

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Тернопільський національний економічний університет
Факультет комп'ютерних інформаційних технологій
Кафедра комп'ютерних наук

ДЗЕБЧУК Олег Михайлович

**Веб-сервіс для каучсерфінгу/ Web service for
couchsurfing**

напрямок підготовки: 6.050103 - Програмна інженерія
фахове спрямування - Програмне забезпечення систем

Бакалаврська дипломна робота

Виконав студент групи ПЗС-41
О. М. Дзєбчук

Науковий керівник:
викладач ПОРПЛИЦЯ Н.П.

Бакалаврську дипломну роботу
допущено до захисту:

" ___ " _____ 20__ р.

Завідувач кафедри
_____ **А. В. Пукас**

ТЕРНОПІЛЬ - 2016

РЕЗЮМЕ

Дипломна робота містить 99 сторінки, 19 таблиці, 48 рисунки, список використаної літератури містить 22 джерела.

Метою дипломної роботи є розробка веб-сервісу для каучсерфінгу.

Об'єктом дослідження є процес пошуку та резервування місця для ночівлі.

Предметом дослідження є застосування сучасних технологій створення веб-орієнтованих додатків.

Методи розробки базуються на використанні мови програмування PHP і системи керування вмістом з відкритим кодом WordPress. Для організації баз даних використано базу даних MySQL.

Одержані результати розроблена веб-орієнтована система каучсерфінг.

Ключові слова: резервування, проектування програмного забезпечення, каучсерфінг, тестування, розробка.

SUMMARY

Thesis contains 99 pages, 19 tables, 48 figures, list of references includes 22 sources.

The aim of the thesis is to develop a service for CouchSurfing.

The object of the research is the process of finding and booking a place to sleep.

The subject of research is the use of technologies of web-oriented applications.

Methods based on development using PHP programming language and development environment WordPress. To organize databases used database MySQL.

The results developed web-oriented system CouchSurfing.

Keywords: backup, software design, CouchSurfing, testing and development.

ЗМІСТ

ВСТУП.....	8
РОЗДІЛ АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ	9
1.1 Коротка характеристика об'єкту управління.....	9
1.2 Опис предметної області.....	10
1.3 Огляд та аналіз існуючих аналогів	16
1.4 Специфікація вимог до програмного продукту	23
РОЗДІЛ 2_ПРОЕКТУВАННЯ WEB-СИСТЕМИ	37
2.1 Розробка програмної архітектури системи	37
2.2 Проектування структури бази даних.....	40
РОЗДІЛ 3 ПРОГРАМНА РЕАЛІЗАЦІЯ.....	47
3.1 Програмна реалізація проекту	47
3.2 Програмна реалізація бази даних	53
РОЗДІЛ 4 ТЕСТУВАННЯ ТА ЗОВНІШНЯ ЕКСПЛУАТАЦІЯ	57
4.1 Тестування	57
4.2 Розгортання програмного продукту	62
4.3 Інструкція користувача	62
ВИСНОВКИ	69
СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ.....	70
ДОДАТОК А ГЛОСАРІЙ	72
ДОДАТОК Б КОД ПРОГРАМНОГО ПРОДУКТУ	74
ДОДАТОК В TEST-CASES	91
ДОДАТОК Г СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ	94
ДОДАТОК Д_ТЕСТ-ПЛАН.....	99

ВСТУП

Актуальність теми

Поняття каучсерфінг переросло з веб-сайту на напрям туризму, тому його популяризація в нашій країні є актуальною темою. Тобто, каучсерфінг зараз – це подорожі з безкоштовним житлом наданим іншими учасниками течії. Оскільки в нашій країні поняття каучсерфінгу не набуло такого значення, як в Європі, тому актуальною постає задача розробки веб-сервісу для каучсерфінгу, яка б дозволила популяризувати це поняття та надати можливість учасникам сервісу подорожувати з меншою затратою коштів.

Мета і задачі розробки

Метою роботи є розробка веб-сайту для каучсерфінгу. Веб-сайт створюється з метою популяризації каучсерфінгу та швидкого доступу до його можливостей, які ми можемо використовувати під час туристичних подорожей. Врахувавши усі плюси та недоліки проаналізованих в подальшому аналогів буде створюватися веб-сервіс для каучсерфінгу.

Методи, засоби та технології розробки

Для реалізації веб-сайту була обрана клієнт-серверна архітектура. Технологією розробки для цього було обрано мову програмування PHP і базу даних MySQL. Для того, щоб процес розробки був структурованим було обрано систему керування вмістом з відкритим кодом WordPress.

Практичне значення одержаних результатів

Результатом розробки є веб-сайт, який буде надавати можливість користувачам резервувати місце для ночівлі у зручний для них час, а також надання власного житла для прийому учасників каучсерфінгу.

РОЗДІЛ 1

АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Коротка характеристика об'єкту управління

CouchSurfing - одна з найбільших гостьових мереж, яка об'єднує більше 10 мільйонів чоловік в 248 країнах. Ідея створення такого ресурсу належить Кейсі Фентону. Вона зародилась в 2000 році, коли він купив дешевий квиток до Ісландії, але у нього не було місця, де можна було зупинитися. Тоді він розіслав листи більш ніж 1500 ісландським студентам з проханням дозволити йому зупинитися у них. В результаті листування він познайомився з декількома групами студентів, готових показати йому їх місто та надати місце, щоб заночувати. Після того, як він провів незабутні вихідні в Ісландії, він вирішив, що більше не буде користуватися послугами туристичної інфраструктури і вирішив створити такий інтернет-ресурс. В січні 2004 сайт став доступний широкому загалу[2].

В країнах Європи набув популярності каучсерфінг, який став справжнім всесвітнім рухом. Цей рух об'єднує людей з різних країн і сприяє обміну культурами і досвідом. Тому з часом поняття – каучсерфінг, яке є назвою ресурсу, переростає у популярну течію, яка переймає назву веб-сайту.

Кожне місто і регіон особливі, зі своєю неповторною культурою, людьми і краєвидами. Кожен, хто хоч раз їздив до інших міст України, зіштовхувався з проблемою пошуку місця, де можна переночувати. Зазвичай все закінчується по добовою орендою.

Основна ідея сервісу для підтримки каучсерфінгу полягає в тому, щоб звести витрати на туристичні подорожі до мінімуму. За основу таких сервісів беруть принцип, де вільне місце, там можна переночувати, при цьому не затримуючись на довго, щоб дати змогу іншим користувачам користуватися сервісом. Тому, створення веб-сайту для каучсерфінгу має багато особливостей і переваг, зокрема:

- учасники безкоштовно діляться один з одним місцем для ночівлі під час подорожей;
- учасники організовують спільні подорожі та розвивають культурний взаємообмін.

Вправно користуючись розробленим сервісом, можна знайти собі місце для ночівлі, за яке не прийдеться платити коштів.

1.2 Опис предметної області

До основних бізнес-процесів, які повністю відображають функціонал сайту з боку користувача та адміністратора, можна віднести:

- процес реєстрації нового користувача;
- процес пошуку місця ночівлі;
- процес резервування місця для ночівлі;
- процес надання місця для ночівлі;
- процес прийняття користувача на сайт.

Розглянемо детальніше функції кожного з вище наведених процесів.

Для того, щоб використовувати усі функції, які надає нам веб-сайт, необхідно бути авторизованим у системі. Для цього необхідно пройти процес реєстрації. Процес реєстрації включає в себе заповнення таких обов'язкових полів як:

- Логін;
- Електронна пошта;
- Пароль;
- ПІП;
- Місце проживання(адрес);
- Хоббі;
- Шкідливі звички;
- Володіння мовами.

Після підтвердження реєстрації буде можливість авторизації.

Всі функції бізнес-процесу «Реєстрація користувача» подано на рисунку 1.1.



Рисунок 1.1 – Функції бізнес-процесу «Реєстрація користувача»

Характеристика бізнес-процесу «Реєстрація користувача» представлена у таблиці 1.1.

Таблиця 1.1

Характеристика бізнес-процесу «Реєстрація користувача»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Реєстрації нового користувача на сайті
Основні учасники	Користувач
Вхідна подія	Необхідність реєстрації
Вхідні документи	Відсутні
Вихідні події	Користувач зареєстрований
Вихідні документи	Відсутні
Клієнт бізнес-процесу	Користувач

Після того, як користувач авторизується у системі він може здійснити пошук місця ночівлі для резервування. Для цього йому необхідно ввести та підтвердити усі необхідні для пошуку поля. Зокрема:

- місто;
- перелік мов;
- фільтр сортування користувачів;
- критерій зайнятості місця.

Функції бізнес-процесу «Пошук» зображено на рисунку 1.2.



Рисунок 1.2 – Функції бізнес-процесу «Пошук»

Характеристика бізнес-процесу «Пошук» представлена у таблиці 1.2.

Таблиця 1.2

Характеристика бізнес-процесу «Пошук»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Пошук
Основні учасники	Користувач
Вхідна подія	Авторизований
Вхідні документи	Відсутні
Вихідні події	Веб-сторінка з список користувачів
Вихідні документи	Відсутні
Клієнт бізнес-процесу	Користувач

Після того, як користувач здійснив пошук, у форму веб-сайту відображається список користувачів за параметрами, які були введені. Після

чого можемо вибрати з відображеного списку будь-який профіль користувача та відправити йому запит, щоб зарезервувати місця для ночівлі.

Функції бізнес-процесу «Резервування місця ночівлі» зображено на рисунку 1.3.



Рисунок 1.3 – Функції бізнес-процесу «Резервування місця ночівлі»

Характеристика бізнес-процесу «Резервування місця для ночівлі» представлена у таблиці 1.3.

Таблиця 1.3

Характеристика бізнес-процесу «Резервування місця для ночівлі»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Резервування місця для ночівлі
Основні учасники	Користувач
Вхідна подія	Результати пошуку
Вхідні документи	Відсутні
Вихідні події	Підтвердження резервування
Вихідні документи	Відсутні
Клієнт бізнес-процесу	Користувач

Також у користувача веб-сайту є можливість запропонувати своє житло для ночівлі. На цьому і базується каучсерфінг. Можливість надавати житло

мають тільки авторизовані користувачі. Для здійснення цієї дії, користувач повинен виконати наступні кроки.

Функції бізнес-процесу «Надання житла » зображено на рисунку 1.4



Рисунок 1.4 – Функції бізнес-процесу «Надання житла »

Характеристика бізнес-процесу «Надання житла» представлена у таблиці 1.4.

Таблиця 1.4

Характеристика бізнес-процесу «Процес надання житла»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Надання житла
Основні учасники	Користувач
Вхідна подія	Заповнений профіль
Вхідні документи	Відсутні
Вихідні події	Житло додано у список
Вихідні документи	Відсутні
Клієнт бізнес-процесу	Користувач

У адміністратора веб-сайту є можливість активувати користувачів, тобто підтверджувати їхню реєстрацію, а також можливість змінити статус користувача, тобто надати підписник, учасник, адміністратор.

Функції бізнес-процесу « Активації користувачів » зображено на рисунку 1.5



Рисунок 1.5 – Функції бізнес-процесу «Активация користувачів»

Характеристика бізнес-процесу «Активация користувачів» представлена у таблиці 1.5.

Таблиця 1.5

Характеристика бізнес-процесу «Процес надання житла»

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Процес активації користувачів
Основні учасники	Адміністратор
Вхідна подія	Користувач авторизований, як адміністратор
Вхідні документи	Відсутні
Вихідні події	Користувачів активовано, змінено статус
Вихідні документи	Відсутні
Клієнт бізнес-процесу	Адміністратор

1.3 Огляд та аналіз існуючих аналогів

Перш ніж почати розробляти певну систему, потрібно спочатку здійснити пошук аналогів, дослідити, проаналізувати їх переваги та недоліки, щоб не допустити ті ж самі помилки, які наявні в системах, які досліджувались.

Проаналізовано 3 веб-сайти, які реалізують аналогічні функції предметної області. Першим аналогом є “Перша українська мережа гостинності” (режим доступу: <http://shuffle.in.ua/>).

На рисунку 1.6 показано головну сторінку “Першої української мережі гостинності”. На цій сторінці відображаються різні вкладки, зокрема: останні додані заявки, авторизація, також відображена карта на якій позначено міста, які надають послуги. Також відображаються профілі випадкових користувачів.

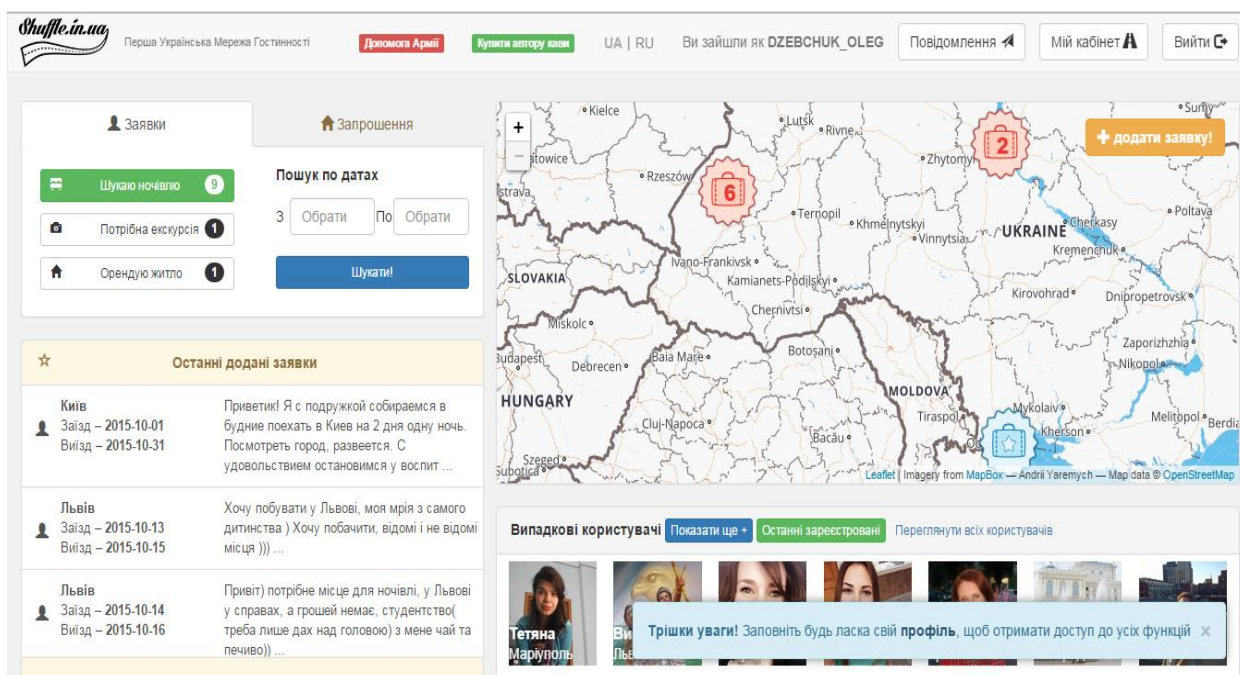


Рисунок 1.6 – Головне вікно веб сайту “Перша українська мережа гостинності”

Натиснувши на кнопку «Мій кабінет» завантажиться веб-сторінка, яка показана на рисунку 1.7. Тут можна “Створити запрошення”, “Створити заявку” та переглянути подані і створені користувачем заявки.

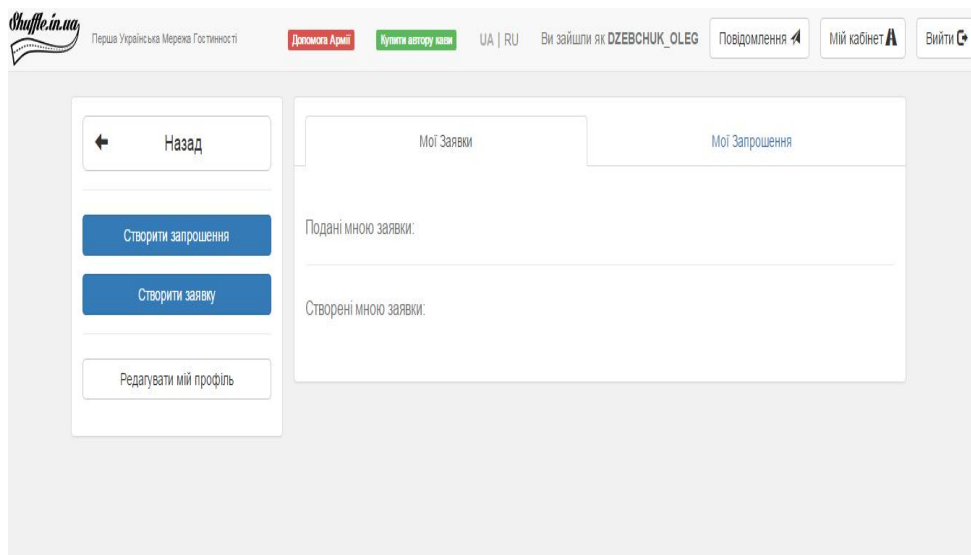


Рисунок 1.7 – Сторінка «Мій кабінет»

Задавши параметри пошуку у вкладці “Заявки” та підтвердивши пошук відображаються результати пошуку, які показані на рисунку 1.8.

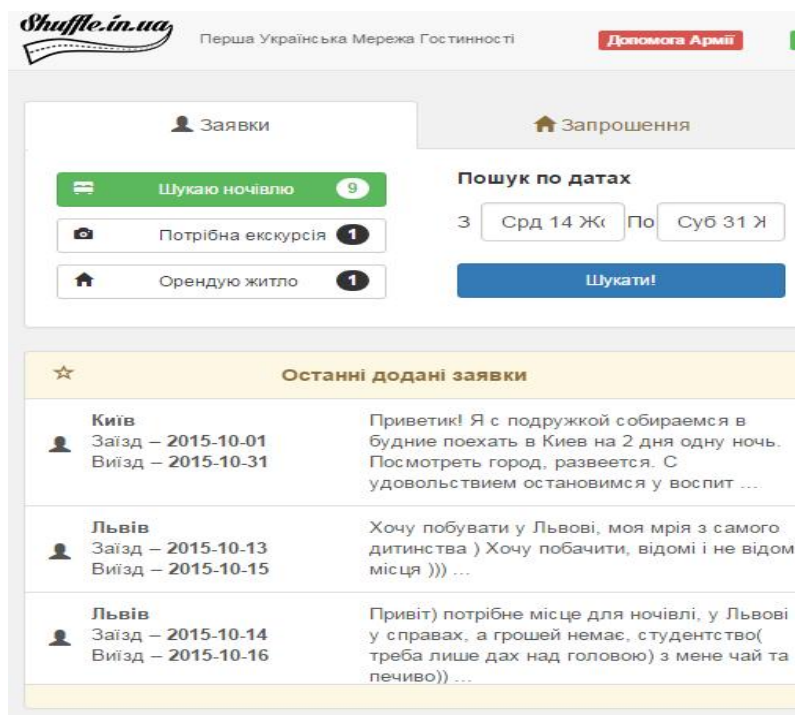


Рисунок 1.8 – Сторінка із результатами пошуку на сайті

Відкривши один із запропонованих варіантів після пошуку, завантажиться веб-сторінка на якій відображається інформація про профіль обраного користувача, який надає послугу, див. рис. 1.9.

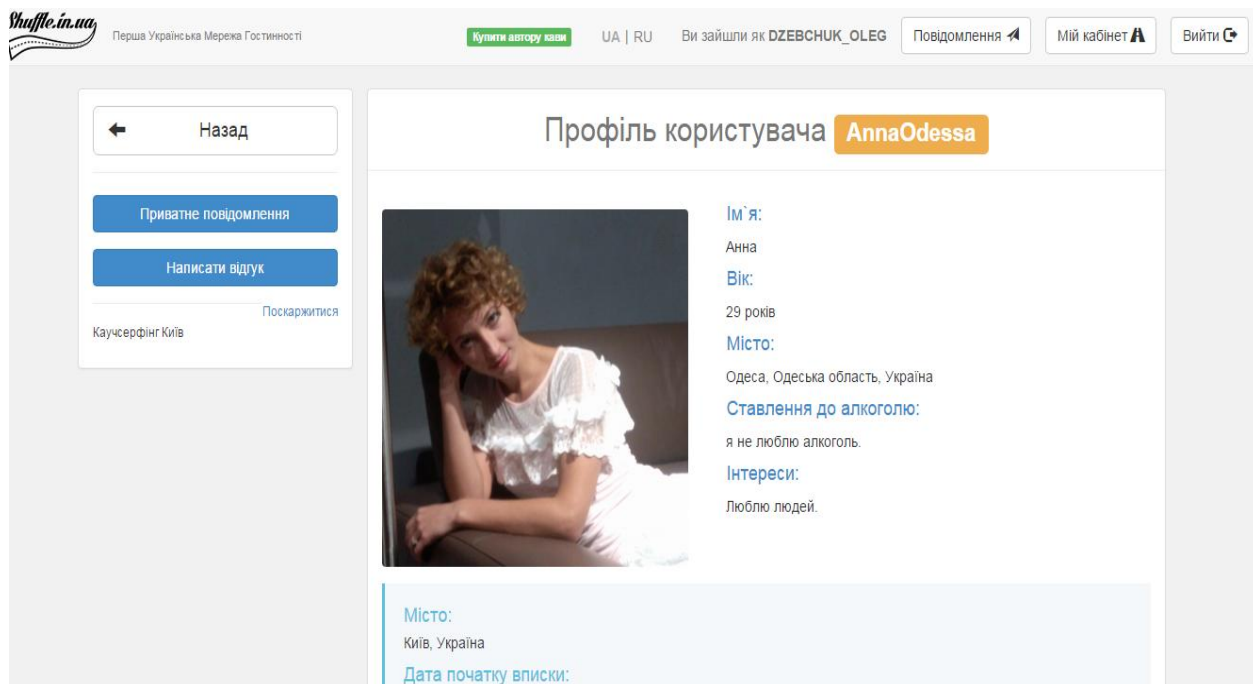


Рисунок 1.9 – Сторінка із профілем одно із користувачів, який надає житло

На рисунку 1.10 показано список користувачів сайту. Для відображення цього списку потрібно натиснути на кнопку “Переглянути всіх користувачів”.

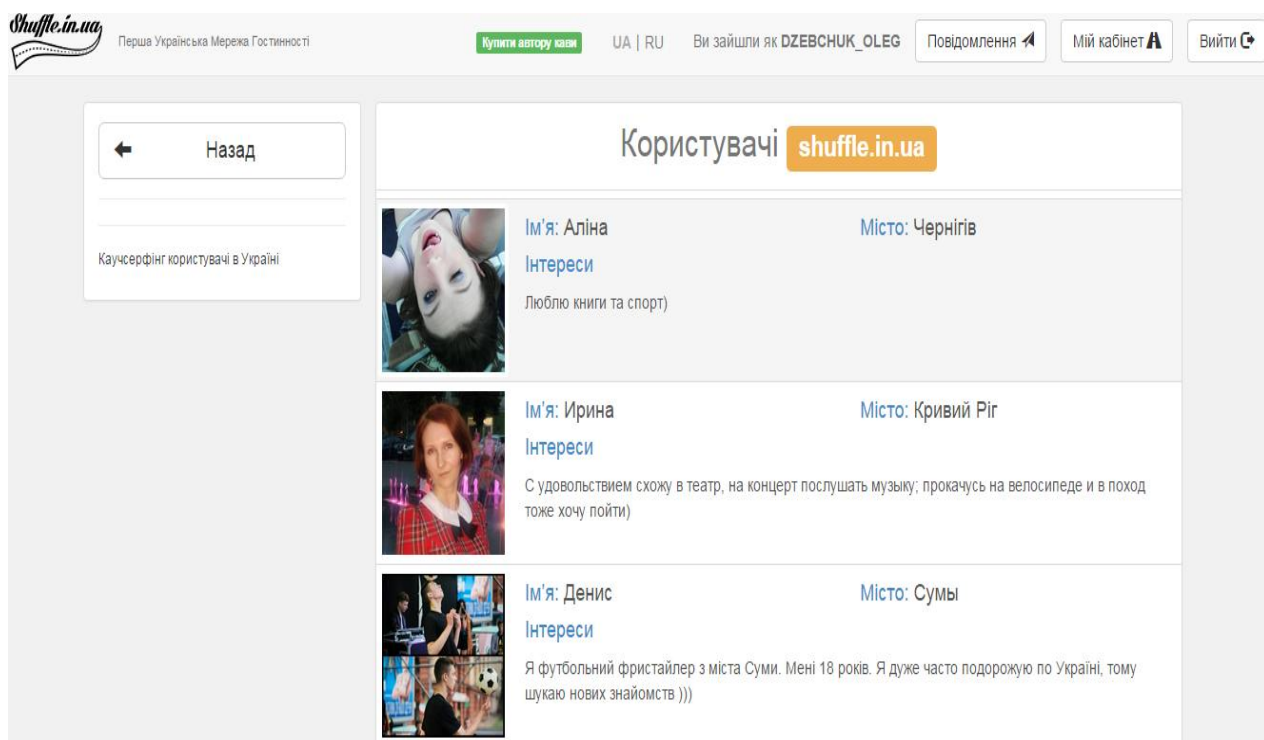


Рисунок 1.10 –Сторінка із списком користувачів сайту

Другим аналогом виступає “Всесвітня мережа Каучсерфінг” (режим доступу: <https://www.couchsurfing.com/>).

На рисунку 1.11 показано головне вікно веб-сайту Couchsurfing. На цьому веб-сайті є можливість авторизації, також на головній сторінці є основна функція сайту “пошук”, в цій функції є можливість обирати пошук по критеріях, також присутні такі вкладки як: групи, події.

Головне вікно сайту показано на рисунку 1.11.

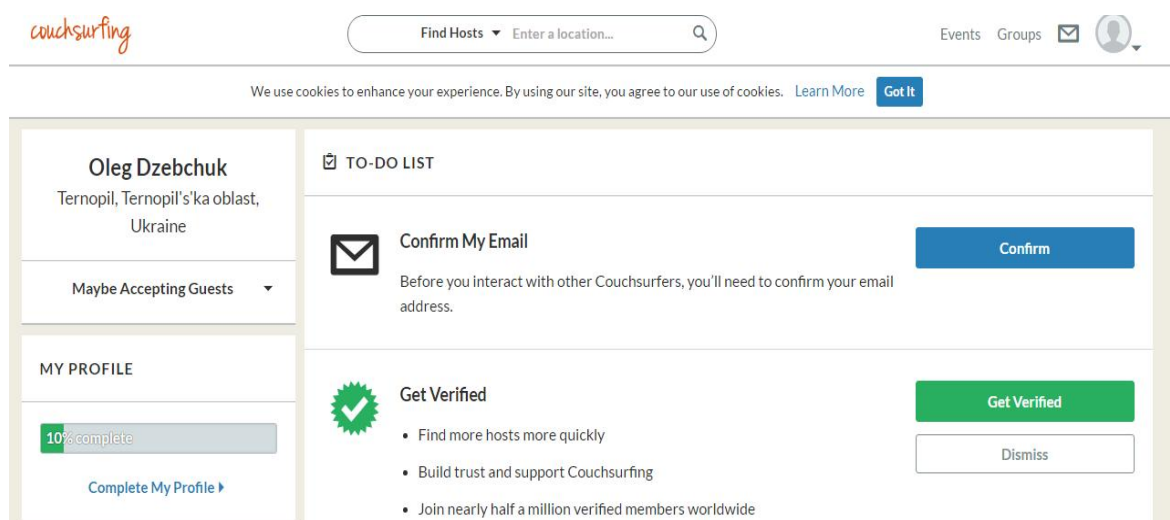


Рисунок 1.11 – Сторінка із головним вікном веб-сайту “Couchsurfing”

Ввівши критерій пошуку та підтвердивши його, відображаються результати пошуку, які показано на рисунку 1.12.

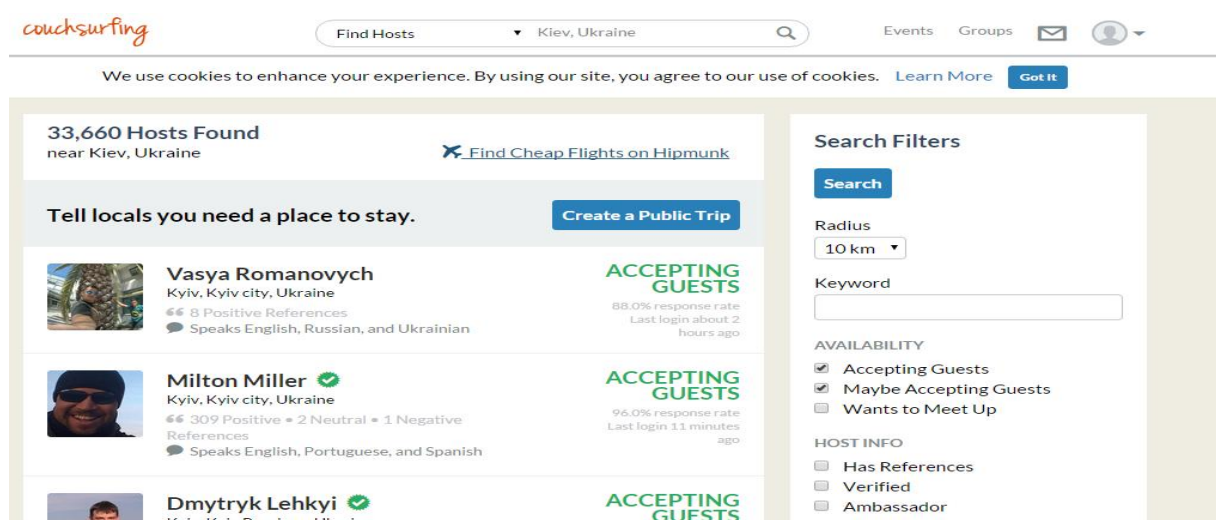


Рисунок 1.12 – Сторінка із користувачами із результатів пошуку

Відкривши вкладку «Groups», показану на рисунку 1.13, нам відображаються існуючі групи для спілкування. Також є можливість створити свою групу натиснувши на кнопку “Create a Group”.

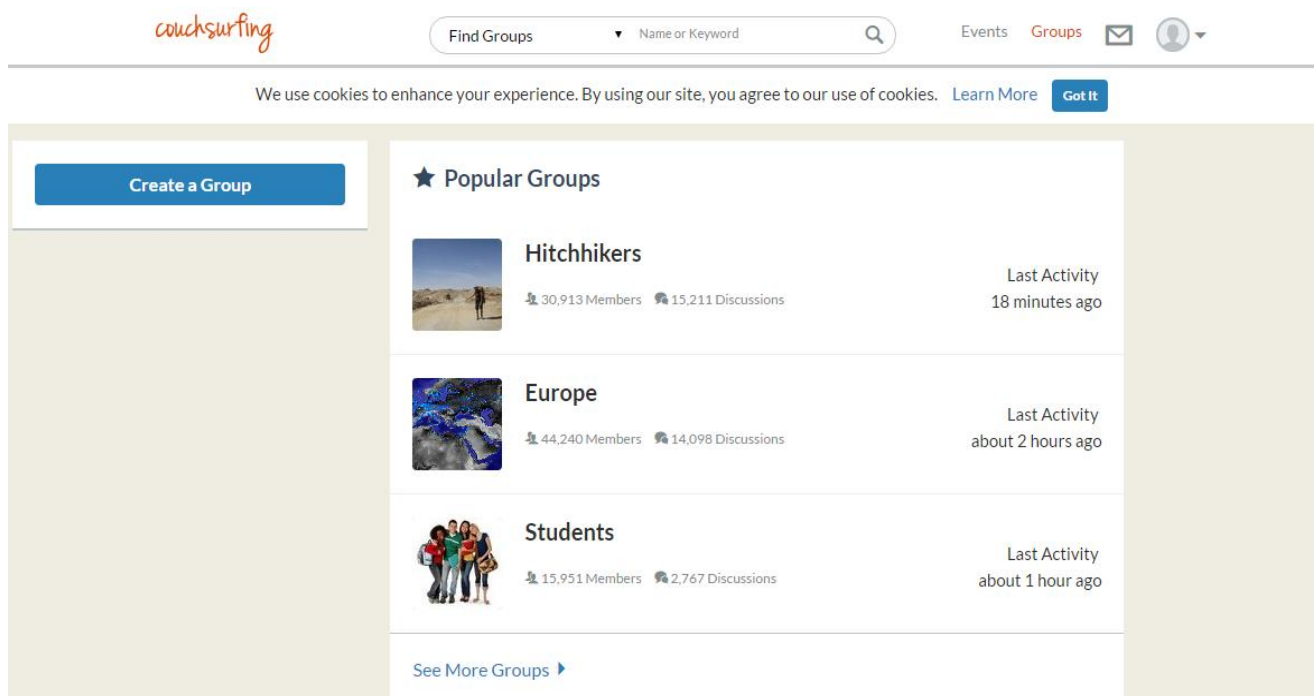


Рисунок 1.13 – Вікно веб-сайту із списком існуючих груп

Відкривши вкладку «Events», показану на рисунку 1.14, відображено існуючі події, а також є можливість створити свою групу натиснувши на кнопку “Create an Event”.

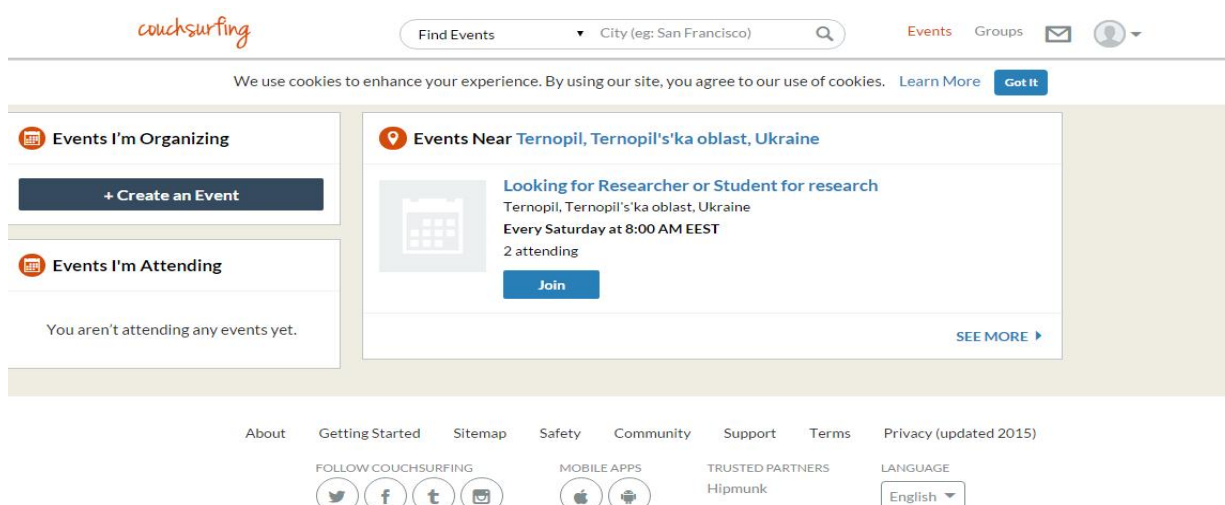


Рисунок 1.14 – Вікно веб-сайту із списком існуючих подій

Третім аналогом є ресурс “Life in Travel” (режим доступу: <http://www.life-in-travels.ru/>).

На рисунку 1.15 показана головна сторінка сайту. На цій сторінці відображаються вкладки з аналогічною функціональністю, як попередні аналоги, також є можливість перейти на блог сайту. Тут наявні статті, які дають допомогу туристу.

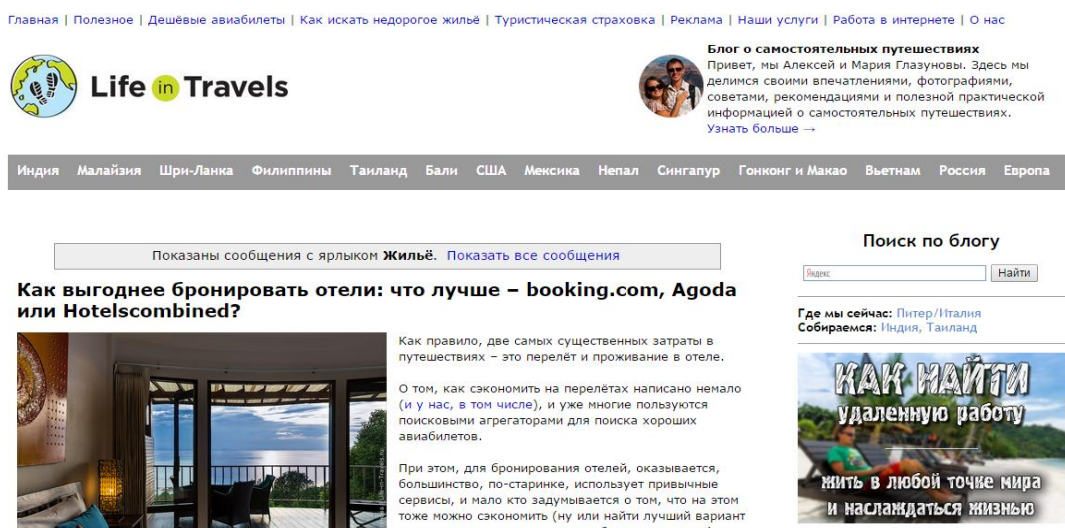


Рисунок 1.15 – Сторінка головного вікна сайту

Перейшовши на вкладку «Наші послуги» показану на рисунку 1.16 відображаються послуги сайту. Нажаль послуги цього сайту є платні, оскільки це ресурс туристичної фірми.

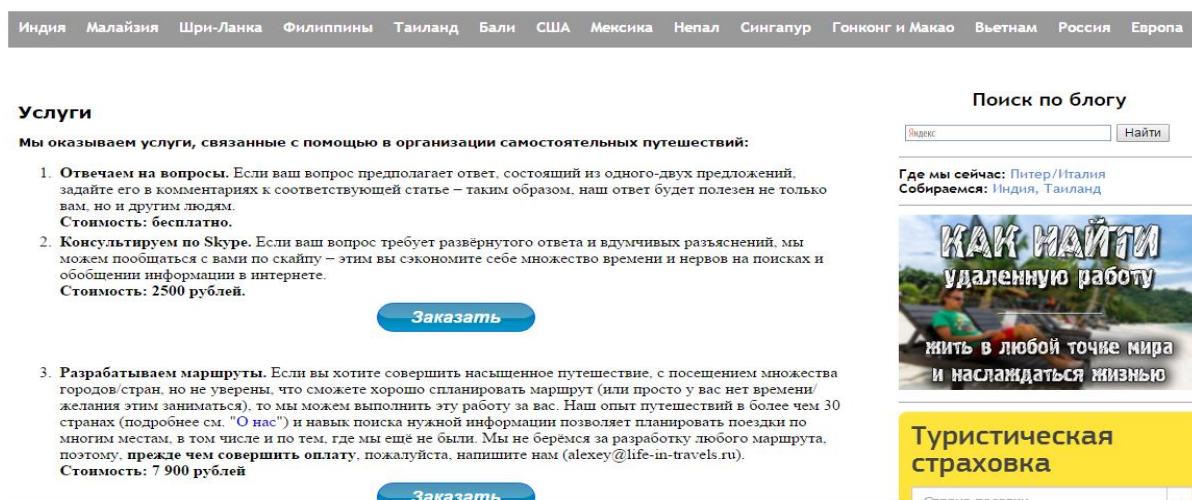


Рисунок 1.16 – Сторінка із послуги сайту

Таблиця 1.6

Порівняльна характеристика програмних продуктів

Фірма-розробник	Не вказано	Не вказано	Не вказано
Назва програмного продукту	Shuffle	Couchsurfing	Live in Travels
Версії продукту	1.0	1.0	1.0
Функціональність	- Пошук - Мій кабінет - Перегляд користувачів - Повідомлення	-Профіль - Пошук -Групи -Події - Обговорення - Перегляд корист	- Блог -Туристична страховка -Наші послуги
Інтерфейс користувача	WEB-система	WEB-система	WEB-система
Допомога користувачеві	Відсутня	Відсутня	Відсутня
Читабельність	Присутня	Присутня	Присутня

Інтерфейс веб-сайту “Першої української гостинності” є зручним і зрозумілим. Проте, є деякі недоліки, такі як: відсутність чату, відсутні групи, пошук не є цілком зручний. Основним недоліком є те, що при великій кількості користувачів маркери на карті наложуються, а також інтерфейс з картою є не легким у розумінні для простого користувача.

Інтерфейс сайту Couchsurfing є зручним і зрозуміли. Тут крім стандартних функцій пошуку є ще додаткові можливості для здійснення пошуку. Присутній чат в групах та можливість створити або прийняти участь в

події. Про те присутні незначні недоліки: відсутня можливість змінити мову сайту, попри те, що веб-сайт є міжнародним.

Інтерфейс третього сайту Live in Travels є не зрозумілим для звичайного користувача, багато платних послуг, але логіка веб-сайт для подорожування зберігається.

Тому метою цієї дипломної роботи є розробка веб-орієнтованої системи каучсерфінг. Під час розробки цієї системи будуть враховані переваги та недоліки кожної із розглянутих альтернатив.

1.4 Специфікація вимог до програмного продукту

Після детального аналізу аналогів та визначення їх основних переваг і недоліків переходимо до етапу опису специфікацій вимог до програмного продукту. Специфікація вимог – це документ, який містить повний опис розроблювального продукту.

Для того, щоб все було зрозуміло, спочатку розробимо глосарій проекту. Глосарій – це словник вузькоспеціалізованих термінів у будь-якій галузі знань з тлумаченням, іноді перекладом на іншу мову, коментарями і прикладами.

Глосарій проекту наведено у додатку А.

Створення веб-сайту досить складний і трудомісткий процес, для того, щоб процес створення не був невпорядкованим, для початку необхідно виділити основні функції системи, а вже після того покроково реалізовувати ці функції на рівні програмування. Для відображення функцій системи нам допоможе діаграма варіантів використання, яка відображає усі функції веб-сайту, які нам потрібно реалізувати.

Діаграма варіантів використання зображена на рисунку 1.17.

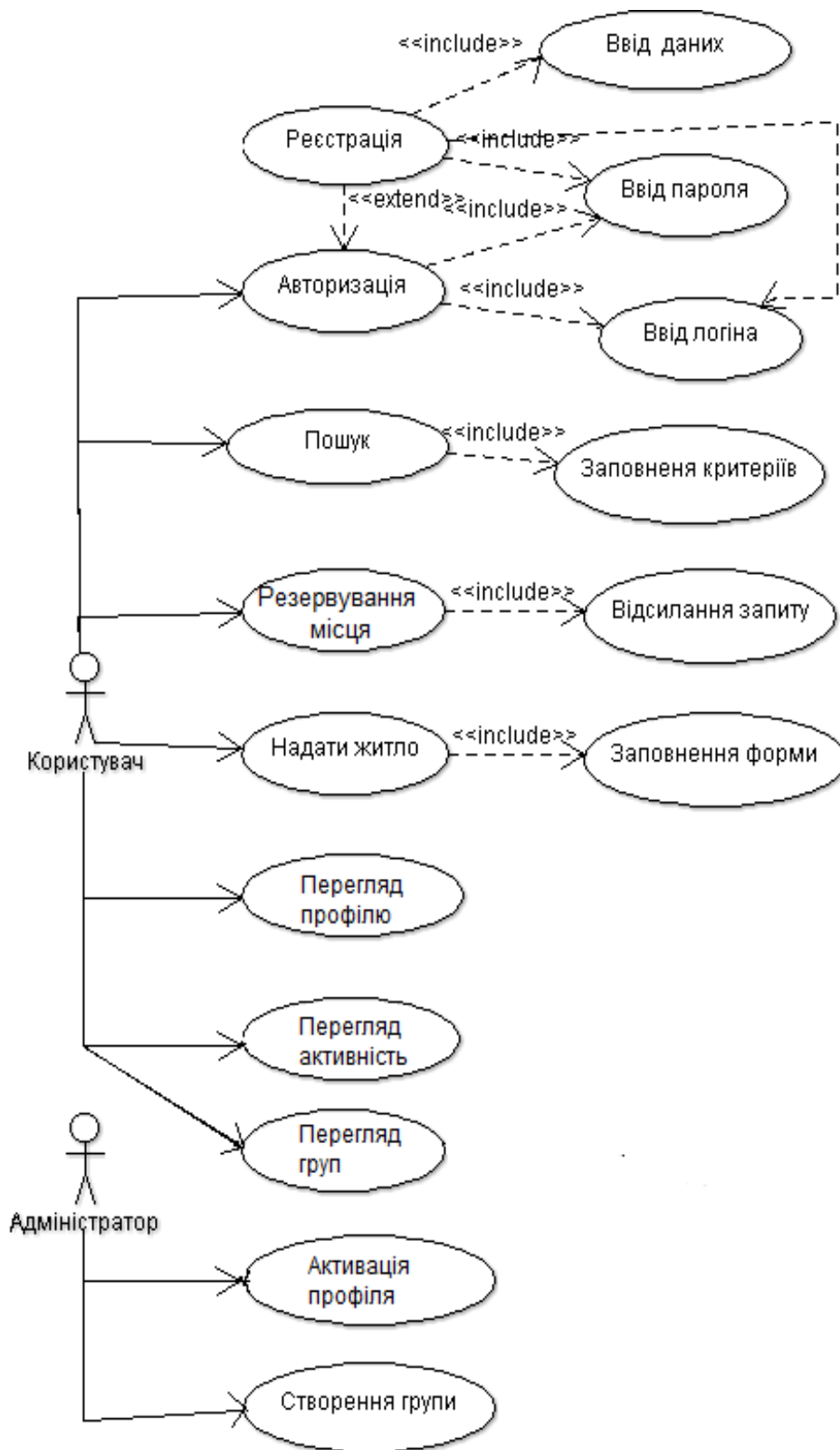


Рисунок 1.17 – Діаграма варіантів використання

Характеристики варіантів використання представленні у таблицях 1.7-1.18

Таблиця 1.7

Характеристика варіанту використання «Авторизація»

Контекст використання	Авторизація
Дійові особи	Користувач
Передумова	Повинен бути зареєстрований
Тригер	Відкрита сторінка авторизації
Сценарій	<ol style="list-style-type: none"> 1. Заповнюємо поле “Логін” 2. Заповнюємо поле “Пароль” 3. Підтверджуємо запам’ятати мене 4. Підтверджуємо вхід натиснувши “Увійти”
Пост-умова	Користувач авторизований

Авторизація дозволить користувачу ідентифікуватися на веб-сайті, що дасть йому права користувача сайту. На рисунку 1.18 відображена розкадровка авторизації.

The image shows a login form with the following elements:

- A label "Ім'я" (Name) above a text input field.
- A label "Пароль" (Password) above a text input field.
- A checkbox labeled "Запам'ятати мене" (Remember me).
- A button labeled "Увійти" (Login).

Рисунок 1.18 – Розкадровка сторінки «Авторизація»

Таблиця 1.8

Характеристика варіанту використання «Пошук»

Контекст використання	Пошук
Дійові особи	Користувач
Передумова	Користувач авторизований
Тригер	Відкрита сторінка пошуку
Сценарій	<ol style="list-style-type: none"> 1. Вводимо потрібне місто для пошуку 2. Натискаємо на кнопку “пошук” 3. Заповнюємо додаткові критерії пошуку 4. Підтверджуємо пошук натиснувши “готово”
Пост-умова	Список користувачів, які підходять під критерії пошуку

Авторизований користувач у системі може здійснити пошук місця на ночівлю. Для цього йому необхідно перейти на вкладку “Пошук” та заповнити усі необхідні поля для здійснення пошуку. На рисунку 1.19 відображено розкадровку сторінки “Пошук”.

пошук користувачів

Місто

Вкажіть місто користувача

Володіння мовою

Перелічіть через кому мови, якими володієте

Є можливість надати ночівлю

Рисунок 1.19 – Розкадровка сторінки «Пошук»

Таблиця 1.9

Характеристика варіанту використання «Резервування місця»

Контекст використання	Резервування
Дійові особи	Користувач
Передумова	Здійснений пошук, а також профіль повинен бути заповнений
Тригер	Відкритий профіль користувача
Сценарій	1. Натискаємо на кнопку “відправити повідомлення” 2. Вводимо повідомлення та підтверджуємо надсилання запиту кнопкою “відправити”
Пост-умова	Підтвердження резервування

Після того, як пошук був здійснений на сторінці веб-сайту відображається список користувачів. Ми можемо вибрати зі списку будь-який профіль користувача та відправити йому запит з проханням переночувати. На рисунку 1.20 відображено розкадровку сторінки “Резервування місця для ночівлі”.



Рисунок 1.20 – Розкадровка сторінки «Резервування місця»

Таблиця 1.10

Характеристика варіанту використання «Надання житла»

Контекст використання	Надати житло
Дійові особи	Користувач
Передумова	Авторизований користувач, а також профіль повинен бути заповнений.
Тригер	Відкрита веб-сторінка
Сценарій	1. Натискаємо на профіль. 2. Підтверджуємо готовність прийняти користувача 3. Підтверджуємо надання житла натиснувши кнопку “готово”
Пост-умова	Житло запропоновано

Також у користувача веб-сайту є можливість запропонувати своє житло для ночівлі. На цьому і базується каучсерфінг. Можливість надавати житло мають тільки авторизовані користувачі. На рисунку 1.21 відображено розкадровку сторінки «Надання житла»

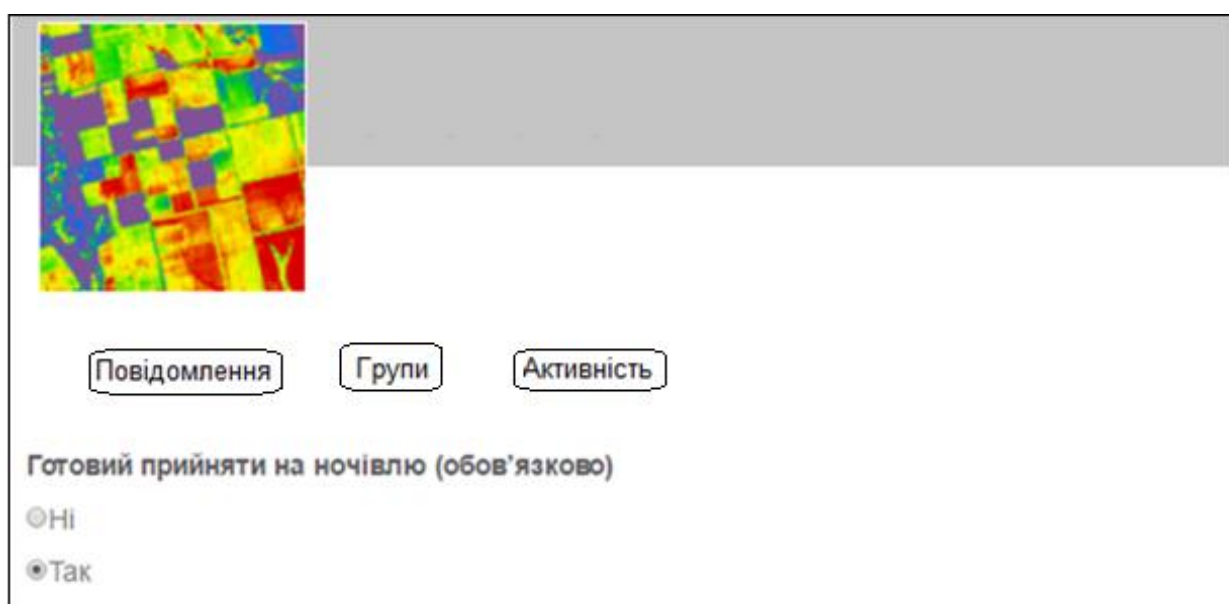


Рисунок 1.21 – Розкадровка сторінки «Надання житла»

Таблиця 1.11

Характеристика варіанту використання «Перегляд профілю»

Контекст використання	Профіль
Дійові особи	Користувач
Передумова	Авторизований користувач
Тригер	Відкритий веб-сайт
Сценарій	<ol style="list-style-type: none"> 1. Відкриття вкладки «Мій профіль» 2. Перегляд інформації профілю 3. Редагування інформації натиснувши “змінити” та підтвердити зміни. 4. Зміна аватару користувача.

Зайшовши на свій профіль можна змінити аватар, а також відредагувати дані, які вводились нами при реєстрації.

На рисунку 1.22 відображено розкадровку сторінки “Перегляд профілю”

The diagram shows a wireframe of a user profile page. On the left, there is a rounded rectangular box labeled 'Аватар' (Avatar) with a 'Змінити' (Change) button below it. On the right, there is a larger rectangular box containing three input fields. The first field is labeled 'ПІП' (First Name), the second 'Місце проживання' (Place of residence), and the third 'Хоббі' (Hobbies).

Рисунок 1.22 – Розкадровка сторінки «Перегляд профілю»

Таблиця 1.12

Характеристика варіанту використання «Активність»

Контекст використання	Активність
Дійові особи	Користувач
Передумова	Авторизований користувач
Тригер	Відкритий веб-сайт
Сценарій	<ol style="list-style-type: none"> 1. Відкриття вкладки «Активність» 2. Перегляд активності користувачів 3. Вибір фільтру відображення контенту активності веб-сайту

Відкривши вкладку активність, нам відобразиться список користувачів та їх дії на сайті. На рисунку 1.23 відображено розкадровку сторінки «Активність».

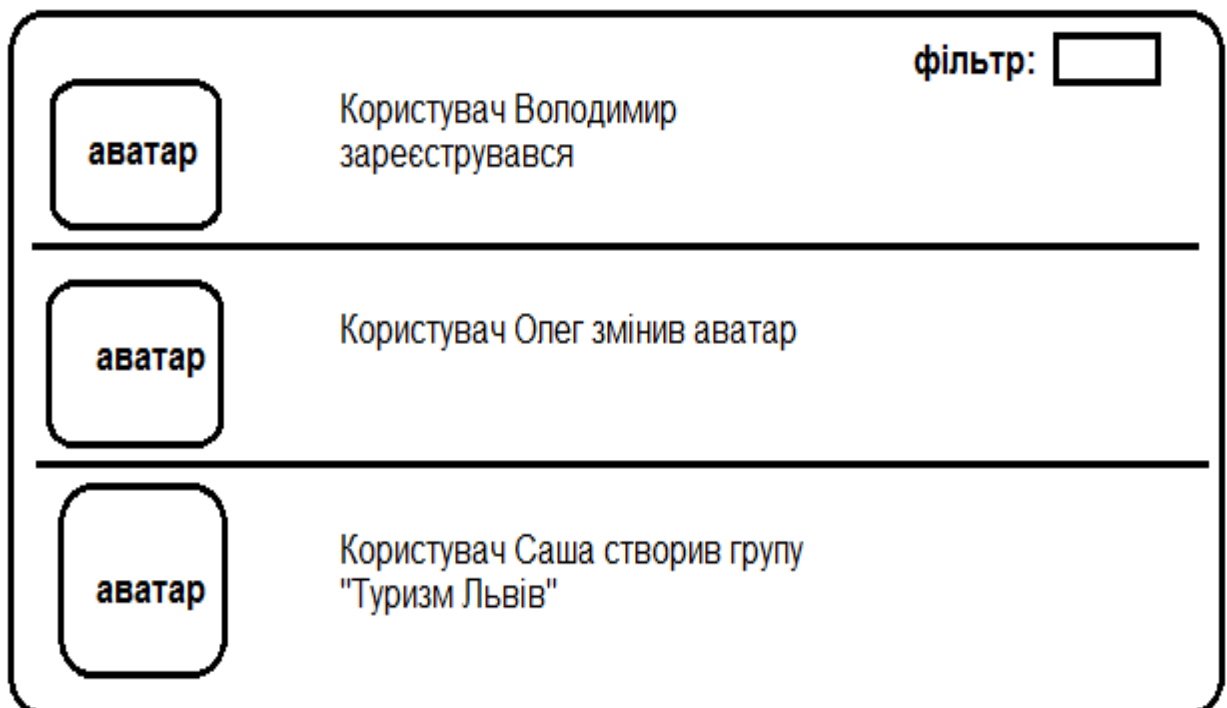


Рисунок 1.23 – Розкадровка сторінки «Активність»

Таблиця 1.13

Характеристика варіанту використання «Групи»

Контекст використання	Групи
Дійові особи	Користувач
Передумова	Авторизований користувач
Тригер	Відкритий веб-сайт
Сценарій	<ol style="list-style-type: none"> 1. Відкриття вкладки “Групи” 2. Перегляд груп 3. Здійснення пошуку груп за її назвою 4. Вибір фільтру сортування груп 5. Створення теми в групі натиснувши
Пост-умова	Групи переглянуті, тему створено.

На рисунку 1.24 відображено розкадровку сторінки “Перегляд груп”. Тут користувач матиме змогу переглянути список груп, а також залишати записи у групах.

Групи

фільтр:

Перегляд 1 групи:

Туризм у Тернополі
Активність: два дні назад

Рисунок 1.24 – Розкадровка сторінки «Перегляд груп»

Таблиця 1.14

Характеристика варіанту використання «Групи»

Контекст використання	Групи
Дійові особи	Адміністратор
Передумова	Права адміністратора
Тригер	Відкритий веб-сайт
Сценарій	1. Відкриття вкладки “Групи” 2. Перегляд груп 3. Створення групи натиснувши “створити групу” в адмін панелі
Пост-умова	Групи переглянуті, групу створено.

На рисунку 1.25 відображено розкадровку сторінки “Груп”. Адміністратор має можливість створювати групи. А також контролювати контент в них.

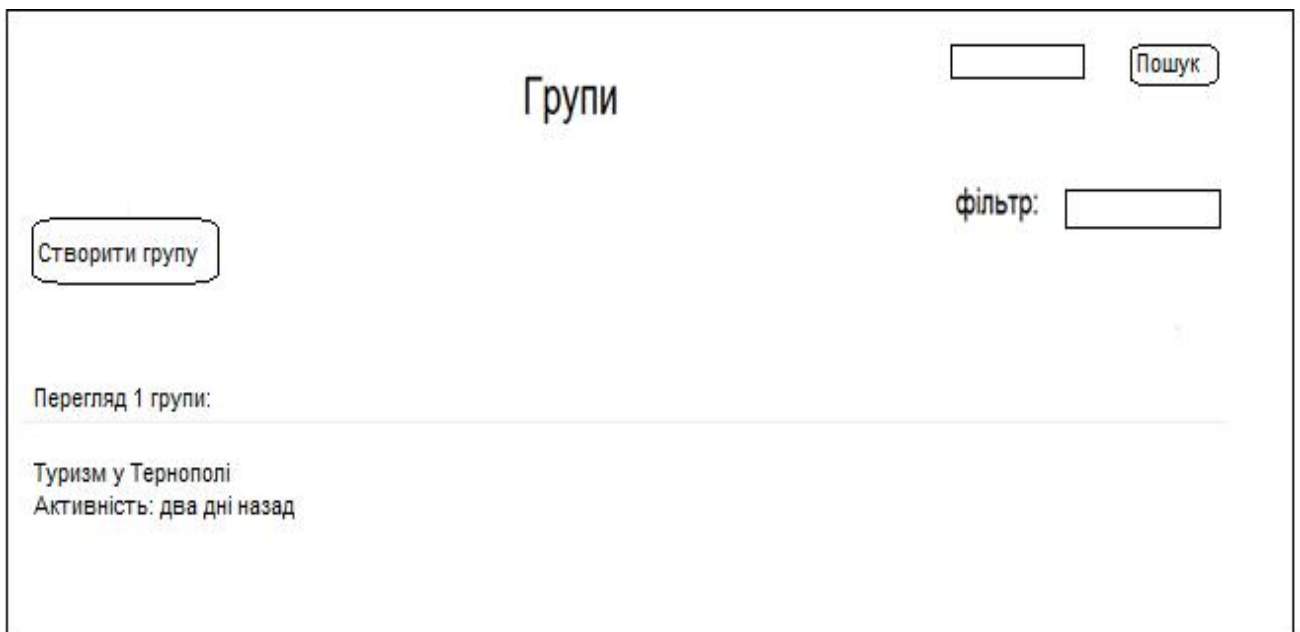


Рисунок 1.25 – Розкадровка сторінки «Групи»

Таблиця 1.15

Характеристика варіанту використання «Активація»

Контекст використання	Активація
Дійові особи	Адміністратор
Передумова	Права адміністратора
Тригер	Відкритий веб-сайт
Сценарій	<ol style="list-style-type: none"> 1. Відкриття вкладки “Користувачі” на адмін панелі 2. Позначення користувачів та вибір дії над ними: активація, видалити. 3. Призначення статусу користувачеві: підписник, учасник, адміністратор.
Пост-умова	Групи переглянуті, групу створено.

На рисунку 1.26 відображено розкадровку сторінки “ Активація”. У адміністратора веб-сайту є можливість активувати користувачів, тобто підтверджувати їхню реєстрацію.

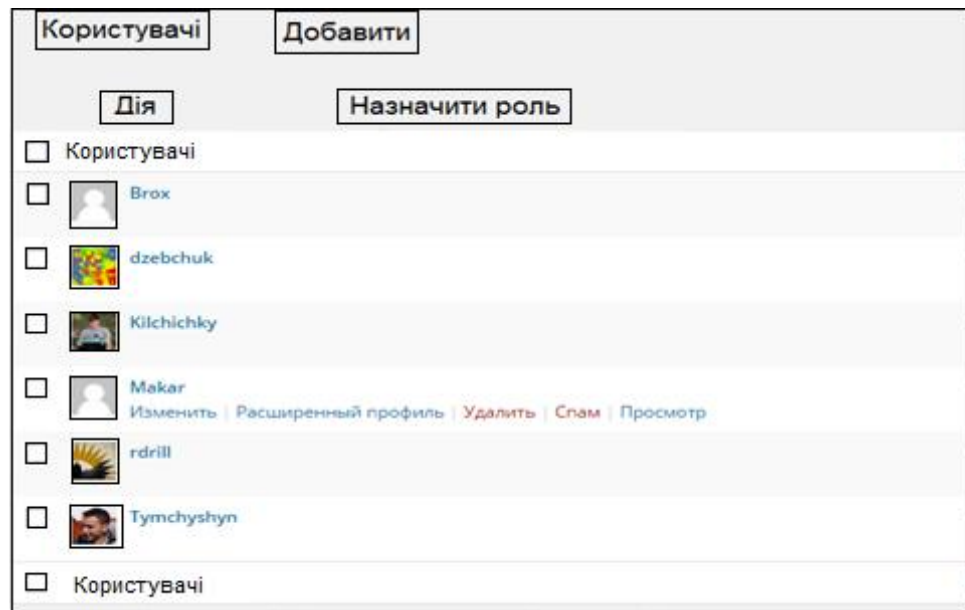


Рисунок 1.26 – Розкадровка сторінки «Активація»

Таблиця 1.16

Специфікація функціональних вимог

Ідентиф. вимог	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
1.	Перегляд головної сторінки	Високий	Середня	Користувач
2.	Перегляд функції «Реєстрація»	Високий	Середня	Користувач
3.	Перегляд функції «Авторизація»	Високий	Середня	Користувач
4.	Перегляд функції «Пошуку»	Високий	Середня	Користувач
5.	Перегляд функції «Резервування місця»	Високий	Середня	Користувач
6.	Перегляд функції «Надання житла»	Високий	Середня	Користувач

7.	Перегляд функції «Створити групу»	Високий	Середня	Адміністратор
8.	Перегляд вкладки «Перегляд активності»	Високий	Середня	Користувач
9.	Перегляд вкладки «Групи»	Високий	Середня	Користувач
10.	Перегляд функції «Активація»	Високий	Середня	Адміністратор

Специфікація нефункціональних вимог наведена у таблиці 1.17

Таблиця 1.17

Специфікація нефункціональних вимог

№	Назва вимоги	Характеристики
1.	Застосовність	
1.1	Час для навчання користувачів	Не більше 10 хв.
1.2	Вимірюваний час відгуку для типових завдань	Не повинно відрізнятися від аналогів
2.	Надійність	
2.1	Середній час безвідмовної роботи	100 год.
2.2	Середнє напрацювання до ремонту	3 місяці
3.	Робочі характеристики	
3.2	Місткість (максимальне значення)	10000 користувачів
4	Проектні обмеження	

4.1	Мова програмування	PHP
5.	Вимоги до документації, призначеної для користувача, і до системи допомоги	Наявність інструкції для користувача

Висновки до першого розділу:

1. Здійснено порівняння трьох веб-сайтів з схожою функціональністю. Було виявлено ряд переваг такі як: простота, зручний інтерфейс, наявність усіх необхідних функцій. Також було виявлено і ряд недоліків: платні функції, відсутність можливості змінити мову.
2. Визначено основні функціональні і нефункціональні вимоги до розроблювальної системи, та запропоновано розробити веб-орієнтовану систему каучсерфінг, в якій врахуються як і переваги так і недоліки аналогових систем.
3. Запроектвані основні бізнес - процеси функціоналу розроблюваної системи, для відображення внутрішніх процесів розроблювальної системи.

РОЗДІЛ 2

ПРОЕКТУВАННЯ WEB-СИСТЕМИ

2.1 Розробка програмної архітектури системи

Для розробки програмної архітектури веб-сайту було обрано клієнт-серверну архітектуру. Цю технологію застосовують для роботи з базами даних в мережі, оскільки з розвитком інтернету величезна кількість даних зберігається в базах даних.

Клієнт-серверна архітектура – це обчислювальна або мережева архітектура, яка передбачає зосередження основної частини ресурсів на серверах, обслуговуючих свої клієнтів. Дана архітектура передбачає такі основні компоненти:

- сервери. Сервер – це об'єкт, який надає інформацію або інші послуги об'єктам мережі, які до них звертаються;
- клієнти. Клієнт – це робоча станція, яка використовує ресурси сервера і надає користувачу зручний інтерфейс;
- мережа. Мережа – це система зв'язку, яка забезпечує взаємодію між клієнтами та серверами[6].

Проект, який розробляється побудований на клієнт-серверній архітектурі. Виділяють його три основні частини:

- зв'язок з БД. БД – це сукупність даних, організованих відповідно до концептуальної структури, яка описує характеристики цих даних і взаємини між ними;
- відображення даних;
- бізнес логіка, вона виконує операції або команди користувача, які він обирає[7].

Обробка та збереження даних відбувається на стороні сервера, відображеннях даних виконується браузером на стороні клієнта.

Вище згадана архітектура показана на рисунку 2.1

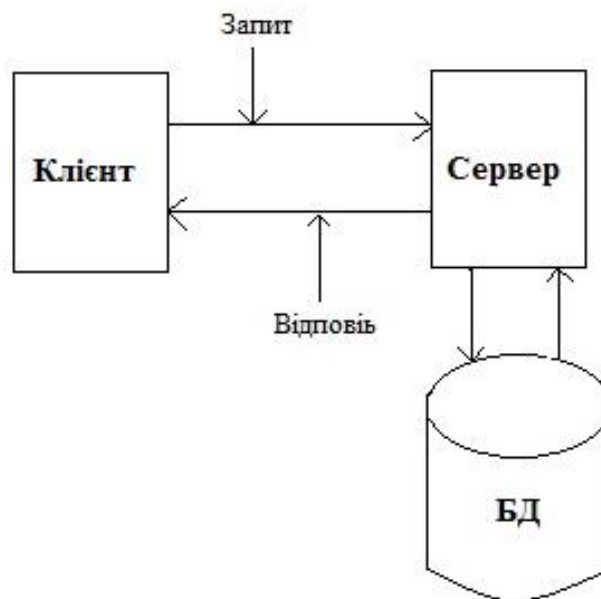


Рисунок 2.1 – Клієнт-серверна архітектура

По даному рисунку зрозуміло, що клієнт – це браузер. Браузер надсилає на сервер запити, у відповідності до дії користувача.

Сервер знаходиться на другому рівні. На сервері відбувається обробка запитів і команд, які надходять з клієнтської сторони.

Третій рівень тут представляє база даних, яка через СУБД приймає sql-запити і повертає потрібні дані на сервер[7].

Наступним етапом проектування буде створення UML-діаграми станів, DFD діаграми.

Діаграма потоків даних — модель проектування, графічне представлення «потоків» даних в інформаційній системі. Діаграма потоків даних також може використовуватись для візуалізації процесів обробки даних (структурне проектування)[9].

На рисунку 2.2 показано діаграму станів системи. Діаграма станів показує, як система переходить з одного стану в інший, а також показує деталі реалізації операцій, виконуваних системою.

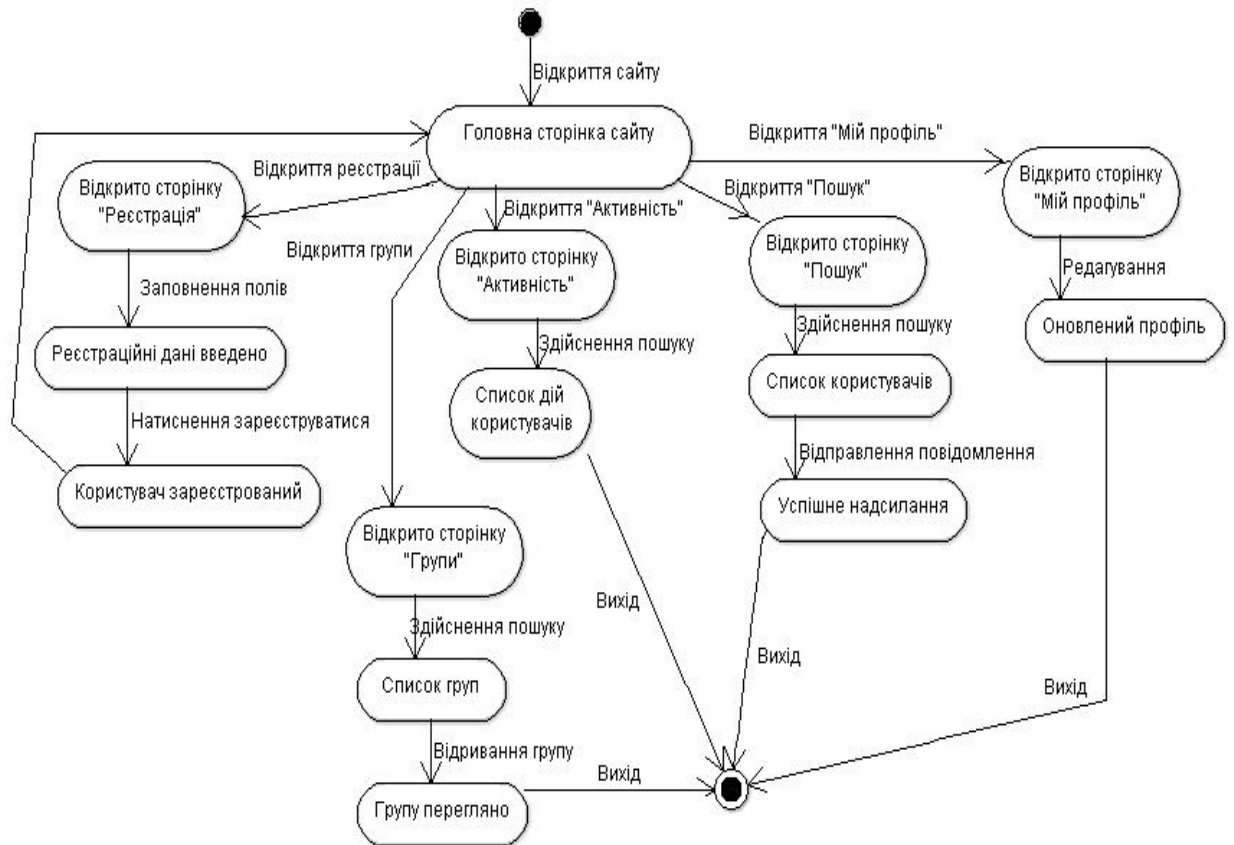


Рисунок 2.2 - UML-діаграма станів

Як ми можемо бачити на діаграмі станів веб-сайту робота починається з відкриття головної сторінки сайту. Далше для того, щоб користувач ефективно використовував функції сайту йому необхідно бути авторизованим. Авторизація необхідна для здійснення пошуку місця ночівлі чи надання місця користувачу.

Далше після авторизація доступна функція пошуку місця для резервування, після заповнення полів пошуку та підтвердження дій.

Також користувачу доступні групи, тут він матиме змогу знайти потрібну йому групи в якій буде надана можливість для спілкування з іншими учасниками.

Перегляд активності – та функція, яка надає змогу переглядати, хто і що з користувачів змінив.

UML-діаграма класів – це діаграма, яка відображає бізнес-логіку розроблюваної системи. На рисунку 2.3 відображена структура класів. Ця діаграма відображає класи спроектованої системи[7].

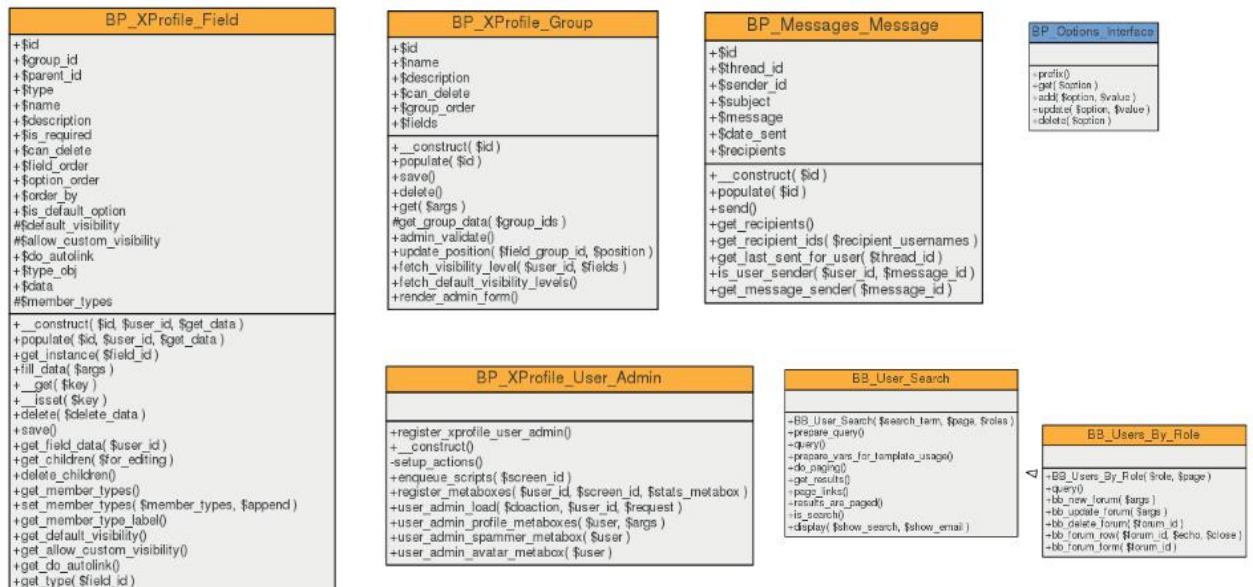


Рисунок 2.3 - Структура діаграми класів

На рисунку 2.3 клас «BP_Xprofile_Field» відповідає за надання таких функцій: відображення інтерфейсу, додавання даних про користувача, редагування та інших функцій пов'язаних з користувачем. Наступний клас «BP_XProfile_Group» тут реалізована функція перегляду, створення груп, можливість у листуванні та інших функцій пов'язаних з контентом групи. У класі «BP_Messages_Message» реалізовані функції перегляду повідомлення, надсилення повідомлення обраному користувачу та інші функції пов'язані з листуванням на сайті. У класі «BP_User_Search» реалізовані функції, які допомагають здійснювати пошук користувачів за обраними критеріями.

2.2 Проектування структури бази даних

У ході розробки системи етапом без якого не обійтись є проектування бази даних, яка забезпечить зберігання даних і дасть змогу проводити операції над ними. Для того, щоб почати проектувати БД нам необхідно

визначити, що саме ми будемо зберігати та для чого, щоб потім в подальшому мати змогу виконувати операції над збереженими даними.

Для здійснення пошуку місця на ночівлі користувачу буде необхідно ввести критерії пошуку у відповідні поля, так як: місто, мова. Дані поля будуть зберігатись у відповідну таблицю, яку ми створимо завчасно. Також усі дані та зміни користувачами профіля будуть зберігатись.

Діаграми потоків даних (Data flow diagram, DFD) використовуються для опису документообігу та обробки інформації. Метою DFD є відобразити, як кожна робота перетворює вхідні дані у вихідні, а також виявити відносини між цими роботами.

Для представлення основного потоку даних в нашій системі використана діаграма DFD, яка представлена на рисунку 2.4.

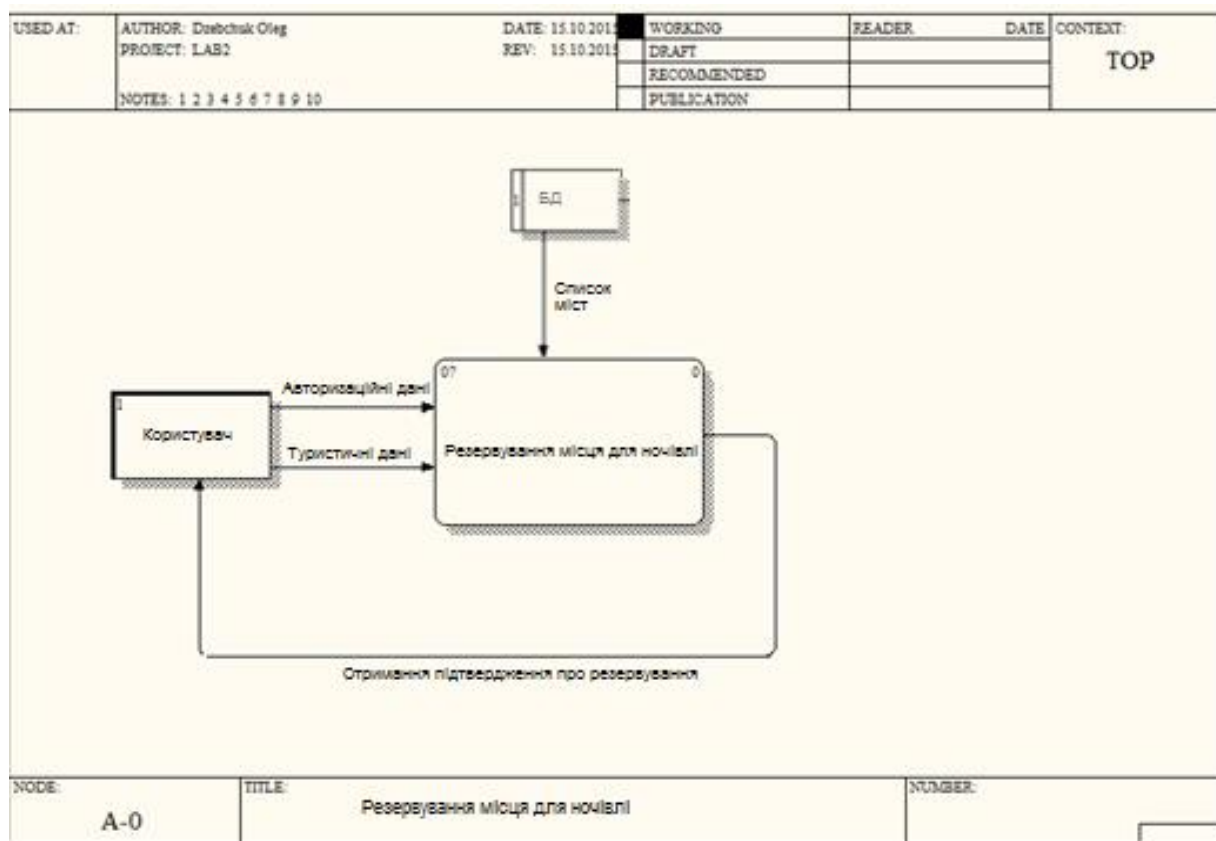


Рисунок 2.4 - Структура діаграми потоків даних DFD

Для опису основних процесів та інформаційних потоків проведемо декомпозицію верхнього рівня діаграми потоків даних, яку зображена на рисунку 2.5.



Рисунок 2.5 - Структура декомпозиція діаграми DFD

Згідно діаграми даних на вході в систему, отримуються користувацькі дані відповідно до яких здійснюється пошук та резервування. Після чого на вихід системи поступає підтвердження резервування від користувача з даними про місце, дату. Відповідно до отриманих даних розроблено відповідну базу даних. Для розробки сховища зберігання даних обрана реляційна модель бази даних.

Реляційна модель бази даних представляє собою централізоване сховище таблиць, що забезпечує швидкий та одночасний доступ до інформації з боку багатьох користувачів. У рядках таблиць частина полів містить дані, стосовно запису, а частина - посилання на записи інших таблиць. Таким чином, зв'язки між записами є невід'ємною властивістю реляційної моделі.

Реляційна модель дає нам структурну та інформаційну незалежність. Тобто тут записи зв'язані так, що зміна одного не впливає на інший, а зміна структури бази даних не приводить до перекомпіляції працюючих з нею додатків[6].

Наступним не менш важливим етапом є розробка діаграми корпоративної моделі даних, яка повністю відобразить елементи, які будуть в нашій базі даних та зв'язки між ними. Діаграма відображена на рисунку 2.6.

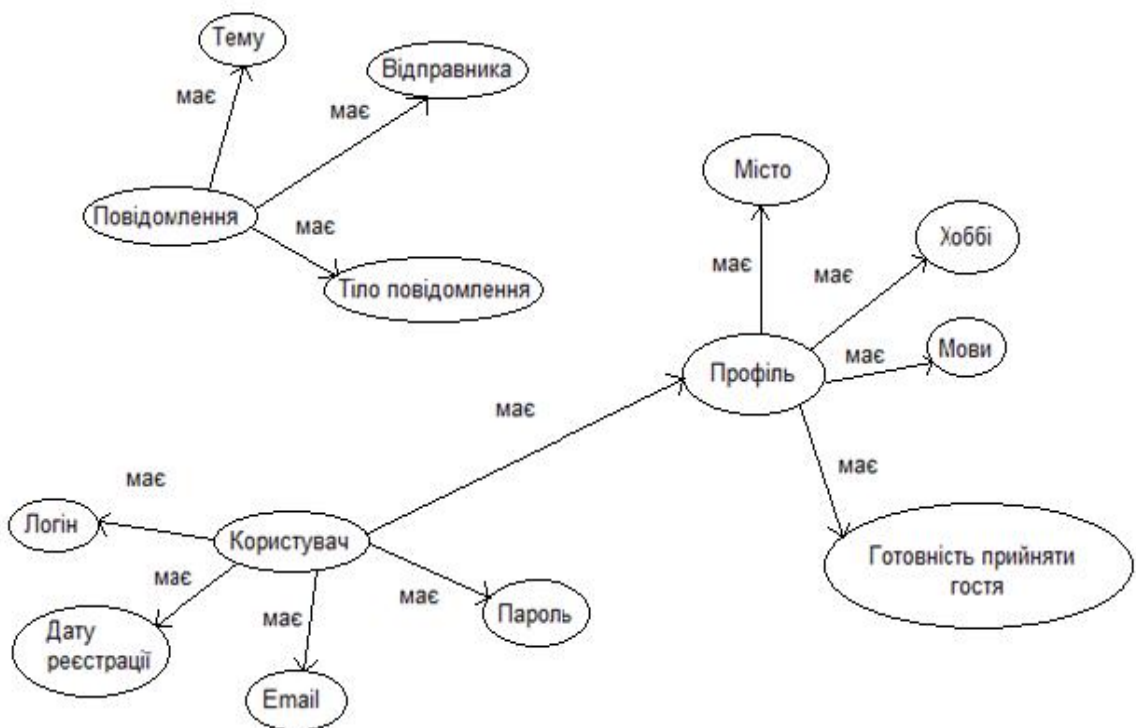


Рисунок 2.6-Структура діаграми корпоративної моделі даних

Отримавши усі необхідні дані розробляється реляційна модель даних, ця модель відображена у вигляді ERD, яка в свою чергу показує відношення та зв'язки між ними. Діаграма ERD відображена на рисунку 2.7.

Фізична реалізація входить у проектування бази даних. Фізичне проектування включає побудову ERD. Для фізичного проектування обрано

базу даних MySQL, оскільки буде використовуватись як сервер в розроблюваній системі.

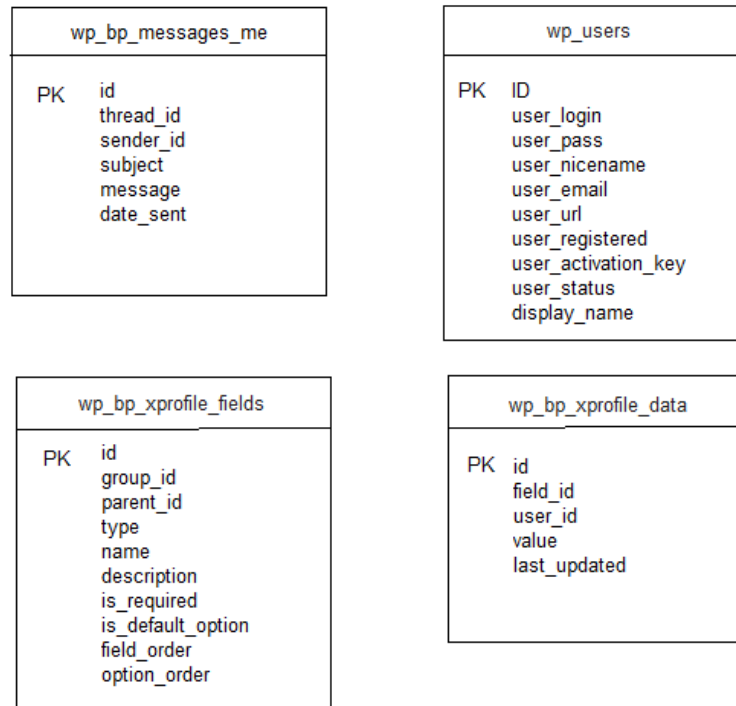


Рисунок 2.7 - Структура діаграми ER

Перед створення веб-сайту спочатку необхідно видалити таблиць в БД для зберігання інформації. Тому під час проектування БД мною були виділено наступні таблиці:

- wp_users . Тут першопочаткові ID, туди реєструються користувачі;

На рисунку 2.8 відображено створення таблиці wp_users.

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

Сортувати за ключем: Жодного

+ Параметри

ID	user_login	user_pass	user_nicename	user_email	user_url	user_registered	user_activation_key	user_status	display_name
1	dzebuchuk	\$P\$BdeDAAI9Tt	dzebuchuk	dzebuchuk13@gmail.com		2016-04-26 16:04:23		0	dzebuchuk
2	Brox	\$P\$BTBy2UzW5n	brox	dzebuchuk@ukr.net		2016-04-27 06:59:42	1461740383:\$P\$BeC	0	Юлер Дзебчук!
3	rdrill	\$P\$Boh8pIkgzhl	rdrill	stpmsuit@gmail.com		2016-05-16 20:49:57		0	Vasya
4	Kilchichky	\$P\$BYMjgSyGAM	kilchichky	Kilchichky@gmail.com		2016-05-17 17:38:05		0	Володимир
5	Makar	\$P\$BA0B0xYmlxt	makar	Natalia@gmail.com		2016-05-17 17:39:52		0	Наталія
6	Tymchyshy	\$P\$BdJdqo15RA	tymchyshyn	Tymchyshyn@gmail.com		2016-05-17 17:41:48		0	Тимчишин Василь

Check all | 3 відміченими: Редагувати | Копіювати | Видалити | Експортувати

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

Рисунок 2.8 – Створення таблиці wp_users.

- wp_bp_xprofile_fields. Тут поля до профілю користувача (країна, місто, інтереси, шкідливі звички);

На рисунку 2.9 відображено створення таблиці wp_bp_xprofile_fields.

	id	group_id	parent_id	type	name	description	is_required	is_default_option	field_order	can_delete	option_order	order_by
	10	1	7	option	Так		0	1	0	1	1	
	11	1	7	option	Ні		0	0	0	1	2	
	7	1	0	radio	Готовий прийняти на ночівлю		1	0	0	1	0	asc
	4	1	0	textarea	Хобі		1	0	4	1	0	
	5	1	0	textarea	Шкідливі звички		1	0	5	1	0	
	6	1	0	textarea	Володіння мовами	Перелічіть через кому мови, якими володієте	1	0	6	1	0	
	1	1	0	textbox	Ім'я		1	0	1	0	0	
	2	1	0	textbox	Країна		1	0	2	1	0	
	3	1	0	textbox	Місто		1	0	3	1	0	

Рисунок 2.9 – Створення таблиці wp_bp_xprofile_fields.

- wp_bp_xprofile_data. Тут дані кастомних полів;

На рисунку 2.10 відображено створення таблиці wp_bp_xprofile_data.

	id	field_id	user_id	value	last_updated
	1	1	1	dzebchuk	2016-06-02 22:41:51
	2	1	2	Юлег Дзєбчук!	2016-05-16 19:52:48
	3	1	3	Vasya	2016-05-17 00:35:13
	4	2	3	Україна	2016-05-17 00:35:13
	5	3	3	Київ	2016-05-17 00:35:14
	6	4	3	Велоспорт	2016-05-17 00:35:14
	7	2	1	Україна	2016-06-02 22:41:51
	8	3	1	Київ	2016-06-02 22:41:51

Рисунок 2.10 – Створення таблиці wp_bp_xprofile_data.

- wp_bp_messages_messages. Тут повідомлення користувачів;

На рисунку 2.11 відображено створення таблиці wp_bp_xprofile_data.

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

Сортувати за ключем: Жодного

+ Параметри

	id	thread_id	sender_id	subject	message	date_sent
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	1	1	1	Лоло	привіт	2016-05-29 23:45:00
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	2	2	1	Ночівля	12.06.2016 буду в 12:00	2016-06-02 21:43:49
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	3	3	1	ніч	резерв	2016-06-02 22:41:29
<input type="checkbox"/> Редагувати <input type="checkbox"/> Копіювати <input type="checkbox"/> Видалити	4	4	1	тест2	прив	2016-06-02 22:51:34

↑ Check all | З відміченими: Редагувати Копіювати Видалити Експортувати

Показати все | Число рядків: 25 | Фільтрувати рядки: Шукати в таблиці

Рисунок 2.11 – Створення таблиці wp_bp_messages_messages.

Висновки до другого розділу:

1. Побудована UML-діаграма станів, яка відображає стани системи при виконанні операцій.
2. Побудована діаграма потоків даних DFD розроблюваної системи, яка відображає роботу системи при виконанні різних операцій.
3. Побудовано діаграму класів, яка дасть змогу програмно реалізувати заплановані функції.
4. Спроектвано таблиці системи дасть змогу розробнику поглянути, які дані мають зберігатися в базі даних і як вони повинні взаємодіяти між собою.

РОЗДІЛ 3

ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Програмна реалізація проекту

Розроблювана система буде представляти собою веб-сайт, який базується на клієнт- серверній архітектурі. Тобто, усі операції, які виконус користувач на сайті, будуть обробляються на серверній частині, де відповідно і зберігатися, а із клієнтської сторони це сайт. Клієнт зможе взаємодіяти із системою через будь-який браузер.

У системі є адміністратор і користувач. Адміністратор має тільки можливість слідкувати за контентом веб-сайту, а також створювати групи.

Всі авторизовані користувачі системи мають рівні права доступу, але для ідентифікації особи, яка зайшла на сайт необхідно пройти авторизацію. Не зареєстрований користувач не матиме можливості авторизуватися, а не авторизований користувач буде позбавлений функції пошуку місця для ночівлі, надання місця для ночів, резервування місця.

Зареєстрований користувач має права на здійснення пошуку користувачів, які надають місце для ночівлі, самостійно надавати місце для ночів, здійснювати резервування місця проживання та всі інші загальнодоступні функції, які надає дана веб-система, зокрема: буде надана можливість брати участь у групах, перебувати у чаті, переглядати останні публікації.

Якщо необхідно розробити інтернет-сайт, відмінний від елементарного ПП, вам не обійтись без використання серверної мови програмування, найбільш популярною серед яких є PHP (Hypertext Preprocessor, попередня

назва: Personal Home Page Tools) – скрипкова мова програмування, призначена для генерації HTML-сторінок на стороні сервера. PHP інтерпретується веб-сервером в HTML-код, який передається у браузер. Основні переваги PHP – гнучкість, ефективність[14].

PHP використовується не лише у поєднанні з HTML, але й із JavaScript, WML, XML та іншими мовами програмування. PHP-код може передаватися будь-яким браузерам і пристроям, в тому числі телефонам, портативним комп'ютерам. PHP-код можна виконувати в режимі командного рядка

Середовище розробки WordPress— це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення веб-сайтів. Сфера застосування — від блогів до складних веб-сайтів. Вбудована система тем і плагінів в поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які веб-проекти. Написана на мові програмування PHP з використанням бази даних.

Дизайн, управління системою та інші можливості :

- простота встановлення, простота налаштувань;
 - підтримка веб-стандартів (XHTML, CSS);
 - модулі для підключення (плагіни) з унікально простою системою їх взаємодії з кодом; можливість автоматичного встановлення та оновлення версії безпосередньо з панелі адміністратора;
 - підтримка так званих «тем», з допомогою яких легко змінюється як зовнішній вигляд, так і способи виведення даних;
 - можливість редагувати шаблони одразу в панелі адміністратора;
 - потенціал архітектури дозволяє легко реалізовувати складні рішення[1];
- Загальна структура проекту.

Веб-система складається з файлів з розширенням “.php”, в яких міститься зовнішній вигляд сайту та його функціонал.

HTML код, який відповідає за візуалізацію нашої системи, а за стиль CSS код. Будь-який сайт, на якій би мові він не був би написаний все одно браузером перекодується в HTML код для відображення сторінки сайту.

Усі сторінки веб-сайту «Couchsurfing по українськи» містить HTML код, який містить зовнішній вигляд системи. Нижче наведений код, який відповідає за частину зовнішнього відображення вигляд сайту:

```
<html >
<style>
  body{
    background-image: url (Psh.jpg);
    background-size: cover;
  }
</style>
<title> Couchsurfing по українське</title>
<form          method="post"          enctype="mul ti part/form-data"
acti on="Avtori z. php">
  <body>
  <meta charset="utf-8">
  <div id="nadpus",div style="text-align: center;">
  <B>
  <span          style="font-family: Monotype          Corsiva; font-
size: 68px; color: black; text-shadow: shadow1, shadow2, shadow3; text-shadow: 4px
3px 0px #fff, 9px 8px 0px rgba(0,0,0,0.15);"> Couchsurfing по українське,
мережа гостинності.
  </span>
  </B> </div> <br> <br> <br> <br>
  <link href="1.css" rel="stylesheet">
  <div id="button",div style="text-align: center;">
  <li id="menu-item-152" class="menu-item menu-item-type-post_type menu-
item-object-page menu-item-152">
    <a href="http://local host/couchsurfing/stream/">Активність</a>
  </li>
    <li id="menu-item-155" class="menu-item menu-item-type-post_type
menu-item-object-page menu-item-155">
  <a
href="http://local host/couchsurfing/%d0%b3%d1%80%d1%83%d0%bf%d0%b8/">Групи</
a></li>
  <li id="menu-item-158" class="menu-item menu-item-type-post_type menu-item-
object-page menu-item-158">
  <a href="http://local host/couchsurfing/members/">Пошук</a></li>
  <li id="menu-item-159" class="bp-menu bp-messages-nav menu-item menu-item-
type-custom menu-item-object-custom menu-item-159">
```

```

<a
href="http://localhost/couchsurfing/members/dzebchuk/messages/">Повідомлення
</a></li>
    <li id="menu-item-160" class="bp-menu bp-profile-nav menu-item
menu-item-type-custom menu-item-object-custom menu-item-160">
<a href="http://localhost/couchsurfing/members/dzebchuk/profile/">Мій
профіль</a></li>
</ul>

```

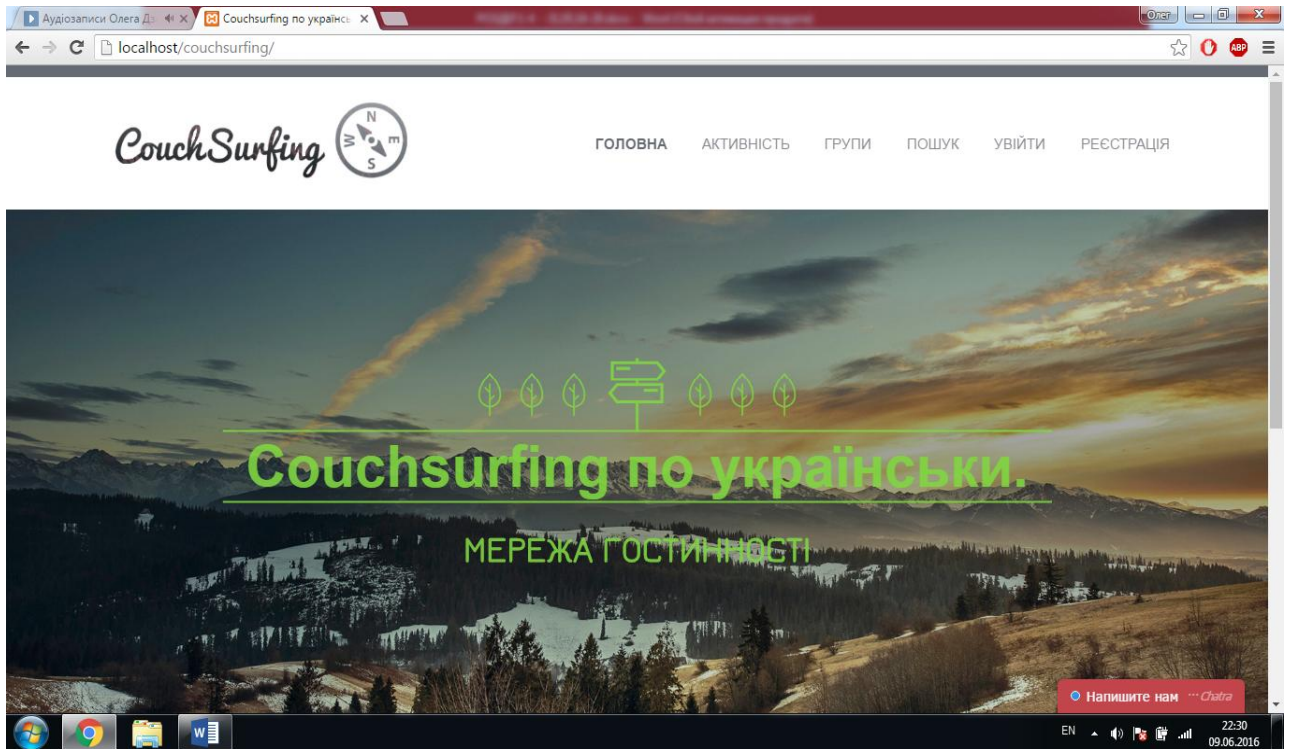


Рисунок 3.1 – Загальна структура сайту

Розглянута частина коду відповідає за розміщення на задньому фоні картинки, розміщення надпису «Couchsurfing» певним стилем, розміром і в певному місці, а також розміщення кнопок, які потрібні, для переходу між сторінками веб-сайту.

За стиль кнопок та фонового відображення, які зображені на рисунку 3.1 відповідає CSS код, який наведений нижче:

```

<style type="text/css">
img.wp-smiley,
img.emoji {
    display: inline !important;
    border: none !important;
    box-shadow: none !important;
    height: 1em !important;

```

```

width: 1em !important;
margin: 0 .07em !important;
vertical-align: -0.1em !important;
background: none !important;
padding: 0 !important;
}
</style>
<link rel='stylesheet' id='theme-my-login-css'
href='http://couchsurfing.local/wp-content/plugins/theme-my-login/theme-my-
login.css?ver=6.4.4' type='text/css' media='all' />
<link rel='stylesheet' id='open-sans-css'
href='https://fonts.googleapis.com/css?family=Open+Sans%3A300italic%2C400ital
ic%2C600italic%2C300%2C400%2C600&#038;subset=latin%2Clatin-
ext%2Ccyrillic%2Ccyrillic-ext&#038;ver=4.5.1' type='text/css' media='all' />
<link rel='stylesheet' id='dashicons-css'
href='http://couchsurfing.local/wp-includes/css/dashicons.min.css?ver=4.5.1'
type='text/css' media='all' />
<link rel='stylesheet' id='admin-bar-css'
href='http://couchsurfing.local/wp-includes/css/admin-bar.min.css?ver=4.5.1'
type='text/css' media='all' />
<link rel='stylesheet' id='dh-fonts-css'
href='//fonts.googleapis.com/css?family=Quattrocento%3A400%2C700%7CQuattrocen
to+Sans%3A400%2C700&#038;subset=latin%2Clatin-ext' type='text/css'
media='all' />
<link rel='stylesheet' id='dh-style-css' href='http://couchsurfing.local/wp-
content/themes/sentio/style.css?ver=35' type='text/css' media='all' />
<style id='dh-style-inline-css' type='text/css'>
.hero-message * { color: #81d742 !important;}
</style>
<script type='text/javascript' src='http://couchsurfing.local/wp-
includes/js/jquery/jquery-migrate.min.js?ver=1.4.0'></script>
<script type='text/javascript' src='http://couchsurfing.local/wp-
content/plugins/theme-my-login/modules/themed-profiles/themed-
<link rel='https://api.w.org/' href='http://couchsurfing.local/?rest_route='
/>
<link rel="EditURI" type="application/rsd+xml" title="RSD"
href="http://couchsurfing.local/xmlrpc.php?rsd" />
<link rel="wlwmanifest" type="application/wlwmanifest+xml"
href="http://couchsurfing.local/wp-includes/wlwmanifest.xml" />
<meta name="generator" content="WordPress 4.5.1" />
<style type="text/css">.recentcomments a{display:inline
!important;padding:0 !important;margin:0 !important;}</style>

```

```

        <style type="text/css" id="custom-background-css">
body.custom-background { background-color: #ffffff; }
</style>
<style type="text/css" media="print">#wpadminbar { display:none; }</style>
<style type="text/css" media="screen">
    * html body { margin-top: 32px !important; }
    @media screen and ( max-width: 782px ) {
        html { margin-top: 46px !important; }
        * html body { margin-top: 46px !important; }
    }
</style>

```

Дальше представлена основна функцію сайту без якої неможливо уявити каучсерфінг – це резервування місця для ночівлі. Для того, щоб зарезервувати місце для ночівлі необхідно здійснити пошук тих користувачів, які надають його, а також підходять під критерії нашого пошуку.

```

function bps_anyfield_search ($f)
{
    global $bp, $wpdb;
    $value = str_replace ('&', '&amp;', $f->value);
    $escaped = '%'. bps_esc_like ($value). '%';
    $sql = array ('select' => '', 'where' => array ());
    $sql ['select'] = "SELECT DISTINCT user_id FROM {$bp->profile->table_name_data}";
    $sql ['where'] [$f->filter] = $wpdb->prepare ("value LIKE %s", $escaped);
    $sql = apply_filters ('bps_field_sql', $sql, $f);
    $query = $sql ['select']. ' WHERE '. implode (' AND ', $sql ['where']);
    $results = $wpdb->get_col ($query);
    return $results;
}

```

Далі ми вибираємо користувача та надсилаємо йому запит

```

$field = wp_cache_get ($field_id, 'bp_xprofile_fields');
if ( false === $field ) {
    $bp = buddypress();

    $field = $wpdb->get_row( $wpdb->prepare( "SELECT * FROM
{$bp->profile->table_name_fields} WHERE id = %d", $field_id ) );

```



```

wp_cache_add( $field->id, $field, 'bp_xprofile_fields' );

if ( ! $field ) {
    return false;
}
}

```

Решта програмного коду реалізації розроблюваної системи знаходиться у додатку Б.

3.2 Програмна реалізація бази даних

Для зберігання даних на веб-сайті, необхідно створити базу даних системи, де будуть зберігатись відповідні дані. Для реалізації системи створено таблиці, де будуть зберігатись відповідні дані.

Приведемо приклад створення відношення бази даних:

```

CREATE TABLE WP_USERS
('id' integer NOT NULL auto_increment,
'user_login' varchar(20) NOT NULL,
'user_pass' varchar(50) NOT NULL,
'user_nicename' varchar(50) NOT NULL,
'user_email' varchar(50) NOT NULL,
'user_url' varchar(50) NOT NULL,
'user_registered' varchar(50) NOT NULL,
'user_status' varchar(50) NOT NULL,
'display_name' varchar(50) NOT NULL,
PRIMARY KEY (`id`),

CREATE TABLE WP_BD_MESSAGES_MESSAGES
('id' integer NOT NULL auto_increment,
'thread_id' varchar(20) NOT NULL,
'sender_id' varchar(50) NOT NULL,
'subject' varchar(50) NOT NULL,
'message' varchar(50) NOT NULL,
'date_send' varchar(50) NOT NULL,
PRIMARY KEY (`id`),

```

```
CREATE TABLE WP_BD_XPROFILE_DATE
('id' integer NOT NULL auto_increment,
'field_id' varchar(20) NOT NULL,
'user_id' varchar(50) NOT NULL,
'value' varchar(50) NOT NULL,
'last_updated' varchar(50) NOT NULL,
PRIMARY KEY (`id`),
```

```
CREATE TABLE WP_BD_XPROFILE_FIELDS
('id' integer NOT NULL auto_increment,
'group_id' varchar(20) NOT NULL,
'parent_id' varchar(50) NOT NULL,
'type' varchar(50) NOT NULL,
'description' varchar(50) NOT NULL,
'is_required' varchar(50) NOT NULL,
'is_default_option' varchar(50) NOT NULL,
'field_order' varchar(50) NOT NULL,
PRIMARY KEY (`id`),
```

Частина функції збереження, оновлення, видалення наведена нижче

```
$is_new_field = is_null( $this->id );
```

```
    if ( ! $is_new_field )
    {
```

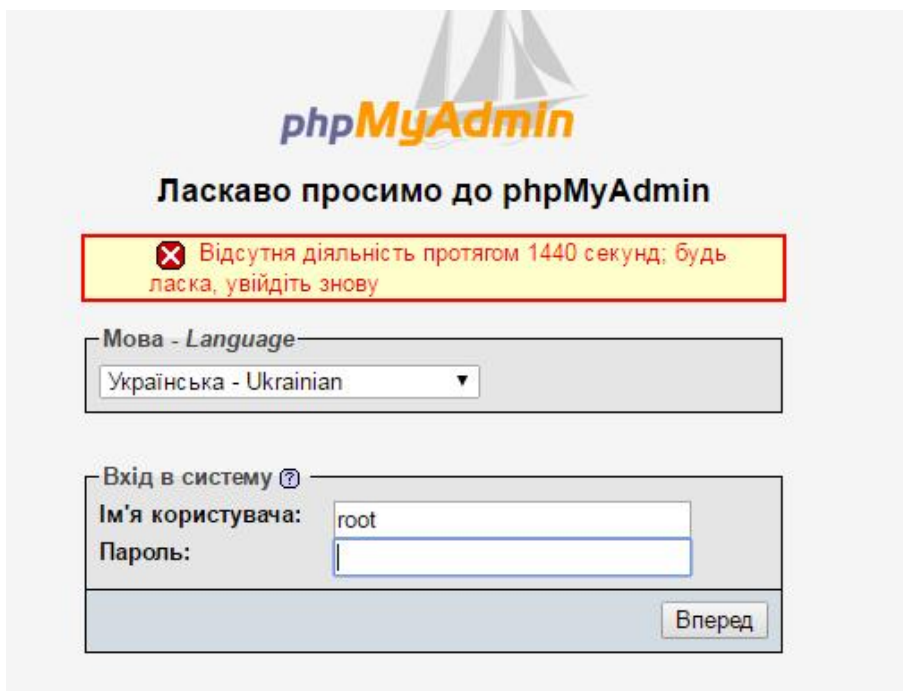
```
        $sql = $wpdb->prepare( "UPDATE {$bp->profile->table_name_fields}
SET group_id = %d, parent_id = 0, type = %s, name = %s, description = %s, is_required
= %d, order_by = %s, field_order = %d, option_order = %d, can_delete = %d,
is_default_option = %d WHERE id = %d", $this->group_id, $this->type, $this->name,
$this->description, $this->is_required, $this->order_by, $this->field_order, $this-
>option_order, $this->can_delete, $this->is_default_option, $this->id );
```

```
    } else {
```

```
        $sql = $wpdb->prepare( "INSERT INTO {$bp->profile-
>table_name_fields} (group_id, parent_id, type, name, description, is_required,
order_by, field_order, option_order, can_delete, is_default_option ) VALUES ( %d, %d,
%s, %s, %s, %d, %s, %d, %d, %d, %d )", $this->group_id, $this->parent_id, $this-
>type, $this->name, $this->description, $this->is_required, $this->order_by, $this-
>field_order, $this->option_order, $this->can_delete, $this->is_default_option );
```

```
$sql = $wpdb->prepare( "DELETE FROM {$bp->profile->table_name_fields} WHERE id = %d OR parent_id = %d", $this->id, $this->id );
```

На рисунках 3.2-3.3 відображено phpMyAdmin- систему для керування таблицями бази даних.



phpMyAdmin

Ласкаво просимо до phpMyAdmin

Відсутня діяльність протягом 1440 секунд; будь ласка, увійдіть знову

Мова - *Language*

Українська - Ukrainian

Вхід в систему ?

Ім'я користувача: root

Пароль:

Вперед

Рисунок 3.2 – Вхід у систему phpMyAdmin

Таблиця	Дія	Рядки	Тип	Порівняння	Розмір	Надмірні видатки
wp_bp_activity	Переглянути Структура Шукати Вставити Очистити Знищити	14	InnoDB	utf8mb4_general_ci	176 КБ	-
wp_bp_activity_meta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_groups	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_groups_groupmeta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_groups_members	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_general_ci	112 КБ	-
wp_bp_messages_messages	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_messages_meta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_messages_notices	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_general_ci	32 КБ	-
wp_bp_messages_recipients	Переглянути Структура Шукати Вставити Очистити Знищити	2	InnoDB	utf8mb4_general_ci	96 КБ	-
wp_bp_notifications	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_general_ci	128 КБ	-
wp_bp_notifications_meta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_xprofile_data	Переглянути Структура Шукати Вставити Очистити Знищити	36	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_bp_xprofile_fields	Переглянути Структура Шукати Вставити Очистити Знищити	9	InnoDB	utf8mb4_general_ci	96 КБ	-
wp_bp_xprofile_groups	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_general_ci	32 КБ	-
wp_bp_xprofile_meta	Переглянути Структура Шукати Вставити Очистити Знищити	18	InnoDB	utf8mb4_general_ci	48 КБ	-
wp_commentmeta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_unicode_ci	48 КБ	-
wp_comments	Переглянути Структура Шукати Вставити Очистити Знищити	1	InnoDB	utf8mb4_unicode_ci	96 КБ	-
wp_fep_messages	Переглянути Структура Шукати Вставити Очистити Знищити	3	InnoDB	utf8mb4_unicode_ci	16 КБ	-
wp_fep_meta	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_unicode_ci	32 КБ	-
wp_links	Переглянути Структура Шукати Вставити Очистити Знищити	0	InnoDB	utf8mb4_unicode_ci	32 КБ	-
Console_options	Переглянути Структура Шукати Вставити Очистити Знищити	256	InnoDB	utf8mb4_unicode_ci	1.9 КБ	-

Рисунок 3.3 – Структура БД

Висновки до третього розділу:

1. Обґрунтовано вибір мови програмування та реалізовано основні функції нашої системи, певні функції описані та наведені прикладом коду, весь код веб-сервісу знаходиться у додатку Б
2. Реалізовано БД, створено ряд таблиць та наведено код пошуку, додавання, видалення з БД.

РОЗДІЛ 4

ТЕСТУВАННЯ ТА ЗОВНІШНЯ ЕКСПЛУАТАЦІЯ

4.1 Тестування

Ключовою частиною реалізації програмного продукту є процес перевірки роботи програми згідно вимог, або тестування.

Тестування ПЗ- це процес дослідження, при якому визначається якість ПП в тому середовищі, яке нам потрібне. Під час перевірки програмного продукту можуть виявлятися дефекти і помилки, які необхідно виправити.

На сьогоднішній день тестування поділяється на два види: ручне тестування та автоматизоване. Ручне тестування – це те тестування, при якому всі операції, які проводяться виконуються вручну, тобто ми самі оцінюємо як працює система, а автоматичне тестування – це те тестування, при якому ПП автоматично проходить по функціях системи, після чого ми матимемо змогу перевірити код реалізації функції і подивитись чи

повертається потрібний нам результат. Цей вид тестування передбачає використання програмних засобів для виконання тестів[13].

Було проведено ручне тестування при якому перевіряли функціональне тестування, а також автоматичне, яке було розроблено за допомогою Selenium.

Тест план по якому проводились усі види тестувань, міститься в додатку В.

Автоматизоване тестування користувацького інтерфейсу проводилось за допомогою програми Selenium, яка встановлена на браузер Mozilla. Суть цього тестування полягає в наступному: програма запам'ятовує дії тестера, які він здійснив швидше в цьому програмному засобі та зможе відтворити їх. Основна перевага полягає в тому, що не прийдеться виконувати тести багато раз, достатньо тільки пройти їх раз, а далі Selenium відтворити їх всі інші рази. Зовнішній вигляд цих всіх операцій і програмного продукту зображено на рисунку 4.1.

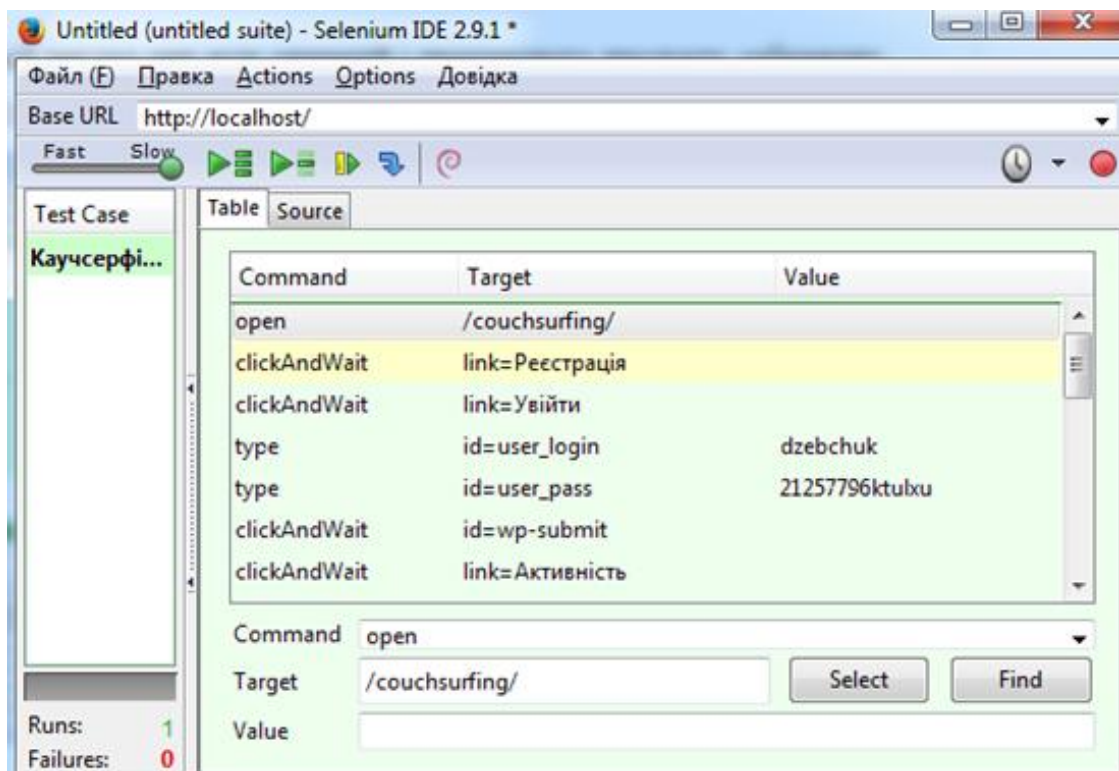


Рисунок 4.1 – Selenium тести

В програмному продукті Selenium , в якому ми проводили тестування нашої системи, тести містять такі команди: open, click and wait, type.

За допомогою цих команд систему можна запрограмувати на виконання тих чи інших функції, які на програмному рівні перевірять систему, чи вона працює вірно[12]. Згідно проведених тестів, якщо перший тест реєстрації буде пройдений з помилкою, то наступні тести пов'язані з реєстрацією будуть не пройдені. Source код одного тесту містить наступну структуру:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
"http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
<head profile="http://selenium-ide.openqa.org/profiles/test-case">
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<link rel="selenium.base" href="http://localhost/couchsurfing" />
<title>New Test</title>
</head>
<body>
<table cellpadding="1" cellspacing="1" border="1">
<thead>
<tr><td rowspan="1" colspan="3">New Test</td></tr>
</thead><tbody>
<tr>
<td>open</td>
<td>/couchsurfing/</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>link=Реєстрація</td>
<td></td>
</tr>
<tr>
<td>clickAndWait</td>
<td>link=Увійти</td>
<td></td>
</tr>
<tr>
<td>type</td>
<td>id=user_login</td>
<td>dzebchuk</td>
</tr>
```

```

<tr>
  <td>type</td>
  <td>id=user_pass</td>
  <td>21257796ktulxu</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=wp-submit</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Активність</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Групи</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>id=groups-order-by</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Nthj</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Пошук</td>
  <td></td>
</tr>
<tr>
  <td>click</td>
  <td>id=field_3</td>
  <td></td>
</tr>
<tr>
  <td>type</td>
  <td>id=field_3</td>
  <td>Одеса</td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>css=input.button</td>
  <td></td>

```



```

</tr>
<tr>
  <td>click</td>
  <td>id=field_7</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Надіслати повідомлення!</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Повідомлення</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=sentbox</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Мій профіль</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>id=edit</td>
  <td></td>
</tr>
<tr>
  <td>clickAndWait</td>
  <td>link=Вийти</td>
  <td></td>
</tr>
</tbody></table>
</body>
</html>

```

Для більш надійнішого результату було проведене ручне тестування, в основу якого покладена перевірка системи з точки зору звичайного користувача, який зможе поставити зауваження до інтерфейсу. Для ручного тестування було використано тест-кейс, в яких описані кроки для проведення перевірки системи. Test-case містить наступну структуру:

- ✓ Id – номер - порядковий номер тесту;
- ✓ Description – опис тесту;
- ✓ Steps to reproduce – кроки для досягнення результату;
- ✓ Expercted Result – очікуваний результат;
- ✓ Actual Result - актувальний результат;
- ✓ Padssed/Failed - показує чи пройшов тест;
- ✓ Enviroment – операційна система.

Тест-кейси до тестування системи знаходяться у додатку В.

Наступним видом тестування було «Тестування безпеки», яке відноситься до ручного тестування. Під час проведення даного виду тестування було перевірено авторизацію, яка згідно ідентифікаторів надає можливість кожному користувачеві увійти в систему під певними правами. Так як в системі є два типи користувачів: учасник і адміністратор, то було проведено авторизацію з кожним з них для перегляду, чи надаються кожному потрібні йому права доступу до системи.

4.2 Розгортання програмного продукту

Для того, щоб розроблювальна система працювала коректно їй необхідний Windows 7 і вище, на яких було перевірено коректність роботи, а також браузер новіших версій. До мінімальних апаратних вимог процесор має бути покоління Intel Celeron 1.8 GHz, оперативної пам'яті 2 Гбайт, відео карта на 512 mb.

4.3 Інструкція користувача

Встановлення:

Користувачу не потрібно нічого встановлювати на ПК, при першому запуску не потрібно встановлювати з'єднання з базою даних MySQL, де буде зберігатися вся інформація, оскільки вона буде зберігатися на сервері. Якщо потрібно змінити структуру бази даних або самої системи, потрібно лише встановити базу даних MySQL, середовище Xhamp, яке дасть змогу компілювати PHP код.

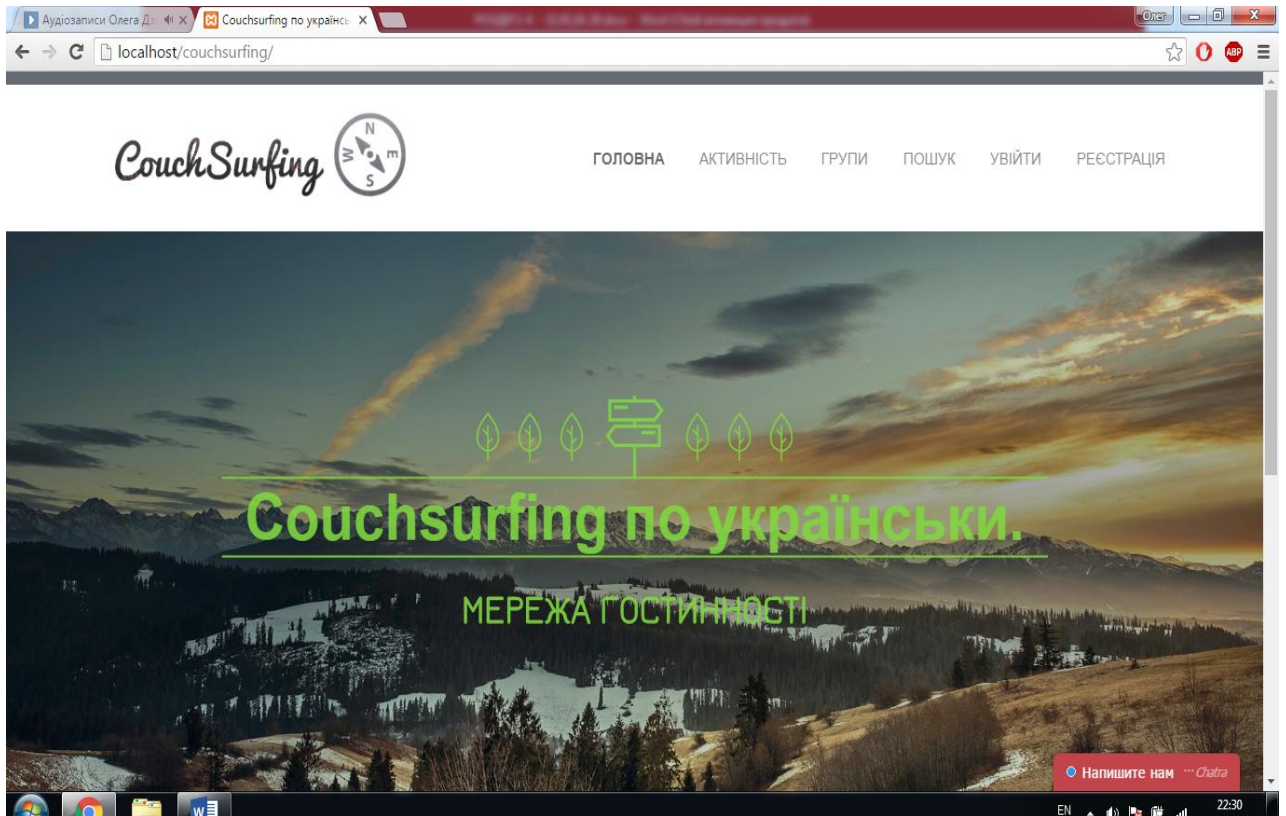
При розробці програмної системи розроблені такі базові функції:

- Реєстрація;
- Пошук місця для ночівлі;
- Надання житла для ночівлі;
- Групи;
- Перегляд активності;
- Активація.

Аналіз помилок:

При виявленні помилок, під час пошуку, перевірте чи правильно ви вводите мови, оскільки там потрібно їх перераховувати через кому, як показано на прикладі під полем вводу.

На рисунку 4.2 відображено вікно головної сторінки сайту. Тут ми бачимо головне меню, а також стіну веб-сайту.



Рисунку 4.2 – Головна сторінка веб-сайту

На рисунку 4.3 відображено вікно “Реєстрація”. Для того, щоб використовувати усі функції, які надає нам веб-сайт, необхідно бути авторизованим у системі. Для цього необхідно пройти процес реєстрації. Процес реєстрації включає в себе заповнення таких обов’язкових полів як:

- Логін;
- Електронна пошта;
- Пароль;
- ПІП;
- Місце проживання(адрес);
- Хоббі;
- Шкідливі звички.

Зареєструватися просто!

Деталі облікового запису

Логін (обов'язково)

Електронна пошта (обов'язково)

Оберіть пароль (обов'язково)

Підтвердіть пароль (обов'язково)

Деталі профілю

Готовий прийняти на ночівлю (обов'язково)

Ім'я (обов'язково)

Країна (обов'язково)

Місто (обов'язково)

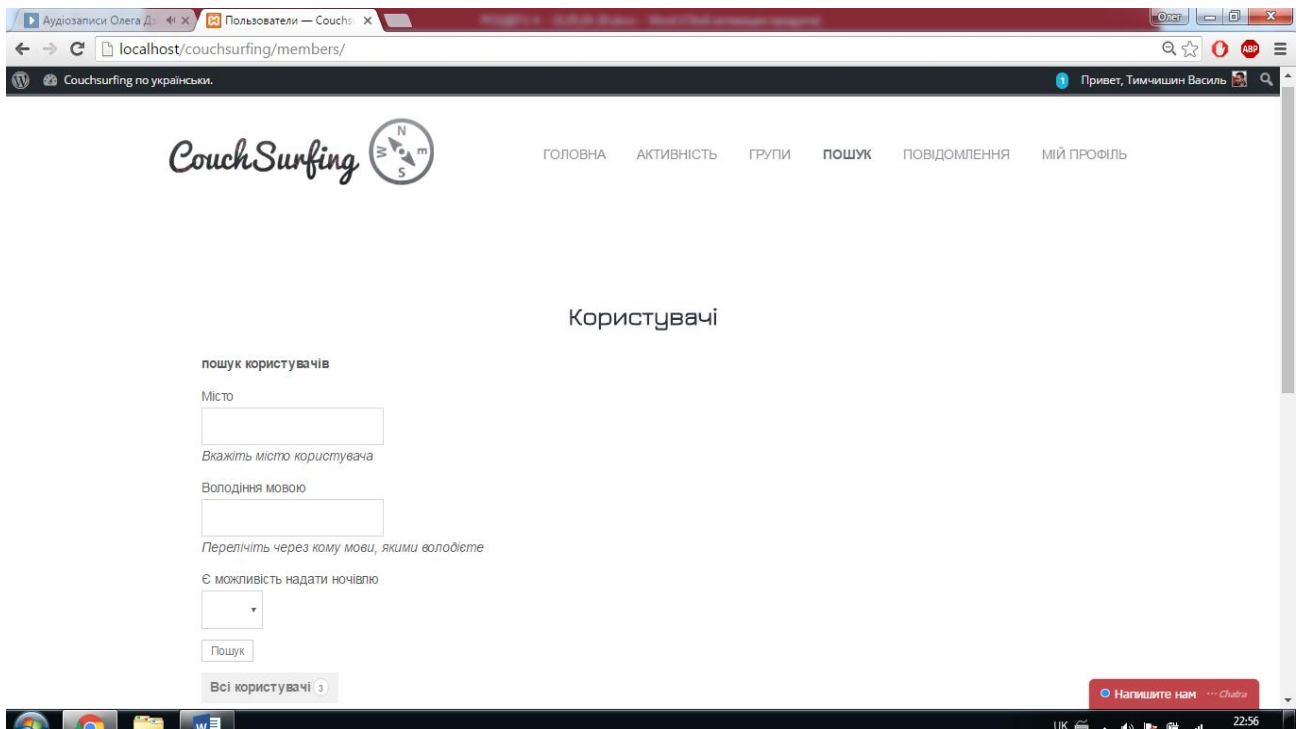
Шкідливі звички (обов'язково)

Володіння мовами (обов'язково)

Рисунку 4.3 – Вікно “Реєстрації” веб-сайту

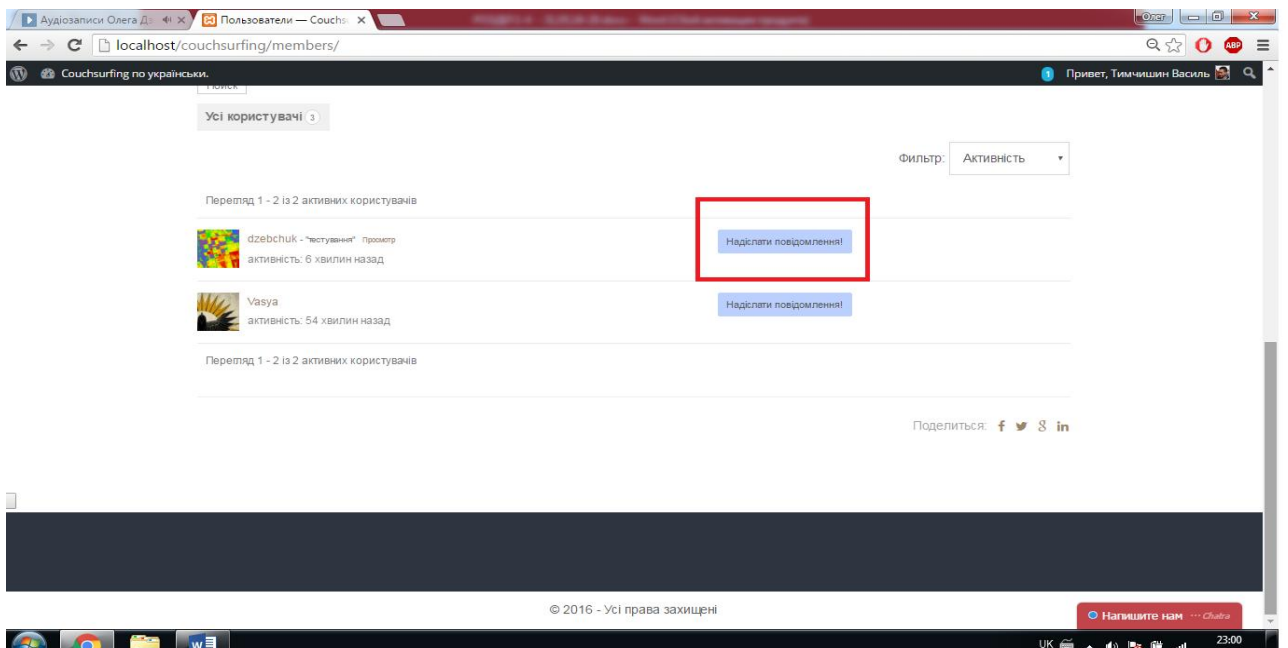
На рисунку 4.4 відображено вікно “Пошук”. Після того, як користувач авторизується у системі він може здійснити пошук місця ночівлі для резервування. Для цього йому необхідно ввести та підтвердити усі необхідні для пошуку поля. Зокрема:

- місто;
- перелік мов;
- фільтр сортування користувачів;
- критерій зайнятості місця.



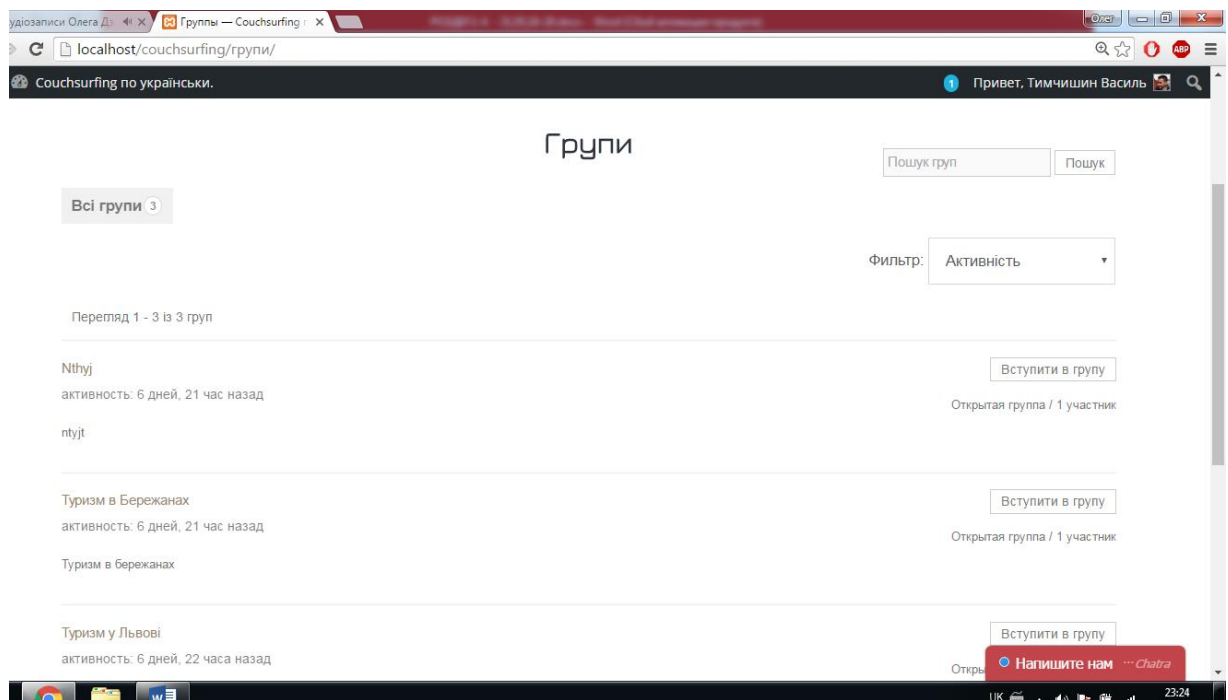
Рисунку 4.4 – Вікно “Пошуку” веб-сайту

На рисунку 4.5 відображено вікно “Надання житла”. Після того, як користувач здійснив пошук, у форму веб-сайту відображається список користувачів. Після чого можемо вибрати з відображеного списку будь-який профіль користувача та відправити йому запит, щоб зарезервувати місця для ночівлі.



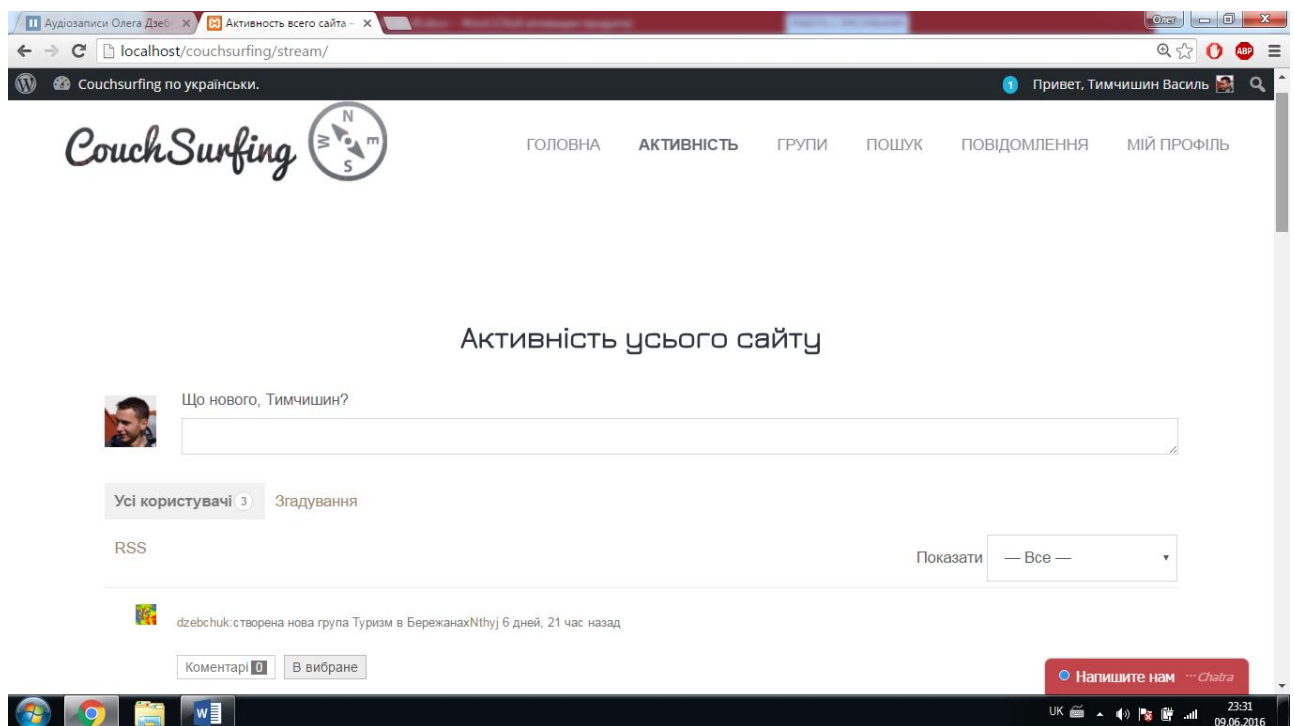
Рисунку 4.5 – Вікно “Резервування житла” веб-сайту

На рисунку 4.6 відображено вікно “Групи”. Групи – це те середовище, де користувачі можуть спілкуватися між собою. Завдяки групам можна знайти попутчиків, що дозволить зекономити кошти.



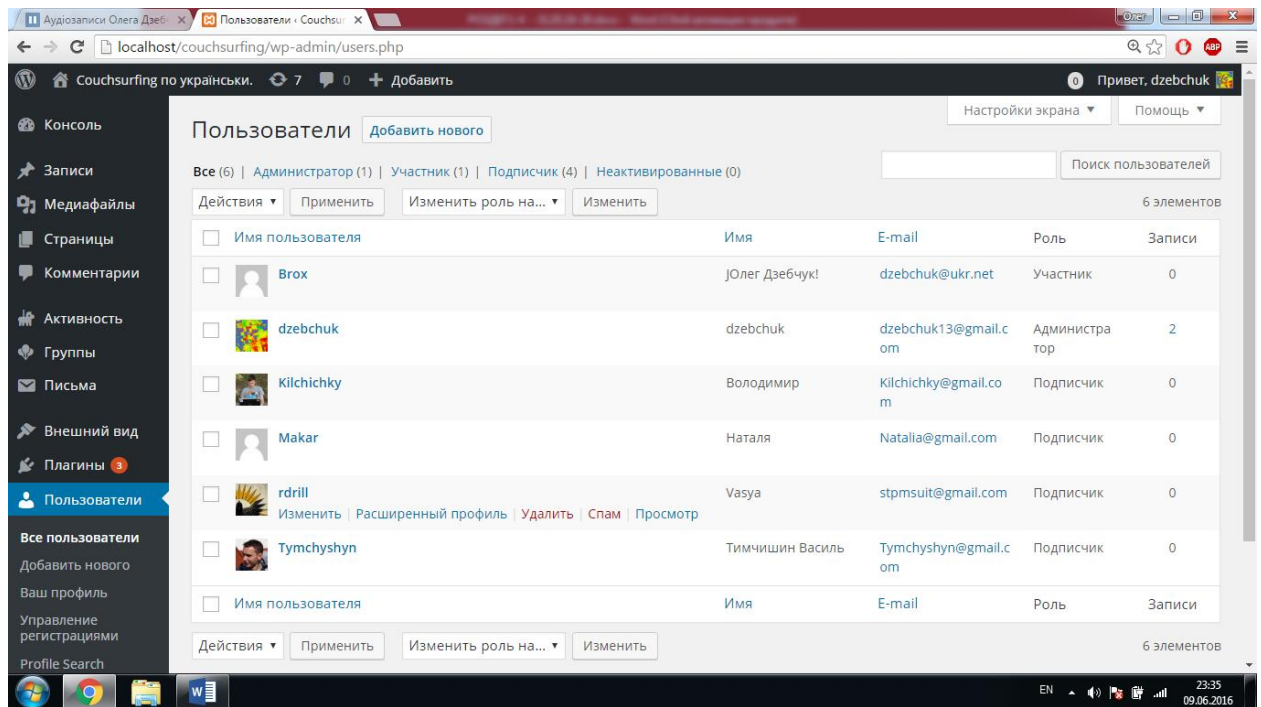
Рисунку 4.6 – Вікно “Групи” веб-сайту

На рисунку 4.7 відображено вікно “Перегляд активності”. Тут є змога відслідковувати за останніми діями користувачів.



Рисунку 4.7 – Вікно “Активність” веб-сайту

На рисунку 4.8 відображено вікно “Активація”. У адміністратора веб-сайту є можливість активувати користувачів, тобто підтверджувати їхню реєстрацію, а також можливість змінити статус користувача, тобто надати підписник, учасник, адміністратор.



Рисунку 4.8 – Вікно адмін панелі “Активація” веб-сайту

Висновки до четвертого розділу:

1. Проведено два види тестування: ручне та автоматичне. Автоматичне проводилось за допомогою Selenium, а ручне проводилось від сторони звичайного користувача по тест-кейсу;
2. Було розроблено інструкцію користувача, в якій описано основні функції системи та необхідні апаратні та програмні засоби для повного функціонування веб-сайту.
3. Складено тест план веб-системи.

ВИСНОВКИ

Під час роботи було розроблено веб-орієнтовану систему каучсерфінг. Реалізовано функції користувача та адміністратора, що забезпечить можливостями здійснення пошуку, резервування житла, надання жила перебування в групах, перегляд активності для користувача. Створена система дасть змогу адміністратору контролювати контент веб-сайту, також є можливість створення груп і активації користувачів. Розроблена система призначена для тих користувачів, які хочуть подорожувати в межах України, бажаючи при цьому економити на житлі.

Веб-сайт розроблявся на основі розглянутих трьох аналогів: “Перша українська мережа гостинності”, “ Всесвітня мережа Каучсерфінг”, та туристична мережа “LiveTravel”. Під час розробки системи було враховано усі плюси та мінуси вище згаданих аналогів.

Розроблено усі необхідні діаграми для проектування та програмування системи та БД. Після реалізації системи на обраних мовах програмування та середовищах, було проведене тестування, яке повинно засвідчити чи виконує система усі нами поставлені функції. Оскільки після проведення тестування ніяких порушень не виявлено, усі функції відповідають поставленим вимогам, тому інтерактивний веб-сайт готовий до повної функціональної роботи, тобто готова система для впровадження.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Середовище WordPress [Электронный ресурс]. – Режим доступа:
<https://uk.wikipedia.org/wiki/WordPress>
2. Н.М. Гаркуша, О.В. Цуканова, О.О. Горошанська. Моделі і методи прийняття рішень в аналізі та аудиті: Навч. посіб. — 2-ге вид. Рекомендовано МОН / Гаркуша Н.М., Цуканова О.В., Горошанська О.О. — К., 2012. — 591 с.
3. Вендров А.М. Проектирование программного обеспечения экономических информационных систем: Учебник. – 2-е изд., перераб. и доп.– М.: Финансы и статистика, 2006. – 544 с: ил.
4. Портал Openbusiness. Портал бізнес-планов, руководств и франшиз для малого и среднего бизнеса. [Электронный ресурс]. — Режим доступа: <http://www.openbusiness.ru/html/dop8/proizvodstvo-spirta.htm>, для доступа до інформаційних ресурсів авторизація не потрібна. — Назва з екрану. www.openbusiness.ru.
5. Д. Пайлон, Н. Питмен. UML 2 для программистов. – Пер. с англ. – Е. Матвеев, 2012. – 240 с.
6. Конноли Т., Бегг К., Строчан А. Базы данных: проектирование, реализация и сопровождение. Теория и практика: Уч. Пособие.- М.: Изддом "Вильяме", 2000.- с.
7. Фаулер М. UML. Основы, 3 е издание. – Пер. с англ. – СПб: Символ Плюс, 2004. – 192 с.
8. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.- М.:ДМК ПРЕСС; СПб.:Питер,2004.-429 с.
9. Бейли Л. - Изучаем SQL. – Исд. Питер, 2012. – 573 с.

10. Сотник С. Л. Конспект лекций по курсу «Основы проектирования систем искусственного интеллекта», 1997-1998.
11. Зандстра М., PHP: объекты, шаблоны и методики программирования, 3-е издание = PHP Objects, Patterns and Practice, Third Edition — М.: «Вильямс», 2010. — С. 560. — ISBN 978-5-8459-1689-1.
12. Суэринг С., Конверс Т., Джойс П. PHP и MySQL. Библия программиста, 2-е издание = PHP 6 and MySQL 6 Bible — М.: «Диалектика», 2010. — 912 с. — ISBN 978-5-8459-1640-2.
13. Кормен Т., Лейзерсон И. Ч., Ривест Р. Л., Штайн К. Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS — 2-е изд. — М.: «Вильямс», 2006. — С. 1296. — ISBN 0-07-013151-1.
14. Кнут Д. Искусство программирования, том 1. Основные алгоритмы = The Art of Computer Programming, vol.1. Fundamental Algorithms — 3-е изд. — М.: «Вильямс», 2006. — С. 720. — ISBN 0-201-89683-4
15. Нильсен Я., Перниче К. Веб-дизайн: анализ удобства использования веб-сайтов по движению глаз = Eyetracking Web Usability — М.: «Вильямс», 2010. — С. 480. — ISBN 978-5-8459-1652-5.
16. Титтел Э., Ноубл Дж. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition — М.: «Диалектика», 2011.

ДОДАТОК А

Глосарій

Таблиця 1

Термін	Опис терміну
Веб- сайт	набір веб-сторінок, які об'єднані за змістом та навігацією. На фізичному рівні сайт може бути як на одному, так і на багатьох серверах.
Веб-додаток	клієнт-серверний додаток, в якому клієнтом виступає браузер, а сервером - веб-сервер. Логіка веб-додатку розподілена між сервером і клієнтом, зберігання даних здійснюється, переважно, на сервері, обмін інформацією відбувається по мережі.
Авторизація	це надане будь-якій особі право на вчинення певних дій в конкретній системі - на сайті або терміналі. Авторизація дозволяє виключити доступ до інформації небажаних лиць та надання певних повноважень на виконання деяких дій у системі.
Реєстрація	процес при якому користувач вказує ПІБ (<u>логін</u>), яким будуть підписуватись його пропозиції, також підтверджується поштова адреса та встановлюється пароль на акаунт.
Користувач	це та особа, яка без авторизації має можливість переглядати сайт, але немає доступу до всіх його функцій.
Авторизований користувач	має можливість використовувати усі функції, які надаються сайтом.

Продовження таблиці 1

Адміністратор	це той користувач, який наділений правами, які дозволяють йому управляти контентом веб-сайт, а також виконувати додаткові функції.
Резервування місця	бронювання місця для ночівлі.
Групи	засіб спілкування користувачів у мережі в режимі реального часу. Також багато цікавої інформації для користувачів.
Каучсерфінг	течія, яка пішла від однойменого сайту каучсерфінг, який призначався для безкоштовного резервування житла та надання житла для гостей.
Запит	лист потенційному хосту, що направляється через спеціальну форму на сайті, з проханням про надання житла. Запит може бути прийнятий (асерт), відхилений (decline).

ДОДАТОК Б

КОД ПРОГРАМНОГО ПРОДУКТУ

Розширений профіль користувача

Авторизація:

Підключення до БД:

```
define(' DB_NAME' , ' couchsurfing' );
/** Користувач */
define(' DB_USER' , ' root' );
/** Пароль */
define(' DB_PASSWORD' , '' );
/** Сервер */
define(' DB_HOST' , ' localhost' );
/** Кодування */
define(' DB_CHARSET' , ' utf8mb4' );
/**Схема співвідношення*/
define(' DB_COLLATE' , '' );
```

Функція перевірки користувача:

```
public function wp_authenticate( &$user_login ) {
    global $wpdb;

    $login_type = $this->get_option( 'login_type' );

    if ( 'default' == $login_type )
        return;

    if ( ! is_email( $user_login ) && 'email' == $login_type ) {
        $user_login = -1;
    } elseif ( is_email( $user_login ) ) {
        if ( $found = $wpdb->get_var( $wpdb->prepare( "SELECT
user_login FROM $wpdb->users WHERE user_email = %s", $user_login ) ) )
            $user_login = $found;
    }
}

<?php global $user_ID, $user_identity, $current_url;

$current_url = 'http'. (empty($_SERVER['HTTPS'])?'':'s').'://'. $_SERVER['SER
VER_NAME']. $_SERVER['REQUEST_URI'];
get_currentuserinfo(); if (!$user_ID) { ?>

<div id="login-register-password">
    <ul class="tabs_login">
        <li class="active_login"><a href="#login" ><?php _e(' Login' ,
' tabbed-login' ); ?></a></li>
        <?php if(get_option('users_can_register')) { ?>
```

```

        <li><a href="#register"><?php _e(' Register' , ' tabbed-Login' )
?></a></li>
        <?php }; ?>
        <li><a href="#forgot_password"><?php _e(' Forgot' , ' tabbed-
Login' ); ?></a></li>
    </ul>
    <div class="tab_container_login">
        <div id="login" class="tab_content_login">

            <?php $register = $_GET[' register' ]; $reset =
$_GET[' reset' ]; if ( $register == true) { ?>

                <h3><?php _e(' Success!' , ' tabbed-Login' ); ?></h3>
                <p><?php _e(' Check your email for the password and then
return to login.' , ' tabbed-Login' ); ?></p>

                <?php } elseif ( $reset == true) { ?>

                <h3><?php _e(' Success!' , ' tabbed-Login' ); ?></h3>
                <p><?php _e(' Check your email to reset your password.' ,
' tabbed-Login' ); ?></p>

                <?php } else { ?>

                <h3><?php _e(' Have an account?' , ' tabbed-Login' );
?></h3>

                <?php } ?>

            <form method="post" action="<?php bloginfo(' url' ) ?>/wp-
login. php" class="wp-user-form">
                <div class="username">
                    <label for="user_login"><?php _e(' Username' ,
' tabbed-Login' ); ?>: </label>
                    <input type="text" name="log" value="<?php
echo esc_attr( stripslashes($user_login)); ?>" size="20" id="user_login"
tabindex="11" />
                </div>
                <div class="password">
                    <label for="user_pass"><?php _e(' Password' ,
' tabbed-Login' ); ?>: </label>
                    <input type="password" name="pwd" value=""
size="20" id="user_pass" tabindex="12" />
                </div>
                <div class="login_fields">
                    <div class="rememberme">
                        <label for="rememberme">
                            <input type="checkbox"
name="rememberme" value="forever" checked="checked" id="rememberme"
tabindex="13" /><?php _e(' Remember me' , ' tabbed-Login' ); ?>

```

```

                </label >
            </div>
            <?php do_acti on(' l o g i n _ f o r m ' ); ?>
            <i n p u t t y p e = " s u b m i t " n a m e = " u s e r - s u b m i t "
value="<?php _e(' Logi n' , ' tabbed-Logi n' ); ?>" t a b i n d e x = " 1 4 " c l a s s = " u s e r -
s u b m i t " / >
                <i n p u t t y p e = " h i d d e n " n a m e = " r e d i r e c t _ t o "
value="<?php echo $current_url ; ?>" />
            <i n p u t t y p e = " h i d d e n " n a m e = " u s e r - c o o k i e "
value="1" />
        </div>
    </form>
</div>

<?php i f ( g e t _ o p t i o n ( ' u s e r s _ c a n _ r e g i s t e r ' ) ) { ?>

    <div i d = " r e g i s t e r " c l a s s = " t a b _ c o n t e n t _ l o g i n "
s t y l e = " d i s p l a y : n o n e ; " >
        <h3><?php _e(' Register for this site!' , ' tabbed-Logi n' );
?></h3>
        <p><?php _e(' Sign up now for the good stuff.' , ' tabbed-
Logi n' ); ?></p>
        <form method="post" action="<?php echo site_url (' wp-
logi n . p h p ? a c t i o n = r e g i s t e r ' , ' l o g i n _ p o s t ' ) ?>" c l a s s = " w p - u s e r - f o r m " >
            <div class="username">
                <label for="user_l o g i n"><?php _e(' Username' ,
' tabbed-Logi n' ); ?>: </label >
                <i n p u t t y p e = " t e x t " n a m e = " u s e r _ l o g i n "
value="<?php echo esc_attr( stripslashes($user_l o g i n) ); ?>" s i z e = " 2 0 "
i d = " u s e r _ l o g i n " t a b i n d e x = " 1 0 1 " / >
            </div>
            <div class="password">
                <label for="user_email"><?php _e(' Your Email' ,
' tabbed-Logi n' ); ?>: </label >
                <i n p u t t y p e = " t e x t " n a m e = " u s e r _ e m a i l "
value="<?php echo esc_attr( stripslashes($user_email) ); ?>" s i z e = " 2 5 "
i d = " u s e r _ e m a i l " t a b i n d e x = " 1 0 2 " / >
            </div>
            <div class="l o g i n _ f i e l d s " >
                <?php do_acti on(' r e g i s t e r _ f o r m ' ); ?>
                <i n p u t t y p e = " s u b m i t " n a m e = " u s e r - s u b m i t "
value="<?php _e(' Sign up!' , ' tabbed-Logi n' ); ?>" c l a s s = " u s e r - s u b m i t "
t a b i n d e x = " 1 0 3 " / >

                <?php $register = $_GET[' register' ];
i f ( $ r e g i s t e r == t r u e ) { e c h o ' < p > C h e c k y o u r e m a i l f o r t h e p a s s w o r d ! < / p > ' ;
} ?>

                <i n p u t t y p e = " h i d d e n " n a m e = " r e d i r e c t _ t o "
value="<?php echo $current_url ; ?>?register=true" />
                <i n p u t t y p e = " h i d d e n " n a m e = " u s e r - c o o k i e "
value="1" />

```



```

        </div>
    </form>
</div>

<?php }; ?>

<div id="forgot_password" class="tab_content_login"
style="display: none;">
    <h3><?php _e('Lost Your Password?', 'tabbed-login');
?></h3>
    <p><?php _e('Enter your username or email to reset your
password.', 'tabbed-login'); ?></p>
    <form method="post" action="<?php echo site_url('wp-
login.php?action=lostpassword', 'login_post') ?>" class="wp-user-form">
        <div class="username">
            <label for="user_login" class="hide"><?php
_e('Username or Email', 'tabbed-login'); ?>: </label>
            <input type="text" name="user_login" value=""
size="20" id="user_login" tabindex="1001" />
        </div>
        <div class="login_fields">
            <?php do_action('login_form', 'resetpass'); ?>
            <input type="submit" name="user-submit"
value="<?php _e('Reset my password', 'tabbed-login'); ?>" class="user-
submit" tabindex="1002" />
            <?php $reset = $_GET['reset']; if($reset ==
true) { echo '<p>.__('A message was sent to your email
address.', 'tabbed-login').'</p>'; } ?>
            <input type="hidden" name="redirect_to"
value="<?php echo $current_url; ?>?reset=true" />
            <input type="hidden" name="user-cookie"
value="1" />
        </div>
    </form>
</div>
</div>

<?php } else { // is logged in ?>

<div id="login-register-password" class="logged-in">
    <div class="sidebox">
        <h3><?php _e('Welcome, ', 'tabbed-login'); ?> <?php echo
$user_identity; ?></h3>
        <?php if (version_compare($GLOBALS['wp_version'], '2.5',
'>=')) {
            if (get_option('show_avatars')) {
                ?>
                <div class="usericon">
                    <?php global $userdata; get_currentuserinfo(); echo
get_avatar($userdata->ID, 50); ?>

```

```

        </div>
        <?php }else{?>
        <style type="text/css">.userinfo p{margin-left: 0px
!important;text-align:center;}.userinfo{width: 100%;}</style>
        <?php }}?>
        <div class="userinfo">
            <p><?php _e('You are logged in as ', 'tabbed-Login'); ?>
<strong><?php echo $user_identity; ?></strong></p>
            <p>
                <a href="<?php echo wp_logout_url($current_url);
?>"><?php _e('Log out', 'tabbed-Login'); ?></a> |
                <?php if (current_user_can('manage_options')) {
                    echo '<a href="" . admin_url() . "'>' .
__( 'Admin', 'tabbed-Login') . '</a>'; } else {
                    echo '<a href="" . admin_url() .
'profile.php">' . __( 'Profile', 'tabbed-Login') . '</a>'; } ?>
            </p>
        </div>
    </div>

    <?php } ?>

</div>

<?php
    echo $after_widget;
}

function form( $instance ) {
?>
    <p>
        <?php _e('No option available for this widget.',
'tabbed-Login'); ?>
        <br/><strong><?php _e('Note : Do not put the same widget
twice in a page.', 'tabbed-Login'); ?></strong>
    </p>

    <?php
    }
}
?>

```

Функція пошуку за полями користувача

```

<?php

add_filter (' bps_fileds_setup', ' bps_xprofile_setup' );
function bps_xprofile_setup ( $fileds )
{

```

```

global $group, $field;

if (!function_exists ('bp_has_profile'))
{
    printf ('<p class="bps_error">. __('%s: The BuddyPress
Extended Profiles component is not active.', 'bp-profile-search').
'</p>',
        '<strong>BP Profile Search '. BPS_VERSION. '</strong>');
    return $fields;
}

$args = array ('hide_empty_fields' => false, 'member_type' =>
bp_get_member_types ());
if (bp_has_profile ($args))
{
    while (bp_profile_groups ())
    {
        bp_the_profile_group ();
        $group_name = str_replace ('&', '&', stripslashes
($group->name));

        while (bp_profile_fields ())
        {
            bp_the_profile_field ();
            $f = new stdClass;

            $f->group = $group_name;
            $f->id = $field->id;
            $f->code = 'field_'. $field->id;
            $f->name = str_replace ('&', '&', stripslashes
($field->name));
            $f->name = bps_wpml (0, $f->id, 'name', $f->name);
            $f->description = str_replace ('&', '&',
stripslashes ($field->description));
            $f->description = bps_wpml (0, $f->id,
'description', $f->description);
            $f->type = $field->type;
            $f->options = bps_xprofile_options ($field->id);
            foreach ($f->options as $key => $label)
                $f->options[$key] = bps_wpml (0, $f->id,
'option', $label);

            $f->filters = bps_xprofile_filters ($field->type);
            $f->display = empty ($f->filters)? '' : $field-
>type;

            $f->search = 'bps_xprofile_search';

            $fields[] = $f;
        }
    }
}

```

```

    }

    return $fields;
}

function bps_xprofile_search ($f)
{
    global $bp, $wpdb;

    $value = $f->value;
    $filter = bps_filterXquery ($f);

    $sql = array ('select' => '', 'where' => array ());

    $sql ['select'] = "SELECT user_id FROM {$bp->profile-
>table_name_data}";
    $sql ['where'] ['field_id'] = $wpdb->prepare ("field_id = %d", $f-
>id);

    switch ($filter)
    {
        case 'range':
            $min = $f->min;
            $max = $f->max;

            if ($min !== '') $sql ['where'] ['min'] = $wpdb->prepare
("value >= %f", $min);
            if ($max !== '') $sql ['where'] ['max'] = $wpdb->prepare
("value <= %f", $max);
            break;

        case 'age_range':
            $min = $f->min;
            $max = $f->max;
            $time = time ();
            $day = date ("j", $time);
            $month = date ("n", $time);
            $year = date ("Y", $time);
            $ymin = $year - $max - 1;
            $ymax = $year - $min;

            if ($max !== '') $sql ['where'] ['age_min'] = $wpdb->prepare
("DATE(value) > %s", "$ymin-$month-$day");
            if ($min !== '') $sql ['where'] ['age_max'] = $wpdb->prepare
("DATE(value) <= %s", "$ymax-$month-$day");
            break;

        case 'contains':
            $value = str_replace ('&', '&amp;', $value);
            $escaped = '%'. bps_esc_like ($value). '%';

```

```

        $sql ['where'] [$filter] = $wpdb->prepare ("value LIKE %s",
$escaped);
        break;

    case 'like':
        $value = str_replace ('&', '&amp;', $value);
        $value = str_replace ('\\\\%', '\\%', $value);
        $value = str_replace ('\\\\_', '\\_', $value);
        $sql ['where'] [$filter] = $wpdb->prepare ("value LIKE %s",
$value);
        break;

    case '':
        $value = str_replace ('&', '&amp;', $value);
        $sql ['where'] [$filter] = $wpdb->prepare ("value = %s",
$value);
        break;

    case 'num':
        $sql ['where'] [$filter] = $wpdb->prepare ("value = %f",
$value);
        break;

    case 'is_in':
        $values = (array)$value;
        $parts = array ();
        foreach ($values as $value)
        {
            $value = str_replace ('&', '&amp;', $value);
            $parts [] = $wpdb->prepare ("value = %s", $value);
        }
        $sql ['where'] [$filter] = '(' . implode (' OR ', $parts) . ')';
        break;

    case 'match_any':
    case 'match_all':
        $values = (array)$value;
        $parts = array ();
        foreach ($values as $value)
        {
            $value = str_replace ('&', '&amp;', $value);
            $escaped = '%: "'. bps_esc_like ($value) . '";%';
            $parts [] = $wpdb->prepare ("value LIKE %s", $escaped);
        }
        $match = ($filter == 'match_any')? ' OR ': ' AND ';
        $sql ['where'] [$filter] = '(' . implode ($match, $parts) . ')';
        break;

    default:
        return array ();

```

```

    }

    $sql = apply_filters ('bps_field_sql', $sql, $f);
    $query = $sql ['select']. ' WHERE '. implode (' AND ',
    $sql ['where']);

    $results = $wpdb->get_col ($query);
    return $results;
}

function bps_xprofile_options ($id)
{
    static $options = array ();

    if (isset ($options[$id])) return $options[$id];

    $field = new BP_XProfile_Field ($id);
    if (empty ($field->id)) return array ();

    $options[$id] = array ();
    $rows = $field->get_children ();
    if (is_array ($rows))
        foreach ($rows as $row)
            $options[$id][stripslashes (trim ($row->name))] =
stripslashes (trim ($row->name));

    return $options[$id];
}

function bps_xprofile_filters ($type)
{
    $filters = array
    (
        'textbox'           => array ('', 'range'),
        'number'           => array ('', 'range'),
        'url'               => array (''),
        'textarea'         => array (''),
        'selectbox'         => array ('', 'range'),
        'radio'             => array ('', 'range'),
        'multiselectbox' => array (''),
        'checkbox'            => array (''),
        'datebox'          => array ('range'),
    );

    if (isset ($filters[$type])) return $filters[$type];
    return array ();
}

function bps_filters_validation ($f)
{

```

```

$type = apply_filters ('bps_field_validation_type', $f->type, $f);
$type = apply_filters ('bps_field_type_for_validation', $type, $f);

return bps_xprofile_filters ($type);
}

function bps_filterXquery ($f)
{
    $type = apply_filters ('bps_field_query_type', $f->type, $f);
    $type = apply_filters ('bps_field_type_for_query', $type, $f);

    if ($f->filter == 'range')
        return ($type == 'datebox')? 'age_range': 'range';

    switch ($type)
    {
    case 'textbox':
    case 'textarea':
    case 'url':
        return bps_text_search ();

    case 'number':
        return 'num';

    case 'selectbox':
    case 'radio':
        return 'is_in';

    case 'multiselectbox':
    case 'checkbox':
        $all = apply_filters ('bps_field_checkbox_match_all', false,
$f->id);
        return $all? 'match_all': 'match_any';
    }

    return false;
}

function bps_displayXsearch_form ($f)
{
    $type = apply_filters ('bps_field_type_for_filters', $f->type, $f);
    $type = apply_filters ('bps_field_type_for_search_form', $type,
$f);

    return $type;
}

add_filter ('bps_fields_setup', 'bps_anyfield_setup', 99);
function bps_anyfield_setup ($fields)
{

```

```

    $f = new stdClass;

    $f->group = __('Other', 'bp-profile-search');
    $f->id = 'any';
    $f->code = 'field_any';
    $f->name = __('Any field', 'bp-profile-search');
    $f->description = __('Search every BP Profile Field', 'bp-profile-search');
    $f->type = 'anyfield';
    $f->options = array ();
    $f->filters = array ('');
    $f->display = 'textbox';
    $f->search = 'bps_anyfield_search';

    $fields[] = $f;
    return $fields;
}

function bps_anyfield_search ($f)
{
    global $bp, $wpdb;

    $value = str_replace ('&', '&', $f->value);
    $escaped = '%'. bps_esc_like ($value). '%';

    $sql = array ('select' => '', 'where' => array ());
    $sql ['select'] = "SELECT DISTINCT user_id FROM {$bp->profile->table_name_data}";
    $sql ['where'] [$f->filter] = $wpdb->prepare ("value LIKE %s", $escaped);

    $sql = apply_filters ('bps_field_sql', $sql, $f);
    $query = $sql ['select']. ' WHERE '. implode (' AND ', $sql ['where']);

    $results = $wpdb->get_col ($query);
    return $results;
}

```

Розширений профіль користувача

```

public function populate( $id, $user_id = null, $get_data = true ) {
    global $wpdb, $userdata;

    if ( empty( $user_id ) ) {
        $user_id = isset( $userdata->ID ) ? $userdata->ID : 0;
    }

    $bp = buddypress();

```



```

        $field = $wpdb->get_row( $wpdb->prepare( "SELECT * FROM {$bp->profile->table_name_fields} WHERE id = %d", $id ) );

        $this->fill_data( $field );

        if ( ! empty( $get_data ) && ! empty( $user_id ) ) {
            $this->data = $this->get_field_data( $user_id );
        }
    }

    public static function get_instance( $field_id ) {
        global $wpdb;

        $field_id = (int) $field_id;
        if ( ! $field_id ) {
            return false;
        }

        $field = wp_cache_get( $field_id, 'bp_xprofile_fields' );
        if ( false === $field ) {
            $bp = buddypress();

            $field = $wpdb->get_row( $wpdb->prepare( "SELECT * FROM {$bp->profile->table_name_fields} WHERE id = %d", $field_id ) );

            wp_cache_add( $field->id, $field, 'bp_xprofile_fields' );
        }

        if ( ! $field ) {
            return false;
        }

        $_field = new BP_XProfile_Field();
        $_field->fill_data( $field );

        return $_field;
    }

    public function fill_data( $args ) {
        if ( is_object( $args ) ) {
            $args = (array) $args;
        }

        foreach ( $args as $k => $v ) {
            if ( 'name' === $k || 'description' === $k ) {
                $v = stripslashes( $v );
            }
            $this->{$k} = $v;
        }
    }

```

```

public function __get( $key ) {
    switch ( $key ) {
        case 'default_visibility' :
            return $this->get_default_visibility();
            break;

        case 'allow_custom_visibility' :
            return $this->get_allow_custom_visibility();
            break;
    }
}

public function __isset( $key ) {
    switch ( $key ) {
        // Backward compatibility for when these were public
        methods.
        case 'allow_custom_visibility' :
        case 'default_visibility' :
            return true;
            break;
    }
}

public function delete( $delete_data = false ) {
    global $wpdb;

    if ( empty( $this->id ) || empty( $this->can_delete ) || (
    $this->parent_id && $this->option_order == 1 ) )
    {
        return false;
    }

    $bp = buddypress();
    $sql = $wpdb->prepare( "DELETE FROM {$bp->profile-
    >table_name_fields} WHERE id = %d OR parent_id = %d", $this->id, $this-
    >id );

    if ( ! $wpdb->query( $sql ) ) {
        return false;
    }

    // Delete the data in the DB for this field.
    if ( true === $delete_data ) {
        BP_XProfile_ProfileData::delete_for_field( $this->id );
    }

    return true;
}

public function save() {
    global $wpdb;

```

```

$bp = buddypress();

$this->group_id      = apply_filters(
'xprofile_field_group_id_before_save',    $this->group_id,    $this->id
);
$this->parent_id     = apply_filters(
'xprofile_field_parent_id_before_save',   $this->parent_id,   $this->id
);
$this->type          = apply_filters( 'xprofile_field_type_before_save',
$this->type,
    $this->id );
$this->name          = apply_filters( 'xprofile_field_name_before_save',
$this->name,
    $this->id );
$this->description   = apply_filters(
'xprofile_field_description_before_save',  $this->description, $this->id
);
$this->is_required   = apply_filters(
'xprofile_field_is_required_before_save',  $this->is_required, $this->id
);
$this->order_by      = apply_filters(
'xprofile_field_order_by_before_save',    $this->order_by,    $this->id
);
$this->field_order   = apply_filters(
'xprofile_field_field_order_before_save', $this->field_order, $this->id
);
$this->option_order  = apply_filters(
'xprofile_field_option_order_before_save', $this->option_order, $this->id
);
$this->can_delete    = apply_filters(
'xprofile_field_can_delete_before_save',   $this->can_delete,   $th
$this->type_obj      = bp_xprofile_create_field_type( $this->type );
o_action_ref_array( 'xprofile_field_before_save', array( $this ) );

    $is_new_field = is_null( $this->id );

    if ( ! $is_new_field ) {
        $sql = $wpdb->prepare( "UPDATE {$bp->profile-
>table_name_fields} SET group_id = %d, parent_id = 0, type = %s, name =
%s, description = %s, is_required = %d, order_by = %s, field_order = %d,
option_order = %d, can_delete = %d, is_default_option = %d WHERE id =
%d", $this->group_id, $this->type, $this->name, $this->description,
$this->is_required, $this->order_by, $this->field_order, $this-
>option_order, $this->can_delete, $this->is_default_option, $this->id );
    } else {
        $sql = $wpdb->prepare( "INSERT INTO {$bp->profile-
>table_name_fields} (group_id, parent_id, type, name, description,
is_required, order_by, field_order, option_order, can_delete,
is_default_option ) VALUES ( %d, %d, %s, %s, %s, %d, %s, %d, %d, %d, %d
)", $this->group_id, $this->parent_id, $this->type, $this->name, $this-
>description, $this->is_required, $this->order_by, $this->field_order,
$this->option_order, $this->can_delete, $this->is_default_option );

```

```

    }
    if ( $wpdb->query( $sql ) !== null ) {

        if ( $is_new_field ) {
            $this->id = $wpdb->insert_id;
        }

        if ( ! $is_new_field ) {
            $this->delete_children();
            if ( $this->type_obj->supports_options ) {

                $parent_id = $this->id;
                $post_option = ! empty( $_POST["{$this->type}_option"] ) ?
                $_POST["{$this->type}_option"] : '';
                $post_default = ! empty( $_POST["isDefault_{$this->type}_option"] ) ? $_POST["isDefault_{$this->type}_option"] : '';
                $defaults = apply_filters( 'xprofile_field_default_before_save',
                $post_default, $this->type );

                $counter = 1;
                if ( !empty( $options ) ) {
                    foreach ( (array) $options as $option_key => $option_value ) {
                        $is_default = 0;

                        if ( is_array( $defaults ) ) {
                            if ( isset( $defaults[ $option_key ] ) ) {
                                $is_default = 1;
                            }
                        } else {
                            if ( (int) $defaults == $option_key ) {
                                $is_default = 1;
                            }
                        }
                    }

                    if ( '' != $option_value ) {
                        $sql = $wpdb->prepare( "INSERT INTO {$bp->profile->table_name_fields}
                        (group_id, parent_id, type, name, description, is_required, option_order,
                        is_default_option) VALUES (%d, %d, 'option', %s, '', 0, %d, %d)", $this->group_id,
                        $parent_id, $option_value, $counter, $is_default );
                        if ( ! $wpdb->query( $sql ) ) {
                            return false;
                        }
                    }

                    $counter++;
                }
            }
        }
    }
    public function get_children( $for_editing = false ) {
        global $wpdb;

```

```

        if ( empty( $for_editing ) && ( 'asc' === $this->order_by ) )
    {
        $sort_sql = 'ORDER BY name ASC';
    } elseif ( empty( $for_editing ) && ( 'desc' === $this-
>order_by ) ) {
        $sort_sql = 'ORDER BY name DESC';
    } else {
        $sort_sql = 'ORDER BY option_order ASC';
    }

    // This eliminates a problem with getting all fields when
there is no
    // id for the object.
    if ( empty( $this->id ) ) {
        $parent_id = -1;
    } else {
        $parent_id = $this->id;
    }

    $bp = buddypress();
    $sql = $wpdb->prepare( "SELECT * FROM {$bp->profile-
>table_name_fields} WHERE parent_id = %d AND group_id = %d {$sort_sql}",
    $parent_id, $this->group_id );

    $children = $wpdb->get_results( $sql );
public function delete_children() {
    global $wpdb;

    $bp = buddypress();
    $sql = $wpdb->prepare( "DELETE FROM {$bp->profile-
>table_name_fields} WHERE parent_id = %d", $this->id );

    $wpdb->query( $sql );
}
public function get_member_types() {
    if ( ! is_null( $this->member_types ) ) {
        return $this->member_types;
    }

    $raw_types = bp_xprofile_get_meta( $this->id, 'field',
'member_type', false );

    // If ` $raw_types ` is not an array, it probably means this is
a new field (id=0).
    if ( ! is_array( $raw_types ) ) {
        $raw_types = array();
    }
}

```

```

// If '_none' is found in the array, it overrides all types.
$types = array();
if ( ! in_array( '_none', $raw_types ) ) {
    $registered_types = bp_get_member_types();

    // Eliminate invalid member types saved in the database.
    foreach ( $raw_types as $raw_type ) {
        // 'null' is a special case - it represents users
without a type.
        if ( 'null' === $raw_type || isset(
$registered_types[ $raw_type ] ) ) {
            $types[] = $raw_type;
        }
    }

    // If no member types have been saved, interpret as *all*
member types.
    if ( empty( $types ) ) {
        $types = array_values( $registered_types );

        // + the "null" type, ie users without a type.
        $types[] = 'null';
    }
}

public function get_member_type_label () {
    // Field 1 is always displayed to everyone, so never gets a
label.
    if ( 1 == $this->id ) {
        return '';
    }

    // Return an empty string if no member types are registered.
    $all_types = bp_get_member_types();
    if ( empty( $all_types ) ) {
        return '';
    }

    $member_types = $this->get_member_types();

    // If the field applies to all member types, show no message.
    $all_types[] = 'null';
    if ( array_values( $all_types ) == $member_types ) {
        return '';
    }

    $label = '';
    if ( ! empty( $member_types ) ) {
        $has_null = false;
        $member_type_labels = array();
        foreach ( $member_types as $member_type ) {

```

```

if ( 'null' === $member_type ) {
$has_null = true;
continue;
} else {
$mt_obj = bp_get_member_type_object( $member_type );
$member_type_labels[] = $mt_obj->labels['name'];
}
}

// Alphabetical sort.
natcasesort( $member_type_labels );
$member_type_labels = array_values( $member_type_labels );

// Add the 'null' option to the end of the list.
if ( $has_null ) {
$member_type_labels[] = __( 'Users with no member type', 'buddypress' );
}

        $label = sprintf( __( '(Member types: %s)', 'buddypress'
), implode( ', ', array_map( 'esc_html', $member_type_labels ) ) );
        } else {
        $label = '<span class="member-type-none-notice">' . __(
'(Unavailable to all members)', 'buddypress' ) . '</span>';
        }

        return $label;
    }
}
public function render_admin_form_children() {
    foreach ( array_keys( bp_xprofile_get_field_types() ) as
$field_type ) {
        $type_obj = bp_xprofile_create_field_type( $field_type
);
        $type_obj->admin_new_field_html( $this );
    }
}
}

```

ДОДАТОК В

TEST-CASES

№	Description	Steps to reproduce	Expected Result	Actual Result	Passed /Failed	Environment
1	Check how the function "Avthorization"	1. Click on the link " Log in 2. Enter login 3. Enter Password. 4. Click to button log in	System must open new page, where user can see, that he is avthorized.	System must open new page, where user can see, that he is avthorized.	Passed	W 7
2	Check how the function "Registration"	1. Click on the link " registration " 2. Enter name 3. Enter login 4. Enter hobby 5. Enter languages 6. Click to button registration	System must show the message, that user successfully registration on the website	System must show the message, that user successfully registration on the website	Passed	W 7
3	Check how the function "Search"	1. Click on the link " search " 2. Enter city 3. Enter languages 4. Click to button search	Members list Showing	Members list Showing	Passed	W 7
4	Check how the function "Reservation"	1. Click on the link " send a message " 2. Enter date 3. Enter message 4. Click to button Send a message	Booking request sent. place reserved.	Booking request sent. place reserved.	Passed	W 7
5	Check how the function "Group"	1. Click on the link "Group" 2. Select group 3. Enter message 4. Click to button Send a message	Review group, messages sent	Review group, messages sent	Passed	W 7

Продовження таблиці 2

6	Check how the function "Activity"	<ol style="list-style-type: none"> 1. Click on the link " Activity " 2. Select user 3. Read the news 4. Click to save 5. Enter message 6. Click to button Send a message 	News Review, messages sent	News Review, messages sent	Passed	W 7
7	Check how the function "My profile"	<ol style="list-style-type: none"> 1. Click on the link " My profile " 2. Select edit 3. Save change 	Profile view editing changes saved.	Profile view editing changes saved.	Passed	W 7
8	Check how the function "Message"	<ol style="list-style-type: none"> 1. Click on the link " Message " 2. Enter the message 	Message sent	Message sent	Passed	W 7

СПЕЦИФІКАЦІЯ ВИМОГ ДО ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

1.1 Призначення, мета

Веб-сайт створюється з метою популяризації каучсерфінгу та надання сервісу для пошуку, резервування місця для ночівлі під час туризму.

Мета даного проекту полягає в тому, щоб надати функціональні вимоги для системи, що розробляється.

1.2 Продукт-аналог

Проаналізовано 3 веб-сайти, які реалізують аналогічні функції предметної області. Першим аналогом є “Перша українська мережа гостинності” (режим доступу: <http://shuffle.in.ua/>).

Другим аналогом виступає “ Всесвітня мережа Каучсерфінг ” (режим доступу: <https://www.couchsurfing.com/>).

Третім аналогом є ресурс “Life in Travel” (режим доступу: <http://www.life-in-travels.ru/>).

2. Загальний опис

2.1 Характеристики продукту

Веб-орієнтована система каучсерфінг повинна надавати два рівня доступу: рівень користувача та адміністратора. Користувач може виконувати наступні дії: зареєструватися в системі, авторизуватися, змінити особисті дані, здійснити пошук, резервування, надіслати повідомлення, переглянути групи, переглянути активність інших користувачів. Адміністратор в свою чергу може: заблокувати зареєстрованого користувача, змінити статус користувачу, створити групу, видаляти пости з груп.

2.2 Класи користувачів та їх характеристики

Авторизований користувач - може змінити особисті дані, здійснити пошук, резервування, надіслати повідомлення, переглянути групи, переглянути активність інших користувачів. Не авторизований користувач - може лише

переглянути інформацію з груп, переглянути активність, переглянути на список користувачів, які надають житло.

Адміністратор - заблокувати зареєстрованого користувача, змінити статус користувачу, створити групу, видаляти пости з груп.

2.3 Середовище функціонування

Вимоги до сервера:

- Програмне забезпечення:
 - ОС Windows 7, Windows 8, Windows 10;
 - БД MySQL;
- Апаратне забезпечення:
 - процесор Intel® Core i3-880M Processor 2,2 ГГц;
 - об'єм оперативної пам'яті 3 Гб;
 - жорсткий диск з вільними 30 Гб пам'яті.

Вимоги до клієнта:

- Програмне забезпечення:
 - ОС Windows XP, Vista, 7, 8,10;
 - браузері Opera, Chrome, Firefox найновіших версій.
- Апаратне забезпечення:
 - процесор починаючи від Intel Pentium 1,5 ГГц;
 - об'єм оперативної пам'яті 1 Гб;

3. Характеристики системи

3.1 Реєстрація.

3.1.1 Реєстрація. Пріоритет – високий.

3.1.2 Послідовності дія / відгук:

1. Відкрити вкладку реєстрації – вікно «реєстрації» відкрито.
2. Ввести необхідні дані – всі поля заповненні.
3. Підтвердити реєстрацію натиснувши кнопку «Зареєструватися» - користувач зареєстрований.

3.1.3 Функціональні вимоги:

1: Необхідно заповнити всі поля.

3.2 Авторизація.

3.2.1 Авторизація. Пріоритет – високий.

3.2.2 Послідовності дія /відгук:

1. Відкрити вкладку «Авторизація» на головній сторінці – вікно «Авторизації» відкрите.
2. Заповнити потрібні поля – всі поля заповненні.
3. Натиснути кнопку «Вхід» - користувач або адміністратор авторизований.

3.2.3 Функціональні вимоги:

1: Необхідно заповнити всі поля.

3.3 Активність.

3.3.1 Активність. Пріоритет – низький.

3.3.2 Послідовності дія /відгук:

1. Відкрити вкладку «Активність» на головній сторінці – вікно «Активність» відкрите.
2. Переглянути активність користувачів – інформацію переглянуто.

3.3.3 Функціональні вимоги:

1: Необхідно відкрити потрібну вкладку.

3.4 Пошук.

3.4.1 Пошук. Пріоритет – середній.

3.4.2 Послідовності дія /відгук:

1. Відкрити вкладку «Пошук» на головній сторінці – вікно «Пошуку» відкрите.
2. Заповнити потрібні поля – всі поля заповненні.
3. Натиснути кнопку «Пошук» - список користувачів по нашому запиту відображений у вікні.

3.4.3 Функціональні вимоги:

1: Необхідно заповнити всі поля.

3.5 Резервування місця для ночівлі.

3.5.1 Резервування місця для ночівлі.. Пріоритет – високий.

3.5.2 Послідовності дія /відгук:

1. Зі списку користувачів, які нам відобразились на екран після пошуку необхідно натиснути «Надіслати запит».- відкрито вікно для заповнення запиту.
2. Заповнити потрібні поля – всі поля заповненні.
3. Натиснути кнопку «Надіслати» - повідомлення надіслано.

3.5.3 Функціональні вимоги:

1: Потрібно, щоб був здійснений пошук, потрібно вибрати певного користувача.

3.6 Групи.

3.6.1 Групи. Пріоритет – низький.

3.6.2 Послідовності дія /відгук:

1. Відкрити вкладку «Групи» на головній сторінці – вікно «Групи» відкрите.
2. Переглянути існуючі групи – групи переглянуті.
3. Натиснути кнопку «Створити групу» - відкриється вкладка для створення групи.
4. Заповнити потрібні поля – всі поля заповненні.
5. Підтвердити створення натиснувши “Створити”- групу створено.

3.6.3 Функціональні вимоги:

1: Необхідно заповнити всі поля.

3.7 Активація користуча.

3.7.1 Активація користуча. Пріоритет – середній.

3.7.2 Послідовності дія /відгук:

1. Авторизуватись як адміністратор – увійшли у систему як адміністратор.

2. Вибрати вкладку «Користувачі» – вкладка «Користувачі» відкрита.
 3. Вибрати зі списку користувачів – користувача обрано.
 4. Вибрати дії активація – користувач обраний.
 5. Підтвердити активацію натиснувши “Активувати”- профіль користувача активовано.
- 3.7.3 Функціональні вимоги:
- 1: Необхідно права адміністратора.

4. Вимоги зовнішніх інтерфейсів

4.1 Користувацькі інтерфейси

Головна інтерфейс форма – це головне вікно веб-сайту. На цьому вікні присутні усі необхідні функції для повноцінного користування системою.

5. Інші нефункціональні вимоги

5.1 Вимоги продуктивності

1. Система повинна функціонувати при вході на неї більше 10000 користувачів.
2. Час відповіді на дію користувача повинен тривати менше 1 сек.

5.2 Вимоги безпеки

1. Всі персональні дані користувачів зашифровані одним із методі шифрування.

5.3 Атрибути якості програмного продукту

1. Система повинна бути зручною та зрозумілою для користувача.

ДОДАТОК Д

ТЕСТ-ПЛАН

1. Вступ

В розроблювальному веб-сайті процес тестування є необхідною частиною, яка дає змогу перевірити усі функції, виявити помилки, дефекти, адже для того, щоб надати веб-сайт у використання потрібно перевірити його згідно поставлених вимог.

2. Розробка тестів.

Були проведенні два види тестування: автоматизоване і ручне. В ручному виді тестування було проведено тестування безпеки та функціональне тестування.

Для перелічених видів тестування розроблені та виконані тестові випадки. Результати тестів задокументовано.

2.1. Функціональне тестування;

Для функціонального тестування було розроблено 12 тестових випадків. Після проходження всіх 12 тестових випадків пройшли успішно.

2.2. Тестування безпеки;

Тестування безпеки було проведено за допомогою ручного виду тестування. Для проведення безпеки було перевірено функцію авторизації, яка ідентифікує користувача та надає йому відповідні права доступу. Перевірка закінчилась успішно, всі функції працюють вірно.

2.3 GUI тестування

Тестування користувацького інтерфейсу пройшло успішно. Що свідчить про те, що користувацький інтерфейс розроблено правильно.

3. Метрики.

Всі заплановані тестові випадки розроблено. Покриття тестами варіантів використання становить 95%.

4. Критерії прийняття відхилення релізу.

Всі розроблені тестові випадки виконано. Результати тестування безпеки показали правильність роботи функції авторизації.

Результати функціональне тестування не виявило ніяких дефектів.

Результати автоматизованого тестування показали, що інтерфейс користувача розроблений правильно

