник роботи: Казимиренко Юлія Алексеевна

Затверджені наказом ректора № 269-уч від "18" травня 2022 року.

2. Термін подання роботи: 20 червня 2022 р.

3. Вихідні дані до проекту (роботи) ДСТУ щодо обробки інформації, літературні джерела, технічна документація на існуючі аналоги систем.

4. Перелік питань, що належать до розробки (найменування розділів)

5. Перелік презентаційних матеріалів _____

6. Консультанти розділів роботи

Розділ	Прізвище, ініціали та посада консультанта	Підпис, дата	
		завдання видав	завдання прийняв
1	Казимиренко Ю.О.	14.02. 2022	21.03.2022
2	Казимиренко Ю.О.	21.03.2022	21.04.2022
3	Казимиренко Ю.О.	21.04.2022	01.06.2022
4	Казимиренко Ю.О.	01.06.2022	07.06.2022

7. Дата видачі завдання 14 лютого 2020 р.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проекту (роботи)	Строк виконання етапів роботи	Примітка
1	Вивчення і опис предметної області	21 березня	
2	Розробка проектних рішень	21 квітня	
3	Реалізація проектних рішень	1 червня	
4	Вирішення питань охорони праці	7 червня	
5	Виконання графічних документів	13 червня	
6	Оформлення пояснювальної записки	19 червня	
7	Подання дипломної роботи (ДР) на кафедральний захист	20 червня	

8	Подання ДР рецензенту	20 червня	
9	Подання ДР на державний захист	27 червня	
10			

Студент

(підпис)

Парубченко В.С.

(ПІБ)

Керівник роботи

(підпис)

Казимиренко Ю.О.

(ПІБ)

АНОТАЦІЯ

У даному дипломному проєкті представлено розробку кросплатформного додатку «Соціальна мережа мисливців та рибалок». Додаток дозволить користувачам, а саме - рибалкам та мисливцям, шукати товаришів, додавати їх до друзів, спілкуватись між собою, вести свій профіль з фото- або відео-галереєю, ділитися цікавими новинами, професійними навичками, створювати теми форуму та групи для ведення свого блогу. Продажі і покупки - невід'ємна частина програмної системи - вона дозволить продавати або робити замовлення товарів.

Дипломний проєкт виконано на 92 сторінках машинописного тексту та містить: 37 рисунків, 47 таблиць та список використаних джерел із 12 найменувань.

Робота викладена українською мовою.

SUMMARY

This diploma project presents the development of a cross-platform application «Social network of hunters and fishermen». The application will allow users, namely fishermen and hunters, to find friends, add them to friends, communicate with each other, maintain a profile with a photo or video gallery, share interesting news, professional skills, create forum topics and groups to run your blog. Sales and purchases are an integral part of the software system - it will allow you to sell or order goods.

The diploma project is made on 92 pages of typewritten text and contains: 37 figures, 47 tables and a list of used sources from 12 names.

The work is presented in Ukrainian.

ЗМІСТ

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ	9
ВСТУП	10
1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	11
1.1 Загальний опис проблеми.....	11
1.2 Аналіз готових рішень.....	14
1.3 Постановка задачі.....	18
2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	20
2.1 Ескізний проєкт програмного забезпечення	20
2.1.1 Побудова контекстної діаграми	22
2.1.2 Розробка діаграм варіантів використання.....	22
2.1.3 Розробка концептуальної моделі.....	33
2.1.4 Розробка діаграми станів.....	39
2.1.5 Проєктування інтерфейсу.....	41
2.2 Технічний проєкт	43
2.2.1 Розробка діаграми послідовності	43
2.2.2 Розробка діаграм класів.....	44
2.2.3 Розробка діаграм діяльності.....	46
2.2.4 Розробка логічної моделі.....	47
2.3 Робочий проєкт	49
2.3.1 Вибір засобів розробки.....	49
2.3.2 Розробка фізичної моделі	50
3 РЕЗУЛЬТАТ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	56
4 ОХОРОНА ПРАЦІ.....	65
4.1 Вступ	65
4.1 Організація робочого місця.....	65
4.2 Мікроклімат робочої зони.....	66
4.3 Освітлення робочого місця	66
4.4 Випромінювання	66
4.5 Організація оптимального режиму праці та відпочинку	66
4.6 Електробезпека.....	67

4.7 Важкість та напруженість праці.....	68
4.8 Висновок	69
ВИСНОВКИ	71
СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ.....	72
ДОДАТОК А – Керівництво користувача.....	73
ДОДАТОК Б – Код модулів	81

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ, СКОРОЧЕНЬ І ТЕРМІНІВ

UML – уніфікована мова моделювання (Unified Modeling Language) - це система позначень, яку можна застосовувати для об'єктно-орієнтованого аналізу і проектування. Його можна використовувати для візуалізації, специфікації, конструювання та документування програмних систем;

БД – база даних;

ADO.NET (ActiveX Data Object для .NET) — технологія, предоставляющая доступ и управление данными, хранящимся в базе данных или других источниках;

ПІ – інформаційна підсистема;

СММР - Соціальна мережа мисливців та рибалок;

MVVM – Model-View-Model.

ВСТУП

Мета дипломного проекту полягає в проектуванні та розробці кросплатформного мобільного додатку «Соціальна мережа мисливців та рибалок».

Додаток дозволить користувачам, а саме - рибалкам та мисливцям, шукати товаришів, додавати їх до друзів, спілкуватись між собою, вести свій профіль з фото- або відео- галереєю, ділитися цікавими новинами, професійними навичками, створювати теми форуму та групи для ведення свого блогу. Продажі і покупки - невід'ємна частина програмної системи - вона дозволить продавати або робити замовлення товарів.

Надати можливість компаніям або ж звичайним зареєстрованим користувачам виставляти свій інвентар або товари на торговельний майданчик по певним категоріям, брендами, моделями і модифікаціям для прискорення продажів і успішного швидкого пошуку.

1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Загальний опис проблеми

Мобільний додаток – це спеціальна програма, яка встановлюється користувачем на мобільний пристрій, через ринки. Найпопулярніші – AppStore, Google Play, де мобільних застосунків налічується уже мільярди з різним призначенням на будь-який смак: різні ігри та розваги, ЗМІ, журнали, книги, соцмережі, гігантські торгові площадки, резервація готелів, замовлення доставки з ресторанів і кафе, мобільні версії служб знайомств, додатки погоди, курсу валют, цін на паливо де дешевше, медичні додатки, що відслідковують самопочуття хворих, доступ до банківських рахунків тощо.

Додатки переслідують, як мінімум, дві явні цілі: не дати власнику нудьгувати та полегшити побут – іншими словами некорисні та корисні мобільні додатки.

Мобільні додатки переважно створюються під якусь конкретну платформу (IOS, Android, Windows Phone та інші).

Основні переваги мобільних додатків:

- повна взаємодія з користувачами, є можливість відправляти Push-повідомлення;
- якщо користувач виконав якусь дію в додатку, яке очікує від нього реакції через певний час, додаток нагадає йому про це;
- так само з'являються більш широкі можливості зворотного зв'язку – користувачі можуть залишити повідомлення, свої відгуки в магазинах додатків, персонально через додаток, соцмережах – і вони автоматично транслюватимуться на усі ресурси;
- якісніший інтерфейс. Всі елементи управління кнопки, текстові поля, посилання повинні бути зручними для натискання пальця на мобільному девайсі, а не для курсора миші;
- високий рівень персоналізації. Мобільні телефони знають про свого власника дуже багато, і використання цієї інформації для збільшення рівня сервісу є однією з ключових причин бурхливого зростання мобільних додатків.

- це ж пункт відноситься і до заліза пристрою – залежності від типу програми, дозволів, підтягуються дані з камери, акселерометра, компаса, барометра і т. ін.;

- локалізація;
- робота в офлайн.

Мобільні додатки – як інструмент для інтернет-магазинів, який дозволяє забезпечити потенційним покупцям унікальний досвід, максимально спростити взаємодію із сайтом і здійснення покупки. У той же час їхнє використання як одного зі способів залучення покупців буде досить витратною інвестицією, з необхідністю регулярно виділяти додаткові кошти на оновлення або внесення будь-яких інших змін.

Мобільні додатки – як інструмент для інтернет-спілкування, який дозволяє забезпечити зареєстрованим користувачам, максимально просту взаємодію між собою і можливість спілкування між собою, переглядати новини тощо.

Соціальна мережа - це домен, вебсайт, має, зазвичай, кілька серверів. Основна мета такого сервісу - зробити інтернет-спілкування між віддаленими Ти один від одного користувачами максимально простим. Деякі такі працюють Тільки в межах однієї країни або популярні у певного мовного сегмента, інші - мають міжнародний формат і на них зареєстровані люди з різних країн.

Принцип роботи:

- Створення сторінки або запису;
- В ньому він може вказати або не вказувати свої інтереси, захоплення, місце роботи та навчання, місце проживання та інші дані;
- Також можна знайти людей, які поділяють ті ж інтереси і захоплення, що і зареєструвався; персональні сайти;
- Тепер з цими людьми можна безперешкодно спілкуватися, а для спрощення цього процесу можна додати в друзі.

Спілкування з допомогою СС доступна тільки зареєстрованим членам, хоча інформація з деяких груп доступна до прочитання і незареєстрованими на сайті людьми.

Перевагами соціальних мереж є:

- Можливість спілкуватися з друзями і колегами на відстані; пошук втрачених контактів і нових знайомств;
- Завдяки соціалкам ви завжди будете в курсі останніх подій, а інколи вони вас попередять задовго до офіційних новинарних ресурсів;

Людина, будь то рибалка або мисливець має потребу в спілкуванні і пошуку нових друзів, товаришів, продаж старих або купівлі нових споряджень, представники компаній так само зацікавлені в найкоротші терміни продати свій товар і отримати вигоду.

Існує не мало додатків, наприклад, Фейсбук. Людина зареєструвалась з метою продати кілька товарів. Розглянемо проблему. Користувач створює звичайну групу, дає їй відповідну назву, оформляє сторінку і додає товари на продаж. Для початку творець цієї групи повинен запросити людей, щоб вони підписалися, іншими словами набрати передплатників. Далі, що йому варто робити і робити - це піарити і розвивати свою групу для просування і для того, щоб якомога більше людей могло побачити його товар і надалі придбати. Якщо людина захотіла продати всього кілька своїх товарів – це проблема, для вирішення якої потрібно чимало часу.

Припустимо, у продавця все склалося, він набрав достатню кількість людей в свою групу і активність в ній росте. І ось, він виставив дуже рідкісний товар і покупці відразу відреагували. Продавець отримав повідомлення відразу від декількох покупців в одне і теж час, але що ж йому робити, як визначити, хто перший йому написав, якщо це відбулося практично одночасно – виявляються труднощі.

Об'єктом розробки є додаток, який дозволить вести спілкування користувачам певної категорії, а саме – рибалок та мисливців. Вони зможуть віддалено знайомитися, переглядати торгову площадку для пошуку потрібних товарів, домовлятися про купівлю / продаж, бронювати товар.

Всі товари повинні бути розділені за категоріями, брендами, моделями і модифікаціями для прискорення продажів і успішного швидкого пошуку. Це дозволить людині ввести ключове слово, будь то категорія, бренд або модель, і

відразу ж знайти товар, якщо такий є. Подальші дії - це забронювати товар і списатися з продавцем для договору про оплату, відправлення і таке інше. Функціонал додатку не передбачає оформлення актів купівлі/продажу.

Інтерфейс повинен бути інтуїтивно зрозумілим, доброзичливим по відношенню до користувача.

1.2 Аналіз готових рішень

Facebook

Соціальна мережа Facebook(логотип представлено на рис. 1.1) - найбільша в світі, щомісячна аудиторія якої досягає 2 млрд користувачів. Ресурс був заснований Марком Цукербергом і його друзями під час навчання в Гарвардському університеті в 2004 році. Офіційно соціальна мережа належить компанії Facebook Inc., яка також володіє сервісами WhatsApp і Instagram. [1]



Рисунок 1.1 – Логотип компанії Facebook

Наведемо приклади найбільших на даний момент соціальних мереж. Найвідвідуванішою є Facebook. Даний ресурс спочатку був закритим і був доступний тільки для аудиторії студентів Гарварда. Протягом місяця більше половини студентів університету зареєструвалися в системі, реєстрація була відкрита також для студентів Стенфорда і Єльського університету. Потім членами соціальної мережі змогли стати всі учні Ліги Плюща, яка налічує 8 кращих університетів США, а поступово і взагалі всі студенти Північної Америки. У вересні 2006 р керівництво Facebook відкрило реєстрацію для всіх користувачів світового інтернету, що призвело до величезної популярності

ресурсу серед широкої публіки. На сьогодні кількість активних користувачів сайту перевищує 175 мільйонів осіб.

OLX

Платформа(логотип представлений на рисунку 1.2) онлайн-оголошень, яка об'єднує людей для покупки, продажу або обміну товарами і послугами. Оголошення класифікуються за такими категоріями, як «Дитячий світ», «Нерухомість», «Транспорт», «Запчастини для транспорту», «Робота», «Тварини», «Будинок і сад», «Електроніка», «Бізнес і послуги», «Мода і стиль», «Хобі, відпочинок і спорт», «Віддам даром» та «Обмін».



Рисунок 1.2 – Логотип OLX

Станом на 2018 рік налічується 35 штаб-квартир інтернет-компанії OLX, розташованих в таких містах, як Буенос-Айрес, Київ, Познань, Ташкент, Бухарест, Софія, Пекін, Алма-Ата, Мумбаї та інших. Щомісяця сайти OLX відвідує понад 330 мільйонів людей з 40+ країн світу.

Робоча версія додатку представлена на рисунку 1.3.

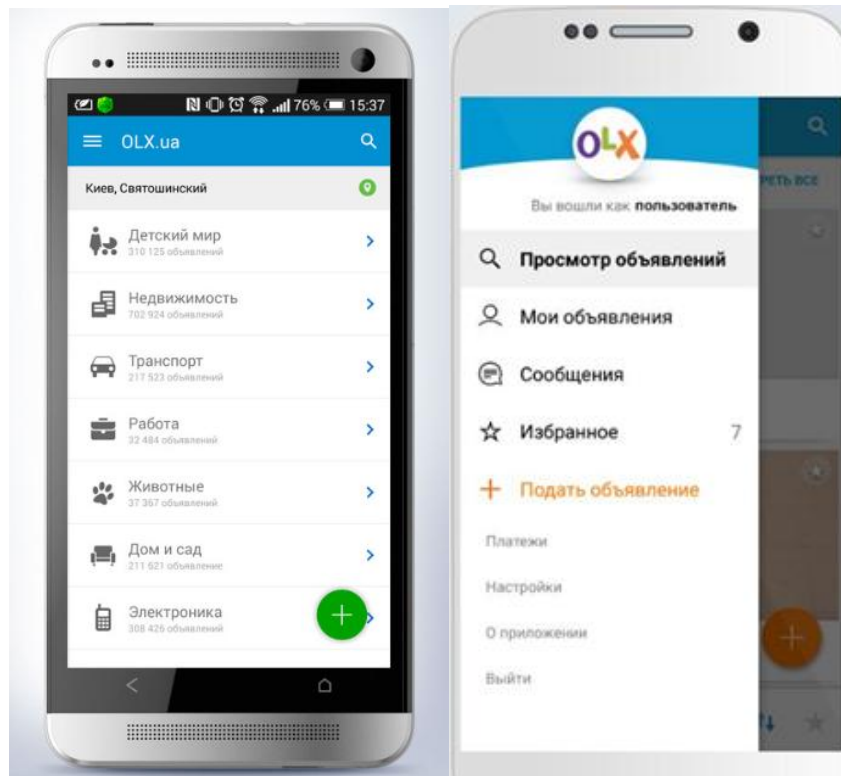


Рисунок 1.3 – Мобільний додаток OLX

Rozetka.com.ua

Напевно, було б дивно, якби найбільший в Україні інтернет-магазин не мав власного мобільного додатка. Він є у версії як для телефонів, так і для планшетів, з різними розширеннями екрана. Видно, що над юзабіліті добре попрацювали, щоб забезпечити потенційним покупцям максимальну зручність при перегляді каталогу й управлінні особистим профілем. [1]

Логотип Rozetka представлений на рисунку 1.4.



Рисунок 1.4 – Логотип Rozetka

Fishbrain

Fishbrain - це онлайн-сервіс для ведення мобільних журналів, обміну фотографіями та соціальних мереж, який дозволяє користувачам записувати дані і робити знімки уловів, а також обмінюватися ними публічно або конфіденційно в додатку, а також за допомогою безлічі інших соціальних мереж. такі як Facebook, Twitter, Tumblr і Flickr.

Fishbrain представлений на рисунку 1.5.

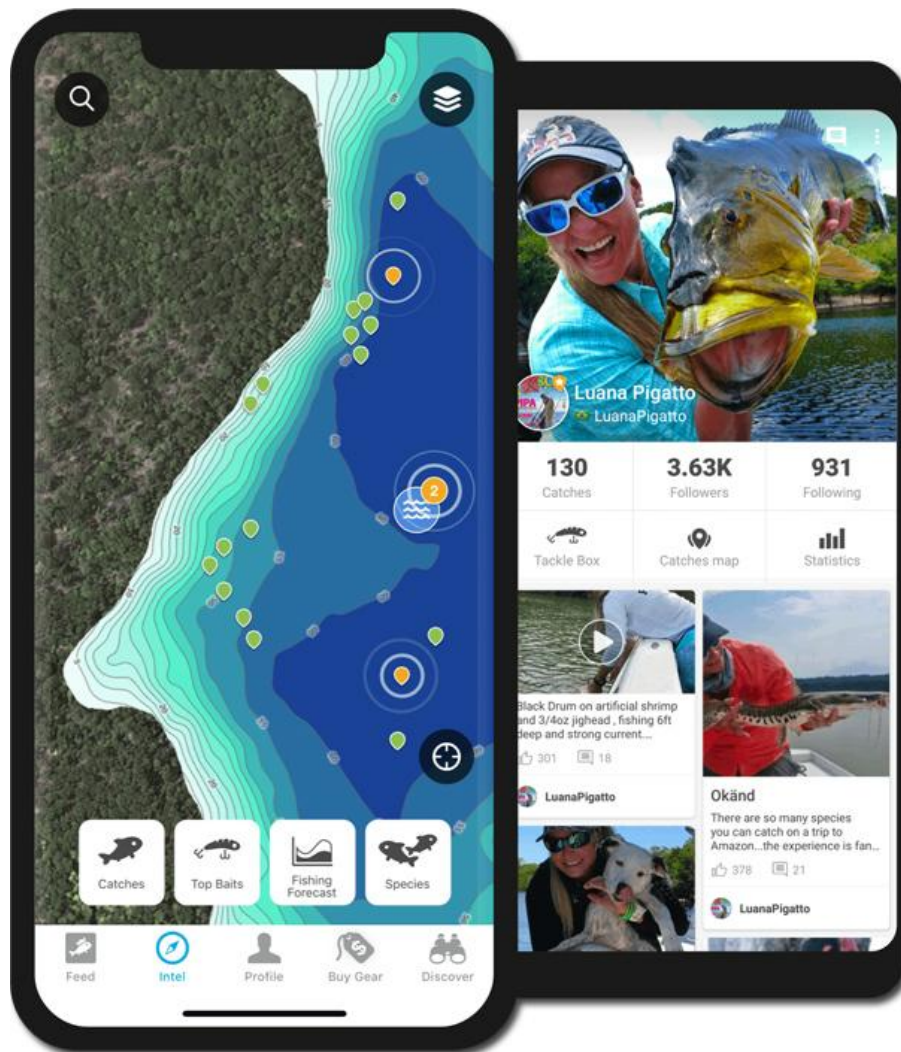


Рисунок 1.5 – Мобільний додаток Fishbrain

Більшість соціальних мереж, які існують на даний час, не є цілеспрямованими на конкретну аудиторію, вони мають зайвий функціонал, який частіше всього не використовують. Тому й виникла ідея, реалізувати дану соціальну мережу, з функціоналом, який люди будуть часто використовувати для задоволення своїх потреб.

1.3 Постановка задачі

Виходячи з аналізу предметної області необхідно виконати проектування і реалізацію кросплатформного додатку «Соціальна мережа мисливців та рибалок». Користувачам буде доступний пошук товаришів з подальшим додаванням їх до друзів; обмін повідомленнями між собою; ведення профілю з фото- або відео- галереєю; ділитися новинами, професійними навичками; створювати теми форуму та групи для ведення свого блогу. Дати можливість компаніям або ж звичайним користувачам виставляти товар на торговельний майданчик по певним категоріям, брендам, моделям і модифікаціям для прискорення продажів і успішного швидкого пошуку.

ПІ має забезпечувати такий функціонал:

- пошук друзів;
- ведення списку друзів;
- обмін повідомленнями;
- ведення форуму;
- ведення груп;
- ведення профілю;
- публікація фото, відео у профіль;
- ведення конкурсів;
- ведення подій;
- публікація товарів на продаж;
- пошук товарів по певним категоріям;
- замовлення товарів.

Вхідні дані:

- інформація про друзів;
- інформація про категорію товару;
- інформація про конкурс;
- інформація про подію;
- інформація про товар;
- інформація про користувача;

- інформація про замовлення;
- інформація про групи.

Вихідні дані:

- список друзів;
- список категорій товарів;
- список конкурсів;
- список подій;
- список товарів;
- список користувачів;
- список замовлень;
- список груп.

2 ПРОЄКТ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

2.1 Ескізний проєкт програмного забезпечення

Для побудови своєї системи була обрана об'єктно-орієнтована методологія. Структура системи описується в термінах об'єктів і зв'язків між ними, поведінка системи - в термінах обміну повідомленнями між об'єктами.

Причини вибору даної методології були наступні:

- Використання об'єктного підходу істотно підвищує рівень уніфікації розробки і придатність для повторного використання;
- Об'єктна декомпозиція дозволяє уникнути створення складних моделей;
- Об'єктна модель природна, оскільки орієнтована на людське сприйняття світу.

Для проєктування ОО системи використовується уніфікована мова моделювання UML.

UML – стандартна візуальна нотація для опису архітектури ІП. Таким чином, всі учасники проєкту спілкуються на одній мові і використовують єдиний інструмент. Всі учасники проєкту можуть використовувати Rational Rose для визначення архітектури програми і спілкування між собою.

Rational Rose – лідируючий засіб візуального моделювання – є простим і повністю інтегрованим рішенням для розробки ІП, включаючи інтернетрешення. Rational Rose є стандартом дефакто серед інструментів проєктування додатків. Жодне інше CASE-засіб не пропонує таку широту і глибину рішень як платформа Rational.

Rational Rose дозволяє візуалізувати, зрозуміти й уточнити вимоги і архітектуру перед початком створення коду. Маючи здатність представляти користувальницький інтерфейс окремо від бізнеслогіки і даних, Rational Rose дозволяє тримати контроль над розробкою ІП. Це дає можливість виключити витрати сил на непотрібні переробки. Використання одного інструмента на протязі всього життєвого циклу розробки допомагає будувати «правильну» систему.

Adobe Experience Design - програма для розробки інтерфейсів від Adobe Systems. Підтримує векторну графіку і веб-верстку і створює невеликі активні прототипи. [2]

Adobe XD створений для команд

Найбільш важливі оновлення вийшли на XD за останні 12 місяців дуже перспективні, і обіцяють нам ще більші зміни в майбутньому. Багато з них створені з урахуванням необхідності роботи в команді. Правда на даний момент такий функціонал доступний тільки в збірці для закритого тестування і в загальнодоступній бета-версії багато ще не реалізовано.

- Інтеграція з бібліотеками Creative Cloud.

У XD легко переміщати поточні проєкти між пристроями і іншими продуктами Adobe. Хмарне управління проєктами також полегшує безперешкодну передачу даних між членами команди.

- URL адреса.

Просте присвоєння проєкту URL адреси для вбудованого, прямого коментування товаришами по команді і клієнтами.

- керівництво по стилю.

Створює і публікує єдине керівництво по стилю для будь-якого файлу з унікальним URL адресою, яким можна поділитися.

- історія змін і управління версіями.

Швидке завантаження і повернення до попередніх версій.

- спільна робота в реальному часі.

Тепер можна працювати одночасно в тому ж самому файлі.

- сумісність.

XD по-справжньому швидкий. Додаток замислювалося розробниками як оперативне в роботі і демонструє це. Особливий акцент творці зробили на реалізації нових можливостей без збільшення обсягу дистрибутива. Навіть завантаження великого файлу не заважає додатком, справлятися з будь-яким завданням миттєво.

Draw.io - це сервіс, призначений для формування діаграм і схем. Сервіс розділений на три частини - меню, панель об'єктів і сам документ.

За допомогою веб-сервісу Draw.io можна створювати:

- діаграми;
- UML-моделі;
- вставка в діаграму зображень;
- графіки;
- блок-схеми;
- форми;
- інше.

2.1.1 Побудова контекстної діаграми

Проектування ІІ починається із розробки контекстної діаграми.

Контекстна діаграма дозволяє швидко, коротко і ємко описати призначення і межі системи, виявити і усунути колективні розбіжності в їх розумінні, показати і домовитися про її масштаби.

Контекстна діаграма представлена діаграмою варіантів використання UML на рисунку 2.1.

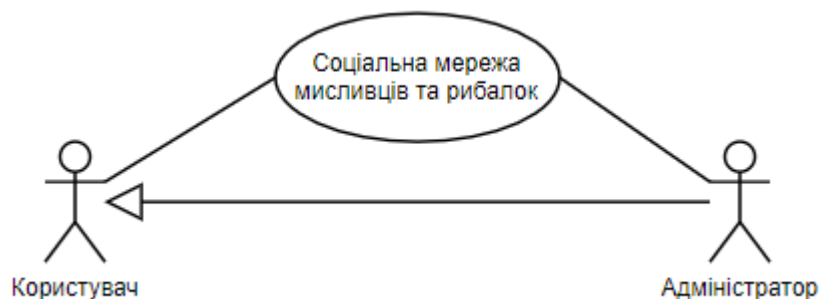


Рисунок 2.1 – Контекстна діаграма UML для ІІ «Соціальна мережа мисливців та рибалок»

2.1.2 Розробка діаграм варіантів використання

Діаграма варіантів використання UML - це граф, що складається з множини акторів, прецедентів обмежених границею системи, асоціацій між акторами та прецедентами, відношень серед прецедентів, та відношень узагальнення між акторами.

Діаграма варіантів використання представлена на рисунку 2.2.

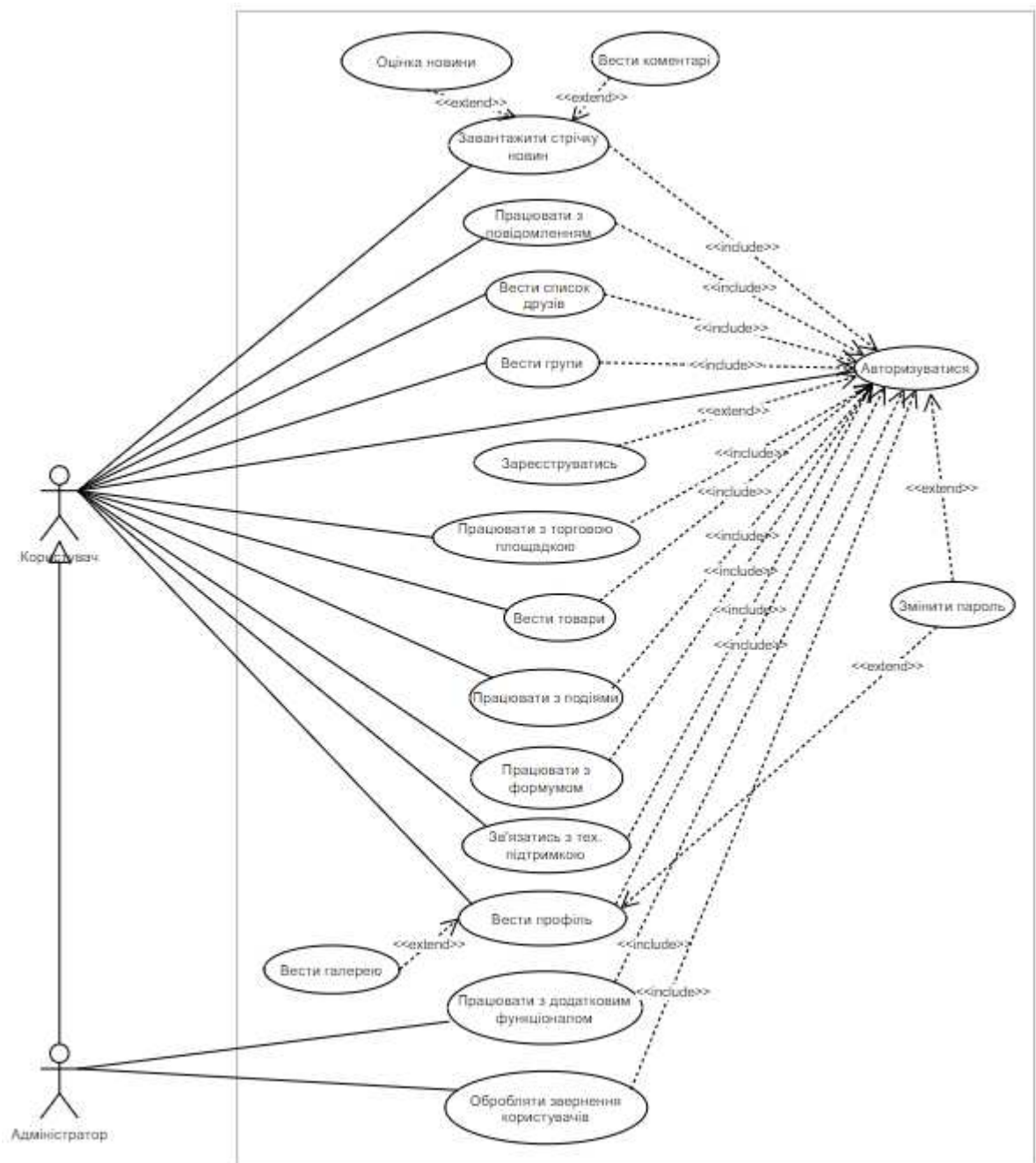


Рисунок 2.2 – Діаграма варіантів використання ІП «Соціальна мережа мислівців та ребалок»

Специфікації варіантів використання наведено в таблицях 2.1-2.18.

Таблиця 2.1 – Специфікація варіанту використання «Завантажити стрічку новин»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу переглядати стрічку новин, ставити «лайк», коментувати.
Передумови	Виконання прецеденту «Авторизуватися»

Продовження таблиці 2.1 – Специфікація варіанту використання
«Завантажити стрічку новин»

Характеристика	Опис
1	2
Основний потік подій:	
Перший потік – потік завантаження новин.	1. Користувач ініціює перехід до інтерфейсу «Новини». 2. Система відображає список новин.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.2 – Специфікація варіанту використання «Працювати з повідомленням»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу переглядати, писати повідомлення.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Повідомлення». Система відображає список діалогів користувача.
Перший потік – потік написання повідомлення	1. Користувач позиціонується на діалозі; 2. Система відображає інтерфейс «Діалог з користувачем»; 3. Користувач позиціонується на полі вводу повідомлення; 4. Користувач ініціює написання коментарю; 5. Користувач ініціює відправлення; 6. Система оновлює інтерфейс та зберігає дані.
Другий потік – потік видалення повідомлення	1. Користувач позиціонується на діалозі; 2. Система відображає інтерфейс «Діалог з користувачем»; 3. Користувач позиціонується на повідомленні, яке потрібно видалити та ініціює видалення; 4. Система видаляє обране повідомлення та оновлює інтерфейс
Третій потік – потік редагування повідомлення	1. Користувач позиціонується на діалозі; 2. Система відображає інтерфейс «Діалог з користувачем»; 3. Користувач позиціонується на повідомленні, яке потрібно редагувати та ініціює редагування; 4. Система зберігає дані та оновлює інтерфейс
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.3 – Специфікація варіанту використання «Вести список друзів»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести список, додавати або видаляти друзів.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Друзі». Система відображає список друзів.
Перший потік – потік додавання друга.	1. Користувач ініціює перехід до вкладки «Пошук»; 2. Користувач вводить інформацію пошуку; 3. Система відображає результати; 4. Якщо результат задовольняє, ініціює додавання до друзів; 5. Система зберігає дані.
Другий потік – потік видалення друга	1. Користувач ініціює перехід до вкладки «Мої»; 2. Система відображає друзів користувача; 3. Користувач позиціонується на другові; 4. Користувач ініціює видалення 5. Система оновлює інтерфейс та зберігає дані.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.4 – Специфікація варіанту використання «Вести групи»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести групи.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Групи». Система відображає обрані групи користувача.
Перший потік – потік створення групи.	1. Користувач ініціює створення групи; 2. Система відображає інтерфейс створення групи; 3. Користувач вводить дані та ініціює створення; 4. Система зберігає дані.
Другий потік – потік видалення групи.	1. Користувач позиціонується на свою групу; 2. Користувач ініціює видалення; 3. Система зберігає дані.
Третій потік – потік підписки на групу	1. Користувач ініціює перехід до вкладки «Пошук»; 2. Користувач вводить інформацію пошуку; 3. Система відображає результати; 4. Якщо результат задовольняє, ініціює підписку на групу; 5. Система зберігає дані.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.5 – Специфікація варіанту використання «Зареєструватись»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу зареєструватись в системі.
Передумови	
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Зареєструватись». Система відображає інтерфейс.
Перший потік – потік реєстрація.	1. Користувач ініціює ввід реєстраційних даних; 2. Користувач ініціює реєстрацію; 3. Система перевіряє введену інформацію на наявність користувача з такими ж даними, якщо інформація задовольняє – система зберігає дані та пропонує виконати прецедент «Авторизуватись», інакше виконується альтернативний потік подій.
Альтернативний потік подій	Система виводить повідомлення користувачу, пропонує ввести нові реєстраційні дані.
Постумови	Не визначені

Таблиця 2.6 – Специфікація варіанту використання «Авторизуватись»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу авторизуватись.
Передумови	
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Авторизуватись». Система відображає інтерфейс.
Перший потік – потік авторизації.	1. Користувач вводить авторизаційні дані та ініціює вхід; 2. Система перевіряє введену інформацію на наявність користувача. Якщо авторизація успішна, система ініціює перехід до інтерфейсу «Головна», інакше виконується альтернативний потік подій.
Альтернативний потік подій	Система повідомляє, що такого користувача в базі немає. Пропонує виконати прецеденти «Зареєструватись» або «Змінити пароль»
Постумови	Не визначені

Таблиця 2.7 – Специфікація варіанту використання «Змінити пароль»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу змінити пароль.
Передумови	
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Змінити пароль». Система відображає інтерфейс.
Перший потік – потік зміни паролю.	1. Користувач вводить запропоновані системою дані та ініціює зміну паролю; 2. Система перевіряє відповідність даних. Якщо дані співпали, система ініціює відправлення коду підтвердження зміни паролю на пошту, яка була використана при реєстрації. 3. Користувач вводить код підтвердження та ініціює зміну

	пароллю.
--	----------

Продовження таблиці 2.7 – Специфікація варіанту використання «Змінити пароль»

Характеристика	Опис
1	2
Перший потік – потік зміни паролю.	4. Система перевіряє відповідність коду підтвердження. Якщо дані співпали, система змінює пароль та відображає повідомлення з успішною зміною паролю, інакше виконується альтернативний потік подій.
Альтернативний потік подій	Система повідомляє, що такого користувача в базі немає. Пропонує перевірити вхідні дані або виконати прецедент «Зареєструватись».
Постумови	Не визначені

Таблиця 2.8 – Специфікація варіанту використання «Працювати з торговим майданчиком»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу працювати з торговою площадкою.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Торговий майданчик». Система відображає інтерфейс.
Перший потік – потік пошук товарів.	1. Користувач позиціонується на полі пошуку; 2. Користувач ініціює написання назви товару; 3. Система відображає товари.
Другий потік – потік пошук товарів за фільтрами.	1. Користувач ініціює вивід фільтрів; 2. Система відображає доступні фільтри; 3. Користувач ініціює вибір фільтрів; 4. Користувач ініціює пошук; 5. Система відображає товари за фільтрами.
Третій потік – потік додавання товару до обраних	1. Користувач позиціонується на товарі та ініціює додавання до обраних; 2. Система додає товар до обраних.
Четвертий потік – потік перегляду товару.	1. Користувач позиціонується на товарі; 2. Система ініціює відображення інтерфейсу «Товар».
П'ятий потік – потік написання повідомлення основнику товару	1. Користувач позиціонується на товарі; 2. Система ініціює відображення інтерфейсу «Товар»; 3. Користувач ініціює написання повідомлення; 4. Система відправляє повідомлення основнику товару.
Шостий потік – потік бронювання товару	1. Користувач позиціонується на товарі; 2. Система ініціює відображення інтерфейсу «Товар»; 3. Користувач ініціює бронювання товару; 4. Система бронює товар, відправляє повідомлення основнику

	товару та оновлює інтерфейс.
--	------------------------------

Продовження таблиці 2.8 – Специфікація варіанту використання
«Працювати з торговим майданчиком»

Характеристика	Опис
1	2
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.9 – Специфікація варіанту використання «Працювати з подіями»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу працювати з подіями.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Події». Система відображає інтерфейс.
Перший потік – потік пошук подій.	1. Користувач позиціонується на полі пошуку; 2. Користувач ініціює написання; 3. Користувач ініціює пошук; 4. Система відображає відповідні події.
Другий потік-потік перегляд події.	1. Користувач позиціонується на події; 2. Система відображає інтерфейс.
Третій потік – потік прийняття участі в події.	1. Користувач позиціонується на події; 2. Система відображає інтерфейс. 3. Користувач ініціює прийняття участі. 4. Система додає користувача до учасників. и
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.10 – Специфікація варіанту використання «Працювати з форумом»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу працювати з форумом, підписуватись на теми, задавати або відповідати на питання.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Форум». Система відображає інтерфейс.

Продовження таблиці 2.10 – Специфікація варіанту використання
«Працювати з формумом»

Характеристика	Опис
1	2
Перший потік – потік створення теми форуму.	1. Користувач ініціює створення теми; 2. Система відображає інтерфейс створення теми; 3. Користувач ініціює введення даних; 4. Користувач ініціює зберігання; 5. Система перевіряє наявність теми у системі. Якщо тема вже існує, система видає попередження, інакше – зберігає дані та оновлює інтерфейс.
Другий потік – потік пошук теми.	1. Користувач позіціонується на полі пошуку та вводить тему; 2. Система відображає відповідні теми форуму;
Третій потік – потік перегляду відповідей теми форуму	1. Користувач позіціонується на темі форуму; 2. Система відображає інтерфейс з відповідями на тему;
Четвертий потік – написання відповіді на тему форуму	1. Користувач позіціонується на темі форуму; 2. Система відображає інтерфейс; 3. Користувач позіціонується на полі вводу відповіді; 4. Користувач ініціює відправлення; 5. Система зберігає дані та оновлює інтерфейс.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.11 – Специфікація варіанту використання «Вести коментарі»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести коментарі.
Передумови	Виконання прецеденту «Завантажити стрічку новин»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Коментарі». Система відображає інтерфейс.
Перший потік – потік написання коментарю.	1. Користувач позіціонується на полі вводу; 2. Користувач ініціює написання коментарю; 3. Користувач ініціює відправлення коментарю; 4. Система зберігає дані та оновлює інтерфейс.
Другий потік – потік редагування коментарю.	1. Користувач позіціонується на коментарі; 2. Користувач ініціює редагування; 3. Система зберігає дані та оновлює інтерфейс.
Третій потік – потік видалення коментарю.	1. Користувач позіціонується на коментарі; 2. Користувач ініціює видалення; 3. Система зберігає дані та оновлює інтерфейс.

Продовження таблиці 2.11 – Специфікація варіанту використання «Вести коментарі»

Характеристика	Опис
1	2
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.12 – Специфікація варіанту використання «Вести галерею»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести галерею.
Передумови	Виконання прецеденту «Вести профіль»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Галерея». Система відображає інтерфейс.
Перший потік – потік додавання посту	1. Користувач ініціює додавання; 2. Система відображає інтерфейс вибору; 3. Користувач ініціює вибір та додає опис; 4. Користувач ініціює підтвердження додавання; 5. Система зберігає дані та оновлює інтерфейс.
Другий потік – потік редагування посту	1. Користувач позиціонується на посту; 2. Користувач ініціює редагування; 3. Користувач ініціює збереження даних; 4. Система зберігає дані та оновлює інтерфейс.
Третій потік – потік видалення посту	1. Користувач позиціонується на посту; 2. Користувач ініціює видалення; 3. Система видаляє запис та оновлює інтерфейс.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.13 – Специфікація варіанту використання «Зв'язатись з тех. підтримкою»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу зв'язатись з тех. підтримкою.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій	Користувач ініціює перехід до інтерфейсу «Тех. підтримка». Система відображає інтерфейс.
Перший потік - потік написання повідомлення	1. Користувач позиціонується на полі вводу повідомлення; 2. Користувач ініціює ввід повідомлення; 3. Користувач ініціює відправлення;

	4. Система зберігає дані
--	--------------------------

Продовження таблиці 2.13 – Специфікація варіанту використання «Зв'язатись з тех. підтримкою»

Характеристика	Опис
1	2
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.14 – Специфікація варіанту використання «Вести профіль»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести свій профіль.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Профіль». Система відображає інтерфейс.
Перший потік – потік редагування профілю.	1. Користувач ініціює редагування профілю; 2. Користувач ініціює збереження інформації; 3. Система зберігає дані.
Альтернативний потік подій	Виконання прецеденту «Змінити пароль»
Постумови	Не визначені

Таблиця 2.15 – Специфікація варіанту використання «Працювати з додатковим функціоналом»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє адміністратору використовувати додатковий функціонал системи.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Додатковий функціонал». Система відображає інтерфейс.
Перший потік – потік вибір функції.	1. Адміністратор позиціонується на функції та ініціює її; 2. Система виконує функцію.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.16 – Специфікація варіанту використання «Обробляти звернення користувачів»

Характеристика	Опис
----------------	------

1	2
Короткий опис	Даний варіант використання дозволяє адміністратору обробляти звернення від користувачів.

Продовження таблиці 2.16 – Специфікація варіанту використання «Обробляти звернення користувачів»

Характеристика	Опис
1	2
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Звернення користувачів». Система відображає інтерфейс зі списком повідомлень користувачів.
Перший потік – потік підтвердження звернення.	1. Адміністратор позиціонується на повідомленні; 2. Адміністратор ініціює підтвердження; 3. Система змінює статус повідомлення.
Другий потік – потік відхилення звернення.	1. Адміністратор позиціонується на повідомленні; 2. Адміністратор ініціює відхилення; 3. Система видаляє запис з БД
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.17 – Специфікація варіанту використання «Вести товари»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести групи.
Передумови	Виконання прецеденту «Авторизуватися»
Основний потік подій:	Користувач ініціює перехід до інтерфейсу «Групи». Система відображає обрані групи користувача.
Перший потік – потік створення товару.	1. Користувач ініціює створення товару; 2. Система відображає інтерфейс створення товару; 3. Користувач вводить інформацію про товар та ініціює створення; 4. Система зберігає дані.
Другий потік – потік редагування товару.	1. Користувач позиціонується на товар; 2. Користувач ініціює редагування; 3. Система зберігає дані.
Третій потік – потік видалення товару.	1. Користувач позиціонується на товар; 2. Користувач ініціює видалення; 3. Система видаляє товар та оновлює інтерфейс..
Альтернативний потік подій	Відсутній
Постумови	Не визначені

Таблиця 2.18 – Специфікація варіанту використання «Оцінка новини»

Характеристика	Опис
1	2
Короткий опис	Даний варіант використання дозволяє користувачу вести групи.
Передумови	Виконання прецеденту «Завантажити стрічку новин»
Основний потік подій: Перший потік – потік оцінка новини.	1. Користувач позиціонується на новині; 2. Користувач ініціює одобрення; 3. Система оновлює інтерфейс та зберігає дані.
Альтернативний потік подій	Відсутній
Постумови	Не визначені

2.1.3 Розробка концептуальної моделі

Концептуальна модель бази даних - це якась наглядна діаграма, намальована в прийнятих позначеннях. Створюється концептуальна модель для подальшого проектування бази даних та переведення її, наприклад, в реляційну базу даних.

Концептуальна модель даних ERD за нотацією Чена для ІП «Соціальна мережа мисливців та рибалок» представлена на рисунку 2.3.

Сутності та їх атрибути концептуальної моделі даних представлені у таблиці 2.19.

Таблиця 2.19 – Таблиця сутностей та їх атрибутів

Сутність	Атрибути
1	2
Зображення	Id Name (Назва) Extension (Розширення файлу)
Користувач	Id FirstName (Ім'я) LastName (Прізвище) Patronymic (По батькові) Login (Логін) Pass_hash (Хеш паролю) Pass_salt («Соль») Email (Пошта) Phone (Телефон)

	DOB (Дата народження) Id_role (Ідентифікатор ролі) Id_feedback (Ідентифікатор відгука) Confirm_code (Код підтвердження)
--	--

Продовження таблиці 2.19 – Таблиця сутностей та їх атрибутів

Сутність	Атрибути
1	2
Користувач	Last_login_DT (Дата останнього входу) Reg_DT (Дата реєстрації) Notify (Уведомлення) Active_status (Статус користувача)
Профіль	Id Id_owner (Ідентифікатор користувача) Id_image (Ідентифікатор зображення) Description (Опис)
Роль	Id Name (Назва)
Конкурс	Id Id_owner (Ідентифікатор користувача) Title (Назва) Description (Опис) Create_DT (Дата створення) Last_DT (Дата закінчення)
Подія	Id Id_owner (Ідентифікатор користувача) Title (Назва) Description (Опис) Max_members (Максимальна кількість учасників) Start_DT (Дата початку події) Create_DT (Дата створення)
Пост	Id Id_profile (Ідентифікатор профілю) Id_dialog (Ідентифікатор діалогу) Description (Опис) Likes (Кількість лайків) Create_DT (Дата створення)
Товар	Id Id_owner (Ідентифікатор користувача) Id_dialog (Ідентифікатор діалогу) Characteristics (Характеристики) Description (Опис) Views (Перегляди) Create_DT (Дата створення)
Форум	Id Id_image (Ідентифікатор зображення) Id_dialog (Ідентифікатор діалогу) Title (Назва) Create_DT (Дата створення)

Група	Id Id_owner (Ідентифікатор користувача) Id_dialog (Ідентифікатор діалогу) Title (Назва) Description (Опис)
-------	--

Продовження таблиці 2.19 – Таблиця сутностей та їх атрибутів

Сутність	Атрибути
1	2
Група	isPrivate Create_DT (Дата створення)
Діалог	Id Id_image (Ідентифікатор зображення) Title (Назва) Create_DT (Дата створення)
Повідомлення	Id Id_dialog (Ідентифікатор діалогу) Id_user (Ідентифікатор користувача) Text (Текст повідомлення) Create_DT (Дата створення) Changed (Змінений)
Оповіщення	Id Id_user (Ідентифікатор користувача) Id_Type (Ідентифікатор типу) Description (Опис)
Тип оповіщення	Id Name (Назва)
Технічна підтримка	Id Id_dialog (Ідентифікатор діалогу)

Сутності та їх зв'язки концептуальної моделі даних відображені у таблиці 2.20.

Таблиця 2.20 – Сутності та зв'язки

Сутності	Тип зв'язку	Зміст зв'язку
1	2	3
Користувач-Профіль	1:M	Профіль може мати тільки одного користувача.
Користувач-Роль	M:1	Користувач може мати одну роль. Одна роль може бути у багатьох користувачів.
Користувач-Користувач	M:M	Користувач може мати багато користувачів у чорному списку. Багато користувачів можуть мати одного користувача у чорному списку.
Користувач-Оповіщення	M:M	Користувач може мати багато оповіщень. Оповіщення можуть бути для безліч користувачів..
Користувач-Оповіщення	1:M	Користувач може бути ініціатором багатьох оповіщень. Оповіщення може мати тільки одного ініціатора.

Оповіщення-Тип оповіщення	M:1	Оповіщення может мати тільки один тип повідомлення Безліч оповіщень можуть мати один тип повідомлення.
---------------------------	-----	---

Продовження таблиці 2.20 – Сутності та зв'язки

Сутності	Тип зв'язку	Зміст зв'язку
1	2	3
Користувач-Повідомлення	1:M	Багато повідомлень може бути відправлено одним користувачем. Відправник може бути тільки один.
Користувач-Діалог	M:M	Користувач може мати безліч діалогів. Діалог может мати безліч користувачів.
Користувач-Діалог	M:1	Користувачі можуть мати діалог з відгуками. Відгук у користувача може бути тільки один.
Користувач-Група	1:M	Користувач може бути власником декількох груп. Власник групи може бути тільки один.
Користувач-Група	M:M	Користувачі можуть бути учасниками декількох груп. Група може мати безліч учасників.
Користувач-Конкурс	1:M	Користувач може бути власником декількох конкурсів. Власник конкурсу може бути тільки один.
Користувач-Конкурс	M:M	Користувачі можуть бути учасниками декількох конкурсів. Конкурс може мати безліч учасників.
Користувач-Подія	1:M	Користувач може бути власником декількох подій. Власник події може бути тільки один.
Користувач-Подія	M:M	Користувачі можуть бути учасниками декількох подій. Подія може мати безліч учасників.
Користувач-Товар	1:M	Користувач може додавати безліч товарів. Власник товару може бути тільки один.
Користувач-Форум	1:M	Засновник форуму може бути тільки один. Багато форумів може мати один користувач.
Діалог-Повідомлення	1:M	Діалог може мати багато повідомлень. Повідомлення може мати діалог.
Діалог-Форум	1:M	Форум може мати тільки один діалог. Форумів може бути безліч.
Діалог-Пост	1:M	Пост може мати тільки один діалог. Постів може бути безліч.
Діалог-Група	1:M	Група може мати тільки один діалог. Груп може бути безліч.
Профіль-Пост	1:M	Профіль може мати безліч постів. Пост може бути тільки в одному профілі.
Технічна підтримка-Діалог	M:1	Технічна підтримка має один діалог
Зображення-Повідомлення	M:M	Багато зображень можуть бути в багатьох повідомленнях.

Зображення-Форум	1:M	Форум може мати тільки одне зображення. Форумів може бути безліч.
Зображення-Товар	M:M	Товар може мати безліч зображень.
Зображення-Подія	1:M	Багато подій можуть мати тільки одне зображення.

Продовження таблиці 2.20 – Сутності та зв'язки

Сутності	Тип зв'язку	Зміст зв'язку
1	2	3
Зображення-Конкурс	M:M	Багато конкурсів можуть мати багато зображень. Багато зображень можуть бути в багатьох конкурсах.
Зображення-Група	M:M	Групи можуть мати безліч зображень. Зображення можуть бути в багатьох групах.
Зображення-Профіль	1:M	Профіль може мати тільки одне зображення.
Зображення-Пост	M:M	Пост може мати безліч зображень. Зображення можуть бути в багатьох постах.
Зображення-Діалог	1:M	Діалог може мати одне зображення.
Товар-Діалог	M:1	Товар може мати тільки один діалог. Багато товарів можуть мати діалог.

Концептуальна модель даних ІІ «Соціальна мережа мисливців та рибалок» представлена на рисунку 2.3.

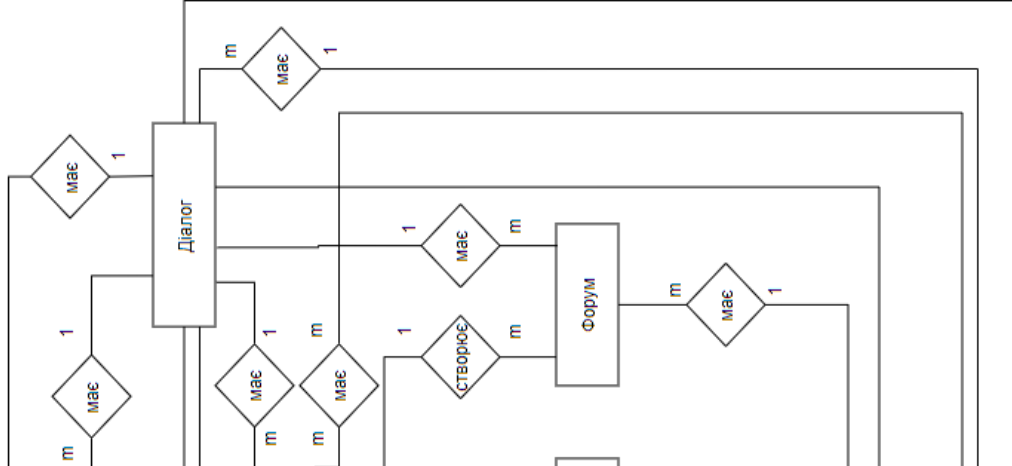


Рисунок 2.3 – Концептуальна модель даних ІП «Соціальна мережа мисливців та рибалок»

2.1.4 Розробка діаграми станів

Діаграма станів UML є графом спеціального виду, який представляє певний автомат. Вершинами графа є можливі стани автомата, зображувані відповідними графічними символами, а дуги позначають його переходи зі стану в стан. Діаграми станів можуть бути вкладені одна в одну для більш детального представлення окремих елементів моделі.

Діаграми станів застосовуються для того, щоб пояснити, яким чином працюють складні об'єкти. Діаграма станів показує, як об'єкт переходить з одного стану в інший. Очевидно, що діаграми станів служать для моделювання динамічних аспектів системи.

Діаграма переходів станів UML представлена на рисунку 2.4.

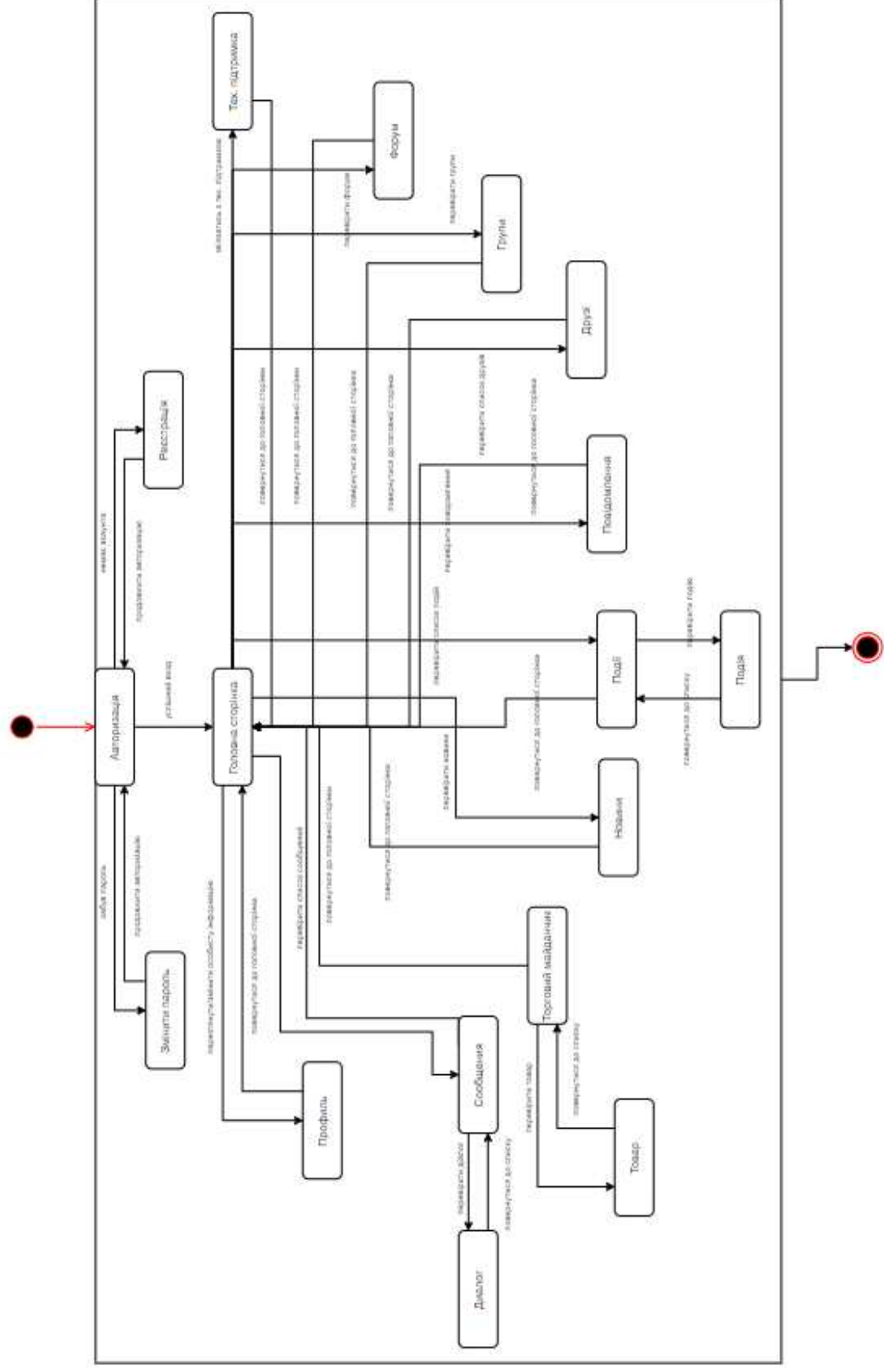


Рисунок 2.4 – Діаграма станів ІІ «Соціальна мережа мисливців та рибалок»
для користувача

2.1.5 Проектування інтерфейсу

На рисунку 2.5 представлені макети трьох форм: реєстрація, авторизація та встановлення паролю.

The image displays three wireframe forms for user authentication and account management, arranged side-by-side. Each form features a header with a logo and name, followed by input fields for registration details (name, surname, phone, email, login, password), login details (login, password), and password recovery details (login, password, confirmation code). A 'Войти' (Login) button is present at the bottom of each form.

Регистрация

Логотип с названием

Имя
Введите свое имя

Фамилия
Введите свою фамилию

Номер телефона
Введите свой номер телефона

Почта
Введите свою почту

Логин
Придумайте логин

Пароль
Придумайте пароль

Пароль
Повторите пароль

Зарегистрироваться

Войти

Авторизация

Логотип с названием

Логин
Введите свой логин

Пароль
Введите свой пароль

Войти

Регистрация Забыли пароль?

Восстановление пароля

Логотип с названием

Логин
Введите свой логин

Пароль
Придумайте новый пароль

Пароль
Повторите новый пароль

Код подтверждения
Введите код подтверждения

Сменить пароль

Войти

Рисунок 2.5 - Проектування інтерфейсу для роботи зі входом

На рисунку 2.6 представлені макети двох форм: повідомлення та діалог, які дозволяють вести повідомлення серед користувачів системи.

The image displays two wireframe forms for messaging and chat. The left form, titled 'Сообщения', shows a list of messages with user avatars, names, and timestamps. The right form, titled 'Переписка', shows a chat conversation with a back arrow, a contact name, and a message input field with a 'Отправить' (Send) button.

Сообщения

Найти...

2 Спивак Филипп
Поехали уже на рыбалку ?! 12:37

1 Бутко Оливер
Вчера успешно продал свои... 14:01

Темченко Йоллика
Вы: Привет, как дела ? 23:59

9+ Плаксий Чеслав
вчера был на рыбалке, в рай... вчера

Логина Иоланта
Вы: Я думаю, ты не права... 12.01.2020

Самсонов Юлиан
Пересланное сообщение 11.11.2019

Переписка

← Спивак Филипп

Привет!
Поехали уже на рыбалку ?!

Здоров, конечно, давай завтра выберемся?

Я свободен, на завтра полностью согласен

Сообщение Отправить

Рисунок 2.6 - Проектування інтерфейсу «Повідомлення» та «Діалог»

На рисунку 2.7 представлений макет «Профіль» в двох станах. Профіль відображає інформацію про користувача, його галерею, а також переглянути список друзів та товарів.

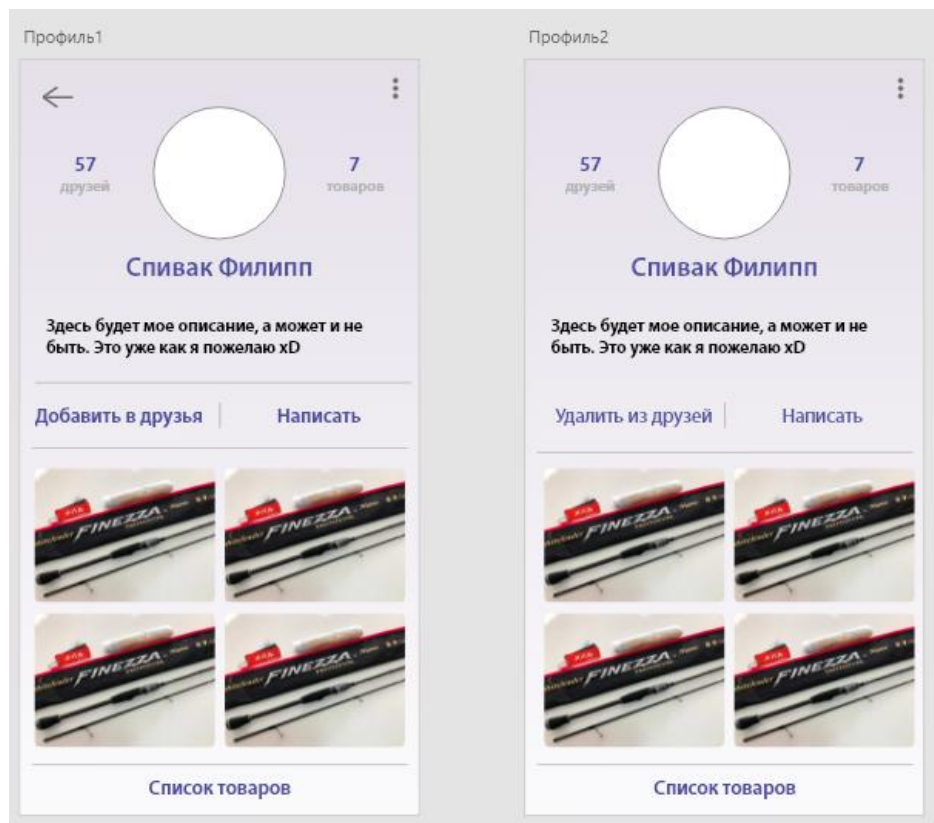


Рисунок 2.7 - Проектування інтерфейсу «Профіль»

На рисунку 2.8 представлені макети двох форм: торговий майданчик та товар. Торговий майданчик дозволяє переглядати товари, вести пошук за назвою або фільтрами. Інтерфейс «Товар» дозволяє переглянути всю інформацію про товар, забронювати або написати повідомлення основнику товару.

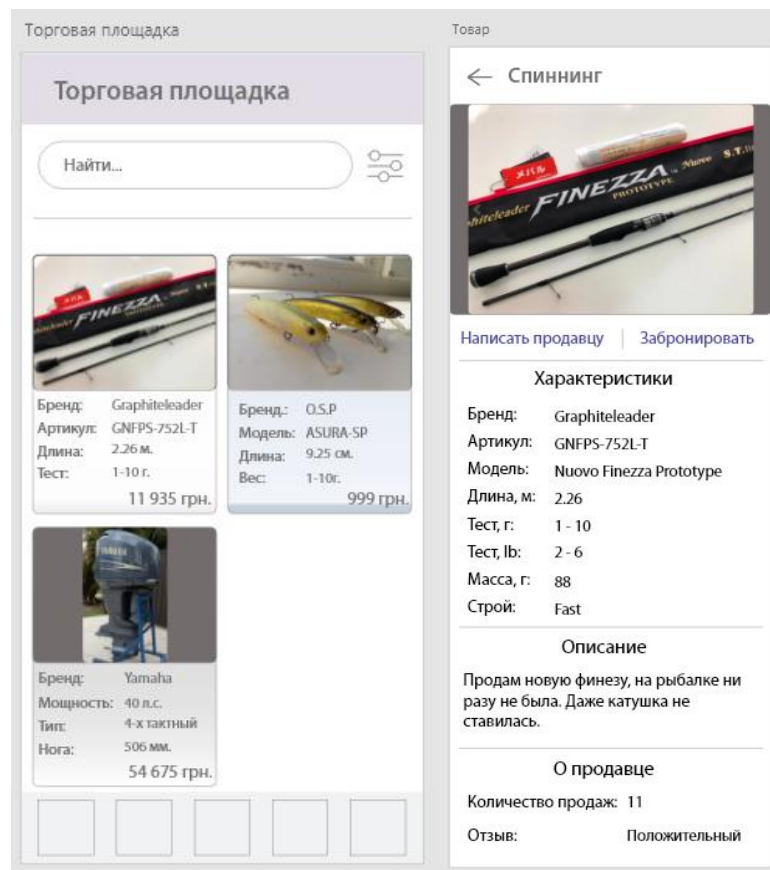


Рисунок 2.8 - Проектування інтерфейсу «Торгова площадка» та «Товар»

2.2 Технічний проєкт

2.2.1 Розробка діаграми послідовності

Діаграми послідовності UML є одним із способів формалізації сценаріїв використання. Її перевага закладається в тому, що на ранніх стадіях опису сценаріїв можливо з'ясувати склад взаємодіючих компонентів та описати потік повідомлень від одних компонентів до інших. Ці компоненти та потоки повідомлень в подальшому будуть трансформовані в конкретні класи (об'єкти), методи цих об'єктів. Відповідно, одразу ж з'ясовується і модель системи подій (Actions), які дані класи (об'єкти) будуть підтримувати та обробляти.

Діаграми послідовності UML для прецедентів представлені на рис. 2.9-2.11.

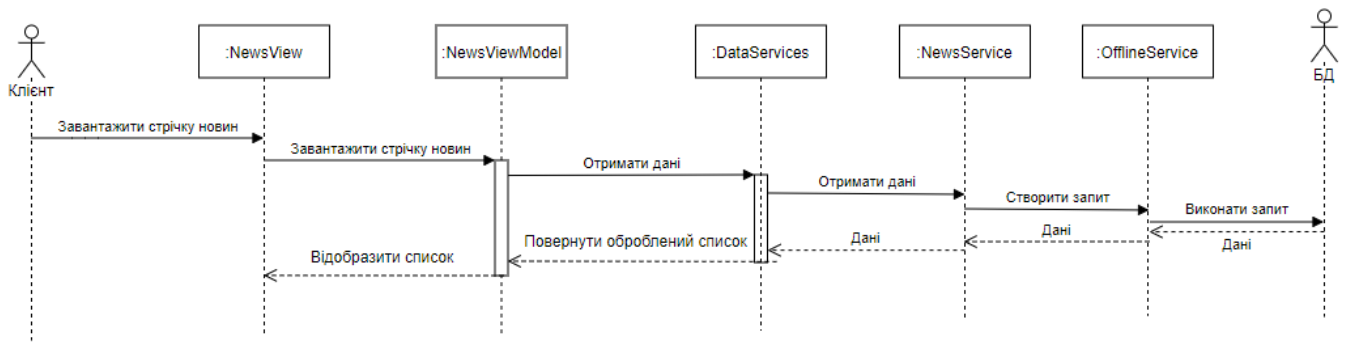


Рисунок 2.9 - Діаграма послідовності для прецедента «Завантажити стрічку новин»

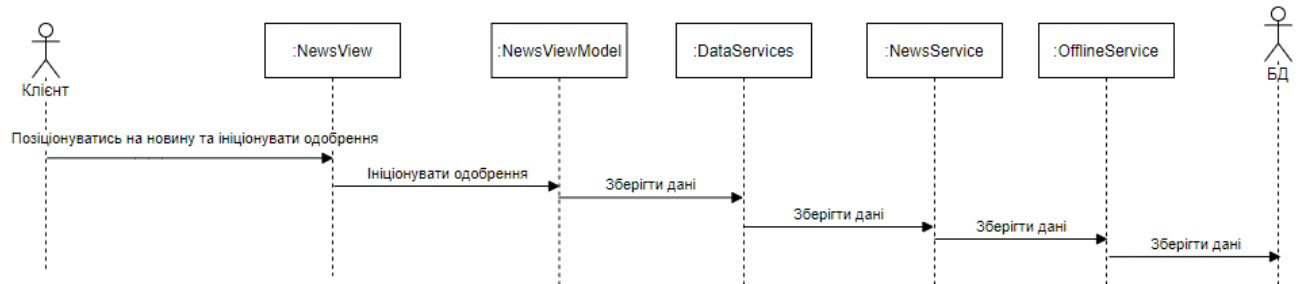


Рисунок 2.10 - Діаграма послідовності для прецедента «Оцінка новини»

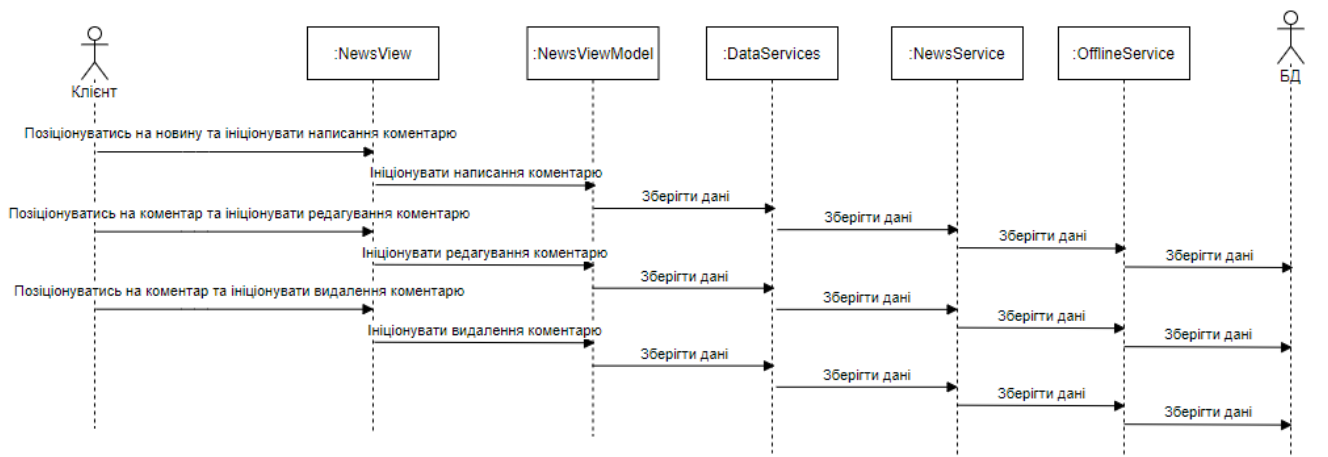


Рисунок 2.11 - Діаграма послідовності для прецедента «Вести коментарі»

2.2.2 Розробка діаграм класів

Діаграма класів UML - це тип діаграм, які частіше за все використовуються при моделюванні об'єктно-орієнтованих систем. Елементи діаграм класів пов'язані різними структурними зв'язками. Такі діаграми використовують для проєктування словника системи чи кооперацій та систем. Діаграми класів мовою UML використовують для того, щоб показати параметри блоків цих діаграм та їх зв'язків.

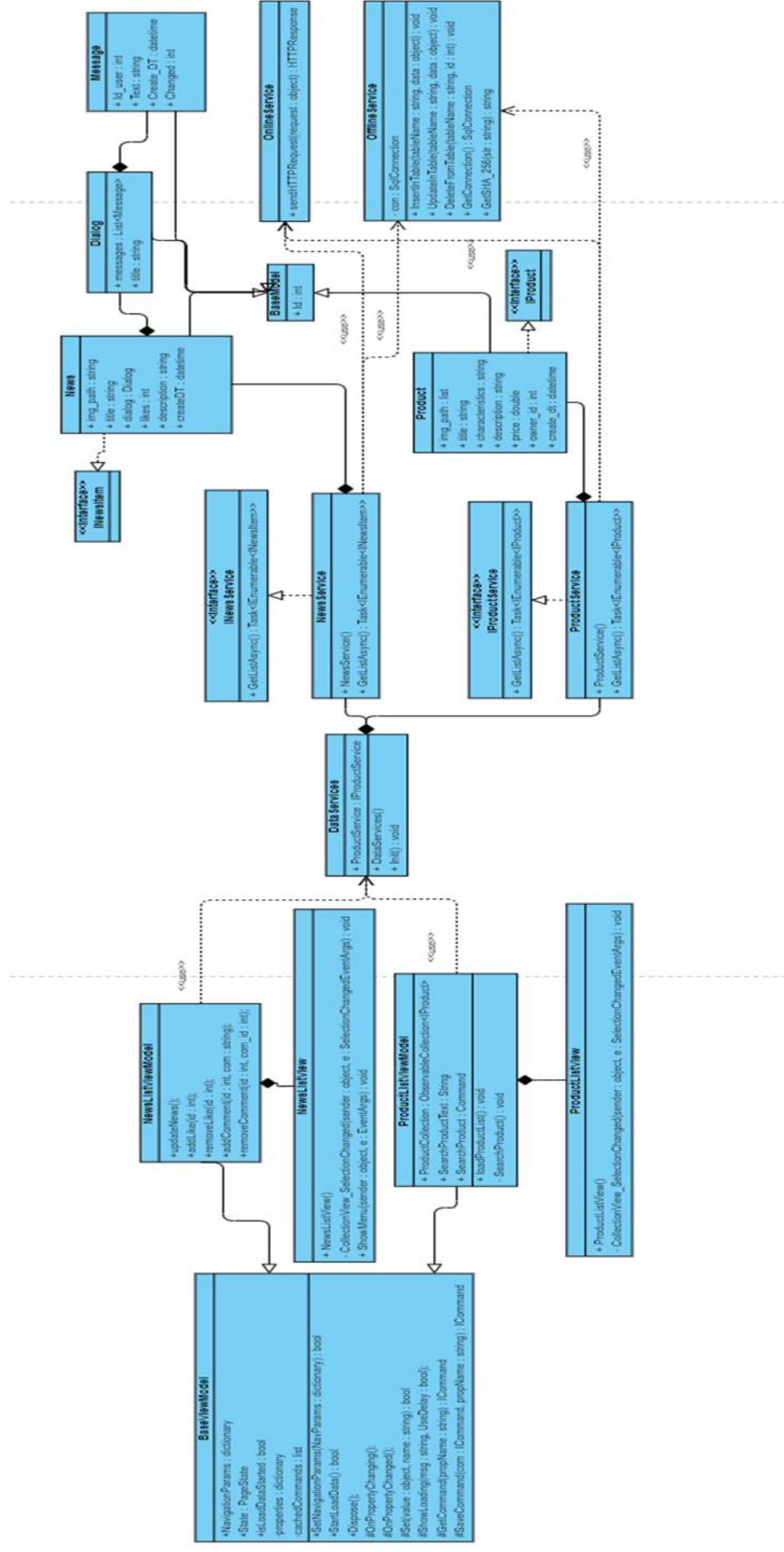


Рисунок 2.12 - Діаграма класів частини ІП «Соціальна мережа мисливців та рибалок»

2.2.3 Розробка діаграм діяльності

Діаграма діяльності UML – діаграма, на якій показані дії, стану яких описано на діаграмі станів. Під діяльністю розуміється специфікація виконуваного поведінки у вигляді координованого послідовного і паралельного виконання підлеглих елементів - вкладених видів діяльності і окремих дій англ. action, з'єднаних між собою потоками, які йдуть від виходів одного вузла до входів іншого.

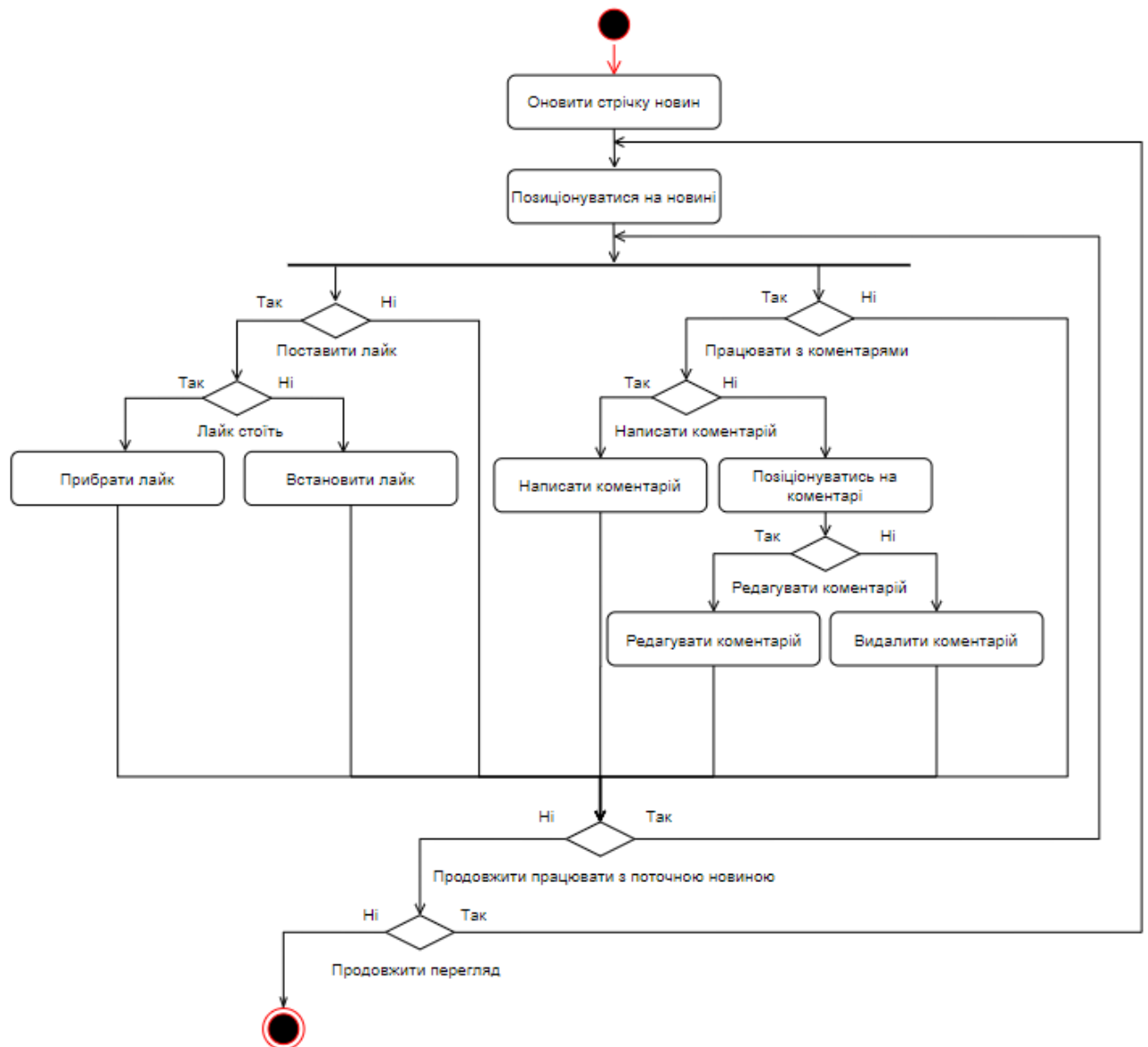


Рисунок 2.13 - Діаграма діяльності UML для роботи зі стрічкою новин

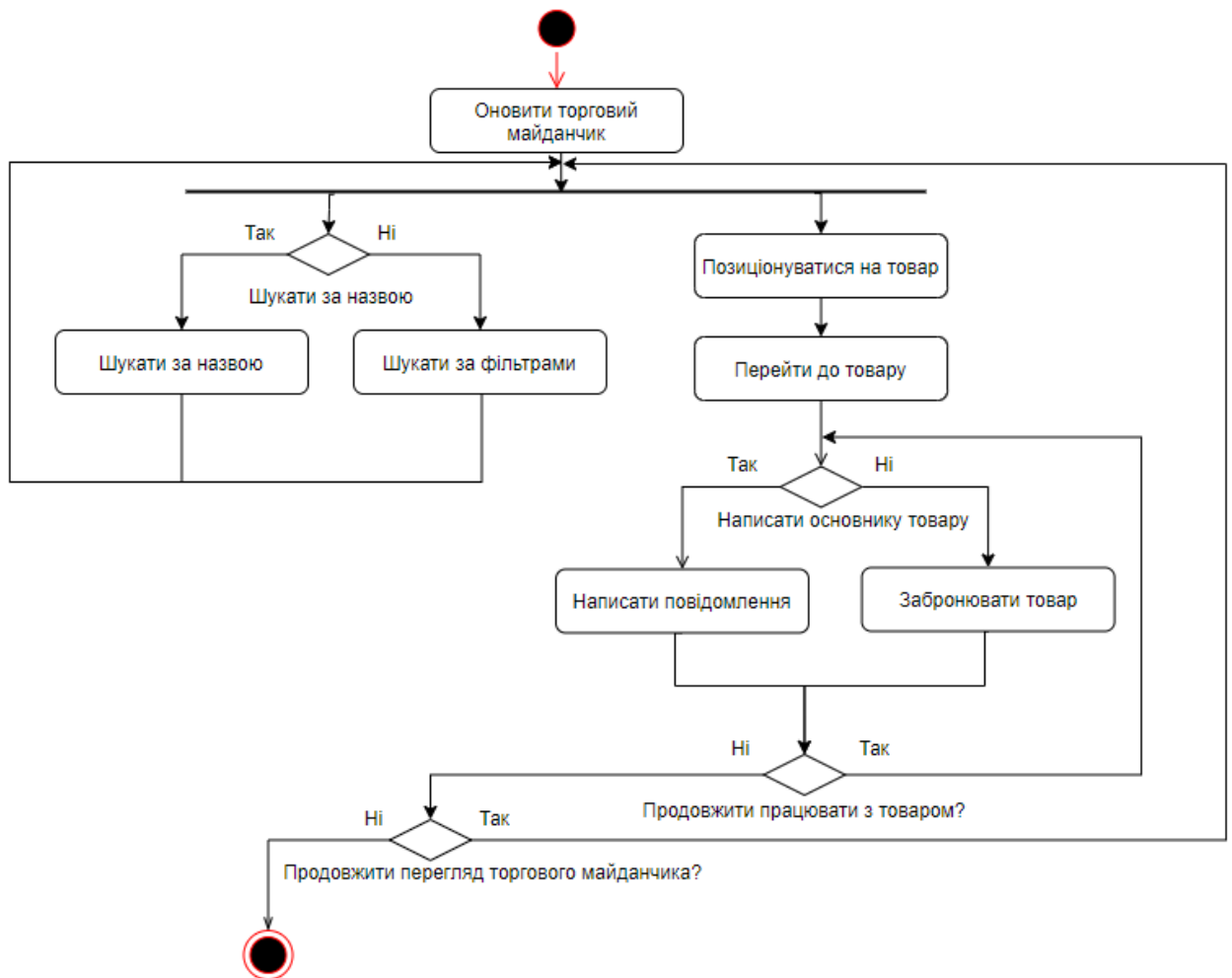


Рисунок 2.14 - Діаграма діяльності UML для роботи з товарами

2.2.4 Розробка логічної моделі

Логічна модель є основою бази даних, вона повинна відображати взаємозв'язки між реляційними таблицями. [7]

Логічна модель даних представлена на таблиці 2.15.

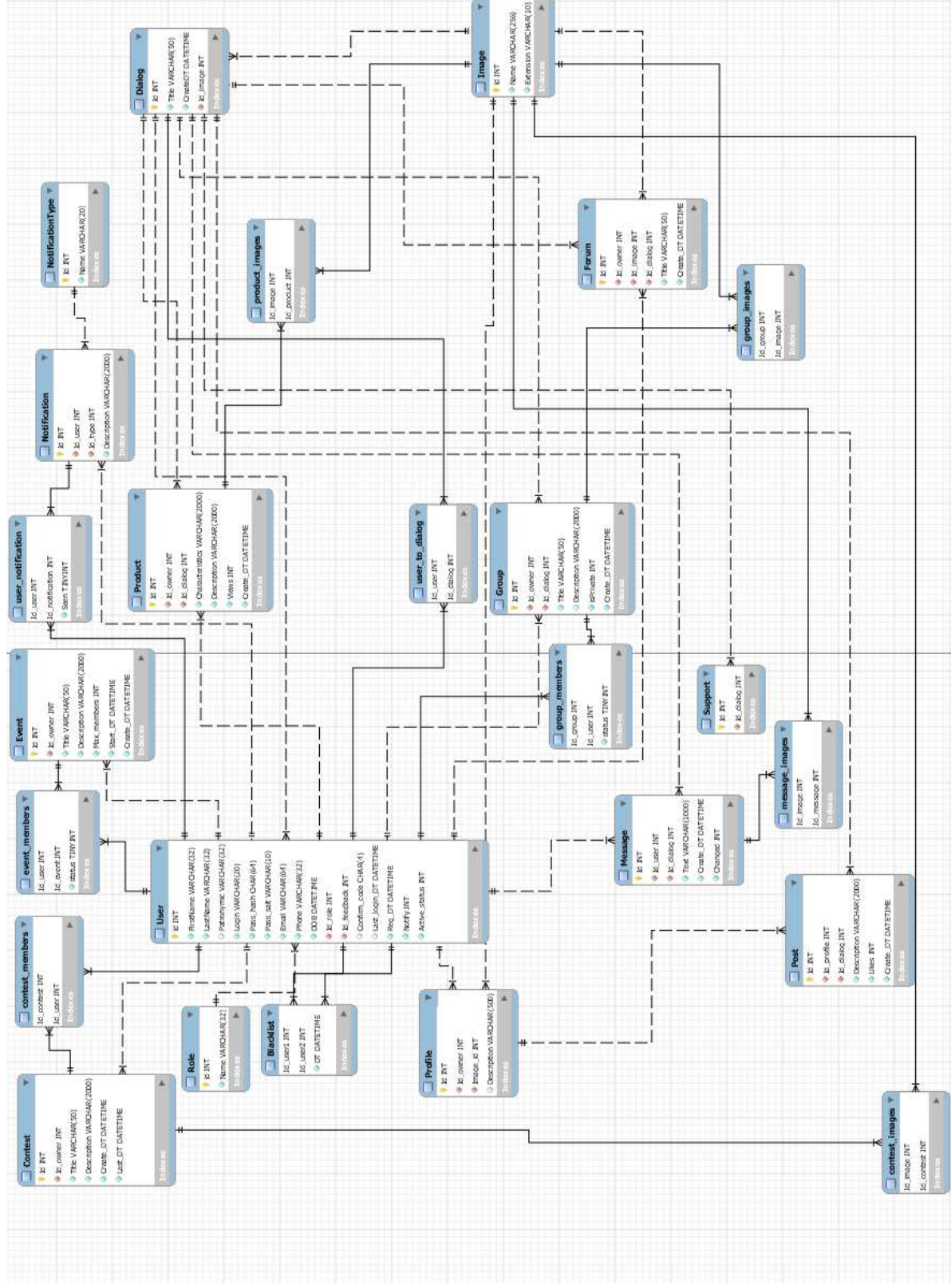


Рисунок 2.15 – Логічна модель даних

2.3 Робочий проєкт

2.3.1 Вибір засобів розробки

Для розробки були використані наступні засоби

Xamarin.Forms - це платформа для користувача інтерфейсу з відкритим вихідним кодом. За допомогою Xamarin.Forms розробники можуть створювати додатки для Android, iOS і Windows на основі загальної бази коду. [3]

Xamarin.Forms дозволяє розробникам створювати призначені для користувача інтерфейси в XAML за допомогою коду програмної частини в C#. Ці інтерфейси на кожній платформі готуються до перегляда як власні елементи управління. [4]

Платформа Xamarin.Forms орієнтована на розробників, перед яким стоять наступні завдання:

- спільне використання макета призначеного для користувача інтерфейсу і розробка на різних платформах;
- спільне використання коду, тестів і бізнес-логіки на різних платформах;
- написання кроссплатформенних додатків на мові C# в Visual Studio.

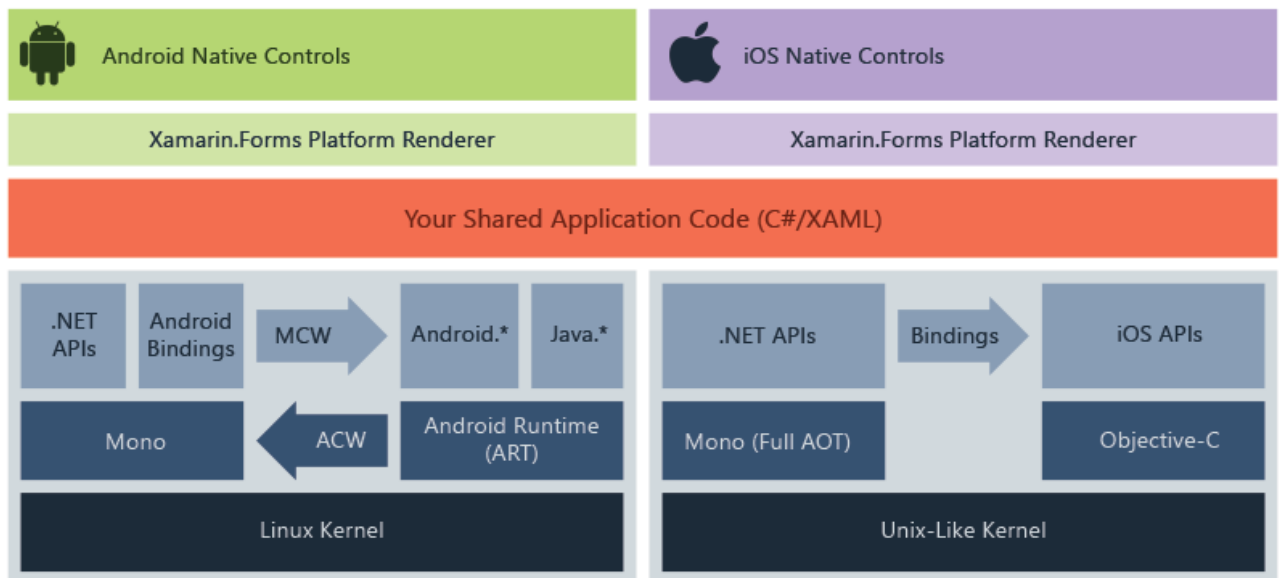


Рисунок 2.16 - Структура Xamarin Forms [3]

Xamarin.Forms надає узгоджений API для створення елементів призначеного для користувача інтерфейсу на різних платформах. Цей API може

бути реалізований в XAML або C # і підтримує прив'язку даних для шаблонів, таких як «Модель - вид – вид моделі» (MVVM). [8]

Під час виконання Xamarin.Forms використовує отрисовщик платформи для перетворення кроссплатформенних елементів призначеного для користувача інтерфейсу в власні елементи управління в Android, iOS і UWP. Дозволяє розробникам добитися звичного зовнішнього вигляду, поведінки і рівня продуктивності, а також реалізувати переваги спільного використання коду на різних платформах. [10]

Додатки Xamarin.Forms зазвичай складаються із загальної бібліотеки .NET Standard і проєктів окремих платформ. Загальна бібліотека містить уявлення XAML або C # і всю бізнес-логіку, наприклад служби, моделі або інший код. Проєкти платформи містять всю залежну від платформи логіку або пакети, необхідні додатком. [3]

MySQL це система керування базами даних з відкритим вихідним кодом (СКБД) з моделлю клієнт-сервер. [9]

СКБД- це програмне забезпечення або служба, яка використовується для створення та управління базами даних на основі реляційної моделі.

MySQL Workbench — інструмент для візуального проєктування баз даних, що інтегрує проєктування, моделювання, створення й експлуатацію БД в єдине безкоштовне оточення для системи баз даних

2.3.2 Розробка фізичної моделі

Фізична модель – описує те, як дані зберігаються в комп'ютері, представляючи інформацію про структуру записів, їх впорядкованість і про існуючі шляхи доступу до даних. [7]

Завершена фізична модель даних включатиме всі артефакти бази даних, необхідні для створення відношень між таблицями чи для досягнення мети продуктивності, як-от індексів, визначень обмежень, зв'язаних і секціонованих таблиць або кластерів.

Фізична модель БД представлена в таблицях 2.21- 2.45.

Таблиця 2.21 - Фізична модель таблиці «Image» (Зображення)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Name (Назва)	VARCHAR (256)		NOT NULL UNIQUE
Extension (Розширення файлу)	VARCHAR (8)		NOT NULL

Таблиця 2.22 - Фізична модель таблиці «User» (Користувач)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
FirstName (Назва)	VARCHAR (32)		NOT NULL
LastName (Прізвище)	VARCHAR (32)		NOT NULL
Patronymic (По батькові)	VARCHAR (32)		
Login (Логін)	VARCHAR (20)		NOT NULL
Pass_hash (Хеш паролю)	CHAR(64)		NOT NULL
Pass_salt («Соль»)	VARCHAR (10)		NOT NULL
Email (Пошта)	VARCHAR (64)		NOT NULL
Phone (Телефон)	VARCHAR (32)		NOT NULL
DOB (Дата народження)	DATETIME		NOT NULL
Id_role (Ідентифікатор ролі)	INT	FK	NOT NULL
Id_feedback (Ідентифікатор відгука)	INT	FK	NOT NULL
Confirm_code (Код підтвердження)	CHAR(4)		DEFAULT NULL
Last_login_DT (Дата останнього входу)	DATETIME		DEFAULT NULL
Reg_DT (Дата реєстрації)	DATETIME		DEFAULT NOW()
Notify (Оповіщення)	INT		DEFAULT 0
Active_status (Статус користувача)	INT		DEFAULT 1

Таблиця 2.23 - Фізична модель таблиці «Profile» (Профіль)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL
Id_image (Ідентифікатор зображення)	INT	FK	NOT NULL
Description (Опис)	VARCHAR (500)		

Таблиця 2.24 - Фізична модель таблиці «Post» (Пост)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_profile (Ідентифікатор профілю)	INT	FK	NOT NULL
Id_dialog (Ідентифікатор діалогу)	INT	FK	NOT NULL
Description (Опис)	VARCHAR (2000)		NOT NULL
Likes	INT		NOT NULL DEFAULT 0
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT

			NOW()
--	--	--	-------

Таблиця 2.25 - Фізична модель таблиці «Role» (Роль)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Name (Назва)	VARCHAR (20)		NOT NULL

Таблиця 2.26 - Фізична модель таблиці «Contest» (Конкурс)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL
Title (Назва)	VARCHAR (50)		NOT NULL
Description (Опис)	VARCHAR (2000)		NOT NULL
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()
Last_DT (Дата закінчення)	DATETIME		NOT NULL

Таблиця 2.27 - Фізична модель таблиці «Event» (Подія)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL
Title (Назва)	VARCHAR (50)		NOT NULL
Description (Опис)	VARCHAR (2000)		NOT NULL
Max_members	TINYINT		NOT NULL DEFAULT -1
Start_DT (Дата початку події)	DATETIME		NOT NULL
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.28 - Фізична модель таблиці «Product» (Товар)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL
Id_dialog (Ідентифікатор діалогу)	INT	FK	NOT NULL
Characteristics (Характеристики)	VARCHAR (2000)		NOT NULL
Description (Опис)	VARCHAR (2000)		NOT NULL
Views (Перегляди)	INT		NOT NULL DEFAULT 0
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.29 - Фізична модель таблиці «Forum» (Форум)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_image (Ідентифікатор зображення)	INT	FK	NOT NULL
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL

користувача)			
--------------	--	--	--

Продовження таблиці 2.29 - Фізична модель таблиці «Forum» (Форум)

Ідентифікатор	Тип	Ключ	Обмеження
Id_dialog (Ідентифікатор діалогу)	INT	FK	NOT NULL
Title (Назва)	VARCHAR (50)		NOT NULL
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.30 - Фізична модель таблиці «Group» (Група)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_owner (Ідентифікатор користувача)	INT	FK	NOT NULL
Id_dialog (Ідентифікатор діалогу)	INT	FK	NOT NULL
Title (Назва)	VARCHAR (50)		NOT NULL
Description (Опис)	VARCHAR (2000)		
isPrivate	INT		NOT NULL DEFAULT 0
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.31 - Фізична модель таблиці «Dialog» (Діалог)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_image (Ідентифікатор зображення)	INT	FK	
Title (Назва)	VARCHAR (50)		NOT NULL
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.32 - Фізична модель таблиці «Message» (Повідомлення)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_dialog (Ідентифікатор діалогу)	INT	FK	NOT NULL
Id_user (Ідентифікатор користувача)	INT	FK	NOT NULL
Text (Текст повідомлення)	VARCHAR (1000)		NOT NULL
Create_DT (Дата створення)	DATETIME		NOT NULL DEFAULT NOW()
Changed (Змінений)	INT		NOT NULL DEFAULT 0

Таблиця 2.33 - Фізична модель таблиці «Notification» (Оповіщення)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_user (Ідентифікатор користувача)	INT	FK	NOT NULL
Id_Type (Ідентифікатор типу)	INT	FK	NOT NULL
Description (Опис)	VARCHAR (2000)		NOT NULL

Таблиця 2.34 - Фізична модель таблиці «NotificationType» (Тип оповіщення)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Name (Назва)	VARCHAR (20)		NOT NULL UNIQUE

Таблиця 2.35 - Фізична модель таблиці «Support» (Технічна підтримка)

Ідентифікатор	Тип	Ключ	Обмеження
Id	INT	PK	NOT NULL UNIQUE AUTO_INCREMENT
Id_dialog	INT	FK	NOT NULL

Таблиця 2.36 - Фізична модель таблиці «contest_members» (Учасники конкурсів)

Ідентифікатор	Тип	Ключ	Обмеження
Id_contest	INT	FK	NOT NULL
Id_user	INT	FK	NOT NULL

Таблиця 2.37 - Фізична модель таблиці «event_members» (Учасники подій)

Ідентифікатор	Тип	Ключ	Обмеження
Id_user	INT	FK	NOT NULL
Id_event	INT	FK	NOT NULL
status	TINYINT		NOT NULL DEFAULT 0

Таблиця 2.38 - Фізична модель таблиці «user_notification» (Повідомлення для користувачів)

Ідентифікатор	Тип	Ключ	Обмеження
Id_user	INT	FK	NOT NULL
Id_notification	INT	FK	NOT NULL
Seen	TINYINT		NOT NULL DEFAULT 0

Таблиця 2.39 - Фізична модель таблиці «Blacklist» (Чорний список)

Ідентифікатор	Тип	Ключ	Обмеження
Id_user1	INT	FK	NOT NULL
Id_user2	INT	FK	NOT NULL
DT	DATETIME		NOT NULL DEFAULT NOW()

Таблиця 2.40 - Фізична модель таблиці «contest_images» (Зображення конкурсу)

Ідентифікатор	Тип	Ключ	Обмеження
Id_image	INT	FK	NOT NULL
Id_contest	INT	FK	NOT NULL

Таблиця 2.41 - Фізична модель таблиці «message_images» (Зображення повідомлення)

Ідентифікатор	Тип	Ключ	Обмеження
Id_image	INT	FK	NOT NULL
Id_message	INT	FK	NOT NULL

Таблиця 2.42 - Фізична модель таблиці «group_members» (Учасники груп)

Ідентифікатор	Тип	Ключ	Обмеження
Id_group	INT	FK	NOT NULL
Id_user	INT	FK	NOT NULL
status	TINYINT		NOT NULL DEFAULT 0

Таблиця 2.43 - Фізична модель таблиці «user_to_dialog» (Листування користувачів)

Ідентифікатор	Тип	Ключ	Обмеження
Id_user	INT	FK	NOT NULL
Id_dialog	INT	FK	NOT NULL

Таблиця 2.44 - Фізична модель таблиці «group_images» (Зображення груп)

Ідентифікатор	Тип	Ключ	Обмеження
Id_group	INT	FK	NOT NULL
Id_image	INT	FK	NOT NULL

Таблиця 2.45 - Фізична модель таблиці «product_images» (Зображення товарів)

Ідентифікатор	Тип	Ключ	Обмеження
Id_product	INT	FK	NOT NULL
Id_image	INT	FK	NOT NULL

3 РЕЗУЛЬТАТ РОЗРОБКИ ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

В результаті роботи над дипломним проектом був розроблений кросплатформений мобільний додаток «Соціальна мережа мисливців та рибалок». Він включає наступні модулі:

Модуль входу до системи

При завантаженні ПО система відображає користувачу інтерфейс «Авторизація» (рис. 3.1). Для входу до системи користувач повинен ввести логін та пароль та натиснути «Увійти». Знизу екрану користувачу доступні посилання до інтерфейсів «Реєстрація» та «Зміна паролю».

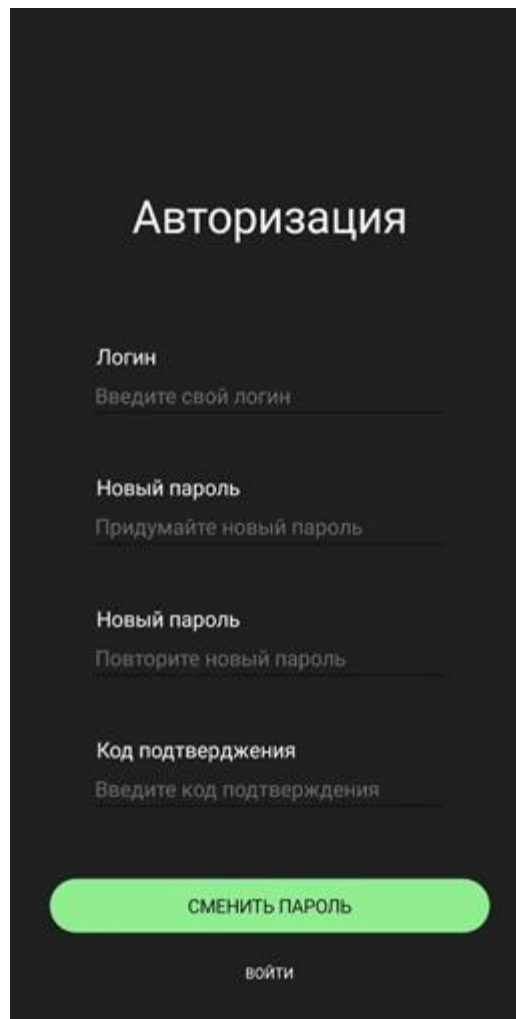


Рисунок 3.1 - Интерфейс «Авторизация»

Для реєстрації (інтерфейс представлений на рисунку 3.2) система відображає перелік полів вводу даних та кнопку для підтвердження реєстрації. Знизу екрану користувачу доступне посилання до інтерфейсу «Авторизація».



Регистрация

Имя
Введите свое имя

Фамилия
Введите свою фамилию

Номер телефона
Введите свой номер телефона

Почта
Введите свою почту

Логин
Придумайте логин

Пароль
Придумайте пароль

Пароль
Повторите пароль

ЗАРЕГИСТРИРОВАТЬСЯ

ВОЙТИ

Рисунок 3.2 - Интерфейс «Реестрація»

Для зміни паролю (інтерфейс представлений на рисунку 3.3) користувач повинен ввести логін та придумати новий пароль, якщо такий логін в системі існує, система відправить код підтвердження на пошту, прив'язану до аккаунту. Знизу екрану користувачу доступне посилання до інтерфейсу «Авторизація».

The image shows a dark-themed user interface for changing a password. At the top, the title 'Смена пароля' (Change password) is displayed in white. Below it, there are four input fields, each with a label and a placeholder text: 'Логин' (Login) with 'Введите свой логин' (Enter your login), 'Новый пароль' (New password) with 'Придумайте новый пароль' (Create a new password), another 'Новый пароль' (New password) with 'Повторите новый пароль' (Repeat the new password), and 'Код подтверждения' (Confirmation code) with 'Введите код подтверждения' (Enter the confirmation code). At the bottom, there is a large green button labeled 'СМЕНИТЬ ПАРОЛЬ' (Change password) and a smaller white button labeled 'ВОЙТИ' (Login).

Рисунок 3.3 - Интерфейс «Зміна паролю»

Модуль роботи з новинами

Інтерфейс «Новини» (рис. 3.4) дозволяє користувачу переглядати оновлення користувачів, на яких він підписан, одобрювати та писати коментарі під новинами.

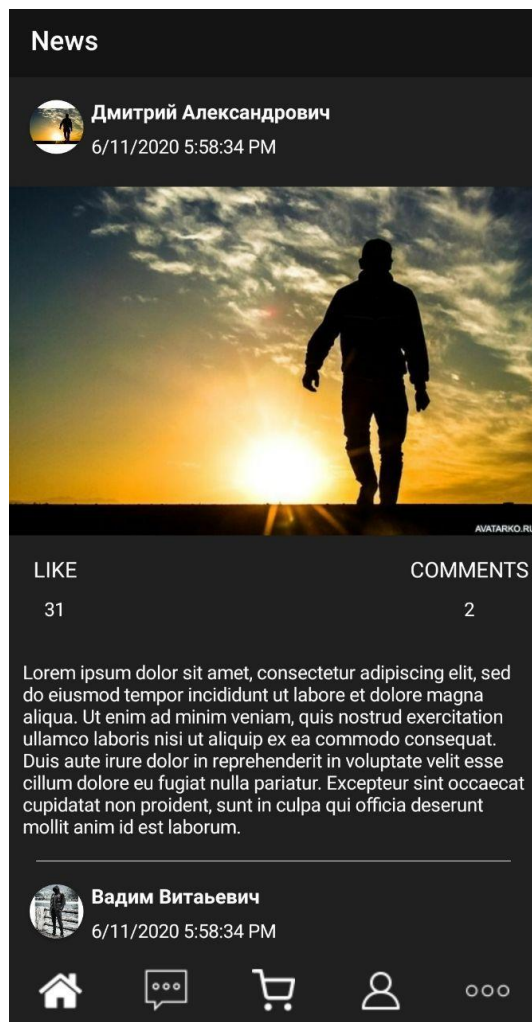


Рисунок 3.4 - Интерфейс «Новини»

Модуль роботи з торговим майданчиком та товаром

При навігації до інтерфейсу «Торговий майданчик» (рис. 3.5) система відобразить список товарів в системі з короткими характеристиками, поле вводу для пошуку товару за назвою та посилання до пошуку за фільтрами. При натисканні на товар, система відобразить інтерфейс «Товар».

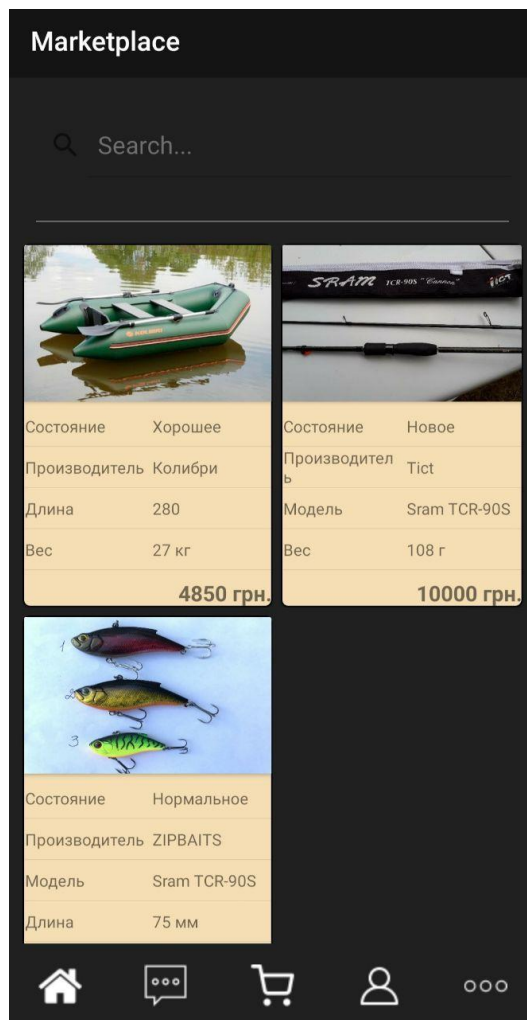


Рисунок 3.5 - Интерфейс «Торговый майданчик»

Інтерфейс «Товар» (рис. 3.6) включає список зображень товару, ціну, назву, характеристики, опис товару та інформацію про продавця з відгуками. Посилання «Написати продавцю» дозволить написати повідомлення основнику товару, а посилання «Забронювати» дозволить покупцеві забронювати товар до з'ясування.

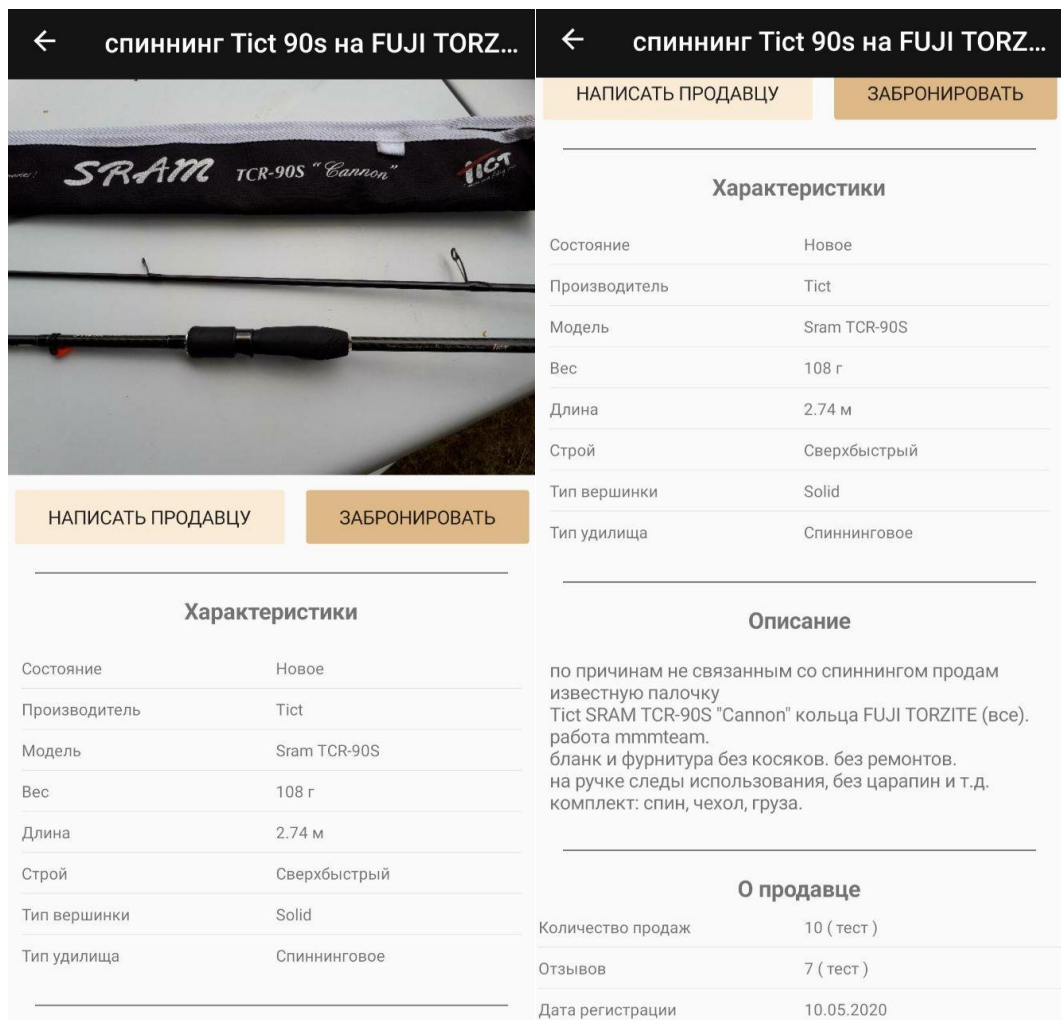


Рисунок 3.6 - Интерфейс «Товар»

Модуль роботи ведення друзів

При навігації до інтерфейсу «Друзі» (рис. 3.7) користувачу буде доступний пошук друзів з можливістю додавання/видалення та поточні друзі користувача з можливістю видалення.

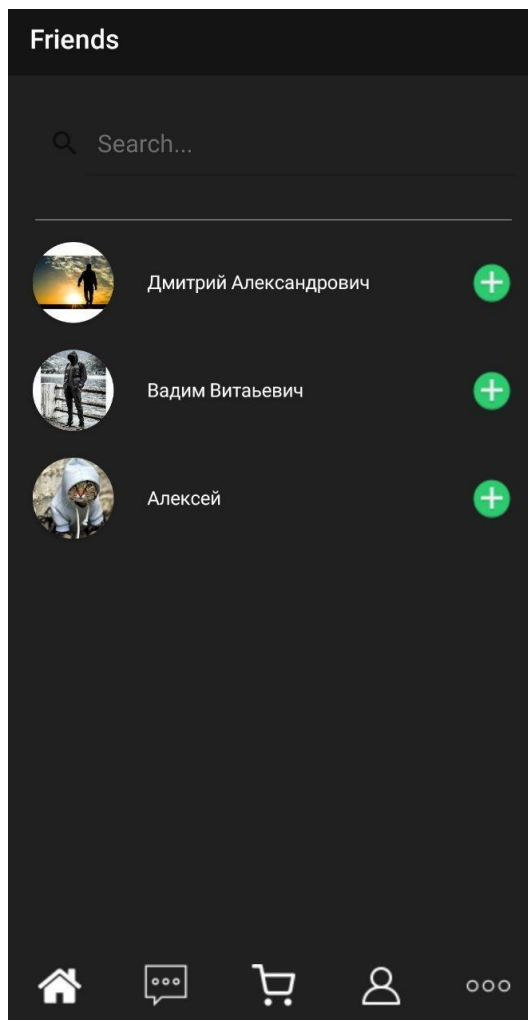


Рисунок 3.7 - Інтерфейс «Друзі»

Модуль роботи з групами

Інтерфейс «Групи» складається з двох вкладок. В першій вкладці система відобразить всі групи користувача, в другій – можливість пошуку груп за назвою та результат пошуку списком груп. При натисканні на групу, система відобразить інтерфейс групи з детальною інформацією, постами та меню для взаємодії з поточною групою.

Модуль роботи з форумом

При навігації до інтерфейсу «Форум» система відобразить користувачу доступні теми форуму з можливістю додавання їх до обраних та посилання на створення своєї теми для обговорення.

Модуль роботи з оповіщеннями

Інтерфейс «Оповіщення» дозволить користувачам бачити оповіщення від системи або інших користувачів за категоріями.

Модуль роботи з повідомленнями

При навігації до інтерфейсу «Повідомлення» (рис. 3.8) користувач побачить список діалогів між користувачами. При переході до діалогу користувач побачить список повідомлень з користувачем або користувачами та поле вводу нового повідомлення з кнопкою відправлення.

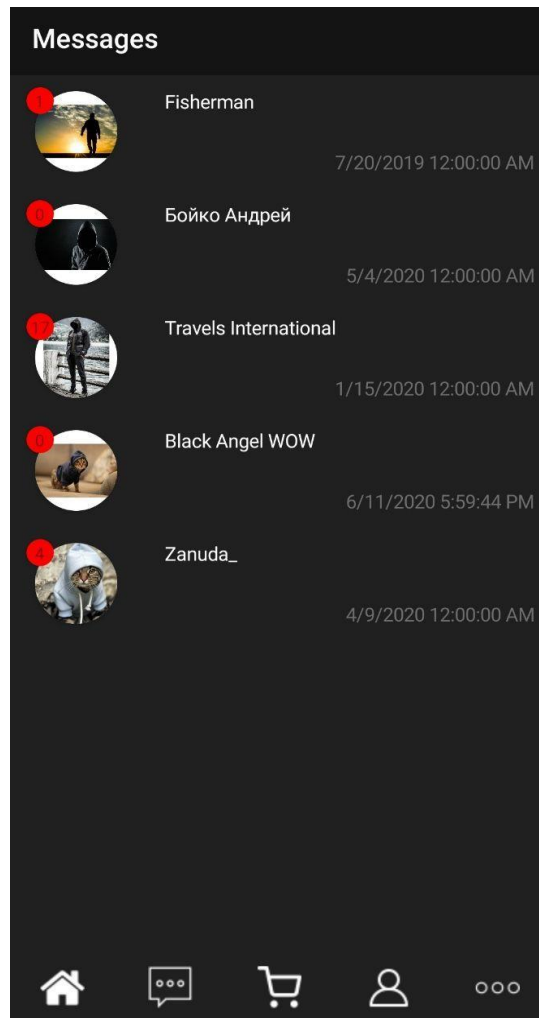


Рисунок 3.8 - Інтерфейс діалоги користувача

Модуль ведення профілю

Інтерфейс «Профіль» відображає всю інформацію користувача з фото- або відео- галереєю. Користувач може редагувати інформацію в своєму профілі, додавати фото або відео. В профілі користувача можна побачити кількість друзів та товарів користувача, при переході система відобразить список цих товарів. Посилання «Додати до друзів» дозволить додати користувача до друзів та «Написати» дозволить написати повідомлення користувачу.

Модуль роботи з конкурсами

При навігації до інтерфейсу «Конкурси» користувач побачить список актуальних конкурсів в системі, в кожному з яких він зможе прийняти участь,

виконавши умови конкурсу. При переході до конкурсу буде доступні умови конкурсу, додатковий опис, дата закінчення конкурсу та кнопка для прийняття участі у конкурсі.

Модуль роботи з подіями

При навігації до інтерфейсу «Події» користувач побачить список подій за певними категоріями. При переході до події користувач побачить детальну інформацію про подію, місце, дату та інше, а також кнопку для прийняття участі (за умови, якщо максимальна кількість учасників Події ще не набрана).

4 ОХОРОНА ПРАЦІ

4.1 Вступ

Охорона праці - це система правових, соціально-економічних, організаційно-технічних, санітарно-гігієнічних і лікувально-профілактичних заходів та засобів, спрямованих на збереження життя, здоров'я і працездатності людини у процесі трудової діяльності.

Даний розділ дипломного проєкту присвячений розгляду питань охорони праці для програміста.

4.1 Організація робочого місця

Робочим місцем протягом усієї робочої зміни для програміста є місце обладнане робочим столом, стільцем та персональним комп'ютером, що складається з монітора, системного блоку, клавіатури та миші. На робочому місці негативно можуть впливати наступні фізичні та психофізіологічні фактори: [11]

- підвищена або знижена температура повітря робочої зони;
- підвищена або знижена вологість повітря;
- недостатня освітленість робочого місця;
- підвищений рівень шуму на робочому місці;
- підвищена іонізація повітря;
- підвищений рівень електромагнітних випромінювань;
- нервово-психічні перевантаження (розумова перенапруга, перенапруга аналізаторів);
- фізичні перевантаження (одноманітна поза викликає статичну втому).

Робота з ПК супроводжується також підвищеним ступенем напруженості трудового процесу. При систематичному впливі факторів, які не відповідають нормативним показникам, зростає рівень професійно зумовленої захворюваності працюючих та можуть виникнути професійні захворювання органів зору, руху, нервової системи.

4.2 Мікроклімат робочої зони

Робота програміста за енерговитратами відноситься до категорії легких робіт Іа, Іб, тому повинні дотримуватися наступні вимоги згідно ДСН 3.3.6.042-99:

- оптимальна температура повітря – 22°C (допустима – 20-24°C);
- оптимальна відносна вологість – 40-60% (допустима – не більш 75%);
- швидкість руху повітря не більш 0,1 м/с.

4.3 Освітлення робочого місця

Нормованим параметром природного освітлення згідно ДБН В.2.5–28 – 2006 є коефіцієнт природного освітлення (КПО). КПО встановлюється в залежності від розряду виконуваних зорових робіт. Робота програміста відноситься до робіт середньої точності (IV розряд зорових робіт, мінімальний розмір об'єкту розрізнення складає 0,5-1,0мм), для яких при використанні бокового освітлення КПО=1,5%. Для штучного освітлення нормованим параметром виступає Емін – мінімальний рівень освітленості, та Кп – коефіцієнт пульсації світлового потоку, який не повинний бути більшим ніж 20%. Мінімальна освітленість встановлюється в залежності від розряду виконуваних зорових робіт. Для IV розряду зорових робіт вона складає 300-500 лк. [11]

4.4 Випромінювання

У приміщенні відсутні інфрачервоні, ультрафіолетові та електромагнітні випромінювання, бо усі монітори ПК вироблені на основі рідкокристалічної матриці, підсвітка якої здійснюється неоновією лампою, що не має сильного електромагнітного випромінювання і сертифіковані в Україні.

4.5 Організація оптимального режиму праці та відпочинку

При організації праці, пов'язаної з використанням ПК, для збереження здоров'я працівника, запобігання професійних захворюванням і підтримки працездатності передбачаються перерви для відпочинку. Нетривалі перерви повинні передувати появі об'єктивних і суб'єктивних ознак стомлення й зниження працездатності. При виконанні робіт, що належать до різних видів

трудової діяльності, за основну роботу з ПК слід вважати таку, що займає не менше 50% робочого часу.

Впродовж робочої зміни мають передбачатися:

- перерви для відпочинку і вживання їжі (обідні перерви);
- перерви для відпочинку й особистих потреб;

У всіх випадках, коли виробничі обставини не дозволяють застосувати регламентовані перерви, тривалість безперервної роботи з ПК не повинна перевищувати 4 години.

При 12-годинній робочій зміні регламентовані перерви повинні встановлюватися в перші 8 годин робота аналогічно перервам при 8-годинній робочій зміні, а протягом останніх 4-х годин роботи, незалежно від характеру трудової діяльності, через кожну годину тривалістю 15 хвилин.

Для зниження нервово-емоційного напруження, стомлення зорового аналізатора, поліпшення мозкового кровообігу, подолання несприятливих наслідків гіподинамії, запобігання втомі доцільно деякі перерви використовувати для виконання комплексу вправ.

4.6 Електробезпека

Приміщення лабораторії за небезпекою ураження електричним струмом можна віднести до 1 класу, тобто це приміщення без підвищеної небезпеки (сухе, без пилу, з нормальною температурою повітря, ізольованими підлогами і малим числом заземлених приладів).

На робочому місці програміста з всього устаткування металевим є лише корпус системного блоку комп'ютера, але тут використовуються системні блоки, що відповідають стандартам фірми IBM, у яких крім робочої ізоляції передбачений елемент для заземлення і провід з жилою, що заземлює, для приєднання до джерела живлення.

Основні причини ураження людини електричним струмом на робочому місці:

- дотик до металевих неструмоведучих частин (корпусу, периферії комп'ютера), що можуть виявитися під напругою в результаті ушкодження ізоляції;

- нерегламентоване використання електричних приладів;
- відсутність інструктажу співробітників з правил електробезпеки.

На протязі роботи на корпусі комп'ютера накопичується статична електрика. На відстані 5-10 см від екрана напруженість електростатичного поля складає 60-280 кВ/м, тобто в 10 разів перевищує норму 20 кВ/м.

4.7 Важкість та напруженість праці

Оцінка напруженості праці здійснювалась на підставі обліку всіх наявних значущих показників, які можуть перевищувати нормативні рівні.

Розподіл функцій за ступенем складності завдання – належить до класу 2 (обробка, виконання завдання та його перевірка). Характер виконуваної роботи – належить до класу 2 (робота за встановленим графіком з можливим його коригуванням у ході діяльності). Навантаження на зоровий аналізатор (при відстані від очей працюючого до об'єкта розрізнення не більше 0,5 м), при тривалості зосередженого спостереження (% часу зміни) – належить до класу 2 (5,0-1,1 мм більше 50 % часу; 1,0-0,3 мм до 50 % часу; менше 0,3 мм до 25 %). Спостереження за екранами відеотерміналів (годин на зміну) – належить до класу 3.2 (більше 4 годин).

Монотонність праці. Кількість елементів (приймів, необхідних для реалізації простого завдання або в операціях, які повторюються багаторазово) – належить до класу 3.1 (5-3 прийоми). Режим праці (фактична тривалість робочого дня (год.) – належить до класу 1 (6-7 годин). Наявність регламентованих перерв та їх тривалість – належить до класу 2 (перерви регламентовані, недостатньої тривалості: від 3 % до 7 % часу зміни). Отже робоче місце за показниками напруженості трудового процесу відноситься до класу 3.1 - Шкідливий (напружена праця).

Важкість праці. Оцінка важкості праці здійснюється на підставі обліку всіх наявних значущих показників. При цьому спочатку встановлюється клас кожного із вимірюваних показників, а кінцева оцінка важкості праці встановлюється за показником, який має найвищий ступінь важкості.

Стереотипні робочі рухи (кількість за зміну): при локальному навантаженні (за участю м'язів кистей та пальців рук) – належить до класу 1 (до

20000). При загальному навантаженні (при роботі з переважною участю м'язів рук та плечового поясу) – належить до класу 1 (до 10000). Робоча поза – належить до класу 2 (періодичне перебування в незручній позі (робота з поворотом) тулуба, незручним розташуванням кінцівок та/або фіксованій позі (неможливість зміни взаєморозташування різних частин тіла відносно одна одної) до 25 % часу зміни.) Нахили корпусу (вимушені, більше 30), кількість за зміну – належить до класу 1 (до 50). Переміщення у просторі (переходи, обумовлені технологічним процесом протягом зміни), км: по горизонталі – належить до класу 1 (до 4). По вертикалі - належить до класу 1 (до 2). Отже робоче місце за показниками важкості трудового процесу відноситься до класу 2 – Допустимий, середнє фізичне навантаження.

4.8 Висновок

При написанні даного розділу було розглянуто питання охорони праці та проаналізовано небезпечні та шкідливі чинники, які супроводжують роботу програміста.

Найбільшому ризику виникнення різноманітних порушень піддаються: органи зору, м'язово-скелетна система, нервово-психічна діяльність. Робота з комп'ютером характеризується значною розумовою напругою і нервово-емоційним навантаженням, високою напруженістю зорової праці та досить великим навантаженням на м'язи рук під час роботи з клавіатурою. У процесі роботи з комп'ютером необхідно дотримуватися правильного режиму праці та відпочинку. Інакше у людини відзначаються значна емоційна напруга зорового апарату, що може призвести до появи головного білю, дратівливості, порушення сну, почуття виснаження. Раціональний режим праці та відпочинку передбачає запровадження регламентованих перерв, рівномірний розподіл навантаження протягом робочого дня, регулярні комплекси вправ для очей, рук, хребта для поліпшення мозкового кругообігу та психофізіологічного розвантаження.

Таким чином, вивчення умов праці на робочому місці є необхідною умовою запобігання негативних наслідків впливу небезпечних та шкідливих факторів.

ВИСНОВКИ

Тема дипломного проєкту полягала в полегшенні роботи мисливців та рибалок в мережі інтернет: обміном досвіду, веденням блогу та зручному продажу або купівлі товарів.

В ході виконання дипломного проєкту було досліджено та зроблено аналіз готових рішень, зроблена та проаналізована постановка задачі. Розроблена пояснювальна записка, яка включає проєктну частину охорони праці та техніко-економічне обґрунтування.

Результатом роботи над дипломним проєктом був розроблений кросплатформений мобільний додаток «Соціальна мережа мисливців та рибалок», який має функціонал, що ставився у постановці задачі.

Самостійно опрацьована тема для створення кросплатформних мобільних додатків на Xamarin Forms.

В результаті роботи значно підвищив рівень навичок у мові програмування C#, мові розмітки XAML та створенні баз даних.

Для експлуатації програмного забезпечення було розроблено керівництво користувача, яке представлено у додатку Б – «Керівництво користувача».

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Мобільний додаток для сайту інтернет-магазину. URL: <https://lemarbet.com/ua/razvitie-internet-magazina/mobilnoe-prilozhenie/> (дата звернення 25.04.2022)
2. Чим Adobe XD так приваблює дизайнерів. URL: <https://web4u.in.ua/blog/chim-adobe-xd-tak-privablyu-dizayner-v-33> (дата звернення 01.05.2022)
3. Xamarin.Forms documentation. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/> (дата звернення 11.05.2022).
4. Creating Mobile Apps with Xamarin.Forms book. URL: <https://docs.microsoft.com/en-us/xamarin/xamarin-forms/creating-mobile-apps-xamarin-forms/> (дата звернення 16.05.2022).
5. Hermes, Dan. Xamarin Mobile Application Development. Cross-Platform C# and Xamarin.Forms Fundamentals. 2015 issue. 389 pages.
6. Роберт и Мика Мартины. Принципы, паттерны и методики гибкой разработки на языке C#. 2017. С. 768.
7. Руководство по MySQL. URL: <https://metanit.com/sql/mysql/> (дата звернення 01.06.2022)
8. Язык программирования C# и .NET. URL: <https://metanit.com/sharp/general.php> (дата звернення 24.05.2022)
9. Руководство по ADO.NET и работе с базами данных. URL: <https://metanit.com/sharp/adonet/> (дата звернення 29.05.2022)
10. Навигация. Основы навигации. URL: <https://metanit.com/sharp/xamarin/5.1.php> (дата звернення: 26.05.2022)
11. Охорона праці та безпека в надзвичайних ситуаціях. URL: <https://studfile.net/preview/5150012/page:12/> (дата звернення 03.05.2022).
12. Техничко-економическое обоснование создания автоматизированных систем и программных продуктов: Методические указания / Самарский государственный аэрокосмический университет; Сост. В.П. Куренкова. Самара, 1997.

ДОДАТОК А – Посібник користувача

Робота починається з установки програмного забезпечення на Android або IOS. При завантаженні ПО система відобразить користувачу інтерфейс входу (рис. А.1) в систему та можливість переходу до інтерфейсу реєстрації або зміни паролю.

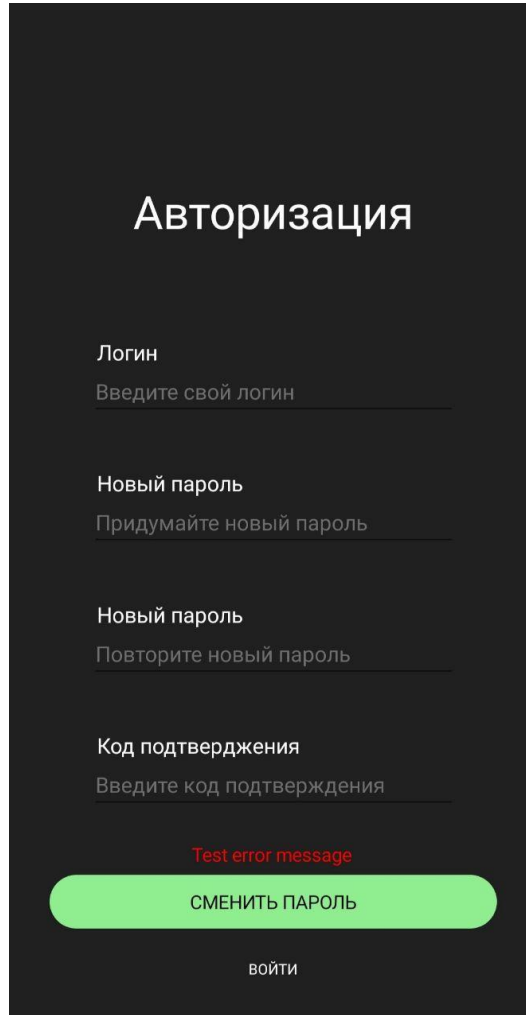


Рисунок А.1 - Интерфейс «Авторизация»

При переході до інтерфейсу реєстрації (рис. А.2) система запросить ввести необхідні дані користувача для внесення до системи.



Регистрация

Имя
Введите свое имя

Фамилия
Введите свою фамилию

Номер телефона
Введите свой номер телефона

Почта
Введите свою почту

Логин
Придумайте логин

Пароль
Придумайте пароль

Пароль
Повторите пароль

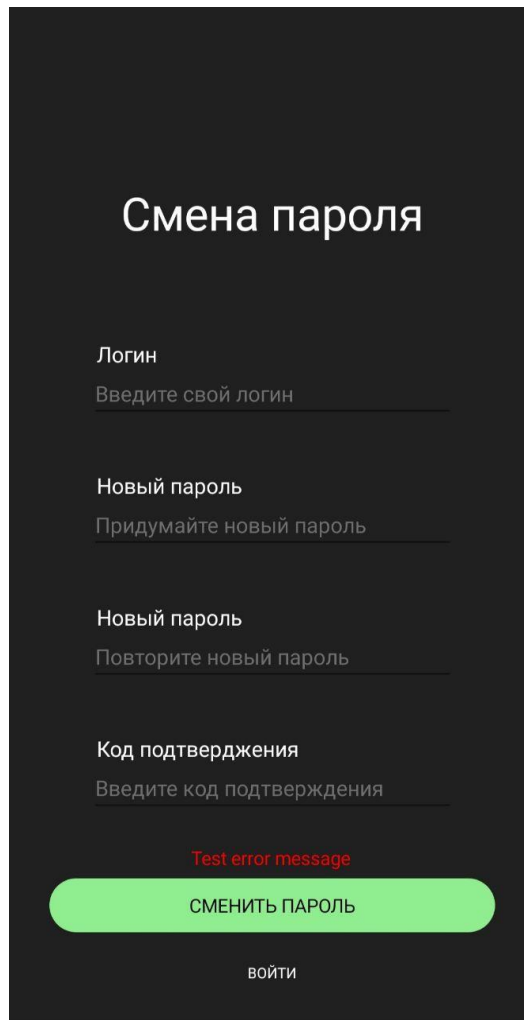
Test error message

ЗАРЕГИСТРИРОВАТЬСЯ

ВОЙТИ

Рисунок А.2 - Интерфейс «Регистрация»

Для зміни паролю (рис. А.3) система запросить логін та придумати новий пароль, якщо такий логін в системі існує, система відправить код підтвердження на пошту, прив'язану до аккаунту.



Смена пароля

Логин
Введите свой логин

Новый пароль
Придумайте новый пароль

Новый пароль
Повторите новый пароль

Код подтверждения
Введите код подтверждения

Test error message

СМЕНИТЬ ПАРОЛЬ

ВОЙТИ

Рисунок А.3 - Интерфейс «Зміна паролю»

Після того, як користувач ввів дані та авторизувався, система відобразить головну сторінку – сторінку новин з навігацією знизу екрану. Інтерфейс новин (рис. А.4) дозволить користувачу переглядати оновлення користувачів, на яких він підписан, одобрювати та писати коментарі під новинами.

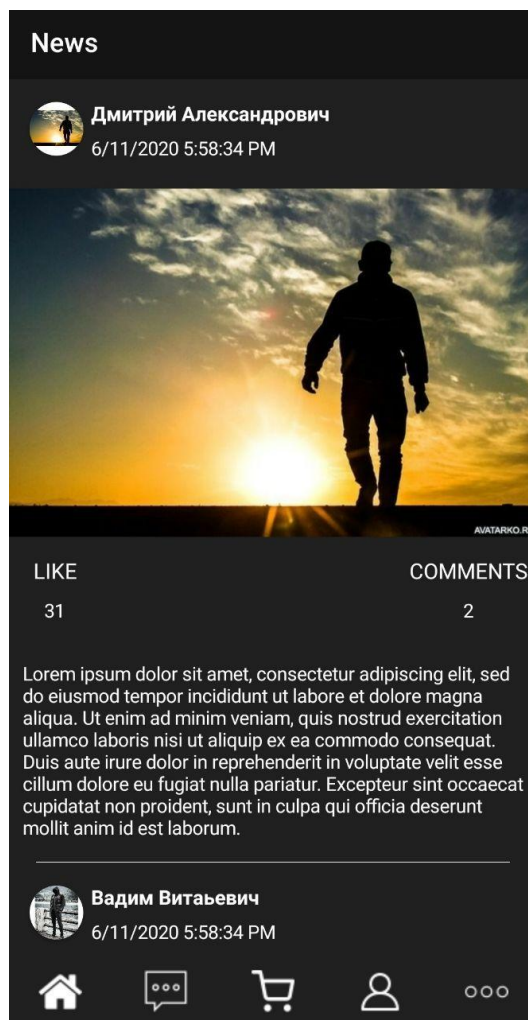


Рисунок А.4 - Интерфейс «Новини»

При навігації до інтерфейсу «Торговий майданчик» (рис. А.5) система відобразить список товарів в системі з короткими характеристиками, поле вводу для пошуку товару за назвою та посилання до пошуку за фільтрами. При натисканні на товар, система відобразить інтерфейс «Товар».

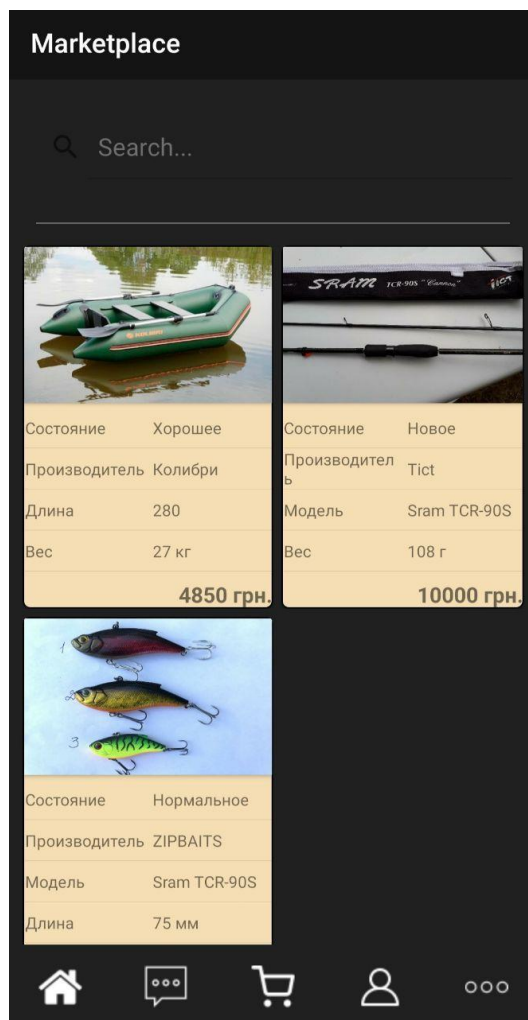


Рисунок А.5 - Интерфейс «Торговий майданчик»

Інтерфейс «Товар» (рис. А.6) включає список зображень товару, ціну, назву, характеристики, опис товару та інформацію про продавця з відгуками. Посилання «Написати продавцю» дозволить написати повідомлення основнику товару, а посилання «Забронювати» дозволить покупцеві забронювати товар до з'ясування.

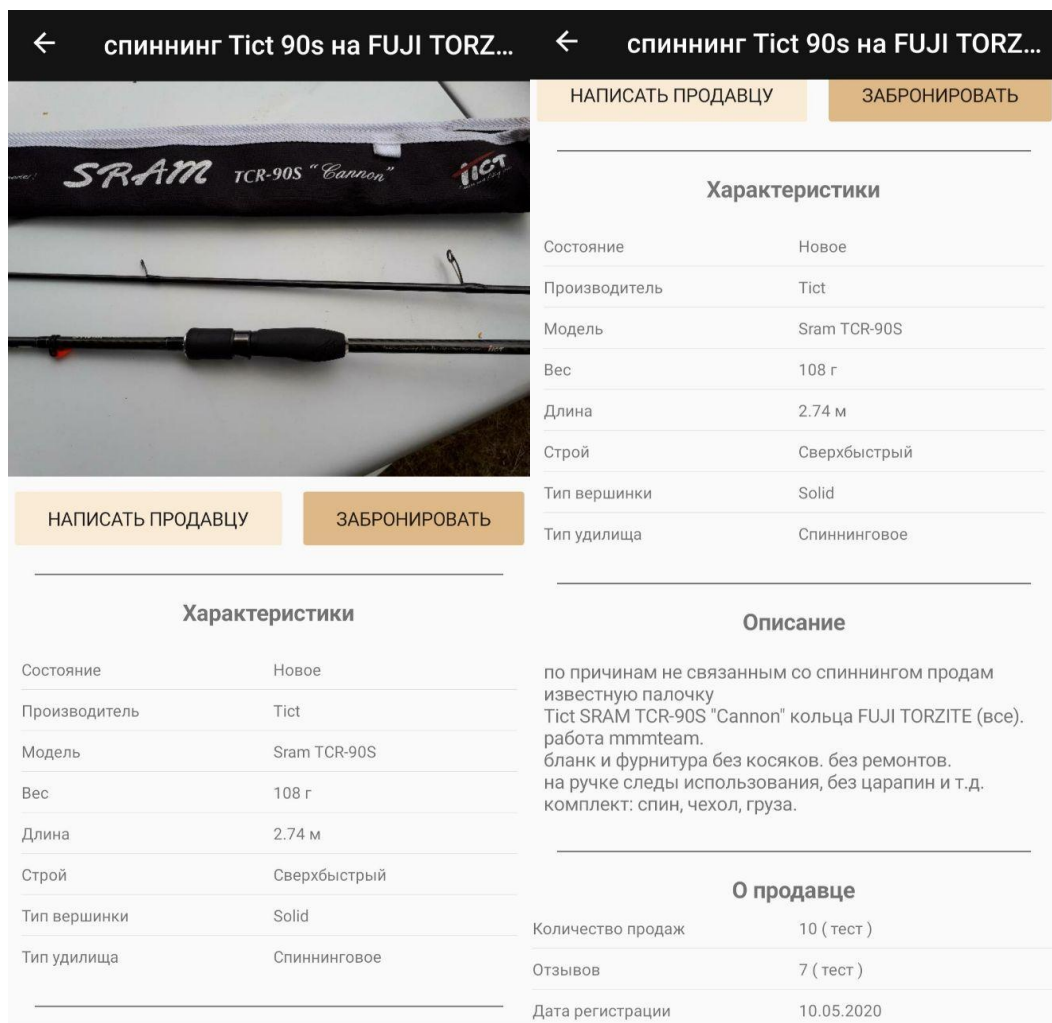


Рисунок А.6 - Интерфейс «Товар»

При навігації до інтерфейсу «Друзі» (рис. А.7) користувачу буде доступний пошук друзів та поточні друзі користувача з можливістю видалення з друзів.

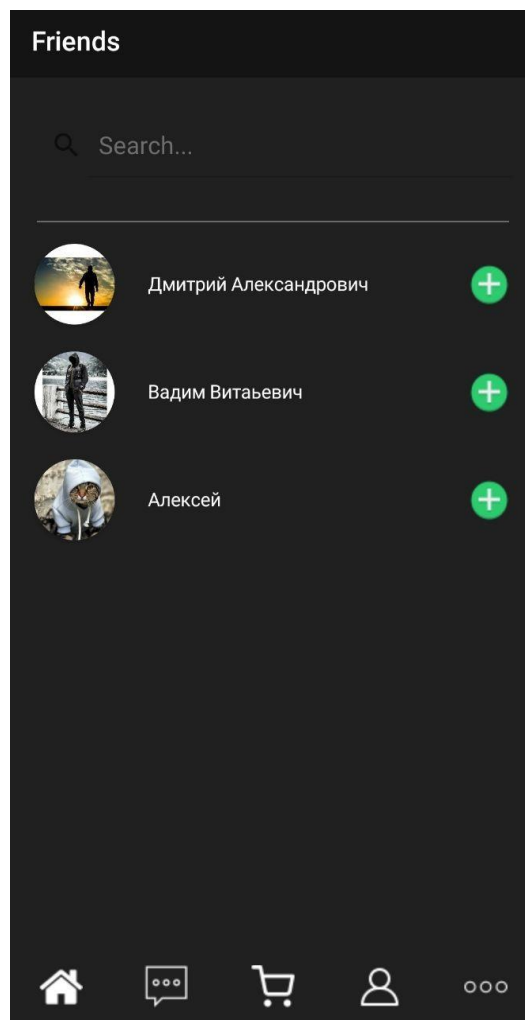


Рисунок А.7 - Інтерфейс «Друзі»

Інтерфейс «Групи» складається з двох вкладок. В першій вкладці система відобразить всі групи користувача, в другій – можливість пошуку груп за назвою та результат пошуку списком груп. При натисканні на групу, система відобразить інтерфейс групи з детальною інформацією, постами та меню для взаємодії з поточною групою.

При навігації до інтерфейсу «Форум» система відобразить користувачу доступні теми форуму з можливістю додавання їх до обраних та посилання на створення своєї теми для обговорення.

Інтерфейс «Оповіщення» дозволить користувачам бачити оповіщення від системи або інших користувачів за категоріями.

При навігації до інтерфейсу «Повідомлення» (рис. А.8) користувач побачить список діалогів між користувачами. При переході до діалогу користувач побачить список повідомлень з користувачем або користувачами.

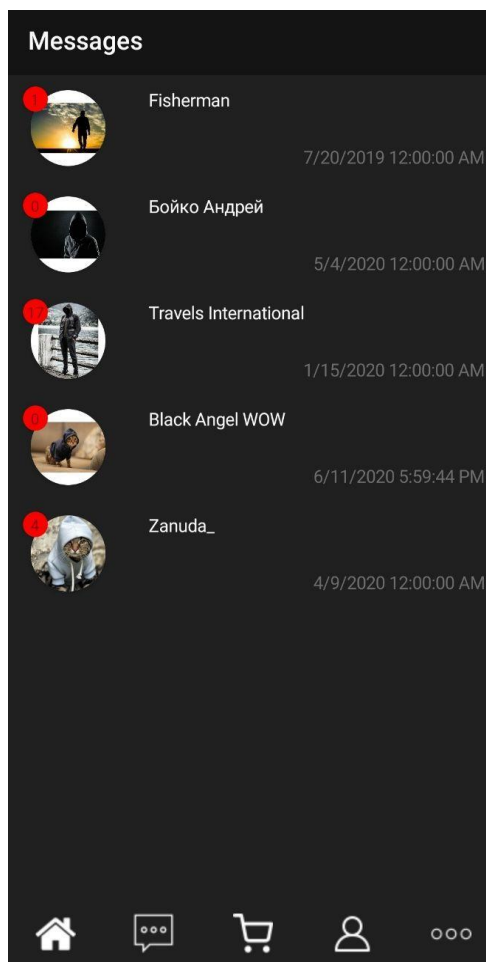


Рисунок А.8 - Інтерфейс діалоги користувача

Інтерфейс «Профіль» відображає всю інформацію користувача з фото- або відео- галереєю. Користувач може редагувати інформацію в своєму профілі, додавати фото або відео. В профілі користувача можна побачити кількість друзів та товарів користувача, при переході система відобразить список цих товарів. Посилання «Додати до друзів» дозволить додати користувача до друзів та «Написати» дозволить написати повідомлення користувачу.

При навігації до інтерфейсу «Конкурси» користувач побачить список актуальних конкурсів в системі, в кожному з яких він зможе прийняти участь, виконавши умови конкурсу. При переході до конкурсу буде доступні умови конкурсу, додатковий опис, дата закінчення конкурсу та кнопка для прийняття участі у конкурсі.

При навігації до інтерфейсу «Події» користувач побачить список подій за певними категоріями. При переході до події користувач побачить детальну інформацію про подію, місце, дату та інше, а також кнопку для прийняття участі (за умови, якщо максимальная кількість учасників Події ще не набрана).

ДОДАТОК Б – Код модулів

Вихідний текст програмних модулів представлено в Листингу Б.1.

Інтерфейсні форми розроблені за допомогою розмітки XAML.

Лістинг Б.1 – Модуль Registration.xaml

```
<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    mc:Ignorable="d"
    x:Class="SocialNetwork_CrossXamarin.Views.Registration">
    <ContentPage.Resources>
        <Style TargetType="StackLayout">
            <Setter Property="Padding" Value="30, 15"/>
        </Style>
    </ContentPage.Resources>

    <ContentPage.Content>
        <ScrollView>
            <Grid BackgroundColor="{DynamicResource Theme}"
                <Grid.RowDefinitions>
                    <RowDefinition Height="160" />
                    <RowDefinition />
                </Grid.RowDefinitions>

                <Image Source="logo.png" HeightRequest="80"
VerticalOptions="Center" HorizontalOptions="Center"/>

                <StackLayout Grid.Row="1" Orientation = "Vertical"
VerticalOptions="FillAndExpand">

                    <Label TextColor="White" FontSize="34"
HorizontalTextAlignment="Center" Margin="0, -20, 0, 40">Регистрация</Label>

                    <StackLayout>
                        <Label Text="Имя" Style="{StaticResource RPA_label}" />
                        <Entry Placeholder="Введите свое имя"
Style="{StaticResource RPA_entry}" Text="{Binding RealName}"/>
                    </StackLayout>

                    <StackLayout>
                        <Label Text="Фамилия" Style="{StaticResource
RPA_label}"/>
                        <Entry Placeholder="Введите свою фамилию"
Style="{StaticResource RPA_entry}" Text="{Binding Patronymic}" />
                    </StackLayout>

                    <StackLayout>
                        <Label Text="Номер телефона" Style="{StaticResource
RPA_label}"/>
                        <Entry Placeholder="Введите свой номер телефона"
Keyboard="Numeric" Style="{StaticResource RPA_entry}" Text="{Binding
PhoneNumber}"/>
                    </StackLayout>

                    <StackLayout>
                        <Label Text="Почта" Style="{StaticResource RPA_label}"/>
                        <Entry Placeholder="Введите свою почту" Keyboard="Email"
Style="{StaticResource RPA_entry}" Text="{Binding Email}" />
                    </StackLayout>
                </Grid>
            </ScrollView>
        </ContentPage.Content>
    </ContentPage>
```

```

        <StackLayout>
            <Label Text="Логин" Style="{StaticResource RPA_label}"/>
            <Entry Placeholder="Придумайте логин"
Style="{StaticResource RPA_entry}" Text="{Binding Login}"/>
        </StackLayout>

        <StackLayout>
            <Label Text="Пароль" Style="{StaticResource
RPA_label}"/>
            <Entry Placeholder="Придумайте пароль" IsPassword="true"
Style="{StaticResource RPA_entry}" Text="{Binding Password}" />
        </StackLayout>

        <StackLayout>
            <Label Text="Пароль" Style="{StaticResource
RPA_label}"/>
            <Entry Placeholder="Повторите пароль" IsPassword="true"
Style="{StaticResource RPA_entry}" Text="{Binding RepeatPassword}" />
        </StackLayout>

        <Label HorizontalOptions="CenterAndExpand"
VerticalOptions="EndAndExpand" Text="{Binding ErrorMsg, Mode=TwoWay}"
TextColor="Red" />

        <StackLayout VerticalOptions="EndAndExpand" Padding="0, 40,
0, 0">
            <Button Text="Зарегистрироваться" Command="{Binding
RegCommand}" Style="{StaticResource RPA_actionbtn}" />
            <Button Text="Войти" Command="{Binding GoToAuth}"
Style="{StaticResource RPA_helpactionbtn}"/>
        </StackLayout>
    </Grid>
</ScrollView>
</ContentPage.Content>
</ContentPage>

```

Authorization.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
    xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
    xmlns:d="http://xamarin.com/schemas/2014/forms/design"
    xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
    mc:Ignorable="d"
    x:Class="SocialNetwork_CrossXamarin.Views.Authorization">
    <ContentPage.Resources>
        <Style TargetType="StackLayout">
            <Setter Property="Padding" Value="30, 15"/>
        </Style>
    </ContentPage.Resources>

    <ContentPage.Content>
        <ScrollView>
            <Grid BackgroundColor="{DynamicResource Theme}">
                <Grid.RowDefinitions>
                    <RowDefinition Height="160" />
                    <RowDefinition />
                </Grid.RowDefinitions>

                <Image Source="logo.png" HeightRequest="100"
VerticalOptions="Center" HorizontalOptions="Center"/>

                <StackLayout Grid.Row="1" Orientation = "Vertical"
VerticalOptions="FillAndExpand" >

```



```

        <Label TextColor="White" FontSize="34"
HorizontalTextAlignment="Center" Margin="0, 0, 0, 80">Авторизация</Label>

        <StackLayout>
            <Label Text="Логин" Style="{StaticResource RPA_label}"/>
            <Entry Placeholder="Введите свой логин"
Style="{StaticResource RPA_entry}" Text="{Binding Login, Mode=TwoWay}"/>
        </StackLayout>

        <StackLayout>
            <Label Text="Пароль" Style="{StaticResource
RPA_label}"/>
            <Entry Placeholder="Введите свой пароль"
IsPassword="True" Style="{StaticResource RPA_entry}" Text="{Binding Password,
Mode=TwoWay}" />
        </StackLayout>

        <StackLayout VerticalOptions="EndAndExpand" Padding="0">
            <Label HorizontalOptions="CenterAndExpand"
VerticalOptions="EndAndExpand" Text="{Binding ErrorMsg, Mode=TwoWay}"
TextColor="Red" />
            <Button Text="Войти" Command="{Binding AuthCommand}"
Style="{StaticResource RPA_actionbtn}" />

            <StackLayout Orientation="Horizontal" Padding="0">
                <Button Text="Регистрация"
HorizontalOptions="StartAndExpand" Command="{Binding GoToRegCommand}"
Style="{StaticResource RPA_helpactionbtn}"/>
                <Button Text="Забыли пароль?"
HorizontalOptions="EndAndExpand" Command="{Binding GoToChangePassCommand}"
Style="{StaticResource RPA_helpactionbtn}"/>
            </StackLayout>
        </StackLayout>
    </Grid>
</ScrollView>
</ContentPage.Content>
</ContentPage>

```

NewsListView.xaml

```

<?xml version="1.0" encoding="utf-8" ?>
<ContentPage xmlns="http://xamarin.com/schemas/2014/forms"
xmlns:x="http://schemas.microsoft.com/winfx/2009/xaml"
xmlns:d="http://xamarin.com/schemas/2014/forms/design"
xmlns:mc="http://schemas.openxmlformats.org/markup-
compatibility/2006"
mc:Ignorable="d"
x:Class="SocialNetwork_CrossXamarin.Views.NewsListView"
x:Name="NewsPage" Title="News">

    <ContentPage.Content >
        <StackLayout Orientation="Vertical" BackgroundColor="{StaticResource
DarkThemeBackground}" >
            <RefreshView >
                <ScrollView Style="{StaticResource newsScrollView}">
                    <CollectionView ItemsSource="{Binding NewsCollection}"
SelectedItem="{Binding SelectedNews}"
SelectionChanged="CollectionView_SelectionChanged" >
                        <CollectionView.ItemTemplate>
                            <DataTemplate>
                                <Grid >
                                    <Grid.RowDefinitions>
                                        <RowDefinition Height="Auto" />
                                        <RowDefinition Height="Auto" />
                                        <RowDefinition Height="Auto" />

```

```

<RowDefinition Height="1" />
<RowDefinition Height="Auto" />
<RowDefinition Height="1" />
</Grid.RowDefinitions>

<StackLayout Style="{StaticResource
newsTopPartPost}" BackgroundColor="{StaticResource DarkThemeBackground}" >
    <Frame Style="{StaticResource
ImageFrame}" >
        <Image HorizontalOptions="Center"
VerticalOptions="Center" Source="{Binding Owner.ProfileImagePath}" />
    </Frame>

    <StackLayout Orientation="Vertical" >
        <Label Text="{Binding
Owner.FullName}" Style="{StaticResource Owner_Label}" />
        <Label Text="{Binding DT}"
TextColor="{StaticResource textDarkTheme}" HorizontalOptions="StartAndExpand" />
    </StackLayout>
</StackLayout>

<Image Grid.Row="1" Source="{Binding
Images_path}" HorizontalOptions="FillAndExpand" VerticalOptions="FillAndExpand"
/>

<StackLayout Grid.Row="2"
Orientation="Horizontal" Margin="5" BackgroundColor="{StaticResource
DarkThemeBackground}" >
    <Frame Style="{StaticResource
newsActionFrame}" HorizontalOptions="StartAndExpand" >
        <StackLayout Style="{StaticResource
newsActionStackLayout}" >
            <Button Text="Like"
Command="{Binding
Path=BindingContext.SetLike, Source={x:Reference NewsPage} }"
Style="{StaticResource
newsActionButton}" />
            <Label Text="{Binding Likes}"
Style="{StaticResource newsActionLabel}" />
        </StackLayout>
    </Frame>

    <Frame Style="{StaticResource
newsActionFrame}" HorizontalOptions="EndAndExpand" >
        <StackLayout Style="{StaticResource
newsActionStackLayout}" >
            <Button Text="Comments"
Command="{Binding
Path=BindingContext.GoToComments, Source={x:Reference NewsPage} }"
Style="{StaticResource
newsActionButton}" Margin="-10" />
            <Label Text="{Binding
Comments.Count}" Style="{StaticResource newsActionLabel}" />
        </StackLayout>
    </Frame>
</StackLayout>

<Frame Grid.Row="4" Padding="10"
BackgroundColor="{StaticResource DarkThemeBackground}" BorderColor="Transparent"
>
    <Label Text="{Binding Description}"
Style="{StaticResource newsDescriptionLabel}" />
</Frame>

```

```

                                <Frame
                                    Grid.Row="5"
                                    HorizontalOptions="FillAndExpand"
                                    VerticalOptions="Center"
                                    Margin="20, 0, 20, 0"
                                    HeightRequest="1"
                                    BackgroundColor="{StaticResource
placeholderDarkTheme}" />

                                </Grid>
                            </DataTemplate>
                        </CollectionView.ItemTemplate>
                    </CollectionView>
                </ScrollView>
            </RefreshView>

            <Grid VerticalOptions="FillAndExpand"
HorizontalOptions="FillAndExpand" BackgroundColor="{StaticResource
DarkThemeBackground}" >
                <Grid.RowDefinitions>
                    <RowDefinition />
                </Grid.RowDefinitions>

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

                <Button Grid.Column="0" Style="{StaticResource menuButton}"
ImageSource="menuHomeLight.png" Margin="2,0,0,5" />
                <Button Grid.Column="1" Style="{StaticResource menuButton}"
ImageSource="menuMessagesLight.png" Margin="0,0,0,5" />
                <Button Grid.Column="2" Style="{StaticResource menuButton}"
ImageSource="menuShopLight.png" Margin="0,0,0,5" />
                <Button Grid.Column="3" Style="{StaticResource menuButton}"
ImageSource="menuProfileLight.png" Margin="0,0,0,5" />
                <Button Grid.Column="4" Style="{StaticResource menuButton}"
ImageSource="menuMoreLight.png" Clicked="Button_Clicked" Margin="0,0,2,5"/>

            </Grid>
        </StackLayout>
    </ContentPage.Content>
</ContentPage>

```

NewsListViewModel.cs

```

using SocialNetwork_CrossXamarin.Base;
using SocialNetwork_CrossXamarin.DAL.DataServices;
using SocialNetwork_CrossXamarin.Interfaces;
using System.Collections.ObjectModel;
using System.Linq;
using System.Windows.Input;
using Xamarin.Forms;

namespace SocialNetwork_CrossXamarin.ViewModels.News
{
    public class NewsListViewModel : BaseViewModel
    {
        private ObservableCollection<INewsItem> _newsCollection;
    }
}

```

```

public ObservableCollection<INewsItem> NewsCollection
{
    get { return _newsCollection; }
    set
    {
        if (_newsCollection != value)
        {
            _newsCollection = value;
            OnPropertyChanged(nameof(NewsCollection));
        }
    }
}

public ICommand SetLike { get; set; }
public ICommand GoToComments { get; set; }

public NewsListViewModel()
{
    NewsCollection = new
ObservableCollection<INewsItem>(DataServices.NewsService.GetListAsync().Result.To
oList());

    SetLike = new Command(SetLikeAsync);
    GoToComments = new Command(GoToCommentsAsync);
}

private void SetLikeAsync()
{
    Application.Current.MainPage.DisplayAlert("Like", "", "Ok");
}

private async void GoToCommentsAsync()
{
    await Application.Current.MainPage.DisplayAlert("Comments", "",
"Ok");
}
}
}

```

NewsObject.cs

```

using SocialNetwork_CrossXamarin.Interfaces;
using SocialNetwork_CrossXamarin.Models.Friends;
using System;
using System.Collections.Generic;

namespace SocialNetwork_CrossXamarin.Models.News
{
    public class NewsObject : BaseModel, INewsItem
    {
        //public List<String> Images_path { get; set; }

        // TODO: init property with user type
        public User Owner { get; set; }
        public String Images_path { get; set; }
        public String Description { get; set; }
        public int Likes { get; set; }
        public List<IComment> Comments { get; set; }
        public DateTime? DT { get; set; }
    }
}

```

BaseViewModel.cs

```
using System.ComponentModel;
using System.Runtime.CompilerServices;

namespace SocialNetwork_CrossXamarin.Base
{
    public class BaseViewModel : INotifyPropertyChanged, INotifyPropertyChanging
    {
        public event System.ComponentModel.PropertyChangingEventHandler
PropertyChanging;
        public event PropertyChangedEventHandler PropertyChanged;

        protected bool CallPropertyChangeEvent { get; set; } = true;

        protected void OnPropertyChanged([CallerMemberName] string propertyName
= null)
        {
            PropertyChanged?.Invoke(this, new
PropertyChangedEventArgs(propertyName));
        }

        protected void OnPropertyChanging([CallerMemberName] string propertyName
= null)
        {
            PropertyChanging?.Invoke(this, new
System.ComponentModel.PropertyChangingEventArgs(propertyName));
        }
    }
}
```