

Міністерство освіти і науки України
Харківський національний університет радіоелектроніки

Факультет _____ ННЦ ЗФН _____
(повна назва)

Кафедра _____ Програмної інженерії _____
(повна назва)

АТЕСТАЦІЙНА РОБОТА
Пояснювальна записка

_____ другий (магістерський) _____
(рівень вищої освіти)

Дослідження алгоритмів пошуку інформації в спеціалізованих соціальних
мережах
(тема)

Виконав: студент 2 курсу, групи ПЗСзм-18-1
спеціальності 121- Інженерія програмного
забезпечення _____

(код і повна назва спеціальності)

освітньо-професійної програми

Програмне забезпечення систем _____

(повна назва освітньої програми)

_____ Хмелевський О.В. _____

(прізвище, ініціали)

Керівник _____ проф. Шубін І.Ю. _____

(посада, прізвище, ініціали)

Допускається до захисту

Зав. кафедри, проф. _____

З.В.Дудар

2019 р.

Харківський національний університет радіоелектроніки

Факультет Комп'ютерних наук

Кафедра Програмної інженерії

Рівень вищої освіти другий (магістерський)

Спеціальність 121– Інженерія програмного забезпечення

(код і повна назва)

Освітньо-професійна програма Програмне забезпечення систем

(повна назва)

ЗАТВЕРДЖУЮ:

Зав. кафедри _____

(підпис)

«____» _____ 20 ____ р.

ЗАВДАННЯ

НА АТЕСТАЦІЙНУ РОБОТУ

Студентові Хмелевському Олегу Віталійовичу

(прізвище, ім'я, по батькові)

1. Тема роботи Дослідження алгоритмів пошуку інформації в спеціалізованих соціальних мережах

затверджена наказом по університету від «____» _____ 2019 р № ____ Стз
заповнюється вручну після отримання наказу

2. Термін подання студентом роботи до екзаменаційної комісії

10 грудня 2019 р.

3. Вихідні дані до роботи проаналізувати існуючі алгоритми, що використовуються для вимоги до веб-сервісу, що буде розроблено, середовище проектування PHPStorm 2018, мова розробки PHP, JavaScript

4. Перелік питань, що потрібно опрацювати в роботі мета роботи, аналіз проблемної галузі і постановка задачі, опис запропонованих варіантів оптимізації, використовувані методи та алгоритми, опис розробленої програмної системи, опис застосованих оптимізацій, аналіз можливих застосувань

5. Консультанти розділів роботи

Найменування розділу	Консультант (посаду, прізвище, ім'я, по батькові)	Позначка консультанта про виконання розділу	
		підпис	дата
Спецчастина	проф. Шубін І.Ю.		

КАЛЕНДАРНИЙ ПЛАН

№	Назва етапів роботи	Терміни виконання етапів роботи	Примітка
1.	Аналіз предметної галузі	10 жовтня 2019 р.	
2.	Огляд існуючих методів	27 жовтня 2019 р.	
3.	Проектування та розробка ПЗ	15 листопада 2019 р.	
4.	Підготовка пояснювальної записки	25 листопада 2019 р.	
5.	Спецчастина	26 листопада 2019 р.	
6.	Підготовка презентації та доповіді	30 листопада 2019 р.	
7.	Попередній захист	10 грудня 2019 р.	
8.	Нормоконтроль, рецензування	11 грудня 2019 р.	
9.	Занесення диплома в електронний архів	12 грудня 2019 р.	
10.	Допуск до захисту в зав. кафедри	14 грудня 2019 р.	
* заповнюється вручну після виконання чергового пункту			

Дата видачі завдання ____ 2019 р.

Студент _____
(підпис)

Керівник роботи _____ проф. Шубін І.Ю.
(підпис) (посада, прізвище, ініціали)

РЕФЕРАТ / ABSTRACT

Пояснювальна записка до атестаційної роботи: 75 с., 31 рис., 3 додатки, 24 джерела.

ВЕБ-СЕРВІС, КЛАСИФІКАЦІЯ, СОЦІАЛЬНА МЕРЕЖА, PHP 7, JAVASCRIPT ES5, JQUERY, HTML5/CSS3.

Метою роботи є створення програмного продукту для автовласників та інших користувачів, який повинен бути зручним та безпечним для користувача .

Соціальна мережа спрямована на побудову спільноти в Інтернеті з людей зі схожими інтересами та діяльністю, а саме автомобілістів.

Результатом є програмний прототип соціальної мережи, в якій єв'язок здійснюється за допомогою сервісу внутрішньої пошти або миттєвого обміну повідомленнями.

WEB SERVICE, CLASSIFICATION, NET, PHP 7, JavaScript ES5, jQuery, HTML5/CSS3.

The goal of the work is to create software for car owners and other users that should be user-friendly and safe.

A social network is aimed at building a community online with people with similar interests and activities, namely motorists.

The result is a software prototype of a social network in which communication is made through an internal mail service or instant messaging..

ЗМІСТ

Вступ.....	6
1 Аналіз проблемної галузі	8
1.1 Аналіз предметної області та аналогів	8
1.2 Аналіз аналогів	10
1.3 Постановка задач дослідження	15
2 Опис проведених теоретичних досліджень	19
3 Аналіз результатів досліджень	28
3.1 Опис вимог до програмного продукту	28
3.2 Аналіз основних функцій системи	29
4 Опис розробленого програмного забезпечення	30
4.1 UML проектування ПЗ	30
4.2 Проектування архітектури ПЗ	35
4.3 Проектування бази даних	37
4.4 Приклади найцікавіших алгоритмів та методів	40
4.5 Проектування UI/UX дизайну системи	41
5 Опис програмної реалізації	46
5.1 Вибір технологій розробки	46
5.2 Програмна реалізація	47
Опис можливості використання отриманих результатів	54
Висновки	56
Перелік джерел посилання	57
Додаток А Фрагмент коду програми	59
Додаток Б Слайди презентації.....	62
Додаток В Відгук і рецензії	72

ВСТУП

Соціальні мережі відіграють важливу роль для сучасної людини. Іноді люди навіть не помічають, як захопившись подіями, що відбуваються в мережі, проводять багато часу в Інтернеті. Адже це не просто так. З нами відбуваються якісь події, і ми поспішаємо цим поділитися з іншими. Ви зробили цікаву фотографію - її неодмінно варто показати друзям. Спілкування з людьми, які зараз далеко від нас і події, що відбуваються в їх житті, захоплюють нас, і ми поспішаємо опинитися в соціальних мережах в вільну хвилину. Іноді складно підтримувати відносини з десятками друзів, з якими спілкуєшся зараз або знав колись, докладніше дізнатися про їхнє життя ви завжди можете, переглянувши стрічку новин своїх друзів.

Соціальні мережі також допомагають нам при пошуку наших давніх знайомих, зв'язок з якими було втрачено багато років назад. Якщо вам необхідно обмінюватися інформацією, робити оголошення та спілкуватися з певною групою, ви можете створити свою групу у мережі, де вся інформація може бути видна тільки цьому колу спілкування. Вступаючи у різні групи, що відповідають вашим інтересам, ви зможете дізнатися багато цікавих фактів, інформацію про ваших кумирів або рецепти улюблених страв. Багато також дуже люблять соціальні мережі за можливість прослуховування музики і перегляд відео, якими ви можете поділитися з друзями і надалі мати багато тем для розмов.

Детальніше вивчити всі можливості, плюси і мінуси соціальних мереж ви можете, вивчивши одну з таких мереж самостійно. Соціальні мережі в наш час для деяких людей - ціле життя. Там вони можуть проводити як кілька хвилин, так і півдня в пошуках цікавої інформації. Однак, знаючи всі плюси віртуального спілкування, не слід забувати про те, що живемо ми у реальному світі і підтримувати відносини зі справжніми друзями краще вживу. Так що насолоджуйтесь спілкуванням в мережі, будьте у курсі подій, нових течій в різних сферах, але не забувайте про реальне життя і близьких людей, що поруч з вами.

Так само більша частина користувачів створюють свій персональний аккаунт у соціальній мережі не тільки для того, щоб почати спілкування з іншим користувачем (людиною) системи, але і знайти товар, який буде йому до смаку, так як соціальні мережі являються на 25-30% торговими майданчиками. Користувач може легко «укласти угоду» з продавцем або ж знайти потенційного покупця. Для всіх не секрет, що існує дуже багато різного виду реклами. Соціальні мережі без цього теж не обійшлися. Будь-який користувач, який має товар, фірму або може запропонувати услуги в пошуку житла, місця для відпочинку, оренди або купівлі автомобіля, то для такого користувача існує можливість розміщення реклами в соціальних мережах, це допоможе швидко та легко поширити інформацію про свою діяльність або товар.

Таким чином, метою магістерської роботи є створення продукту для авто користувачів, який повинен бути зручним та безпечним для користувача.

1 АНАЛІЗ ПРОБЛЕМНОЇ ГАЗУЗІ

1.1 Аналіз предметної області та аналогів

Соціальна мережа спрямована на побудову спільнот в Інтернеті з людей зі схожими інтересами та/або діяльністю. Зв'язок здійснюється за допомогою сервісу внутрішньої пошти або миттєвого обміну повідомленнями. Зазвичай на сайті мережі можна вказати інформацію про себе (дату народження, школу, вуз, улюблені заняття та інше), по якій аккаунт користувача зможуть знайти інші учасники. Розрізняються відкриті і закриті соціальні мережі. Одна із звичайних рис соціальних мереж – система «друзів» і «груп».

Сторінка користувача – уявлення, яке може містити фото користувача, ім'я, стіну, посилання на друзів користувача, посилання на відправку повідомлення користувачу, дані про користувача: дату народження, інтереси, особиста інформація та інше.

Багато соціальні мережі дозволяють завантажувати фотографії, переглядати як свої, так і чужі фотографії. Писати повідомлення можна як особисто - коли тільки одержувач читає повідомлення, так і на стіну - коли безліч людей можуть прочитати це повідомлення. Стіна – область на екрані, де користувачі можуть залишати повідомлення. У деяких соціальних мережах крім повідомлень на стіну можна залишати музичні треки, відео, картинки. Кожен користувач може мати статус – текст, який він сам пише. Це може бути текст, що описує те, чим він зайнятий або де він, його емоції або взагалі будь-який текст. Статус зазвичай розташовується поруч з фотографією користувача і ім'ям. Ім'я користувач може писати будь-: як реальне, так і прізвисько або їх комбінацію. Друзі користувача – це інші користувачі, які за обопільною згодою стали «друзями» в соціальній мережі. Це можуть бути люди, які часто спілкуються один з одним або просто хотіли б, щоб вони були «друзями» в соціальній мережі.

У соціальних мережах можна здійснювати пошук. Пошук зазвичай можна виробляти за різними критеріями – полях, які утримуються в особистих сторінках користувача. Зазвичай найголовніший критерій пошуку – ім'я користувача.

Щоб користуватися соціальною мережею потрібно пройти реєстрацію - процес, де відвідувач вводить логін, пароль та e-mail, після якого йому надаються права на користування соціальною мережею. Після реєстрації відвідувач стає користувачем і може зайти на сайт після того, як пройде авторизацію. Авторизація - це не те ж саме що ідентифікація та аутентифікація: ідентифікація – це називання особою себе системі; аутентифікація – це встановлення відповідності особи, призначеному ним самим ідентифікатором; а авторизація – надання цій особі можливостей у відповідність до покладених йому правами або перевірка наявності прав при спробі виконати будь-яку дію.

В якості необхідних складових соціальних мереж виділяються наступні:

- наявність спектра дій (оцінок, коментарів, виправлень і т.п.) по відношенню до редакційного змісту (якщо таке в принципі є);
- простір власних дій «спільника» (свої публікації, наповнення проекту інформацією і т.п.);
- механізми особистої представленості «спільника» на сайті (всякі особисті кабінети, профілі, персональні стрічки і т.п.), що складаються з двох базових речей - самопрезентація (в т.ч. посилання на особисті профілі на інших ресурсах) і «слід» діяльності людини на даному проекті;
- механізми зв'язності, взаємних дій, об'єднання і роботи в групах і т.п., тобто вибудовування спільнот поверх загальної тематики проекту;
- механізми оцінок, статусів, репутації, автомодерації і т.п., спрямовані перш за все на змістовне взаємодію «спільників» всередині проекту.

1.2 Аналіз аналогів

На сьогоднішній день існує велика кількість соціальних мереж: drive2.ru, vk.com, facebook.com, instagram.com, twitter.com, telegram, тощо.

Drive2.ru - це сайт продажу автомобілів і наполовину соціальна мережа. За допомогою drive2 можна знайти підтримку автомобіля в хорошому або не дуже хорошому стані, залежить від того, що саме ви шукаєте і для яких цілей (рис.1.1).

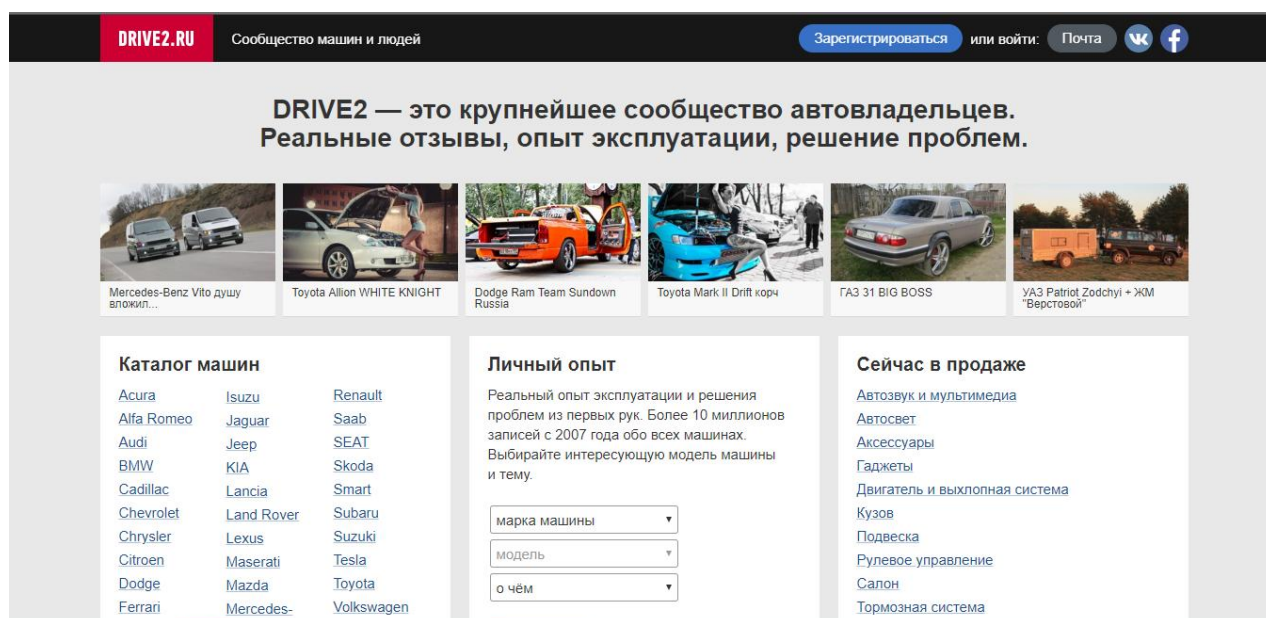


Рисунок 1.1 – Головна сторінка сайту Drive2.ru

Vk.com – це гігант соціальних мереж створений Павлом Дуровим у 2006 році для країн СНД (рис. 1.2). Завдяки Павлу Дурову і його проекту звичайні люди (користувачі) почали активно спілкуватися один з одним на величезній відстані і заводили нові знайомства. Цей проект активно розвивався в кращу сторону, але нікому не секрет, що 2014 року компанія Mail.ru купила цього «гіганта» за дуже невеличку суму грошей, і взяла розвиток vk.com у свої руки (рис. 1.3).

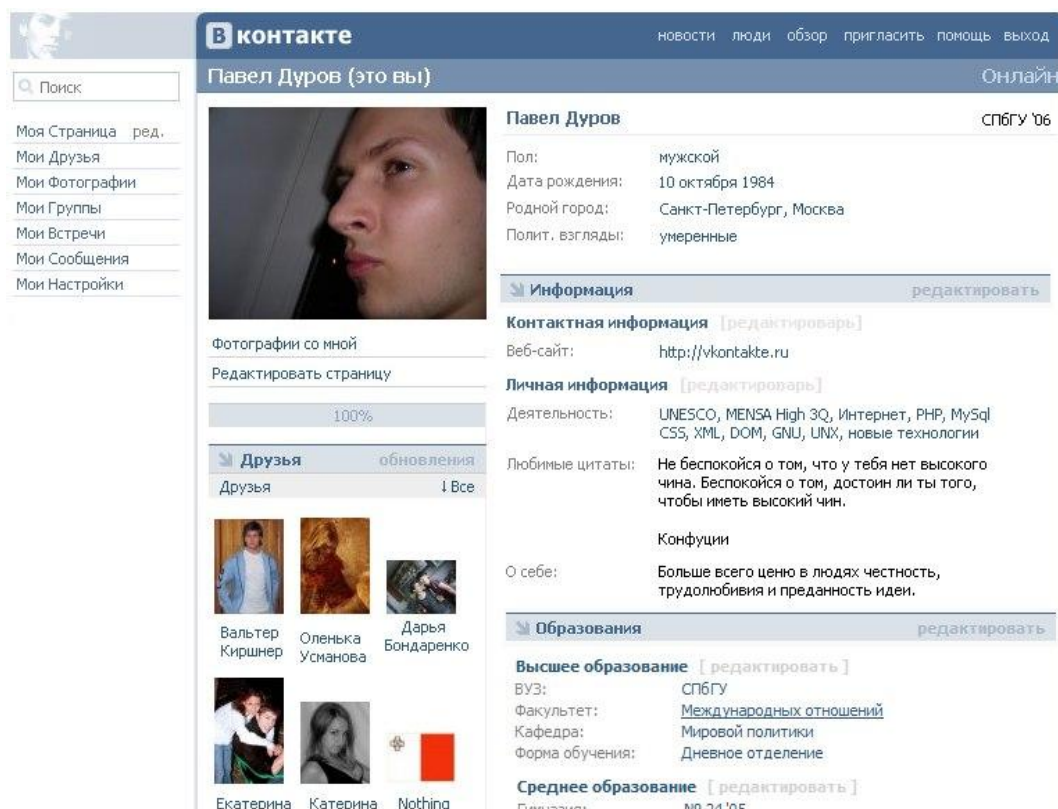


Рисунок 1.2 – Интерфейс соціальної мережі vk.com 2006 рік

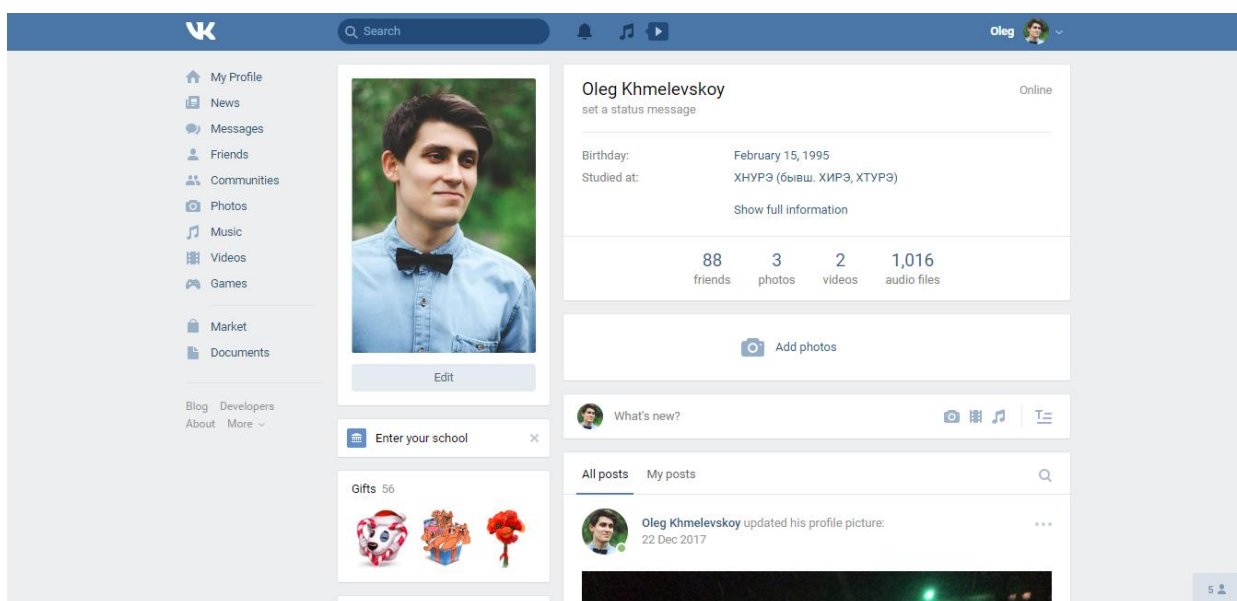


Рисунок 1.3 – Интерфейс соціальної мережі vk.com 2018 рік

Facebook – найбільша у світі соціальна мережа, що почала працювати 4 лютого 2004 року як мережа для студентів деяких американських університетів. Засновником та головою сервісу є Марк Цукенберг. За даними Alexa, сайт

Facebook.com займає у світі 3 місце за відвідуваністю. Станом на липень 2016 року, кількість користувачів становила 1,7 млрд, з яких 1,1 млрд відвідує свої аккаунти щодня. Кількість українських користувачів, станом на травень 2014 року, становила близько 6 млн. 30 червня 2017 року кількість користувачів досягла 2 млрд чоловік у всьому світі, в Україні 10 млн чоловік (рис 1.4).

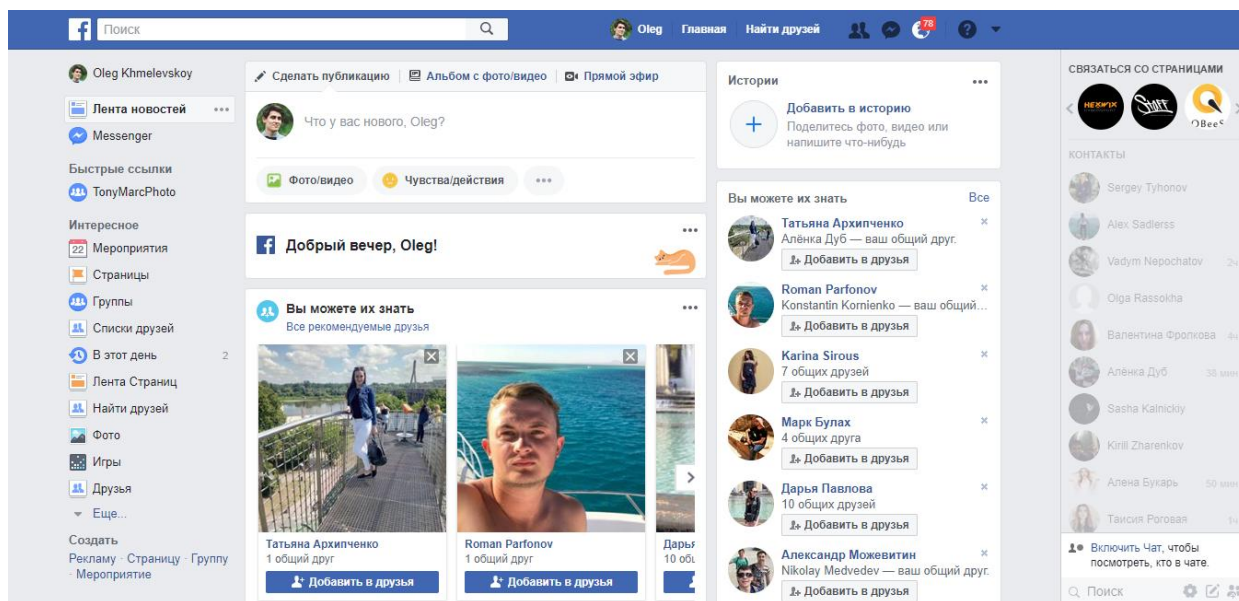


Рисунок 1.4 – Інтерфейс соціальної мережі facebook.com

Instagram – соціальна мережа, що базується на обміні фотографіями, дозволяє користувачам робити фотографії, застосовувати до них фільтри, а також поширювати їх через свій сервіс і низку інших соціальних мереж. Є одним із найпопулярніших сервісів у мистецтві айфонографії. Instagram робить фотографії в квадратній формі - як камери KodakInstamatic і Pakaroid. Більшість же мобільних фоторедакторів використовує співвідношення сторін 3:2 (рис. 1.5).

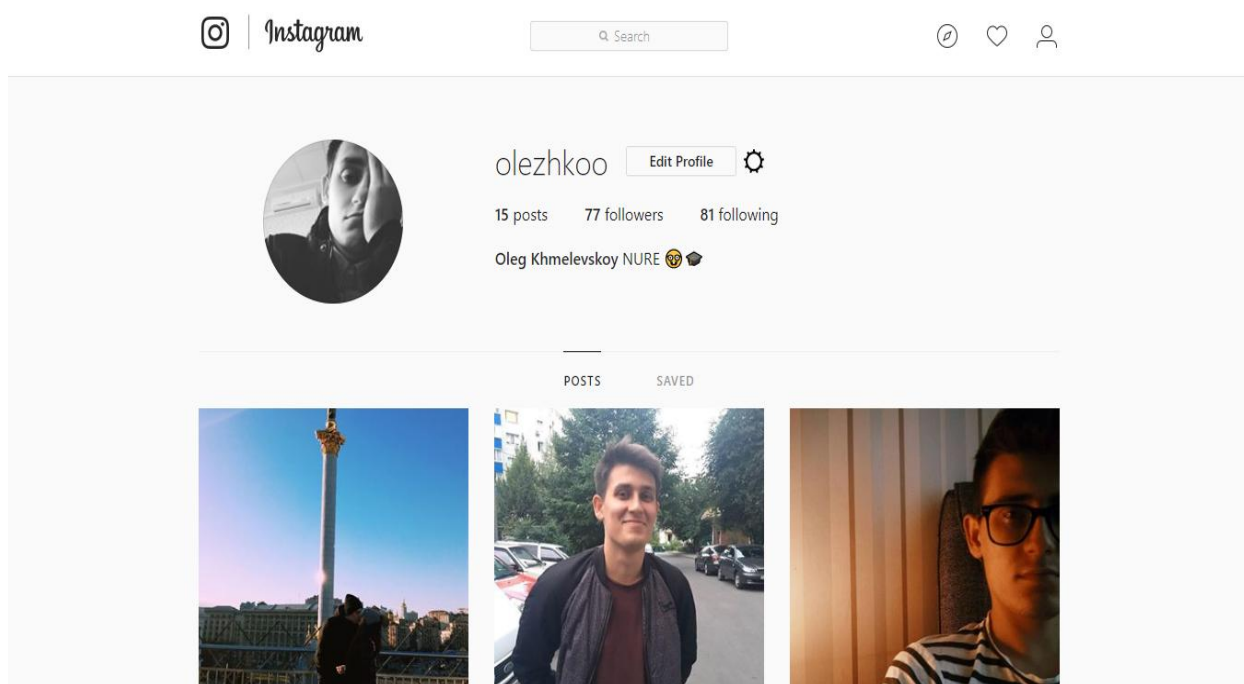


Рисунок 1.5 – Інтерфейс соціальної мережі instagram.com

Twitter – соціальна мережа, яка є мережею мікроблогів, дає змогу користувачам надсилати короткі текстові повідомлення (до 280 символів), використовуючи, служби миттєвих повідомлень і сторонні програми-клієнти (рис 1.6).

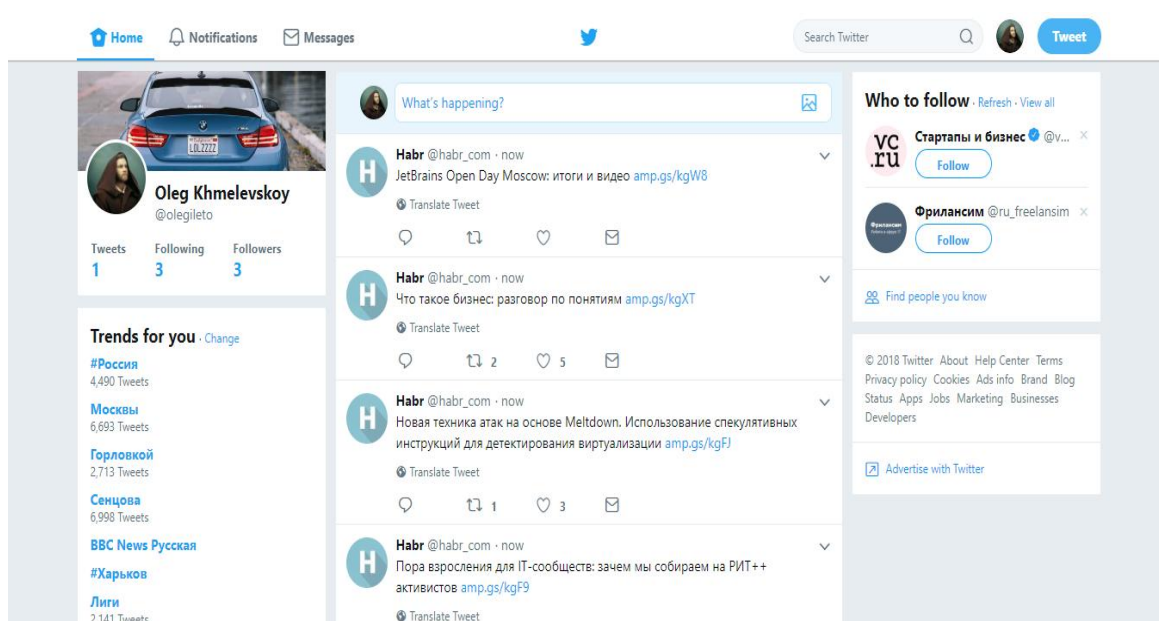


Рисунок 1.6 – Інтерфейс сайту twitter.com

Telegram – месенджер, програмне забезпечення для смартфонів, планшетів, ПЕ, яке дозволяє обмінюватися текстовими повідомленнями та різноманітними файлами, зокрема графічними файлами та відеофайлами, а також безкоштовно телефонувати іншим користувачам програми.

Обліковий запис користувача прив'язується до номеру мобільного телефону: щоб авторизуватися, потрібно ввести код авторизації з СМС. Такі коди мають обмежені терміни придатності. Таким чином, користувач позбавляється необхідності запам'ятовувати чи зберігати десь свій пароль (рис 1.7).

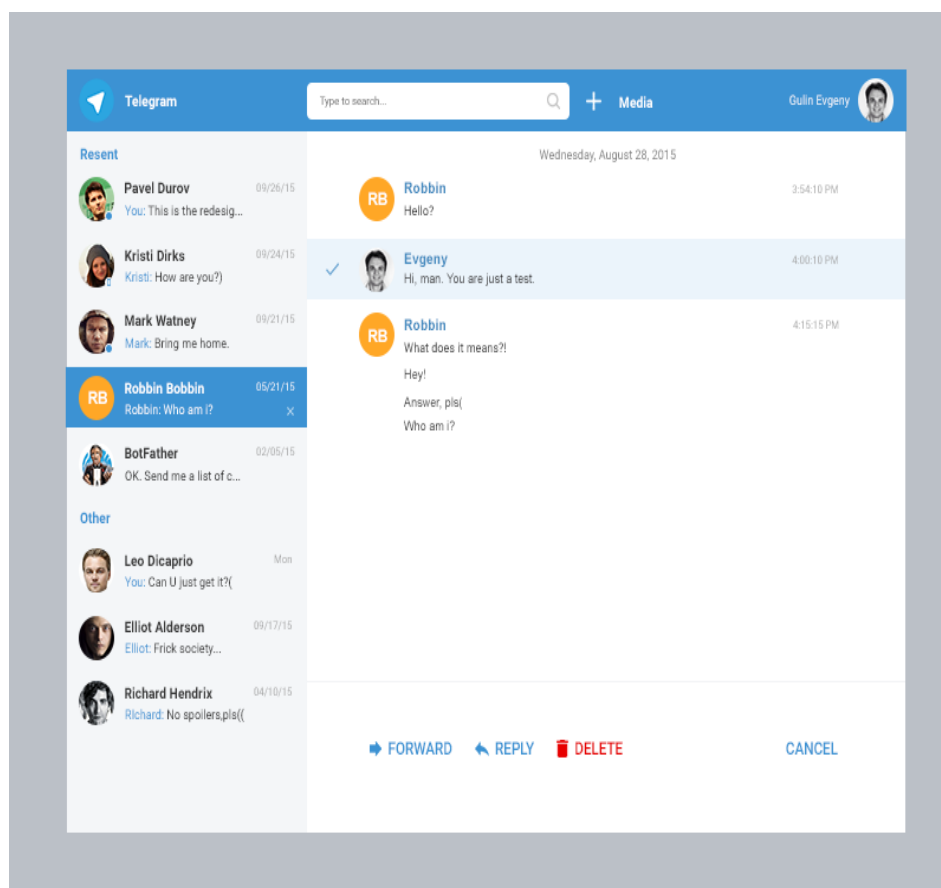


Рисунок 1.7 – Інтерфейс telegram web

Найважливішою особливістю, що відрізняє Telegram від своїх конкурентів, можна вважати інструмент комунікації в форматі публічних каналів (рис. 1.8).

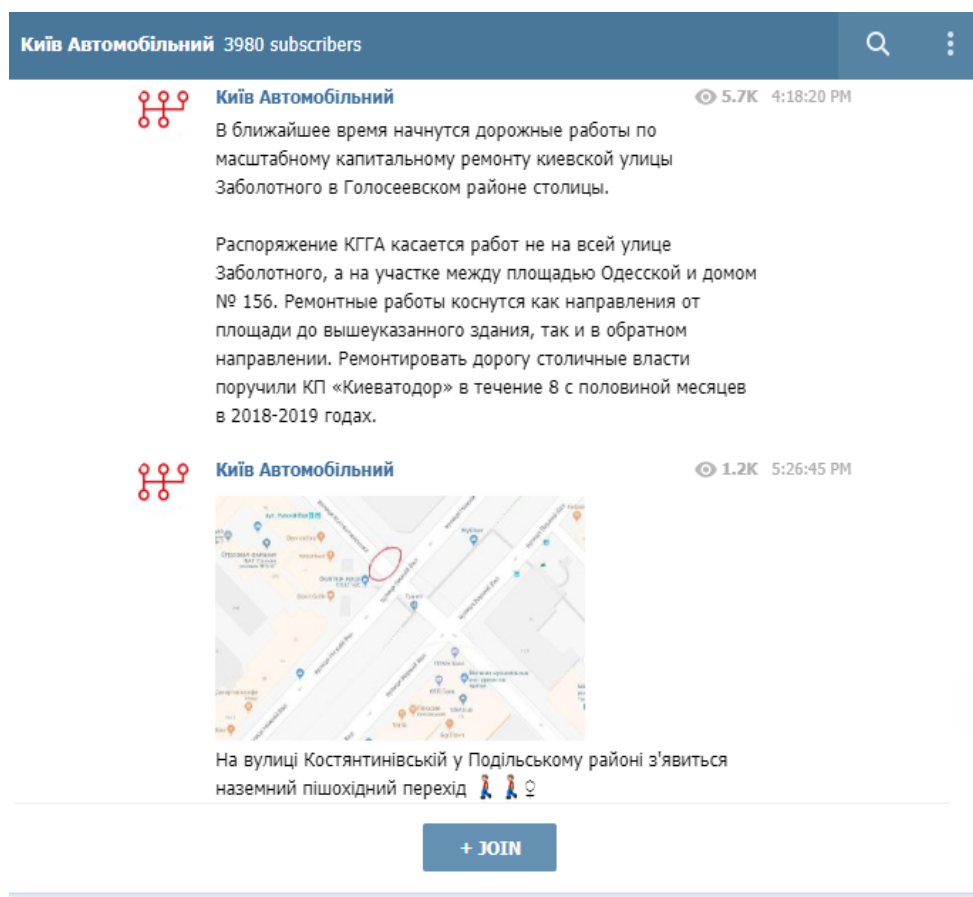


Рисунок 1.8 – Приклад тематичного блогу в Telegram

Такий спосіб дозволяє автору або групі авторів ділитися інформацією з необмеженим колом осіб з мінімальною дистанцією між читачем і контентом, але і зберігати при цьому анонімність.

1.3 Постановка задач дослідження

Перед початком роботи над програмною системою необхідно визначитися з вимогами до неї. У загальному випадку під вимогами розуміють сукупність властивостей, які повинна мати система, що реалізується.

Метою роботи є створення соціальної мережі для автомобілістів.

Данна соціальна мережа повинна стати зручним інструментом для користування. Для цього користувачеві надається можливість створити

персональний акаунт, за допомогою якого він має можливість робити усі дії які передбачає собою мережа.

Таким чином для виконання задачі, що поставлена, система повинна підтримувати наступні основні функції:

- створення персонального профілю;
- додавання або видалення друзів;
- додавання відео;
- зміна інформації о користувачеві;
- зміна статусу користувача;
- додавання або видалення аватару користувача;
- додавання або видалення зображення заднього фону профілю користувача;
- пошук друзів;
- надіслання повідомлення друзям;
- додавання окремих фото користувача;
- додавання інформації щодо автомобіля користувача;
- перегляд інформації друзів;
- додавання або видалення постів;
- перегляд стрічки постів користувача;
- додавання або видалення заміток;
- відправлення листів друзям;
- оповіщення щодо додавання користувача до друзів;
- перегляд іншим користувачем друзів;

Також, не зважаючи на повну працездатність системи, в майбутньому

планується додавання наступних функцій:

- систему лайків на постах користувачів;
- перенесення системи на мобільні пристрої;
- підтвердження реєстрації листом на електронну пошту;
- блокування небажаних користувачів;
- власний плейліст;
- стрічка новин;
- оповіщення про отриманий лист;
- видалення певних листів;
- редагування вже відправлених листів;
- окрема сторінка для пошуку друзів;
- впровадження реклами;
- торговий майданчик для продажу автівок;
- реєстрація за допомогою іншої соціальної мережі;
- видалення відеозаписів зі сторінки;
- видалення окремих фотографій;
- видалення автівок, доданих користувачем.

Адаптація в гіперпросторі полягає в обмеженні видимості окремих елементів вузлів і можливостей навігації. У порівнянні з ІСМ інших типів, АНГС використовують спрощені моделі користувача та соціальної мережі. Підвищення їх ефективності можливо за рахунок застосування різних методів, зокрема при створенні бази знань про предметну область і про соціальну мережу.

З метою вирішення зазначеної проблеми пропонується формалізація структури гіперпростору шляхом виділення різних типів вузлів та зв'язків між ними залежно від їх дидактичного статусу. На цій основі побудовані моделі адаптивної навігації у соціальної мережі гіперпросторі, що дозволяють враховувати логічні залежності та специфічні характеристики елементів соціальної мережі, а також поточний і цільовий рівень вподобань користувача. У запропонованому підході передбачається керована навігація, згідно якої порядок

переходів визначається сценарієм подання інформації – послідовністю вузлів, яка враховує задану користувачем логіку представлення матеріалу і стан вподобані та інтересів користувача.

Сценарій будується на основі гіпертекстової бази знань і вибраної моделі навігації, а потім динамічно адаптується до змін моделі користувача.

2 ОПИС ПРОВЕДЕНИХ ТЕОРЕТИЧНИХ ДОСЛІДЖЕНЬ

В галузі розробки та застосування адаптивних гіпермедіа систем знайшли своє використання багато відомих технологій, а також нові і ті, що розроблюються. Виключно для систем адаптації розроблюються алгоритми, що перш за все враховують особливості надання інформації – середовище навчання (гіпермедіа-сторінки, Інтернет-ресурси, та інші). Але питання ефективності алгоритмів, що розроблюються постає все актуальніше, адже основна мета адаптивних навчальних систем – це допомога користувачеві знайти правильну інформацію у зручному для нього вигляді та у найшвидший час. Для досягнення мети адаптації система повинна мати інформацію для аналізу інтересів та вподобань користувача, історії його взаємодії з системою, будь-яку інформацію, до якої вона може адаптуватися. Окремим питанням постають види та варіанти надання інформації, вирішення на конкретному етапі питання релевантності наданої інформації, та інше. За своєю структурою адаптивні системи дуже різноманітні, від надзвичайно складних до більш простих, з меншою кількістю компонентів та параметрів, що враховуються при побудові соціальної мережі. Основними компонентами соціальних мереж є моделі користувача та предметної галузі, а також, в залежності від типу системи, база знань.

Такі дослідження перш за все пов'язані з соціальною мережею та впровадженням мультимедійних технологій. Відбуваються дослідження різних алгоритмів та технологій, триває розробка інструментальних програмних засобів.

Оскільки існує вже велика кількість алгоритмів адаптації, вони використовуються у сучасних системах, але, нажаль, не завжди вдається правильно обрати алгоритм, що буде давати найбільш ефективні результати адаптації системи до потреб соціальної мережі. Об'єктом дослідження є ефективність алгоритмів адаптації. Як предмет дослідження обрано алгоритми адаптації, їх аналіз та розробка. Тому завданням є як дослідження існуючих алгоритмів для АНС та можливостей сучасних мультимедіа систем у соціальній

мережі та аналіз методів, технологій та засобів адаптивної гіпермедіа, так і вибір алгоритму, що легко може бути інтегрований соціальної мережі, розробка архітектури системи, що призначена для дослідження ефективності алгоритмів, забезпечення можливості визначення різних видів навігаційних правил для моделі навчальної системи, розробка компонентів системи, що забезпечує адаптацію до параметрів суб'єкта соціальної мережі та іншої інформації, яка має значення для алгоритму адаптації, розробка моделі системи, що адаптована до параметрів користувача, структуру якої можна дослідити, дослідження можливостей розробленої системи, та методів використання.

Адаптивні гіпермедіа-системи (АГС) – це усі гіпермедіа-системи, які зберігають особливості користувача в моделі користувача і застосовують цю модель для адаптації до користувача різних візуальних аспектів системи. Іншими словами, система повинна задовольняти трьом критеріям: вона має бути гіпертекстовою або гіпермедійною, вона повинна мати модель користувача і вона повинна адаптувати свій гіпермедіа-простір, використовуючи цю модель.

Важливими елементами адаптивних гіпермедіа-систем є такі:

- модель предметної галузі – описує, яким чином інформація системи структурується і поєднується;
- модель користувача – описує те, яка інформація про користувача повинна зберігатися в системі. Це включає подання цільових для користувача знань, а також і інформацію про вже відвідані ним сторінки;
- модель викладання, або модель адаптації – містить педагогічні правила, які визначають, яким чином модель предметної області і модель користувача поєднуються для забезпечення поточної адаптації;
- механізм адаптації – безпосередньо виконує адаптацію через адаптування або динамічну генерацію контенту сторінок, а також налаштування адрес та типів посилань, щоб супроводжувати кожного користувача індивідуально.

Аналізуючи теперішній стан розвитку освітніх систем, зокрема в контексті безперервної освіти, можна зробити висновок про те, що системи соціальних мереж зайняли своє місце у світі інформаційних технологій та потребують все

більше уваги. Розробка нових технологій та покращення існуючих може суттєво змінити ставлення до соціальних мереж.

Зазначимо, що під адаптивними гіпермедіа-системами (АГС) ми розуміємо всі гіпертекстові і гіпермедіа-системи, які відображають в моделі користувача деякі особливості користувача в моделі користувача і застосовують цю модель для адаптації до користувача різних видимих аспектів системи.

Іншими словами, система повинна задовольняти трьом критеріям:

- бути гіпертекстовою або гіпермедіа-системою;
- мати модель користувача;
- бути здатною адаптувати гіпермедіа, використовуючи цю модель (тобто одна і та ж система може виглядати по-різному для користувачів з різними моделями).

Застосування адаптивної гіпермедіа доцільно в тих випадках, коли система призначена для обслуговування великої групи користувачів з різними цілями, знаннями та досвідом і коли лежить в основі досить великий гіперпростір. Користувачі з різними цілями і знаннями можуть цікавитися різними частинами наданої на гіпермедіа-сторінці інформації і можуть використовувати різні посилання для навігації.

Розрізняють такі типи адаптації систем:

- адаптовані (пристосовані) гіпермедіа-системи – системи, в яких адаптація надана системі самим розробником після фази тестування; у цьому випадку адаптація не може бути коректною для кожного індивідуального користувача;
- гіпермедіа-системи, що адаптуються – системи, які можуть модифікуватися тільки за явною вимогою користувача; системи, що адаптуються дозволяють користувачу явно встановлювати переваги або надають профіль за допомогою заповнення форми; вся інформація, надана користувачем, зберігається в моделі користувача, яка модифікується тільки за його явним запитом; надання інформації потім адаптується до цієї моделі; деякі системи можуть мати дуже складні моделі користувача, в той час як інші розрізняють лише декілька стереотипних користувачів типу «початківця», «середнього» і «експерта»;

- адаптивні гіпермедіа-системи – системи, які самі можуть адаптуватися до потреб користувача; адаптивні системи формують модель користувача, відстежуючи навігацію користувача у інформаційному просторі, а також за допомогою тестів в системах навчання; надання матеріалів адаптується до моделі користувача, і модель користувача постійно оновлюється, у міру того як він переглядає інформацію.

Аналіз існуючих АГС дозволяє виділити п'ять видів гіпермедіа-систем, які використовуються в даний час як прикладні галузі у більшості науково-дослідних проектів з адаптивної гіпермедіа:

- гіпермедіа-соціальної мережі;
- мережеві інформаційні системи;
- інформаційно-пошукові системи;
- мережеві довідкові системи;
- системи для керування індивідуалізованим поданням.

Найбільш популярна область дослідження для адаптивної гіпермедіа – гіпермедіа-соціальної мережі. Існуючі гіпермедіа-соціальної мережі мають відносно невеликий гіперпростір подання окремої інформації. Метою користувача зазвичай є вивчення всього матеріалу або значної його частини. Форма гіпермедіа підтримує кероване користувачем оволодіння соціальної мережі. Найбільш важлива характеристика користувача в гіпермедіа- соціальної. Методи адаптивної гіпермедіа можуть бути корисні для вирішення ряду проблем, пов'язаних з використанням гіпермедіа-соціальної мережі. По-перше, знання різних користувачів можуть достатньо відрізнятися, і знання окремого користувача може дуже швидко рости.

Розглянемо навчання як процес керування складною системою, в якій користувач є об'єктом керування, а ІНС – джерелом керування. В модель адаптивного керування навчанням включені:

- об'єкт керування (ОК) – користувач;
- середовище, в якому здійснюється функціонування системи;
- пристрій адаптивного керування (ПАК);

- бази ресурсів системи;
- модель користувача.

Модель користувача дозволяє врахувати не тільки поточний стан об'єкта керування, але і передісторію навчання, що зберігається у вигляді набору параметрів моделі ОК (модель користувача).

Тут $X=(x_1, x_2, x_3, \dots, x_N)^T$, – вектор, що визначає фактичний стан середовища, що впливає на вектор стану $Y=(y_1, y_2, y_3, \dots, y_M)^T$ об'єкта; m – операція транспонування векторів. Інформація про ці стани вимірюється датчиками D_X і D_Y , які інформують пристрій адаптивного керування.

Вочевидь, що вектор вимірюваних датчиком D_X параметрів станів середовища $X'=(x'_1, x'_2, x'_3, \dots, x'_n)^T$ не дорівнює вектору його фактичних станів ($X' \neq X$), а вектор вимірюваних датчиком D_Y параметрів станів об'єкта керування $Y'=(y'_1, y'_2, y'_3, \dots, y'_m)^T$ не дорівнює вектору його фактичних станів ($Y' \neq Y$) у силу того, що датчики вимірюють тільки те, що використовується в процесі керування. Однак $X' \in X$ і $Y' \in Y$, тобто одержувана інформація деяким чином (але далеко не повністю) відбиває дійсний стан об'єкта й середовища, що доповнюється параметрами $\Omega=(\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_p)^T$ моделі учня.

Ресурси R , що виділяються на створення системи адаптивного керування, значною мірою визначають і обсяг інформації X' і Y' , що збирається датчиками D_X і D_Y . Пристрій адаптивного керування, отримуючи на вході інформацію про середовище X , об'єкт Y , мету Z^* і ресурси системи R (база даних соціальної мереж, методів її обробки, правил і шаблонів, ситуацій, а також функціональні модулі соціальної мережі), видає на виході керуючий вплив $U=(u'_1, u'_2, \dots, u'_k)^T$, за допомогою якого можливо досягти мети Z^* і перевести об'єкт в шуканий стан Y^* , відповідно до виконання заданої мети Z^* в рамках ресурсів $R=(r'_1, r'_2, r'_3, \dots, r'_p)^T$, тобто $\langle X', Y', Z^*, \rangle \rightarrow U^* \rightarrow Y^*$.

Результатом розв'язання задачі може бути: «вірно», «неправильно». Кожен терм описується у своїй функції приналежності $\mu_i(y)$, яка може приймати значення від 0 до 1. На основі значень параметрів стану об'єкта керування y'_i , в блоці фазифікації обчислюються значення $\mu_i(y)$ кожного терму. Результатом

застосування правила є величина, яка є ступенем істинності, вона може приймати значення від 0 до 1 ($0 \leq \mu_i(y) \leq 1$).

На другому етапі виконується активізація висновків правил нечіткої логіки. На підставі аналізу ступеня істинності висновків, в залежності від значень функції приналежності $\mu_i(y)$ вибирається те чи інше правило. Наприклад, було встановлено, що користувач «знає соціальну мережу», але «не зовсім». У цьому випадку система збирає об'єкти з соціальної мережі матеріалом, який недостатньо добре засвоєний користувачем. У випадку якщо встановлено, що користувач «не знає навчальний матеріал», то ІНС збирає з об'єктів НМІ необхідний користувач НММ. Але ця дія виконується після акумуляції.

На третьому етапі виконується акумуляція висновків для кожної лінгвістичної змінної. Висновки з кожного правила збираються разом для кожної лінгвістичної змінної та обробляються спільно.

На четвертому етапі виконується дефазифікація, в результаті якої відповідно до користувачів істинності для кожного терма вихідної змінної розраховується її числове значення і формується керуючий вплив. Таким чином, в результаті роботи МКНЛ буде сформовано керуючий вплив $U = (u'_1, u'_2, \dots, u'_k)^T$.

Незалежно від обраного ПАК методу керування соціальної мережі, на підставі вимірюваних датчиками D_X і D_Y параметрів станів середовища $X' = (x'_1, x'_2, x'_3, \dots, x'_n)^T$ і об'єкта керування $Y' = (y'_1, y'_2, y'_3, \dots, y'_m)^T$ і параметрів моделі учня $\Omega = (\sigma_1, \sigma_2, \sigma_3, \dots, \sigma_p)^T$ формується керуючий вплив $U = (u'_1, u'_2, \dots, u'_k)^T$. У критичних випадках, при виникненні нерозв'язних ситуацій, коли керування $U = (u'_1, u'_2, \dots, u'_k)^T$.

Структурована соціальної мережі інформація, що зберігається в базі даних НМІ у вигляді об'єктів «нульового» рівня O_0 , збирається ІНС в об'єкти «високого» рівня O_i , що представляють собою систематизований і впорядкований НММ. Цей НММ ще не призначений безпосередньо для роботи користувача, оскільки не містить зовнішнього оформлення і засобів навігації, але використовується для подальшої обробки в ІНС. Об'єкти «високого» рівня використовуються також як заготовки для компіляції електронних документів з

соціальними мережами для їх передачі іншим користувачам в різних форматах представлення даних: гіпертекстовий документ (*.html), Microsoft Word (стандартний формат *.doc або універсальний формат текстових документів *.rtf), Adobe Acrobat (формат *.pdf), Windows help (формат *.chm).

Гіперпростір це зазвичай множина гіпертекстових статей – вузлів, між якими встановлені асоціативні гіпертекстові зв'язки, і використовується вільний порядок навігації шляхом переходів по цим зв'язкам. Адаптація в гіперпросторі полягає в обмеженні видимості окремих елементів вузлів і можливостей навігації. У порівнянні з ІНС інших типів, АНГС використовують спрощені моделі користувача та соціальної мережі. Підвищення їх ефективності можливо за рахунок застосування різних методів, зокрема при створенні бази даних про предметну область.

З метою вирішення зазначеної проблеми пропонується формалізація структури гіперпростору шляхом виділення різних типів вузлів та зв'язків між ними залежно від їх дидактичного статусу і від особливостей застосування в соціальній мережі. На цій основі побудовані моделі адаптивної навігації у соціальному гіперпросторі, що дозволяють враховувати логічні залежності та дидактичні характеристики елементів соціальної мережі, а також поточний і цільовий рівень знань користувача. У запропонованому підході передбачається керована навігація, згідно якої порядок переходів визначається сценарієм мережі – послідовністю вузлів, яка враховує задану викладачем логіку викладу матеріалу і стан знань користувача.

Сценарій будується на основі гіпертекстової бази знань і вибраної моделі навігації, а потім динамічно адаптується до змін моделі користувача.

У процесі автоматизованого навчання необхідна динамічна адаптація складу соціальної мережі до змін знань користувача шляхом зміни рівня складності та обсягу пропонованої навчальної та контролюючої інформації. Для реалізації такої адаптації потрібно розробити формальне представлення стану знань користувача і знань, викладених у вузлі навчального гіперпростору, а також представлення правил відбору навчальних матеріалів, відповідних стану знань

користувача. Навчальні матеріали гіперпростору повинні бути структуровані за темами й упорядковані за послідовністю вивчення, між ними є логічні, смислові, причинно-наслідкові та інші зв'язки. Формальне представлення знань системи про предметну галузь, що досліджується (про її об'єкти, процеси та закономірності і їх взаємозв'язки) назвемо предметними знаннями. До складу предметних знань входять також контрольні завдання і знання про методи їх виконання. Для автоматизованого планування процесу навчання необхідні метазнання – опис властивостей навчальних матеріалів, перелік необхідних попередніх знань і очікуваний ефект від навчання.

Для організації навчальних матеріалів в ІНС використовуються різні підходи: від повнотекстових баз даних або гіпермедіа-технологій до баз знань та експертних систем. Предметні знання в ІНС частіше представляються моделями на основі семантичних мереж або фреймів, що дозволяє істотно інтелектуалізувати обробку запитів і ведення діалогу з учнем. Проте створення БЗ до теперішнього часу є досить важким і дорогим процесом, що вимагає високої кваліфікації розробників.

Методи гіпертекстового моделювання предметної області є менш важкими. Відомо, що гіпертекстова технологія виникла як результат вивчення механізмів пам'яті і процесів створення текстових документів шляхом упорядкування та структуризації мережі ідей, що подаються автором гіпертексту. Ця технологія близька своїми цілями дослідження до теорій навчання та когнітивної психології і є найбільш прийнятною з існуючих засобів моделювання предметних знань, а модель гіпертексту розглядається як узагальнена модель знань в соціальній мережі. Предметні знання в АНГС представляються у вигляді множини взаємопов'язаних текстових фрагментів, що описують об'єкти предметної області.

На кожному кроці динамічно адаптивного навчання повинні враховуватися результати всіх попередніх кроків, представлені у вигляді історії навчання і моделі стану знань користувача. Методи керування процесом навчання (навігації) в ІНС представляються у вигляді методичних знань – формалізованого опису

методики викладання та оцінювання знань, яка регламентує порядок вивчення соціальної мережі та вимоги до підготовки користувача.

Таким чином, можна стверджувати, що підтримка процесу навігації у навчальному гіпертекстовому середовищі має свої особливості, що перебувають в необхідності врахування логічних зв'язків між вузлами для забезпечення принципу поступового вивчення і додаткового розрізнення вузлів і зв'язків по їх освітньому статусу. Тому статистичні підходи до планування послідовності вузлів гіперпростору, не можуть бути застосовані. Отже формальне представлення моделі користувача позначимо через λ , а множину всіх таких моделей, допустимих гіперпростором, – через Λ . Правило формування сценарію задамо за допомогою дидактичної функції – часткового мультівідображення $\pi(\lambda) : \lambda \rightarrow O$, де $\lambda \in \Lambda$, а O – множина елементів сценарію навчання, $O \subseteq F$. Ця функція встановлює відповідність між поточною моделлю користувача і множиною вузлів, які підлягають вивченню.

3 АНАЛІЗ РЕЗУЛЬТАТІВ ДОСЛІДЖЕННЯ

3.1 Опис вимог до програмного продукту

Соціальна мережа для власників автомобілів призначена для швидкого спілкування з користувачами, які мають автомобіль або, які бажають придбати «нове» авто. Також у майбутньому планується торговий майданчик інструментами та деталями для авто.

Перелічено вимоги, що залежать від специфіки серверної частини програми:

- інтерактивність - ряд додатків потрібно забезпечити механізмом нотифікацій - повідомити додатком (користувачеві) про настання певних подій, передати повідомлення користувачу;
- перегляд інформації інших користувачів системи;
- обробка інформації користувача, коли користувач буде додавати свої данні до системи, то серверна частина повинна обробити ці данні та передати їх до бази даних;
- додавання фото, відео файлів до системи, серверна частина повинна кодувати фото та відео файли, для меншого використання пам'яті бази даних. Це забезпечую швидку роботу системи;
- спілкування між користувачами системи;
- забезпечувати авторизацію доступу до даних для запобігання несанкціонованого доступу;

Система повинна складатися з клієнта і сервера. Клієнтом повинен виступати браузер з доступом до інтернету. Сервер має виконувати основні функцію працездатності сайту.

Клієнтська частина взаємодіє безпосередньо з користувачем, тому вона, насамперед, повинна мати привабливий та інтуїтивно зрозумілий інтерфейс. Також, окрім цього, інтерфейс має підлаштовуватися під різну ширину екрану ПК та бути доступним для основних браузерів.

3.2 Аналіз основних функцій системи

Основні функції системи мають бути такими:

- можливість створити власний акаунт у системі;
- можливість увійти до системи;
- перегляд профайла користувача;
- додавання або видалення відео файлів;
- додавання або видалення фото зображень;
- додавання або видалення аудіо файлів;
- додавання або видалення друзів;
- написання повідомлення іншому користувачеві.

Дана система, як будь-яка інша потребує обслуговування, тому її користувачі повинні дотримуватися деяких вимог.

До цих вимог належать такі:

- для використання системи необхідно підключення до мережі інтернет;
- користувачеві необхідно мати браузер.

4 ОПИС РОЗРОБЛЕНОГО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

4.1 UML проектування ПЗ

Під час моделювання програмного продукту були побудовані п'ять діаграм, що описують систему з різних сторін. Проаналізувавши кожну з них, можна отримати всю необхідну для розробки системи інформацію, тому що побудовані UML-діаграми охоплюють всі аспекти роботи і весь необхідний функціонал програмного продукту.

Концептуальною основою об'єктно-орієнтованого аналізу і проектування ПЗ (ООАП) є об'єктна модель. Її основні принципи (абстрагування, інкапсуляція, модульність і ієрархія) і поняття (об'єкт, клас, атрибут, операція, інтерфейс і ін.). Більшість сучасних методів ООАП засновані на використанні мови UML.

Уніфікована мова моделювання UML являє собою мову для визначення, уявлення, проектування і документування програмних систем, організаційно-економічних систем, технічних систем та інших систем різної природи. UML містить стандартний набір діаграм і нотацій самих різноманітних видів.

Для того щоб більш точно зрозуміти як повинна працювати система, все частіше використовується опис функціональності системи через варіанти використання (Use Case або прецеденти). Варіанти використання - це опис послідовності дій, які може здійснювати система у відповідь на зовнішні впливи користувачів або інших програмних систем.

Варіанти використання відображають функціональність системи з точки зору отримання відчутного результату для користувача, тому вони точніше дозволяють ранжувати функції за значимістю одержуваного результату.

Варіанти використання призначені в першу чергу для визначення функціональних вимог до системи і керують усім процесом розробки. Усе основні види діяльності такі як аналіз, проектування, тестування виконуються на основі варіантів використання. Під час аналізу і проектування варіанти використання дозволяють зрозуміти, як результати, які хоче отримати користувач впливають на

архітектуру системи і як повинні вести себе компоненти системи, для того щоб реалізувати потрібну для користувача функціональність.

В процесі тестування, описані раніше варіанти використання дозволяють простіше оцінити точність реалізації вимог користувачів і дозволяють провести покрокову перевірку цих вимог.

Для розробки моделей використовувалося програмне забезпечення “Draw.io”. Виходячи з наведених нижче даних можемо скласти діаграму прецедентів, що наведено на рисунку 4.1.

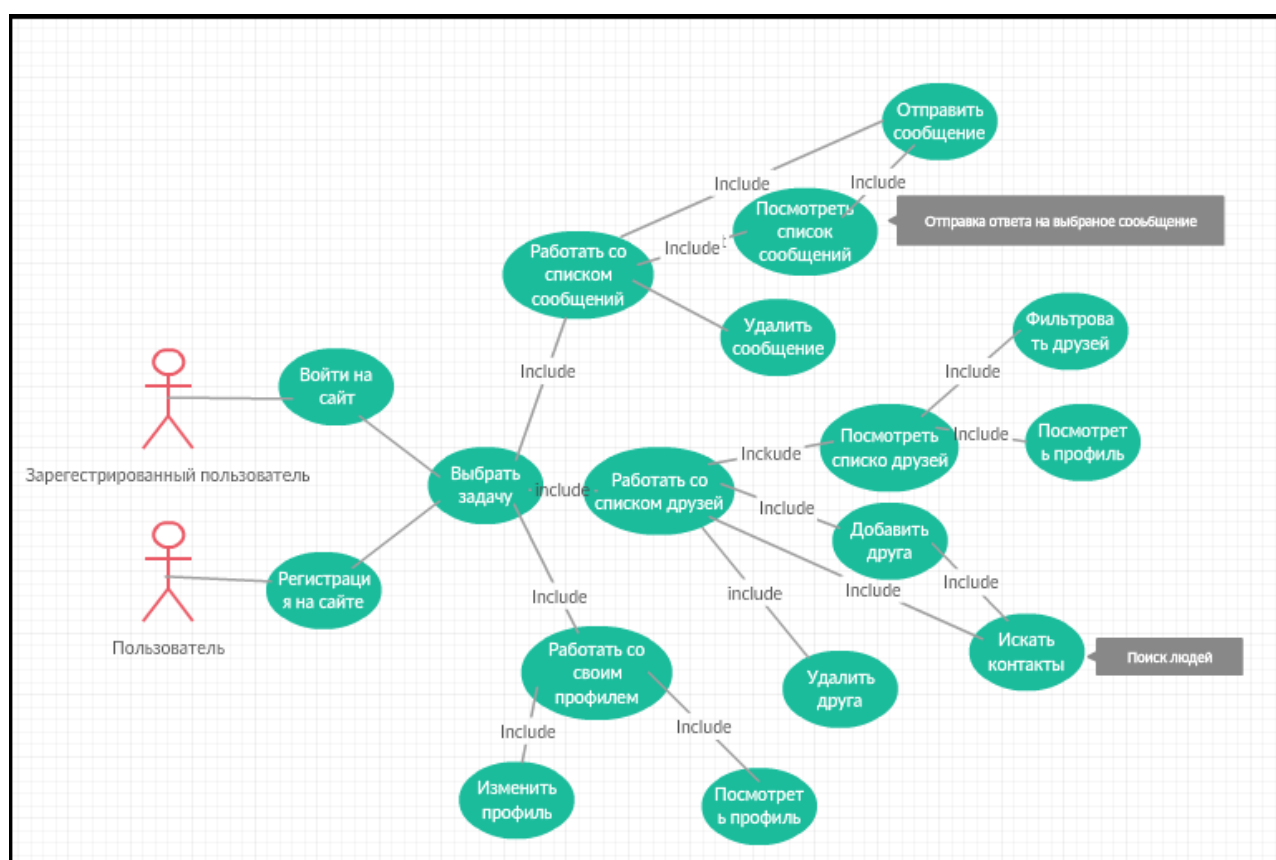


Рисунок 4.1 – Діаграма прецедентів

На рисунку 4.1 зображено два актори: гість (Користувач) і користувач (Зареєстрований користувач). Різниця між ними полягає в тому, що гість - звичайний відвідувач інформаційного веб-сайту, який має можливість лише переглянути інформацію про наданий продукт, оцінити його переваги і недоліки, спираючись виключно на інформацію, надану на сайті. Переглянувши її, людина

приймає рішення, наскільки цей предмет Інтернету речей поліпшить його життя, і чи потрібно йому ставати іншим актором, тобто користувачем.

Другий актор, користувач – людина, яка заєрестувалася у системи для того, щоб зроби свій власний профіль и спілкуватися з іншими заєрестрованими користувачами соц. мережи.

Побудуємо діаграму класів (рис. 4.2) – статичну структурну діаграму, яка описує структуру системи, демонструє класи системи, їх атрибути, методи і залежності. Діаграма класів – статичне представлення структури моделі. Відображає статичні (декларативні) елементи, такі як: класи, типи даних, їх зміст та відношення

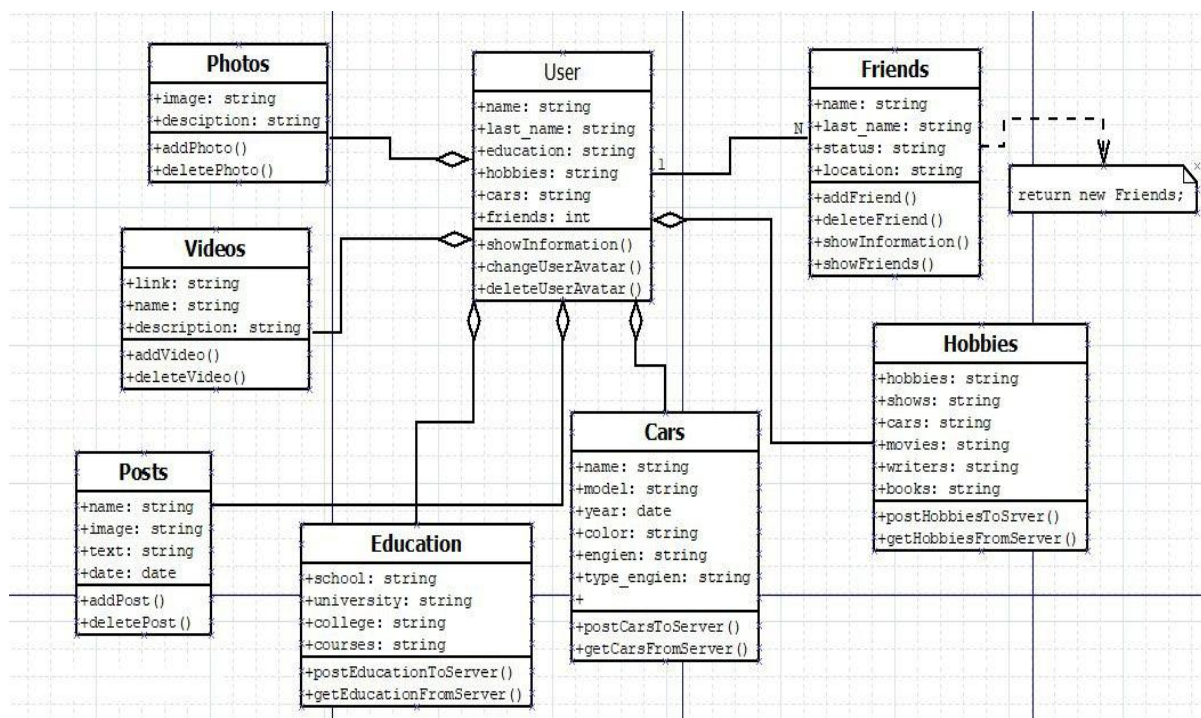


Рисунок 4.2 – Діаграма класів

На рисунку зображені класи сутностей БД і класи-регулятори до кожного з них. Сутності включають в себе перелік полів і конструкторів для себе, а регулятори – методи регулювання бази даних з використанням наданих сутностей.

Побудуємо діаграму об'єктів (див. рис. 4.3) – статичну діаграму, яка демонструє повний або частковий знімок модельованої системи в заданий момент часу і відображає об'єкти класів і їх зв'язку.

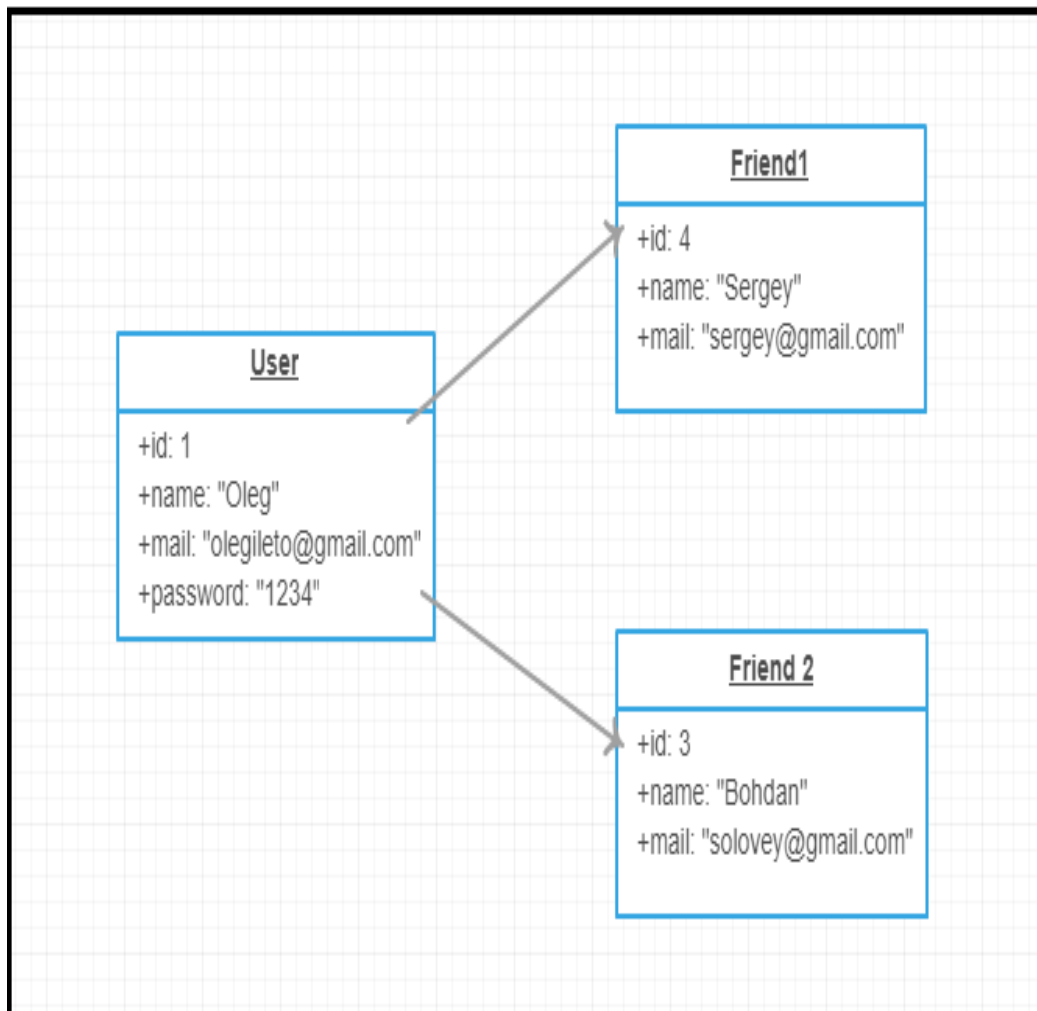


Рисунок 4.3 – Діаграма об'єктів

Побудуємо діаграму послідовностей (рис. 4.4) – діаграму послідовностей, яка зображує впорядковану в часі взаємодію об'єктів в динаміці і послідовні повідомлення, якими вони об'єднані. А саме взаємодії між користувачами щодо додавання один одного у друзі, а також повідомлення щодо цієї дії і, власне, додавання або зміну інформації, яка буде відображатися для всіх, хто зайде на сторінку певного користувача.

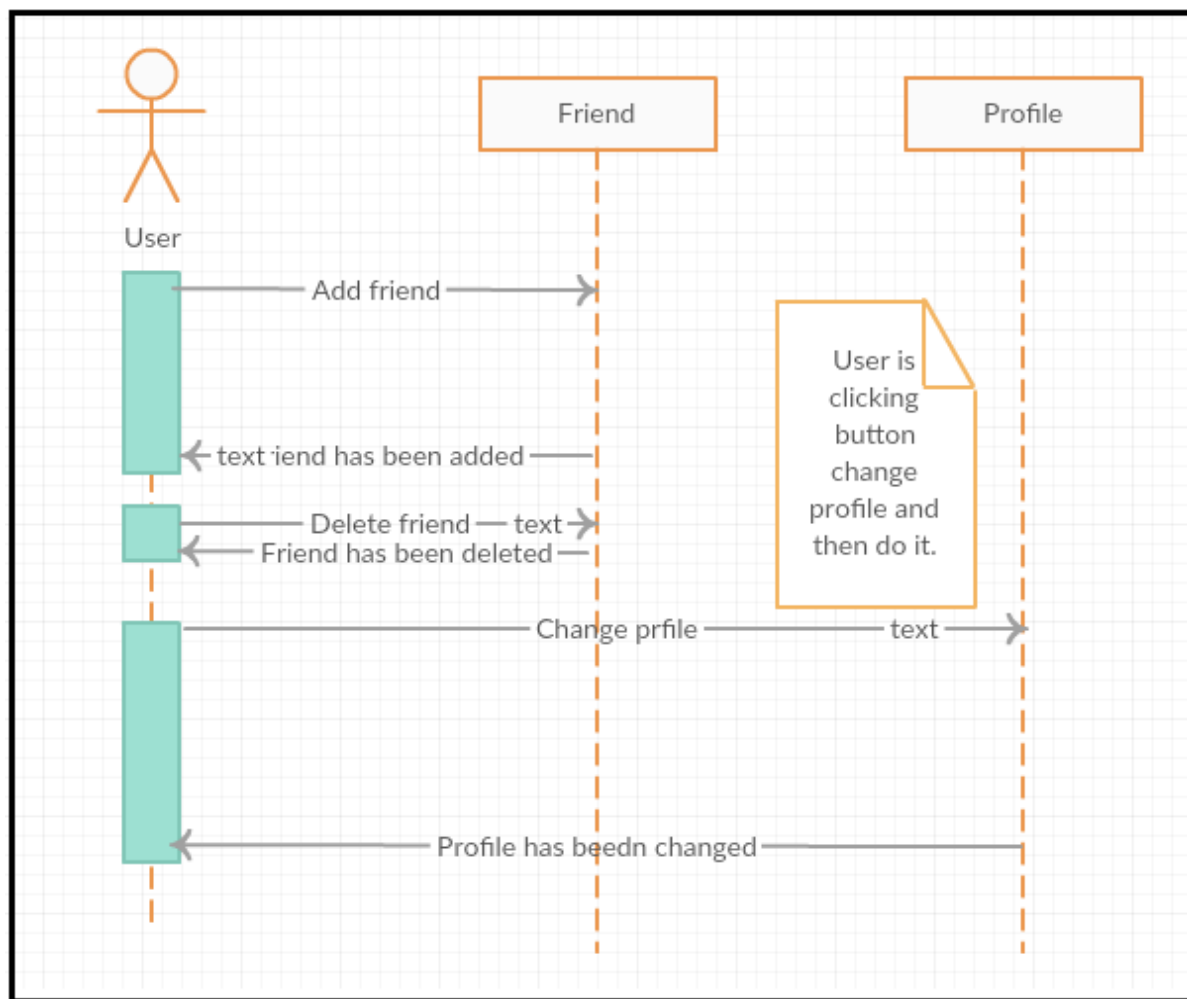


Рисунок 4.4 – Діаграма послідовностей

На діаграмі можна побачити користувача, що робить запит до серверу щодо додавання або видалення друга. Після того, як користувач додав певну людину в друзі, ця людина отримає повідомлення на власній сторінці з інформацією про додавання її. Це дозволить швидше та зручніше знаходити друзів або заводити нові знайомства в соціальній мережі. Також діаграма ілюструє принцип зміни або додавання інформації у профілі. Користувачу достатньо перейти в певну вкладку та заповнити необхідні йому поля.

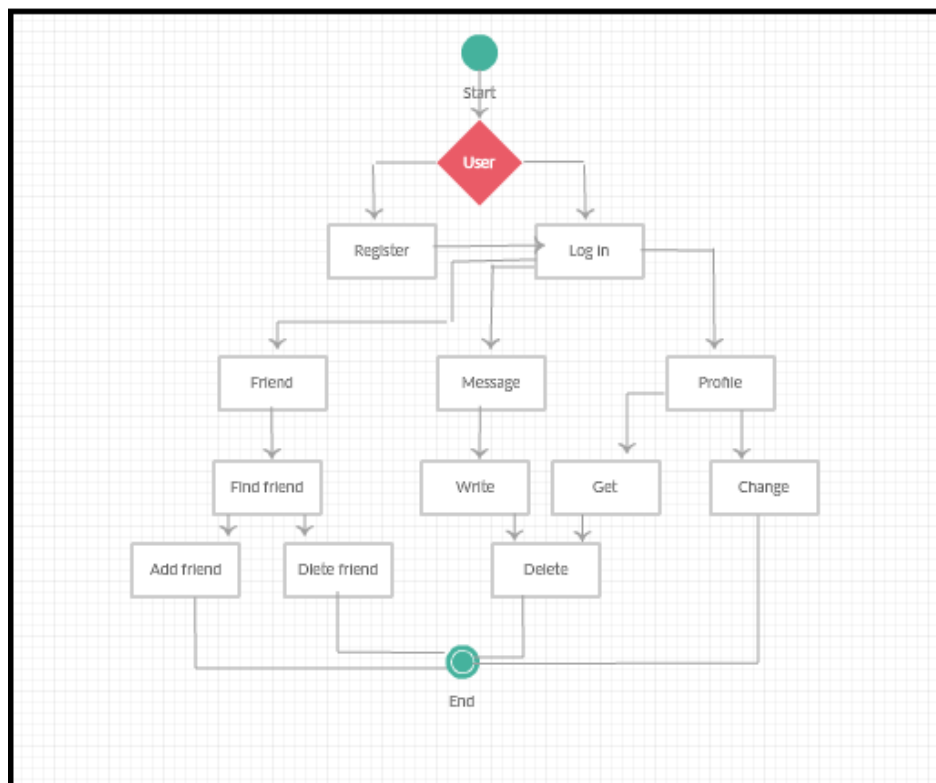


Рисунок 4.5 – Діаграма станів

На діаграмі станів зображено, як користувач може робити свої дії. Якщо користувач не має власного аккаунту, тоді йому потрібно його створити. А якщо користувач вже має свій аккаунт, тоді він може переходити з одного стану на інший. Користувач переходить на стан «Друзі», там він може переходити на стан «Знайти друга», а далі «Добавити друга» чи «Видалити друга», після цього можете завершити свої дії. Переходячи на стан «Повідомлення» користувач має можливість отримати повідомлення від іншого користувача чи написати іншому користувачеві у відповідь. На стані «Профіль» користувач має можливість повністю змінити свій профіль так також завершити свої дії (рис. 4.5).

4.2 Проектування архітектури ПЗ

Для реалізації проекту було вирішено використовувати трирівневу архітектуру. В якості серверної мови програмування було обрано мову PHP. PHP

найбільш широко використовується саме для створення скриптів для виконання на стороні сервера. Все, що вам знадобиться, це парсер PHP (у вигляді програми CGI або серверного модуля), вебсервер і браузер. Щоб Ви могли переглядати результати виконання PHP-скриптів в браузері, вам потрібен працюючий вебсервер і встановлений PHP.

Трирівнева архітектура – архітектурна модель програмного комплексу, що передбачає наявність у ньому трьох компонентів: клієнта, сервера додатків (до якого підключено клієнтську програму) і сервера баз даних (з яким працює сервер додатків).

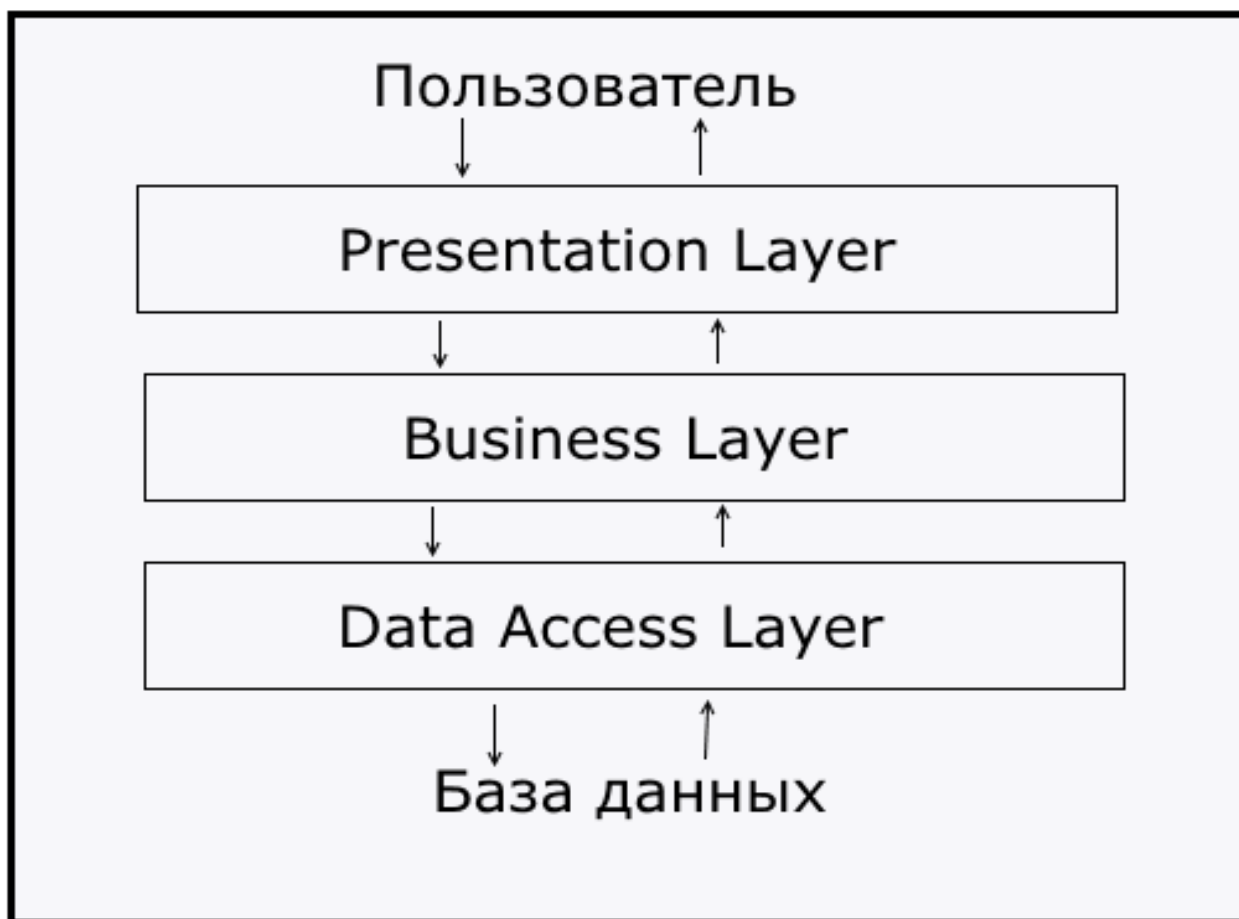


Рисунок 4.6 – Схематичне зображення трирівневої архітектури

Як видно з рисунку 4.6, трирівнева архітектура складається з таких компонентів:

Клієнт – це інтерфейсний (зазвичай графічний) компонент, який представляє перший рівень, власне застосунок для кінцевого користувача. На

перший рівень може бути винесена і зазвичай виноситься найпростіша бізнес-логіка: інтерфейс авторизації, алгоритми шифрування, перевірка значень, що вводяться, на допустимість і відповідність формату, нескладні операції (сортування, групування, підрахунок значень) з даними, вже завантаженими на термінал.

Сервер застосунків розташовується на другому рівні. На другому рівні зосереджена більша частина бізнес-логіки. Поза ним залишаються такі фрагменти, що експортуються на термінали, а також розміщені в третьому рівні збережені процедури і тригери.

Сервер бази даних забезпечує зберігання даних і виноситься на третій рівень. Зазвичай це стандартна реляційна або об'єктно-орієнтована СУБД. Якщо третій рівень являє собою базу даних разом з збереженими процедурами, тригерами і схемою, яка описує застосунок в термінах реляційної моделі, то другий рівень будується як програмний інтерфейс, що зв'язує клієнтські компоненти з прикладною логікою бази даних.

У простій конфігурації фізично сервер застосунків може бути поєднаний з сервером бази даних на одному комп'ютері, до якого по мережі підключається один або декілька терміналів.

У «правильної» (з точки зору безпеки, надійності, масштабування) конфігурації сервер бази даних міститься на виділеному комп'ютері (або кластері), до якого по мережі підключені один або кілька серверів застосунків, до яких, в свою чергу, по мережі підключаються термінали.

4.3 Проектування бази даних

Для зберігання інформації була вибрана СУБД MySQL. А також ORM бібліотека RedBean, яка працює, власно, з bean'ами. Для початку ви можете просто створити bean. Це об'єкт з публічними властивостями і типом. Тип bean'a

використовується для зіставлення таблиці в БД і, опціонально, для вказівки вашого класу. Bean створюється таким чином:

```
$book = R::dispense( 'book' );
```

Тепер у нас є порожній bean типу book. Задамо йому пару властивостей:

```
$Book ->title = 'Boost development with RedBeanPHP';
$Book->author = 'Charles Xavier';
```

І збережемо в БД:

```
$Id = R::store($ book);
```

Перед створенням бази даних слід визначити основні сутності та їх атрибути. Основний сутністю є користувачі і йому відповідають такі атрибути:

- id - унікальний ідентифікатор користувача;
- name - ім'я користувача;
- last_name - прізвище користувача;
- password - пароль користувача;
- email - адрес пошти користувача;
- birthday - дата народження;
- avatar - головна фотографія користувача;
- background - зображення заднього фону профіля;
- status – статус користувача;
- is_married – сімейний стан користувача;
- info – коротка інформація щодо користувача;
- occupation - інформація щодо освіти користувача;
- number – номер мобільного телефону.

Схема бази даних наведена на рис. 4.7.

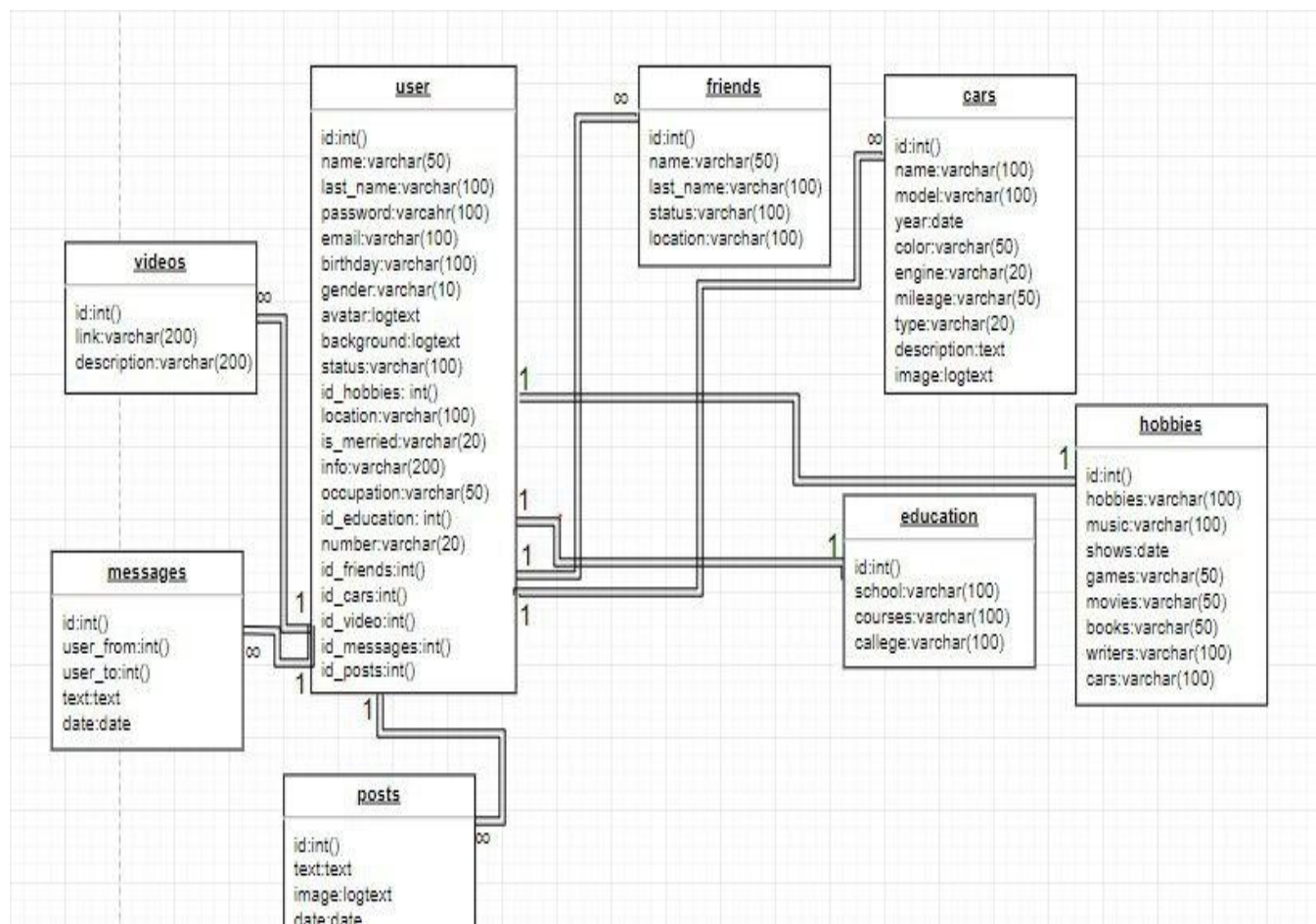


Рисунок 4.7 – Схема бази даних

Також база має такі таблиці як:

- cars – інформація щодо авто власника;
- education – інформація щодо освіти певного користувача;
- friends – інформація щодо відносин між користувачами;
- hobbies – інформація про хоббі користувача;
- messages – листи користувачів;
- photos – фото користувачів;
- posts – пости користувачів;
- videos – відеозаписи користувачів.

4.4 Приклади найцікавіших алгоритмів та методів

Одним з цікавих методів системи є створення постів. За допомогою цього методу користувач може залижати повідомлення як на своїй сторінці, так і на сторінках інших користувачів. Повідомлення окрім тексту можуть мати в собі певні зображення, завантажені з персонального комп'ютера. Ці повідомлення мають змогу бачити всі користувачі соціальної мережі, котрі зайшли на певну сторінку. Людина, на сторінці якої розміщують пости, може видаляти їх на свій розсуд, а також розміщувати свої власні пости у своєму профілі.

Основні дані, за допомогою яких пост додається до бази даних це:

- `id_user` – унікальний ідентифікатор сторінки поста;
- `id_writer` – унікальний ідентифікатор користувача, що написав пост;
- `text` – текстовий зміст посту;
- `image` – додане до посту зображення;
- `date` – дата створення посту.

Нижче наведено лістинг коду, що реалізує створення поста:

```
$post = R::dispense('posts');
$data = $_POST;
if (isset($_GET['id'])) {
    $post->id_user_page = $_GET['id'];
} else {
    $post->id_user_page = $_SESSION['logged_user']->id;
}
$post->id_writer = $_SESSION['logged_user']->id;
if (isset($data['post_text'])) {
    $post->text = $data['post_text'];
} else {
    $post->text = "";
}
$post->date = date("Y-m-d H:i:s");
if (getimagesize($_FILES['post_image']['tmp_name']) == false) {
    $post->image = "";
} else {
```



```

$image = addslashes($_FILES['post_image']['tmp_name']);
$image = file_get_contents($image);
$image = base64_encode($image);
$post->image = $image;
}
R::store($post);

```

4.5 Проектування UI/UX дизайну системи

Оскільки дана система розрахована на великий круг користувачів, то дизайн її користувацького інтерфейсу повинен бути доступним та зрозумілим кожному.

Проектуючи інтерфейс багато уваги приділялося саме зручності системи. Так як в системі існує багато різних розподілень за пріоритетністю та родом занять, було вирішено виділити основні кольори для різних категорій. Та вести ці кольори в дизайн застосунку.

Для користуванням соціальною мережою потрібно зробити свій персональний акаунт. На сторінці Log in можна створити новий акаунт (рис. 4.8) або увійти вже від існуючим акаунтом (рис. 4.9)

Рисунок 4.8 – Сторінка реєстрації

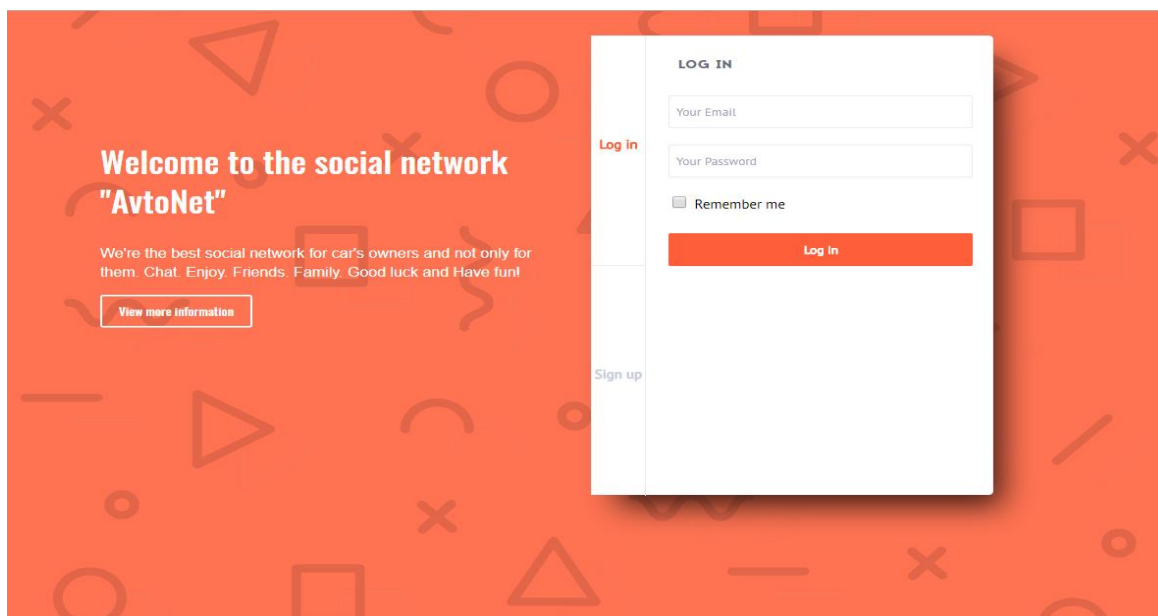


Рисунок 4.9 – Сторінка входу

Коли користувач увійшов до свого акаунту на сайті, то він бачить перед собою головну сторінку соц. мережі (Профіль користувача). На цій сторінці користувач може переходити на інші сторінки сайту завдяки навігаційній панелі, яка знаходиться у лівій частині екрану. Також існує панель на якій можна подивитися друзів які зараз на сайті, ця панель знаходиться у правій частині екрану. На сторінці користувача можна побачити усіх друзів, які користувач має, подивитися відео, фото, а також усі автомобілі, які користувач додавав для перегляду (рис. 4.10).

У шапці на сторінці користувача можна створити пошук по друзям, побачити повідомлення, які були надіслані користувачеві, а так маніпулювати своїм акаунтом (рис. 4.11).

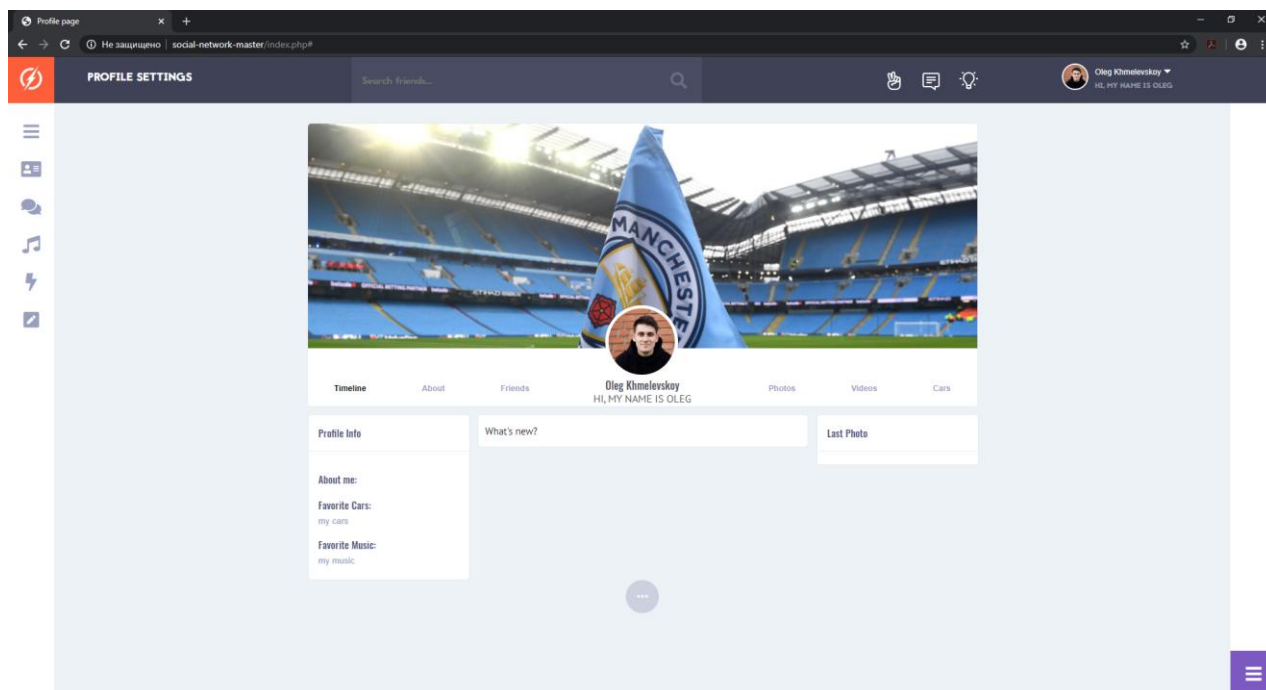


Рисунок 4.10 – Сторінка користувача

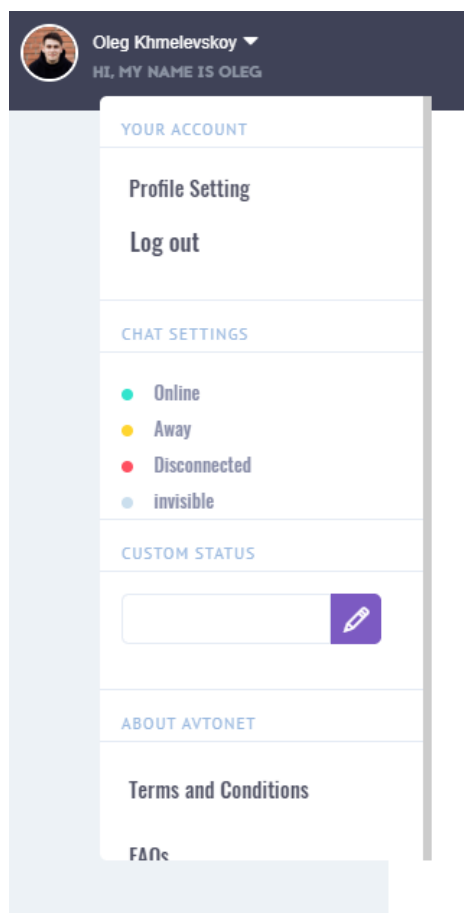


Рисунок 4.11 – Маніпуляція аккаунтом

На сторінці налаштувань користувач може повністю змінити свій акаунт (рис. 4.12), а саме:

- Змінити повністю інформацію користувача;
- Змінити свій пароль, імейл або мобільний телефон.

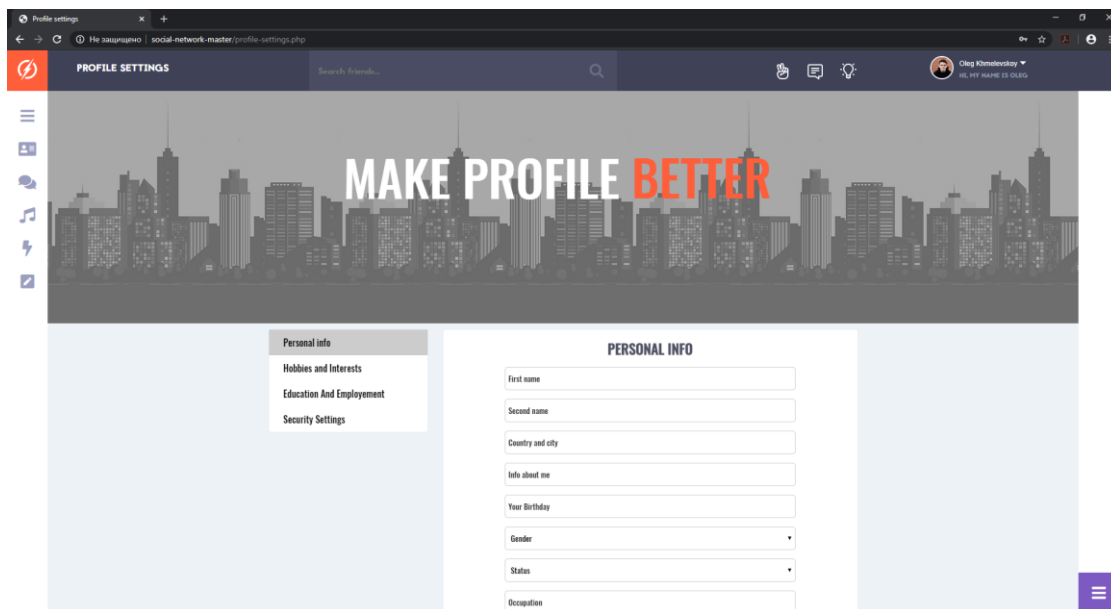


Рисунок 4.12 – Сторінка налаштувань

На сторінці повідомлень користувач може спілкуватися з іншими користувачами соц. мережі, які знаходяться у користувача у друзях (рис. 4.13).

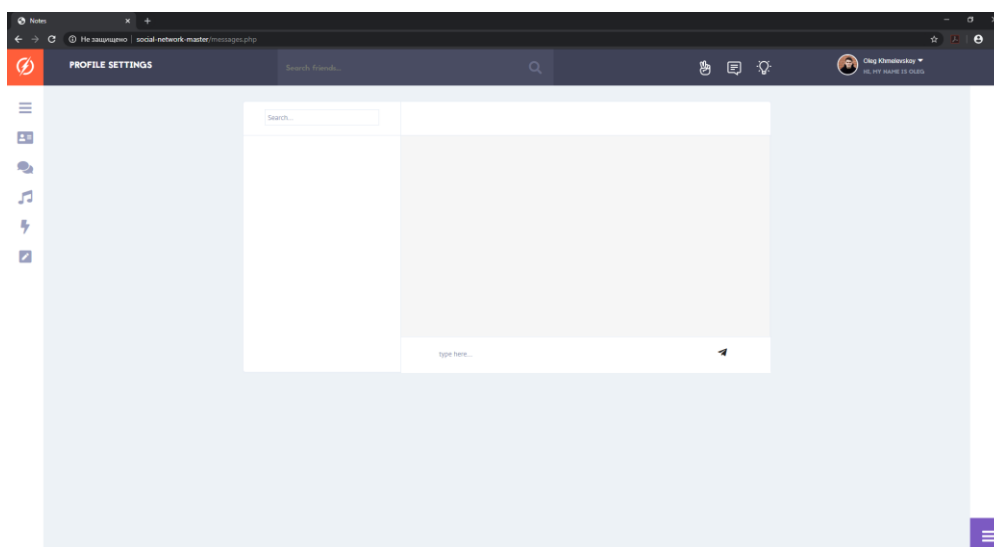


Рисунок 4.13 – Сторінка повідомлень

На сторінці друзів користувач може переглядати доданих друзів та переходити до їх сторінок.(рис. 4.14).

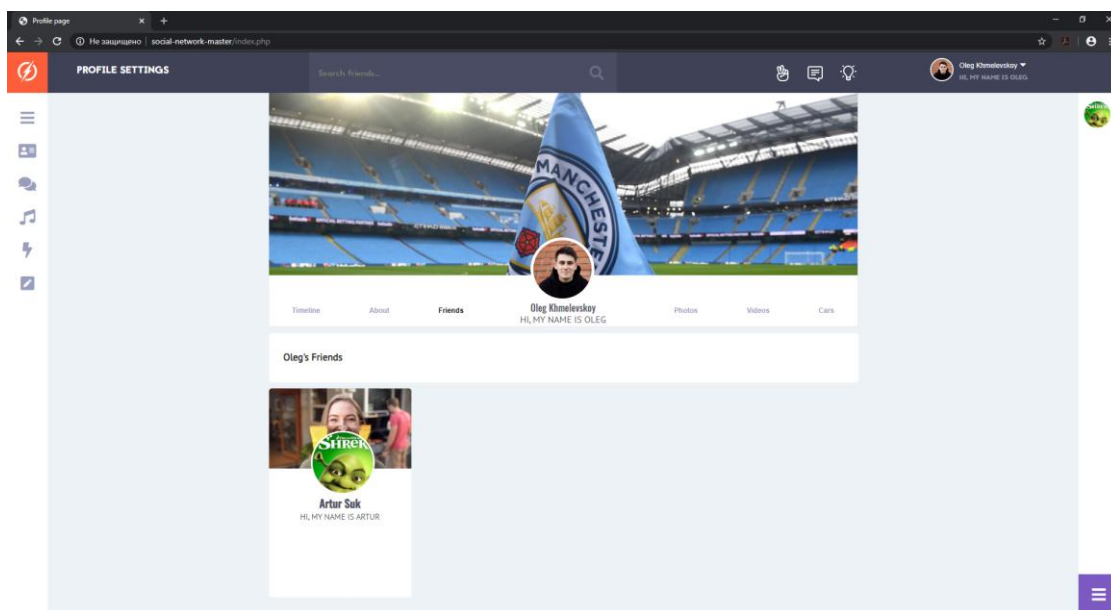


Рисунок 4.14 – Сторінка друзів

На рисунку 4.14 можна побачити друзів, доданих користувачем, що залогінівся. Щоб додати певну людину у друзі користувачу не потрібно відправляти йому запит та чекати на підтвердження. Для того щоб передивидись доданих друзів (це також можливо на сторінках інших користувачів) необхідно натиснути на певну вкладку профілю. Подорожуючи між вкладками ми можемо передивлятись файли, додані користувачем, такими як відео, власні фото або ж фото автівок, а також інформацію, яку користувач додав до своєї сторінки, від дати народження до освіти та особистої інформації.

5 ОПИС ПРОГРАМНОЇ РЕАЛІЗАЦІЇ

5.1 Вибір технологій розробки

Додаток було реалізовано мовами: JavaScript (клієнська частина), PHP (серверна частина).

Model-view-controller (MVC, «модель-уявлення-контролер», «модель-вид-контролер») – схема використання декількох шаблонів проектування, наведене на рисунку 5.1, за допомогою яких модель додатки, призначений для користувача інтерфейс і взаємодія з користувачем розділені на три окремих компонента таким чином, щоб модифікація одного з компонентів надавала мінімальний вплив на інші. Дана схема проектування часто використовується для побудови архітектурного каркаса, коли переходять від теорії до реалізації в конкретній предметній області. Основна частина програмної системи, сервер - написана з використанням шаблону MVC.

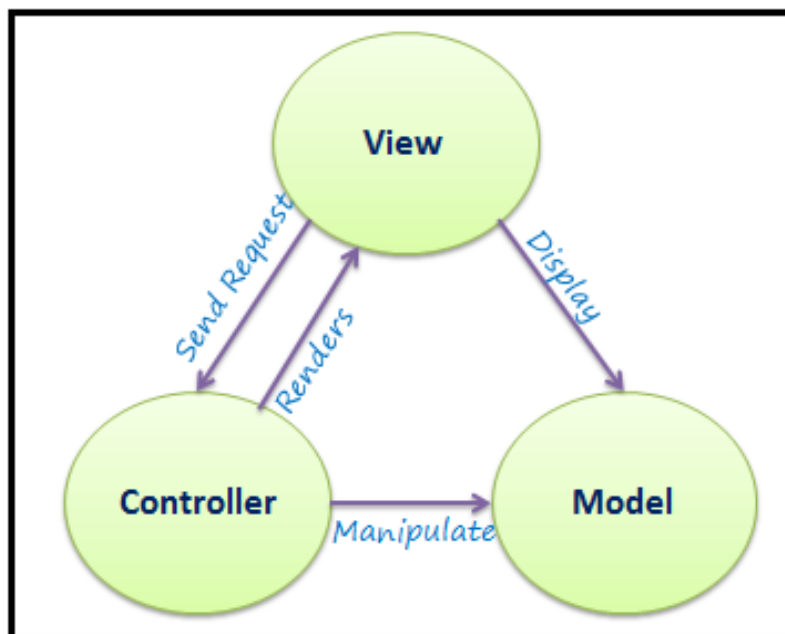


Рисунок 5.1 – Шаблон проектування MVC

5.2 Програмна реалізація

Сервер написаний на PHP з використанням MySQL мовою запитів та PHPMyAdmin для реалізації самої бази даних. У проекті існує своя схема структури проекту (рис. 5.2).

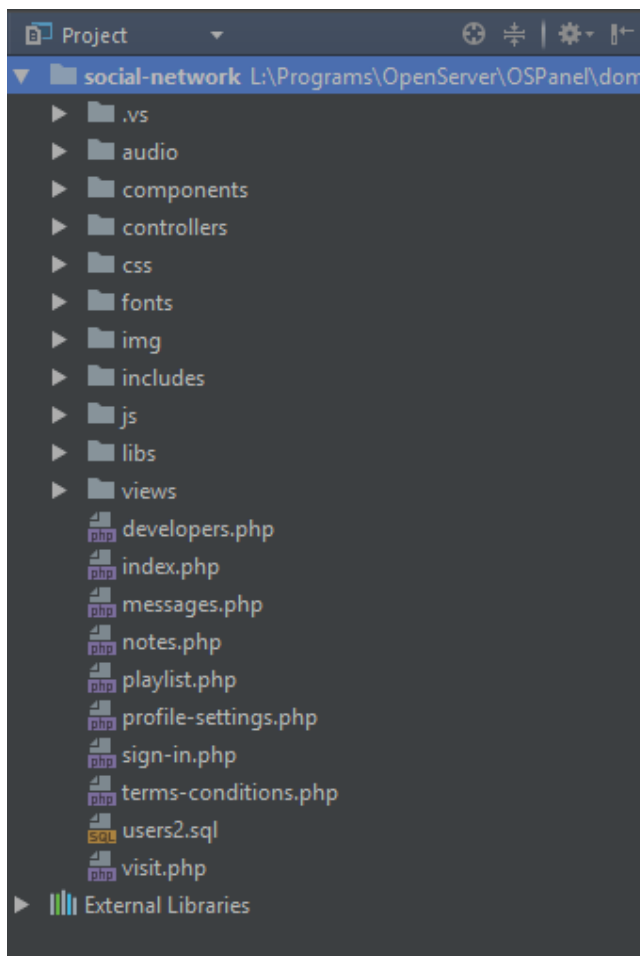


Рисунок 5.2 – Схема структури проекту

Користувач може залогінитись та переглянути або додати свої фотографії, відео матеріали, а також змінити інформацію, розташовану в його власному профілі, додати інформацію щодо автівок, список друзів, інформацію про навчання, тощо. На рисунку 5.3 можна побачити як переглянути основну інформацію про користувача.

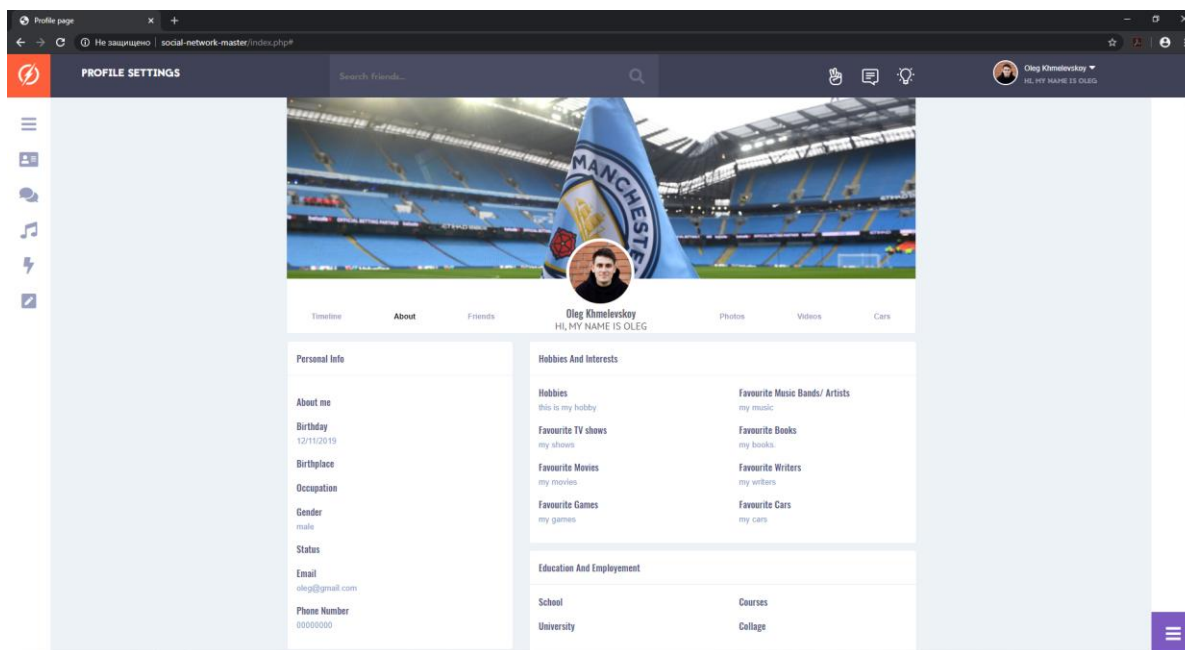


Рисунок 5.3 – Інформація про користувача

На рисунку 5.4 ми можемо побачити додані користувачем відеоролики.

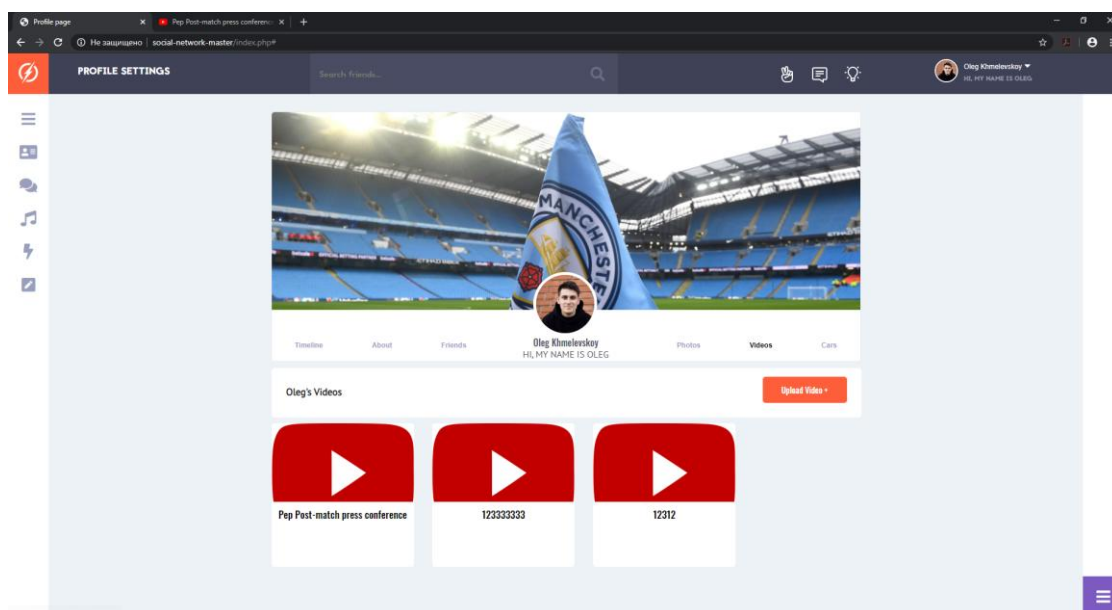


Рисунок 5.4 – Перегляд відео

Щоб додати власне відео користувач повинен натиснути на певну кнопку та додати посилання та назву або короткий опис ролика.

Наступний лістинг коду ілюструє цей процес:

```
$data = $_POST;
```



```

if (isset($_data['send_video']))
{
    $video = R::dispense('videos');
    $video->id_user = $_SESSION['logged_user']->id;
    $video->link = $_data['video_link'];
    $video->description = $_data['video_description'];
    R::store($video);
}

```

На рисунку 5.5 можна побачити, як виглядає вікно додавання відеозапису.

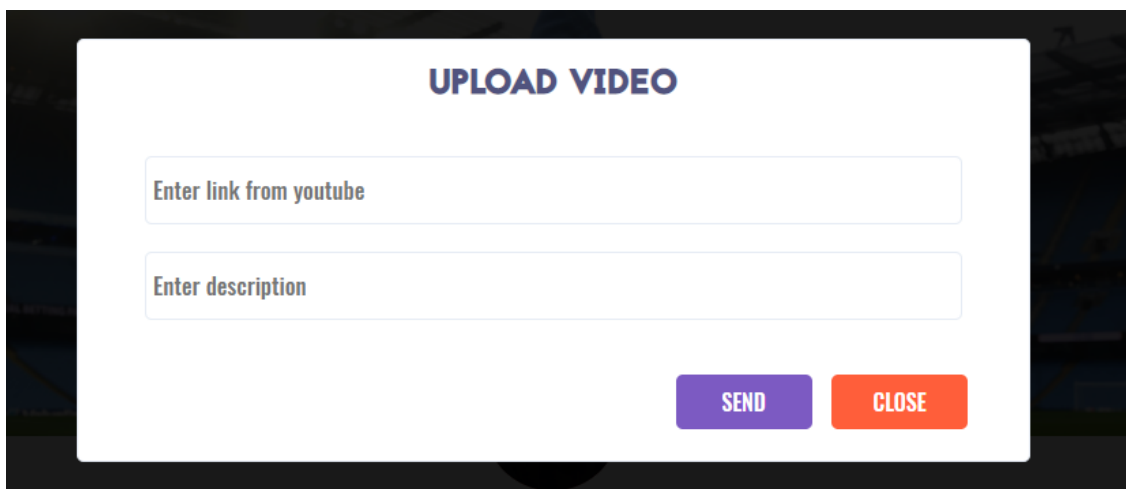


Рисунок 5.5 — Вікно додавання відеоряду

Також користувач може додавати фотографії власних автівок. Наступний лістинг коду ілюструє цей процес:

```

if (isset($_POST['send_car']))
{
    addCar();
}
function addCar() {
    $data = $_POST;
    $car = R::dispense("cars");
    $car->id_user = $_SESSION['logged_user']->id;
    $car->name = $data['car_name'];
    $car->model = $data['car_model'];
    $car->year = $data['car_year'];
    $car->color = $data['car_color'];
    $car->engine = $data['car_engine'];
}

```

```

$car->mileage = $data['car_mileage'];
$car->type = $data['car_type'];
$car->description = $data['car_descrtiption'];
$carImage = addslashes($_FILES['car_image']['tmp_name']);
$carImage = file_get_contents($carImage);
$carImage = base64_encode($carImage);
$car->image = $carImage;
R::store($car);
}

```

На рисунку 5.6 ми бачимо вікно додавання автівки користувачем.

Рисунок 5.6 – Додавання автівки

Для того, щоб відправити повідомлення нам необхідно знати унікальні ідентифікатори двох користувачів, за допомогою глобальної змінної `$_SESSION` ми зберігаємо ідентифікатор користувача, що збирається відправити повідомлення та додаємо його у таблицю бази даних, якщо повідомлення створено. Ідентифікатор профілю, якому ми доставляємо повідомлення ми беремо за допомогою методу `$_GET`.

Наступний лістинг коду ілюструє створення повідомлення:

```

$message = R::dispense('messages');

```

```

$message->user_from = $_SESSION['logged_user']->id;
$message->user_to = $_GET['sendTo'];
$message->text = $data['message_text'];
$message->time = date("H:i:s");
R::store($message);

```

На рисунку 5.7 зображено створення повідомлення користувачем.

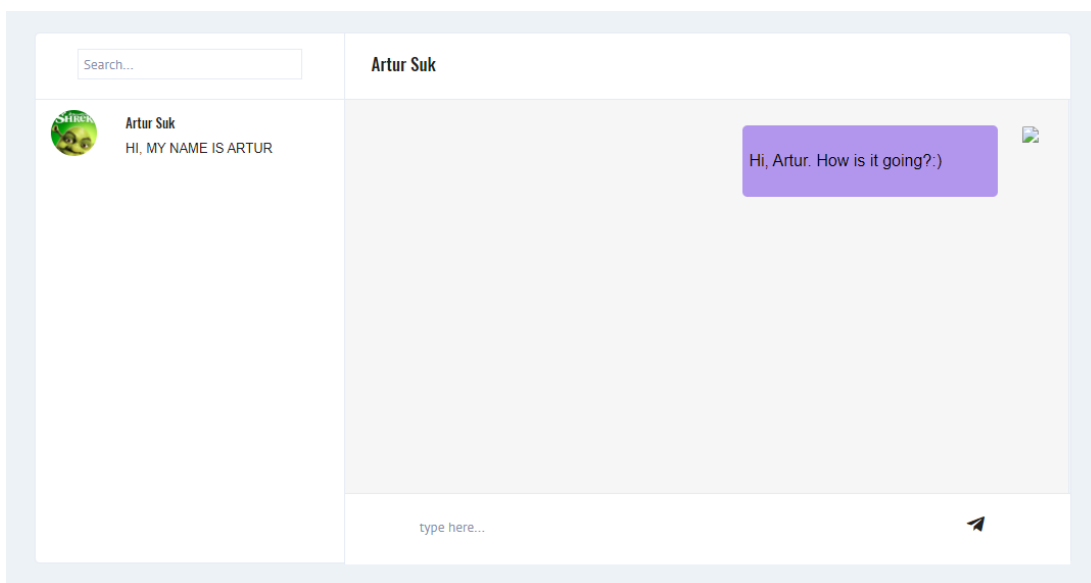


Рисунок 5.7 – Створення повідомлення.

Користувачі мережі можуть відправляти та переглядати повідомлення лише тих користувачів, котрих додали до списку друзів. Для перевірки цих відносин ми маємо окрему таблицю в БД.

Наступний лістинг коду ілюструє додавання у друзі:

```

$friends = R::dispense('friends');
$friends->id_user = $_SESSION['logged_user']->id;
$friends->id_friend = $_GET['id'];
$friends->status = 1;
R::store($friends);

```

На своїй власній сторінці користувач може змінювати головне фото профілю а також задній фон, котрий інші користувачі будуть бачити позаду аватару. Для економії пам'яті усі фото файли зберігаються у форматі base64.

Наступний лістинг коду ілюструє зміну користувачем аватару:

```

$avatar = addslashes($_FILES['change_ava']['tmp_name']);
$avatar = file_get_contents($avatar);
$avatar = base64_encode($avatar);
saveAvatar($avatar);

```

На рисунку 5.8 можна переглянути аватар та задній фон користувача.

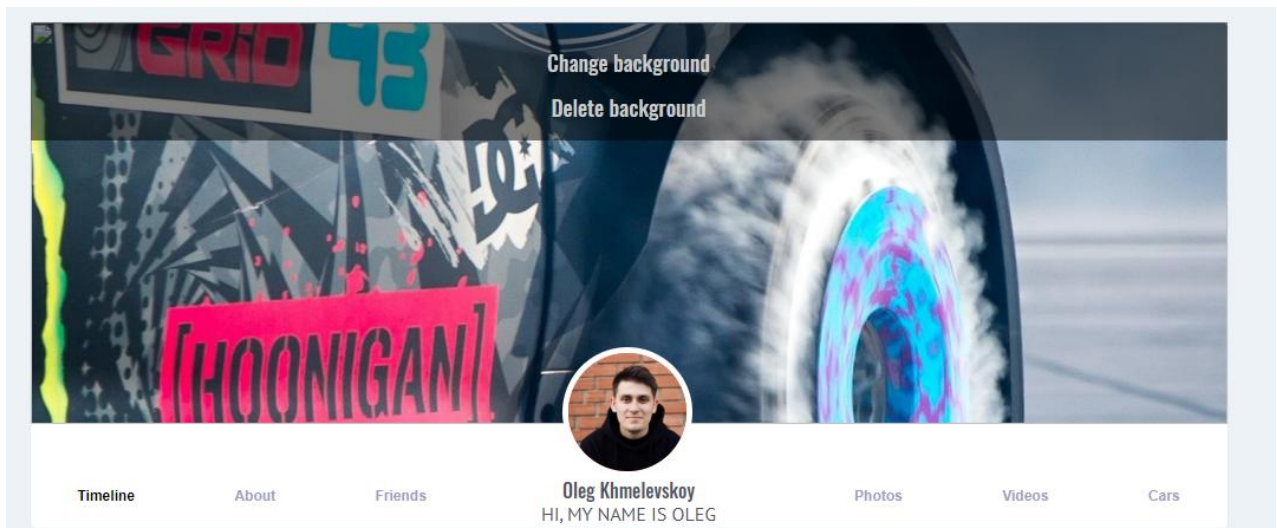


Рисунок 5.8 – Аватар і задній фон профілю.

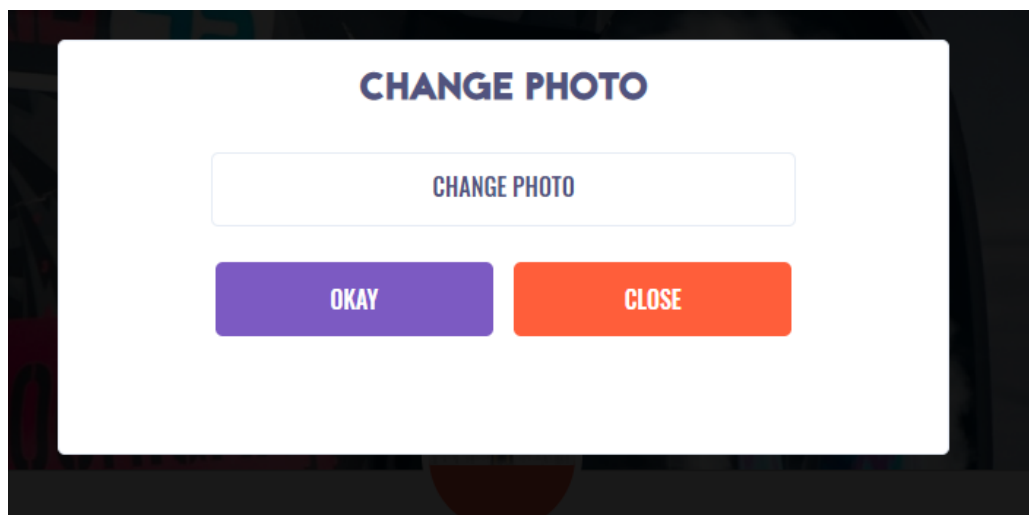


Рисунок 5.9 – Зміна аватару користувача

Користувач може додавати будь-яку фотографію на свій вибір, це може бути як власне фото, так і будь-яке зображення з інтернету. Це також стосується заднього фону профілю. Також на рисунку 5.8 можна побачити статус користувача, що знаходиться прямо під його ім'ям та прізвищем. Цей статус

користувач може змінити в будь-який момент і написати в ньому про своє самопочуття, місце роботи, місцезнаходження або взагалі якусь філософську цитату.

Тільки що зареєстрованим користувачам присвоюється типова фотографія, фон профілю та статус який згодом користувач може змінити за власним бажанням.

6 ОПИС МОЖЛИВОСТІ ВИКОРИСТАННЯ ОТРИМАНИХ РЕЗУЛЬТАТІВ

Тестування програмного забезпечення – процес дослідження, випробування програмного продукту, який має на меті перевірку відповідності між реальною поведінкою програми і її очікуваним поведінкою на кінцевому наборі тестів, обраних певним чином (ISO / IEC TR 19759: 2005) [32].

Тестування програмного забезпечення є процесом технічного дослідження, призначеного для виявлення інформації про якість продукту відносно контексту, в якому він має використовуватись. До техніки тестування відносять процес пошуку помилок або інших дефектів, а також випробування складових програмного забезпечення.

Оскільки число можливих тестів навіть для нескладних програмних компонент практично нескінченне, тому стратегія тестування полягає у тому, щоб провести всі можливі тести з урахуванням наявного часу та ресурсів, а також покриття тестами основної частини функціоналу системи. Як результат програмне забезпечення тестується стандартним виконанням програми з метою виявлення дефектів, тому для тестування системи, що розроблюється у даній роботі, було обрано метод мануального, тобто ручного тестування.

Проводилося мануальне тестування інтерфейсу користувача для виявлення помилок у функціональності за допомогою інтерфейсу, необроблених виключеннях при взаємодії з інтерфейсом, втрати або перекручення даних, переданих через елементи інтерфейсу, а також помилок в інтерфейсі, таких як відсутність елементів. Ручне тестування (manual testing) - частина процесу тестування на етапі контролю якості в процесі розробки програмного забезпечення. Воно проводиться тестувальником без використання програмних засобів, для перевірки програми або сайту шляхом моделювання дій користувача.

Реалізувавши програмне забезпечення, необхідно перевірити його працездатність і порівняти отримані експериментальні результати з теоретичними, і провести їх комплексний аналіз.

Результати тестування відображено у вигляді таблиці 6.1.

Таблиця 6.1 – Результати тестування ПЗ

№ стадії тестуван ня	Виявлені дефекти, к-ть	Виправлені дефекти, к-ть	Опис
1	2	2	Виявлено та виправлено дефект з створенням постів.
2	3	3	Виявлено та виправлено помилки при роботі повідомленнями
3	1	1	Виявлено та виправлено дефект не коректного відображення друзів на сторінці користувача.
4	1	1	Виявлено та виправлено дефект дублікації фото

З таблиці видно, що усього було виконано чотири стадії тестування, під час кожної з яких були знайдені та виправлені певні дефекти. Це дозволило перевірити відповідність між реальною поведінкою програми і її очікуваним поведінкою на кінцевому наборі тестів, виконаних певним чином, виявити ситуації, в яких поведінка програми є неправильною, небажаною або не відповідає специфікації, а також виконати процес, що містить у собі всі активності життєвого циклу, як динамічні, так і статичні, що стосуються планування, підготовки та оцінки програмного продукту і пов'язаних з цим результатів робіт з метою визначення відповідності описаним вимогам, а саме демонстрації, що вони підходять для заявлених цілей і для визначення дефектів

ВИСНОВКИ

В ході виконання атестаційної роботи була спроектована і розроблена програмна система, яка повинна зручним у використанні для автоволодарів.

На початку роботи було проведено аналіз предметної області, а також аналіз систем-аналогів, які вже присутні на ринку, і успішно реалізують свою продукцію.

Також були визначені вимоги до програмного продукту і до потенційних користувачів.

На етапі проектування було виконано концептуальне моделювання предметної області і визначена архітектура проекту. Також була спроектована і розроблена модель бази даних і визначені технології, використовувані при розробці програмного продукту.

Наступним етапом – був етап розробки, на якому була реалізована дана система у вигляді сервера та клієнтського додатку, використовуючи засоби і технології згідно з постановою на етапі проектування системи. В процесі розробки проводилося мануальне тестування.

Програмна система потребує певної доробки основного функціоналу і має простір для розширення та поліпшення. Перш за все, необхідно провести більш широке тестування і виправити знайдені помилки.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Кафедра програмної інженерії [Електронний ресурс] URL: <http://software.nure.ua/> (дата звернення: 10.05.2018).
2. Создаем динамические веб-сайты с помощью PHP, MySQL, JavaScript, CSS и HTML5. 3-е издание, Робин Никсон, 688 стр., 2015 рік.
3. PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide, Ларі Улман, 1021 стр., 2012 рік.
4. PHP Advanced and Object-Oriented Programming: Visual QuickPro Guide, Мэт Зандстра, 560 стр., 2011 рік.
5. Предметно-ориентированное проектирование (DDD). Структуризация сложных программных систем, Эрик Эванс, 448 стр., 2010 рік.
6. Eloquent JavaScript, Марин Хавербеки, 1130 стр., 2017 рік.
7. Learning JavaScript Design Patterns, Єдди Османі, 448 стр., 2017 рік.
8. Сайт з інформацією [Електронний ресурс] URL: <http://javascript.info/> (дата звернення: 19.05.2018).
9. Book on Building Front-End Web Apps with Plain JavaScript
10. Герд РНР 7, Дмитро Котеров, 1088 стр., 2016
11. Вергер, 333 стр., 2015 рік.
12. Самоучитель РНР 7. 448 стр., 2018 рік.
13. Дэвид Скляр - Изучаем РНР 7. 382 стр., 2017 рік.
14. Learning management system // Wikipedia. URL: https://en.wikipedia.org/wiki/Learning_management_system (дата звернення: 26.01.2017).
15. Бейкер, Р. Educational data mining and learning analytics/Р. Бейкер, Г. Сіменс – The Cambridge handbook of the learning sciences, 2014. – 274 с.
16. Ржеуцька С. Опыт применения методов кластеризации для анализа результатов дистанционного обучения // Информатизация инженерного

образования: материалы международной науч.-практ. конф. 2016. № 56. С. 617 – 620.

17. Expectation-maximization algorithm // Wikipedia. URL: https://en.wikipedia.org/wiki/Expectation%E2%80%93maximization_algorithm (дата звернення: 23.11.2019).

18. K-means clustering // Wikipedia. URL: https://en.wikipedia.org/wiki/K-means_clustering (дата звернення: 25.11.2019).

19. Кукієр, К. Big Data: A Revolution That Will Transform How We Live, Work, and Think/К. Кукієр, В. Штойнберг, 2013. – 236 с.

20. Лам Ч. Hadoop in Action/Ч. Лам, 2010. – 365 с.

21. Дашкевич О. Аналіз можливостей Apache Kafka в рамках забезпечення стрімінгу Big Data // Інформаційні системи і технології: матеріали 7-й Міжнарод. науч.-техн. конф. 2018. № 12. С. 34-35.

22. Data Mining with WEKA MOOC – Material // Machine Learning at Waikato University. URL: <https://www.cs.waikato.ac.nz/ml/WEKA/mooc/dataminingwithWEKA> (дата звернення: 26.11.2019).

23. A Simple Step by Step Guide to WEKA // Analyticscosm. URL: <https://analyticscosm.com/a-simple-step-by-step-guide-to-WEKA/> (дата звернення: 12.11.2019).

24. Shubin I., Turevska I. Improving the Automated Testing of Web-based Services by Reflecting the Social Habits of Target Audiences /I.Shubin, I. Turevska // Proceeding of 2015 Information Technologies in Information Business Conference (ITIB) 7 – 9 October, 2015, IEEE Catalog Number CFP15D13-PRT pp. 93-96