

**НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ
імені ІГОРЯ СІКОРСЬКОГО»
ФАКУЛЬТЕТ ІНФОРМАТИКИ ТА ОБЧИСЛЮВАЛЬНОЇ ТЕХНІКИ
КАФЕДРА АВТОМАТИКИ ТА УПРАВЛІННЯ В ТЕХНІЧНИХ СИСТЕМАХ**

«На правах рукопису»

УДК _____

«До захисту допущено»

Завідувач кафедри

(підпис)

(ініціали, прізвище)

“ ” _____ 20__ р.

Магістерська дисертація

зі спеціальності (спеціалізації) 121, інженерія програмного забезпечення (інженерія програмного забезпечення комп'ютерних систем)

(код і назва спеціальності)

на тему: Веб-сервіс для управління стартапами

Виконав (-ла): студент (-ка) 2 курсу, групи ІТ-83мп

(шифр групи)

Хуссейн Джагангір Саржієв

(прізвище, ім'я, по батькові)

[підпис]

(підпис)

Науковий керівник

професор кафедри ІТ-83мп, укр. [підпис]

(посада, науковий ступінь, вчене звання, прізвище та ініціали)

[підпис]

(підпис)

Консультант

(назва розділу)

(науковий ступінь, вчене звання, прізвище, ініціали)

(підпис)

Рецензент

(посада, науковий ступінь, вчене звання, науковий ступінь, прізвище та ініціали)

(підпис)

Засвідчую, що у цій магістерській дисертації немає запозичень з праць інших авторів без відповідних посилань.

Студент

[підпис]
(підпис)

Київ – 2019 року

Резюме

Магістерська робота містить 101 сторінка, 16 таблиць, 44 рисунки, список

використаних джерел із 20 найменувань та 8 додатків АЗ.

Метою магістерської роботи є розробка веб-орієнтованої системи управління стартапами.

Об'єкт дослідження - процес керування стартапами.

Предметом дослідження є застосування сучасних технологій створення веб-сервісу для розробки веб-додатку керування стартапами. Розробка виконується на технологіях Hypertext Preprocessor (PHP), сервер бази даних SQL і веб-сервісу Open Server.

Одержані результати полягають в розробці веб-додатку для керування стартапами.

Ключові слова: управління стартапами, веб-ресурс.

Summary

The master's thesis contains 101 pages, 16 tables, 44 drawings, a list used a source of 20 titles and 8 additional A3.

The technique of magister work is an extended web system of a stored system managing startups.

The object of research is the process of managing startups.

We assume that modern technologies created on the Internet are being used and additional devices are being developed for sending out an online service for managing startups. The distribution methods are based on PHP technology, MySQL database server and Open Server web server.

The data used is web-based resources for managing startups.

Keywords: web application, startup management.

ВСТУП

Актуальність обраної тематики полягає в тому, що, як сьогодні, так і на наступні часи, створення нових стартапів проектів є невід'ємною частиною. Розробка інтернет-порталу, який допомагає не відходячи від свого місця, або дому, управляти власними проектами та робітниками є дуже зручним сервісом, тому що, де б ми не знаходились, ми завжди зможемо працювати над нашим стартапом. Для цього нам необхідний лише доступ до мережі інтернет. Однак вагомим мінусом таких систем є те, що спілкування між учасниками команди, відбувається завжди на відстані і в різний час, тому в них немає можливості особисто при зустрічі обговорити всі деталі роботи над проектом. Тому виникає проблема організації роботи команди над проектом.

На організацію проекту потрібно не мало часу , щоб зібрати програмістів в один час, для вирішення всіх моментів щодо стартапу . Тому розробка веб-додатку, який би дозволяв швидко організувати комунікацію усієї команди, є важливою задачею.

Мета магістерської роботи полягає в аналізі та детальному дослідженні уже існуючих інтернет-сайтів для створення проектів (якщо такі існують в даний час) та додатків, які дозволяють створювати власні стартапи, і розробка власного інтернет-додатку, який дозволив би розробляти та управляти командою для стартапу.

Отож, дивлячись на мету, можемо обрати такі основні питання, які ставляться до розроблюваного проекту:

- 1) проаналізувати існуючі сервіси для управління стартапом.
- 2) на основі дослідження створити функціональні та не функціональні вимоги до створюваного проекту.
- 3) створити модель БД розроблюваного проекту.
- 4) визначити необхідний стек технологій для створення обраного функціоналу.
- 5) створення веб-сервісу для кращого управління проектами згідно з визначених задач, описаних в розділах 1-4.

Об'єкт дослідження – процес управління стартапами.

Предмет дослідження – впровадження сучасних технологій створення веб-орієнтованих додатків для створення веб-додатку керування стартапами.

Під час створення системи використовуватимемо такі ресурси необхідні для розробки інтернет-додатку:

Hypertext Preprocessor– скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. Hypertext Preprocessor є однією з найпоширеніших мов, що використовуються у сфері веб-розробок

Java Script – мова для написання скриптів, призначена для створення інтерактивних веб-сайті, не потребує додаткового компілятора.

MySQL – програмування баз даних, яка дає можливість редагувати та працювати з БД, котрі є набором зв'язаної інформації, що зберігається в таблицях Практична цінність розроблюваного інтернет-додатку полягає в швидкому та зручному редагуванню стартапами.

РОЗДІЛ 1 АНАЛІЗ ПРЕДМЕТНОЇ ОБЛАСТІ

1.1 Коротка характеристика об'єкту управління

Основним напрямом діяльності об'єкту управління стартапами є створення, пошук та реалізація проектів різного типу. За допомогою даної системи користувачі мають можливість створювати проекти, команди, власні компанії.



Рисунок 1.1 - Схема організаційної структури управління підприємством

В наш час є дуже багато сайтів, де ми можемо придбати певну продукцію (сайт, програму) за відповідну суму грошей. Однак людей не можуть собі дозволити витратити великі суми на реалізацію цих проектів. Тому доцільно створити систему, яка надасть нам можливість реалізувати свої ідеї без фінансових вливань з нашої сторони, а навпаки зібрати кошти на реалізацію та подальший розвиток даного проекту.

Людина, яка створила проект, автоматично стає адміністратором проекту.

Адміністратор – це людина, яка створила проект, займається пошуком команди, фінансування. Також ця людина може назначати на посаду адміністратора учасників проекту.

Існує три типи користувачів:

- звичайний користувач;

- спеціаліст;
- компанія.

Якщо користувач має цікаву ідею, але не має достатньо знань в певній сфері для її реалізації, він може зареєструватися як звичайний користувач, створити проект, описати свою ідею і всі зацікавлені користувачі зможуть зробити свій внесок у це проект, якщо це спеціаліст, та допомогти з розробкою, якщо компанія то фінансова допомога проекту, або викупити проект повністю на певних умовах, про які домовляються компанія і користувач, що створив даний проект.

Є тип користувачів “спеціалісти”, які мають знання в конкретній області, а саме в програмуванні, але вони не можуть себе реалізувати. Більшість з яких працює на не цікавій для них роботі, або взагалі фрілансерами і давно вже хотіли створити свій проект, але не мали власної ідеї і тому для такого типу користувачів є “звичайні користувачі”. Спеціаліст підбирає для себе цікавий проект і пропонує свої послуги, якщо “звичайний користувач” не проти його допомоги, вони домовляються між собою на яких умовах вони будуть співпрацювати і розпочинають розробку.

Компанія – юридична особа, що ставить собі за мету знайти спеціаліста, або проект для подальшої роботи з ними. Головною метою компанії є отримання прибутку.

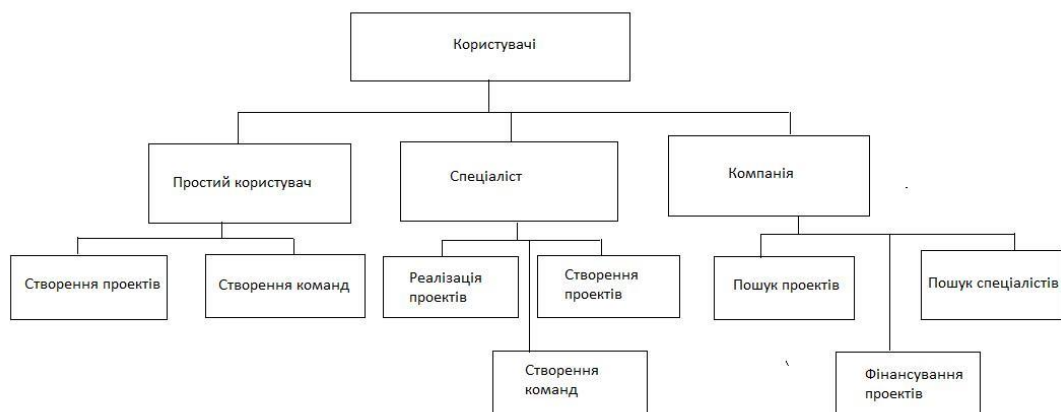


Рисунок 1.2 - Схема організаційної структури управління підприємств

1.2 Опис предметної області

Для організації роботи системи управління стартапами необхідне виконання таких основних бізнес-процесів (рисунок 1.3):

- процесу формування календарного плану;
- процесу управління командою;
- процесу управління стартапом;
- процесу фінансування стартапу;
- процесу управління зустрічами.

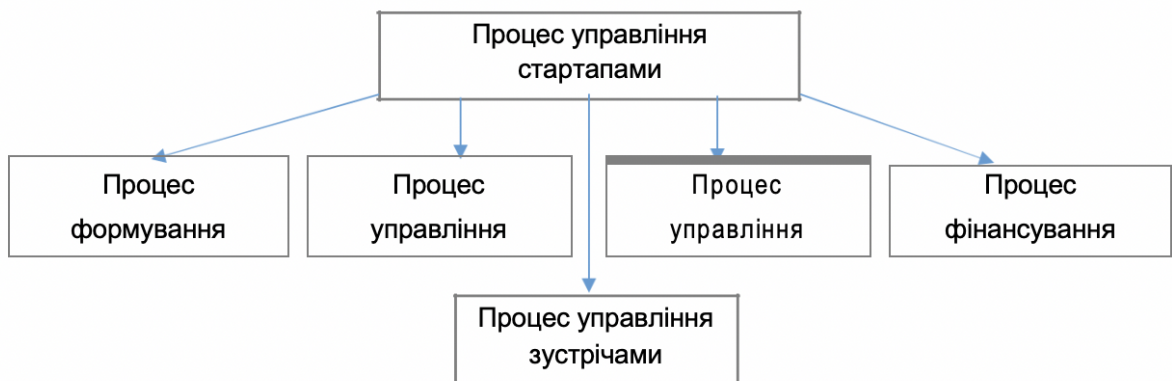


Рисунок 1.3 - Діаграма дерево функцій

Розглянемо детальніше кожен з вище представлених бізнес-процесів. На рисунку 1.4 бачимо діаграму функцій процесу формування календарного плану.

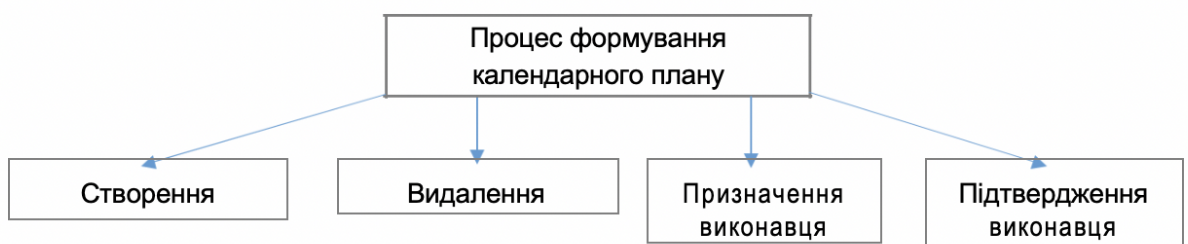


Рисунок 1.4 – Діаграма функцій процесу формування календарного плану

У кожному проекті є завдання, терміни виконання цих завдань та відповідальні за їх виконання, тому для зручності створення нових проектів буде використовуватися процес формування календарного плану. Цей процес буде поділятися на такі етапи:

- створення завдання (вводиться назва завдання і його короткий опис);
- видалення завдання (видалення уже створених завдань);
- призначення виконавця (вибирається зі списку учасників проекту).
- підтвердження виконавця (керівник проекту затверджує виконання завдання за конкретним користувачем після його згоди).

Характеристику бізнес-процесу формування календарного плану наведено в таблиці 1.1

Таблиця 1.1 - Характеристика бізнес-процесу формування календарного плану

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Формування календарного плану
Основні учасники	Керівник
Вхідна подія	Запит на створення завдання
Вхідні документи	-
Вихідна подія	Повідомлення про статус завдання (помилка, завдання створене)
Вихідні документи	-

Клієнт бізнес-процесу	Процес управління командою, процес управління стартапом, процес управління зустрічами
-----------------------	---



Рисунок 1.5 – Діаграма функцій процесу управління командою

В більшості випадків для створення проекту потрібні люди зі спеціальними навиками в різних областях (дизайнери, програмісти, тестери тощо). Разом вони будуть утворювати команду, яка розроблятиме конкретний продукт, тому необхідно організувати процес управління командою . Цей процес буде поділятися на такі етапи:

- відображення списку учасників (всі користувачі, що приймають участь у розробці проекту);
- призначення на посаду (призначати на посаду може лише керівник проекту);
- додавання нового учасника (додаванням нового учасника займається керівник проекту);
- видалення учасника (видаляти учасників може лише керівник проекту).

Характеристику бізнес-процесу формування календарного плану наведено в таблиці 1.2

Таблиця 1.2 - Характеристика бізнес-процесу управління командою

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Управління командою
Основні учасники	Керівник, учасник команди
Вхідна подія	Запит на управління учасником
Вхідні документи	-
Вихідна подія	Повідомлення про статус запиту (помилка)
Вихідні документи	-
Клієнт бізнес-процесу	процесу формування календарного плану, процесу управління стартапом, процесу управління зустрічами.

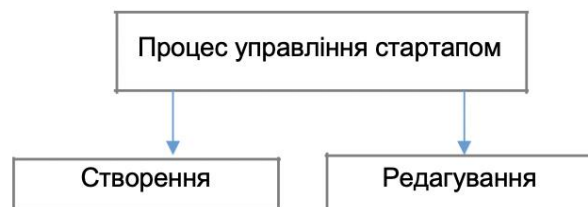


Рисунок 1.6 – Діаграма функцій процесу управління стартапом

Для створення нового проекту необхідно його описати, вказати учасників, які необхідні для його реалізації. Ці дані можна змінювати під час розробки проекту, для цього був реалізований процес управління стартапом. Цей процес буде поділятися на такі етапи:

- створення стартапу (користувач що створює новий проект, автоматично стає керівником, при створенні він вказує назву проекту, короткий опис);
- редагування (редагуванням займається керівник проекту).

Характеристику бізнес-процесу управління стартапом наведено в таблиці 1.3

Таблиця 1.3 - Характеристика бізнес-процесу управління стартапом

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Управління стартапом
Основні учасники	Керівник
Вхідна подія	Запит на створення стартапа
Вхідні документи	-
Вихідна подія	Повідомлення про статус стартапа (помилка, проект створений)
Вихідні документи	-
Клієнт бізнес-процесу	Процес формування календарного плану, процесу управління командою, процес управління зустрічами.

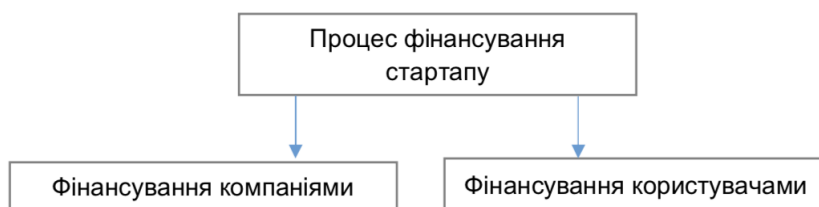


Рисунок 1.7 – Діаграма функцій процесу фінансування стартапу

Кожен проект потребує фінансування. Фінансування може відбуватися з двох сторін: компаніями та звичайними користувачами. Для зручності був

розроблений процес фінансування стартапу. Цей процес буде поділятися на такі етапи:

- фінансування компаніями;
- фінансування користувачами (звичайні користувачі фінансують проект).

Характеристику бізнес-процесу формування календарного плану наведено в таблиці 1.4

Таблиця 1.4 - Характеристика бізнес-процесу фінансування стартапа

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Фінансування стартапа
Основні учасники	Керівник, простий користувач, компанія
Вхідна подія	Запит на можливість фінансування
Вхідні документи	-
Вихідна подія	Повідомлення про статус фінансування (помилка, дозволено)
Вихідні документи	-
Клієнт бізнес-процесу	Процес управління стартапом.

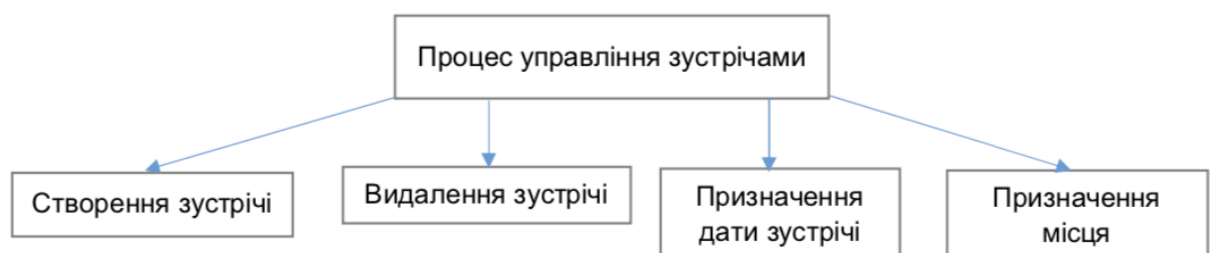


Рисунок 1.8 – Діаграма функцій процесу управління зустрічами

Спілкування через інтернет в наш час є дуже зручним, але все таки він не може замінити спілкування “в живу”. Тому за допомогою процесу управління зустрічами, учасники команди можуть домовитися про зустріч, де вони зможуть обговорити всі питання стосовно проекту, який вони розробляють. Цей процес буде поділятися на такі етапи:

- створення зустрічі (створювати зустрічі може лише керівник);
- видалення зустрічі (видаляти зустрічі може лише керівник проекту);
- призначення дати зустрічі (встановленням дати займається керівник проекту);
- призначення місця проведення (місце проведення вирішує керівник проекту).

Характеристику бізнес-процесу управління зустрічами наведено в таблиці 1.5

Таблиця 1.5 - Характеристика бізнес-процесу управління зустрічами

Назва характеристики	Значення характеристики
Ім'я бізнес-процесу	Управління зустрічами
Основні учасники	Керівник
Вхідна подія	Запит на створення зустрічі
Вхідні документи	-
Вихідна подія	Повідомлення про статус зустрічі (помилка, зустріч створена)

Вихідні документи	-
Клієнт бізнес-процесу	Процес управління командою.

1.3 Огляд і аналіз існуючих аналогів

Indiegogo — це сайт для фінансування проектів засхемою краудфандингу. Сервіс заснований в 2008 році. Головний офіс знаходиться в Сан-Франциско, Каліфорнія. Близько дев'яти мільйонів людей з усього світу відвідують сайт щомісяця.

Indiegogo є одним з найпопулярніших веб-сайтів краудфандингу в Америці. Сайт дозволяє кожному зібрати гроші для власної ідеї.

Даний сайт дозволяє користувачам створити сторінку для фінансування свого проекту, налаштувати обліковий запис PayPal, скласти список "пільг" для різних рівнів вкладень. Користувачі мають можливість публікувати свої проекти через соціальні мережі, такі як Facebook, Twitter та аналогічні платформи.

Сайт отримує 4% комісії з компаній що досягли своєї цілі. Для кампаній, що не змогли зібрати вказану суму, користувачі мають можливість безкоштовного відшкодування всіх грошей їх вкладникам, або можуть зберегти всі зібрані гроші, але з комісією у 9%.

Indiegogo виплачує кошти негайно, як тільки внески збираються на PayPal рахунку користувача. Станом на січень 2014 року, було проведено більше 200 тисяч рекламних компаній та зібрано «мільйони доларів»

Основним функціоналом на Indiegogo є створення проекту, який потім зможе фінансувати велика аудиторія людей, якщо цей проект їх зацікавить.

При створенні проекту, користувач повинен вказати наступні пункт:

- суму, яку необхідно зібрати для створення даної ідей;
- назву проекту;
- зображення проекту;
- вибрати категорію, до якої буде відноситися даний проект:
 - а) тварини,
 - б) святкування,
 - в) навчання,

- г) медицина,
- д) спорт,
- е) волонтерство;
- вказати місто проживання;
- описати даний проект (Розказати, чому потрібні кошти, на що саме. Що спонукало вас, щоб розпочати збір коштів? Як люди можуть допомогти?);
- додати фото;
- вказати одержувача коштів. (Можна вказати що це ви, або вказати іншу особу).

Після заповнення всіх пунктів, потрібно натиснути на кнопку створити, після чого створиться сторінка з вашим проектом і буде доступна всім користувачам для фінансування.

Start a campaign

Raise funds for creative, entrepreneurial, or other passion projects.

How much money would you like to raise?

\$ 500 USD ▾

Minimum \$500.

What is the title of your campaign?

My campaign title...

50 characters maximum.

How would you like to get started?



Quick launch editor

Focus on essentials and launch within minutes. You can always switch to the standard campaign editor to add more details later.



Standard campaign editor

Access all functionality, such as perks, campaign analytics, and campaign team setup, before you launch your campaign.

CREATE MY CAMPAIGN

Рисунок 1.9 - Створення проекту

Під час створення нового проекту на Indiegogo потрібно вказати скільки грошей необхідно для створення даного проекту, назву проекту .

The screenshot shows the 'Itema' campaign creation interface. At the top, there's a header with the 'Itema' logo and an 'Edit' link. Below it are input fields for 'Country' (a dropdown menu) and 'City'. A funding amount of '\$500 USD' is displayed with an 'Edit' link. A vertical sidebar on the left contains a list of tabs: '1 Basics' (active), '2 Story', '3 Perks', '4 Team', '5 Funding', '6 Extras', and '7 InDemand'. Below the sidebar are three buttons: 'REVIEW & LAUNCH' (pink), 'VIEW CAMPAIGN' (dark grey), and 'SAVED' (pink). The main content area is titled 'Basics' and includes a descriptive paragraph: 'Make a good first impression: introduce your campaign objectives and entice people to learn more. This basic information will represent your campaign on your campaign page, on your campaign card, and in searches.' Below this is a 'Tagline' input field. A 'Campaign Card Image' section features a large placeholder image with the word 'GO' and an 'UPLOAD AN IMAGE' button. A character count '100/100' is visible next to the tagline field.

Рисунок 1.10 - Інформація про проект

В формі інформація про проект необхідно заповнити всі поля, описати проект так, щоб зацікавити користувачів, щоб вони захотіли вкласти гроші у ваш проект.

This screenshot is identical to the one above, showing the 'Itema' campaign creation form. It displays the 'Basics' tab with the same input fields for 'Country', 'City', and funding amount, the sidebar with tabs 1 through 7, and the main content area with the 'Basics' section, 'Tagline' field, and 'Campaign Card Image' placeholder. The 'REVIEW & LAUNCH', 'VIEW CAMPAIGN', and 'SAVED' buttons are also present.

Рисунок 1.11 - Вигляд проекту після створення

Після створення проекту ми можемо бачити, як бачать наш проект всі користувачі і скільки грошей проект отримав.

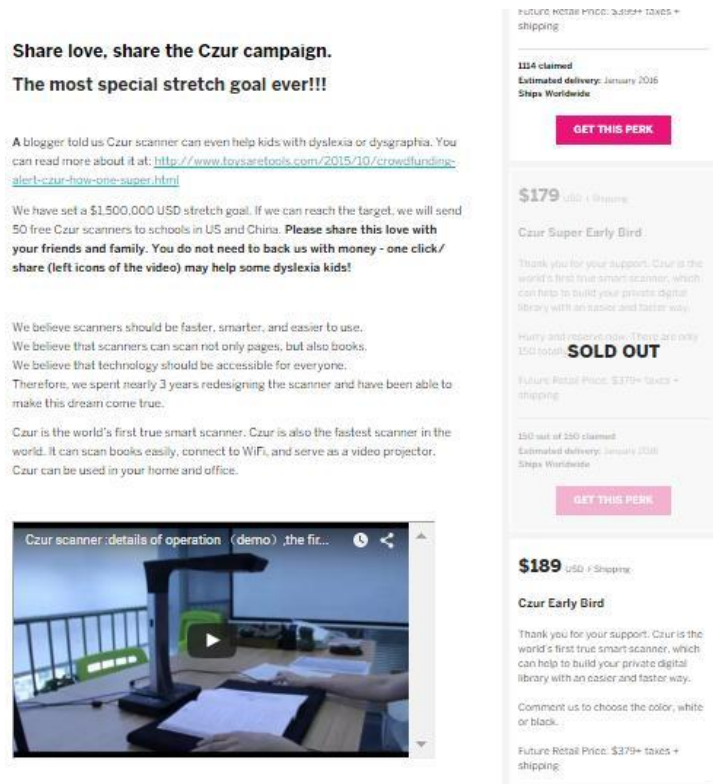


Рисунок 1.12 - Опис проекту

В описі проекту відображається вся інформація, яку ми вводили під час створення проекту, влюбий момент ми можемо її змінити.

Teamfinding - це “добровільний” соціальний сервіс, орієнтований на тих користувачів, які хочуть свої ідеї і тих, хто хоче приймати участь в реалізації таких ідей.

Teamfinding вирішує дві проблеми.

Перша: проблема людей, які хочуть прийняти участь в реалізації цікавих проектів, приймати участь на зустрічах на різну тематику, і в цілому, де застосувати свої навички та досвід (крім своєї постійної роботи) – тобто самореалізуватися, але пошук таких проектів і однодумців важкий і не цікавий процес.

Teamfinding вирішує дану проблему, і рішення заключається в можливості швидкого пошуку цікавих проектів, груп, зустрічей, в які можна подати заявку на участь, познайомитися з новими людьми і отримати безцінний досвід.

Друга: проблема тих, хто хоче реалізувати дані ідеї, але зустрічається з тим, що в певній області вони не достатньо кваліфіковані, мають малий досвід або просто не вистачає часу реалізувати самостійно.

Рішення заключається в можливості створити і стати координатором проекту / зустрічі / спільноти і знайти людей, яким буде цікаво приймати участь у проекті. Можливість знайти і скоординувати навиків і досвід спеціалістів для реалізації ідей.

Участь в проектах добровільна, про компенсацію в тому чи іншому проекті можна домовлятися лише з координатором проекту.

Teamfinding – для створення нової ідеї і пошуку людей для її реалізації необхідно вказати наступні пункти:

- назва ідеї;
- опис ідеї;
- веб-сайт ;
- тип ідеї;
- проект;
- стартап;
- знімальна група;
- об'єднання;
- спільнота;
- зустріч;
- стадію виконання;
- ідея;
- задокументована ідея;
- проектування;
- розробка;

- тестування;
- реалізована;
- пошук інвесторів;
- розширення команди;
- пошук команди;
- роль людини, що створила ідею.

Після того ,як будуть вказані всі пункти, буде створена ідея, до якої можна буде запрошувати людей та приймати зацікавлених осіб до себе в команду.

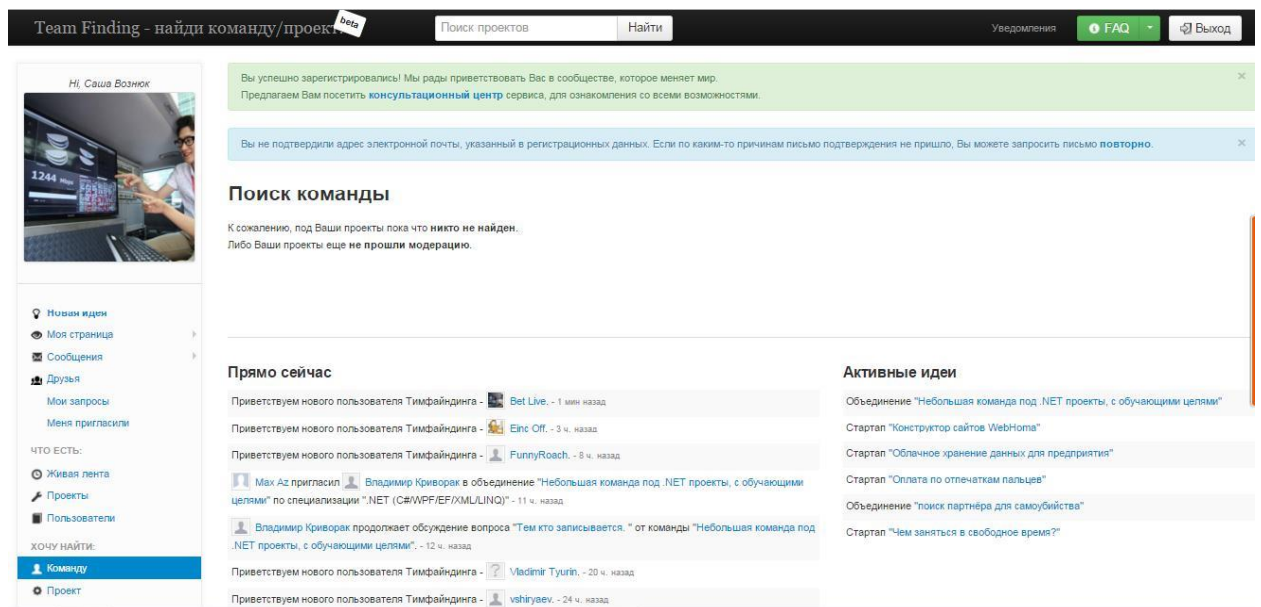


Рисунок 1.13 - Головна сторінка користувача

На головній сторінці користувача ми бачимо всі новини сайту (нових зареєстрованих користувачів, створені проекти, нових учасників проекту тощо), активні дії (як розроблюються проекти).

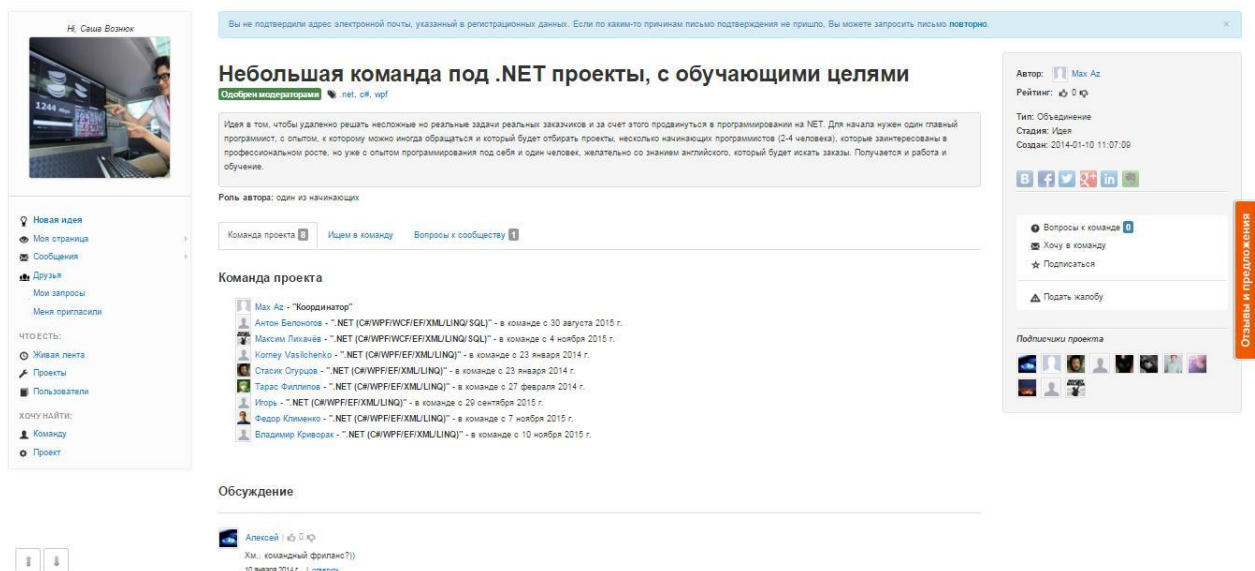


Рисунок 1.14 - Вигляд створеного проекту

Після створеного проекту ми бачимо його сторінку, де відображається інформація про проект, короткий опис, хто створив даний проект, список учасників та інформація про проект.

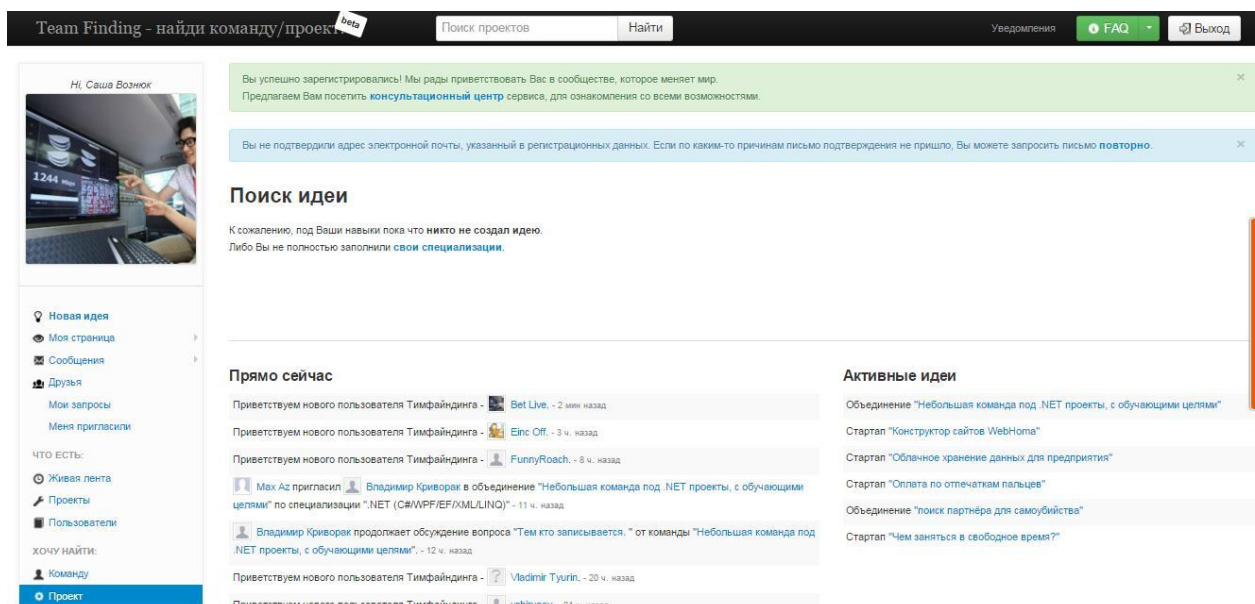


Рисунок 1.15 - Пошук ідей

Пошук ідеї. Якщо новий користувач хоче створити новий проект, але він не впевнений, що такий проект вже існує, то він може скористатися пошуком ідей, де користувач вводить назву ідеї і якщо така ідея існує, то вона буде відображена.

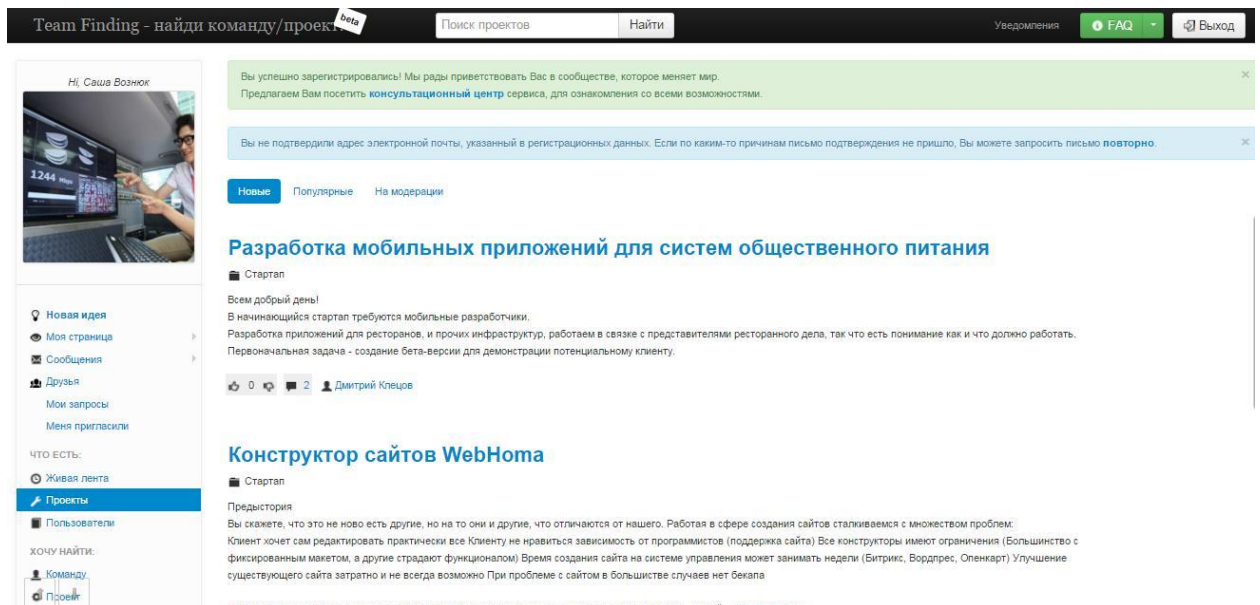


Рисунок 1.16 - Список проектов

Список проектов. На даній сторінці користувач може переглянути всі проекти і вибрати той проект в якому він би хотів приймати участь.

Startup fellows – сайт призначений для пошуку стартапів та людей, які можуть їх реалізувати.

Для створення нового стартапу потрібно виконати наступні кроки:

- натиснути на кнопку “новая запись” після чого заповнити наступні

поля:

а) тип запису:

- стартап шукає людей;
- шукаю стартап.

б) заголовок;

- в) короткий опис;
- г) опис;
- натиснути кнопку опублікувати.

Для того, щоб знайти стартап, потрібно вибрати в меню “Стартап ищет людей”, після чого зі списку стартапів вибрати той який вам подобається і вибрати “Написать автору”.

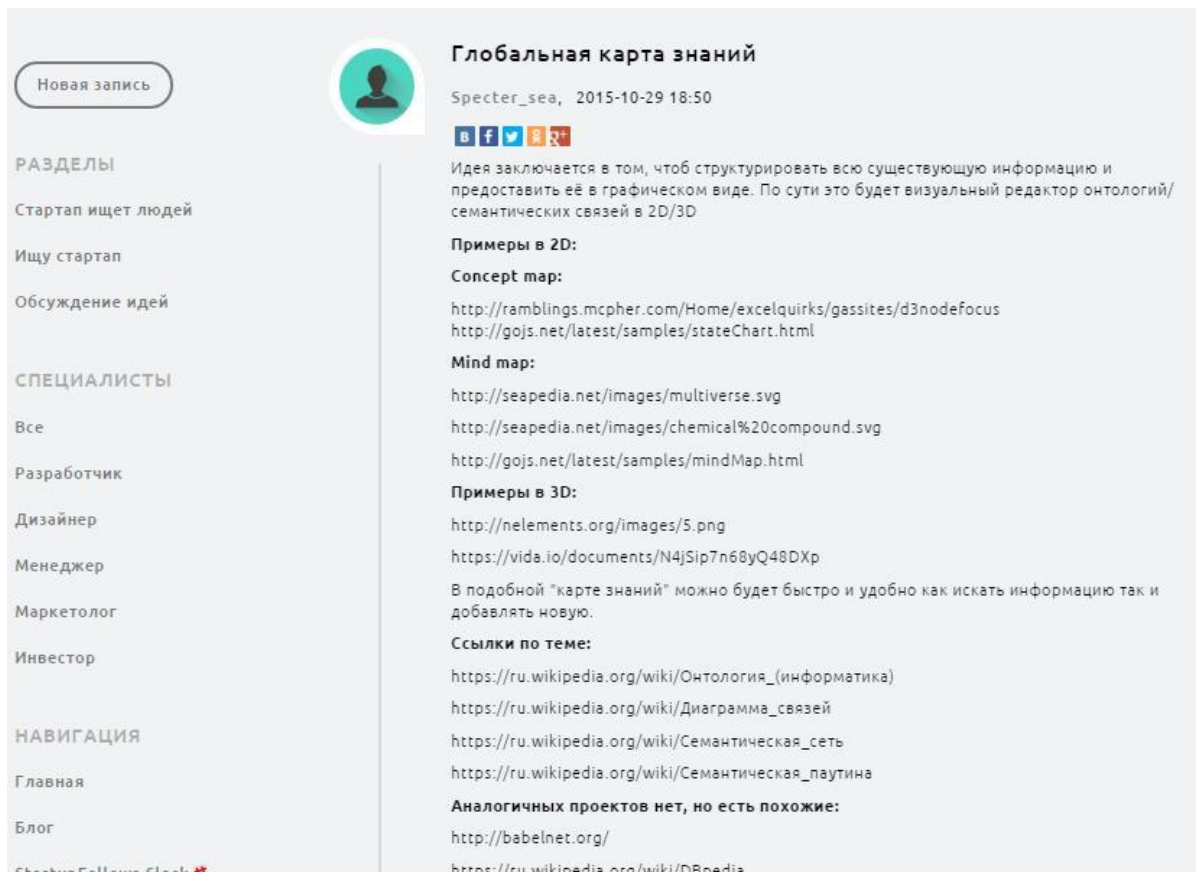


Рисунок 1.17 - Створений проект

В створеному стартапі можна переглянути короткий опис про проект, посилання в соціальних мережах та посилання на:

- приклади;
- посилання по темі;
- аналоги.

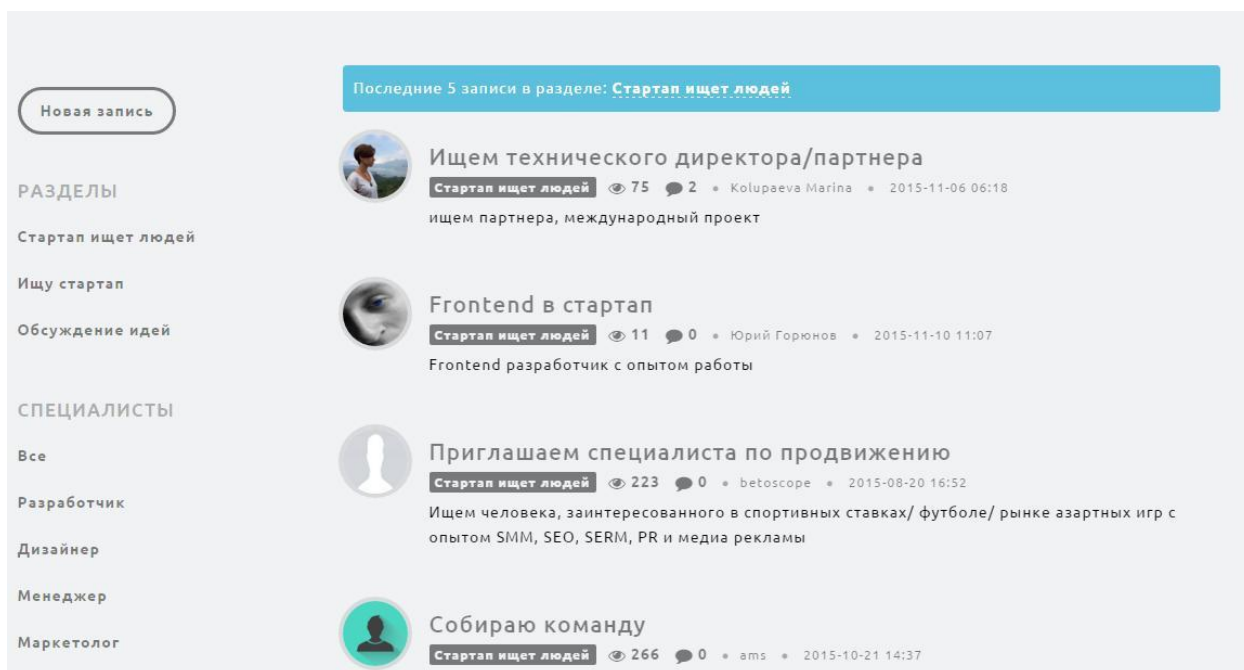


Рисунок 1.18 - Список проектов

В списке проектов ми бачимо усі створені проекти та інформацію про них:

- назва;
- чи шукає стартап людей;
- короткий опис.

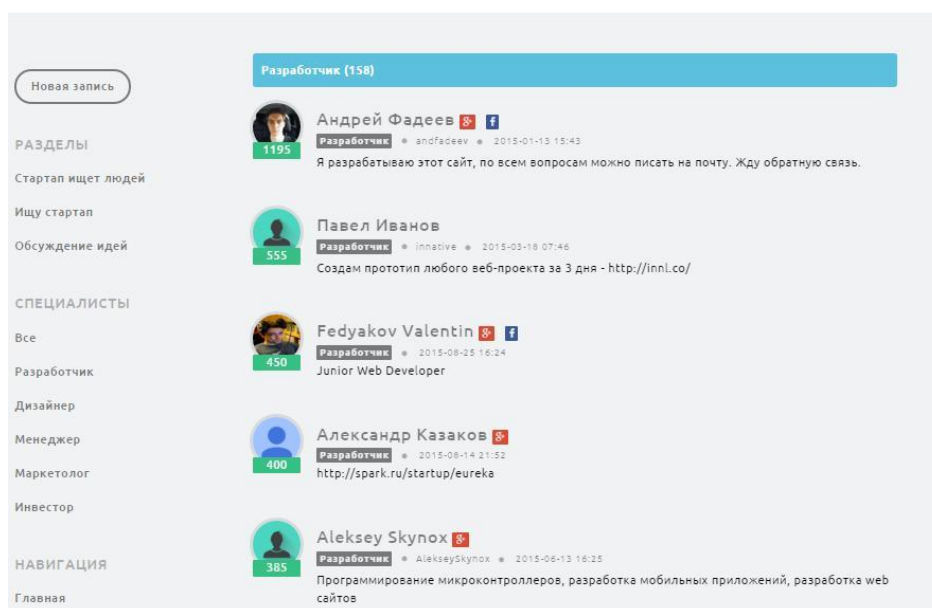


Рисунок 1.19 - Список користувачів

В списку користувачів ми можемо переглянути всіх, хто зареєстрований на сайті і деяку інформацію про них, а саме:

- ім'я та прізвище;
- статус користувача (розробник, звичайний користувач);
- коротка інформація про користувач;
- посилання на сторінки в соціальних мережах.

Таблиця 1.6 - Порівняльна характеристика програмних продуктів

Фірма-розробник	indiegogo	teamfinding	startup fellows
Тип продукту	Краудфандінг	Управління стартапами	Пошук стартапів та людей для стартапів
Дата заснування	19-04-2007	21-08-2012	25-12-2014
Місце знаходження	США, Каліфорнія, Сан-Франциско	Україна, Київ	США, Ашберн
Назва програмного продукту	Indiegogo.com	teamfinding.com	startupfellows.ru
Версії продукту	2015	2015	2015
Функціональність	Створення проектів для їх фінансування	Створення проектів для спільної розробки з користувачами	Створення проектів для спільної розробки з користувачами
Інтерфейс користувача	Приємний	Помірний	Не зручний
Допомога користувачу	Присутня	Відсутня	Відсутня

Порівнявши дані програмні продукти можна сказати, що startup fells і teamfinding виконують одні і ті ж функції, дають можливість користувачам створювати проекти, команди. Недоліком цих систем є їхній інтерфейс, який не є зручним, тому користувачу на перший погляд може здатись, що створення або пошук проекту є важким процесом. До переваг можна віднести створену систему коментарів під проектами, де користувачі можуть обговорити даний проект.

Indiegogo - є потужною системою по збору коштів для реалізації проектів. До недоліком можна віднести назвати недостатню кількість систем оплати.

На даний момент існує дуже багато сервісів для оплати в мережі інтернет і кожен користувач вибирає той сервіс, який йому найбільше до вподоби, тому я вважаю, що в системі, яка працює з сервісами оплати, необхідно по максимуму підключити ці сервіси для зручності користувачів. До переваг можна віднести інтерфейс, який дуже зручний і зрозумілий для користувача. Також однією з переваг є те, що вже за декілька хвилин ми можемо створити власний проект і розпочати збір коштів.

Після аналізу програмних продуктів, можна прийняти наступні рішення для розробки модуля системи:

- здатність поширювати проекти за допомогою соціальних мереж, що надасть можливість збільшити кількість користувачів та збільшити фінансування проектів;
- створити систему для підтвердження особи, щоб зменшити кількість людей, які хочуть “нажитися”, створивши проект, який неможливо реалізувати.

1.4 Специфікація вимог до модуля (системи)

Нижче наведено словник термінів розробленої системи.

Основні поняття та критерії предметної області та проекту:

Краудфандінг - об'єднання людей, які добровільно залучають свої гроші або інші ресурси, як правило через Інтернет, щоб підтримати розробки інших

людей. Краудфандингове фінансування виконує різні функції— допомога постраждалим, підтримка з боку вболівальників чи фанатів, підтримка політичних кампаній, фінансування стартап-компаній та малого бізнесу, створення програмного забезпечення і багато ін. Для стартування збору коштів обов'язково повинна бути мета, ціна її досягнення, а обрахунок усіх витрат і процес збору мають бути відкриті для публіки у вільному доступі.

Стартап – нещодавно створена компанія (можливо, ще не зареєстрована офіційно, але серйозно планує стати офіційною), що будує свій бізнес на основі інновацій або інноваційних технологій, не вийшла на ринок або ледве почала на нього виходити і що володіє обмеженим набором ресурсів. Часто стартап-компанії називають «гаражними». Особливо часто термін «стартап» застосовується відносно інтернет-компаній і інших фірм, що працюють всфері ІТ, проте, це поняття розповсюджується і на інші сфери діяльності.

Користувачі системи:

Простий користувач - створює проекти, навчається програмуванню, створює команду.

Програміст - спеціаліст в певній області, це може бути (UX дизайнер, UI дизайнер, php програміст і т.д). Шукає цікаві проекти, приймає участь у реалізації проектів, може створювати проекти, команди.

Компанія - шукає цікаві проекти, фінансує, купує них. Також наймає спеціалістів у свою компанію

У таблиці 1.8 наведена специфікація вимог до системи.

Таблиця 1.8 - Специфікація функціональних вимог

Ідентифікатор вимог	Назва (варіанту вимоги)	Атрибути вимог		
		Пріоритет	Складність	Контакт
1	Створення стартапу	Обов'язкове	середня	Користувач, Спеціаліст

2	Пошук стартапу	Обов'язкове	середня	Користувач, Компанія, Спеціаліст
3	Реєстрація	Обов'язкове	низька	Користувач, Компанія, Спеціаліст
4	Вхід	Обов'язкове	низька	Користувач, Компанія, Спеціаліст
5	Обмін повідомленнями	Обов'язкове	висока	Користувач, Компанія, Спеціаліст

У таблиці 1.9 наведена специфікація нефункціональних вимог системи.

Таблиця 1.9 - Специфікація нефункціональних вимог

Ідентифікатор вимоги	Назва вимоги	Атрибути вимог		
		Пріоритет	Складність	Контакт
Застосовність				
1	Час необхідний для навчання звичайних і досвідчених користувачів	Середній	Середня	Адміністратор
Надійність				
1	доступність	Високий	Середня	
2	середній час безвідмовної роботи	Високий	Висока	
Робочі характеристики				
1	швидкодія для транзакції	Високий	Середня	
2	продуктивність	Високий	Висока	

Вимоги до документації, призначеної для користувача, і до системи				
1	Зрозумілість	Високий	Низька	Адміністратор
Інтерфейси				
1	Зручність	Високий	Середня	
2	Зрозумілість	Високий	Середня	

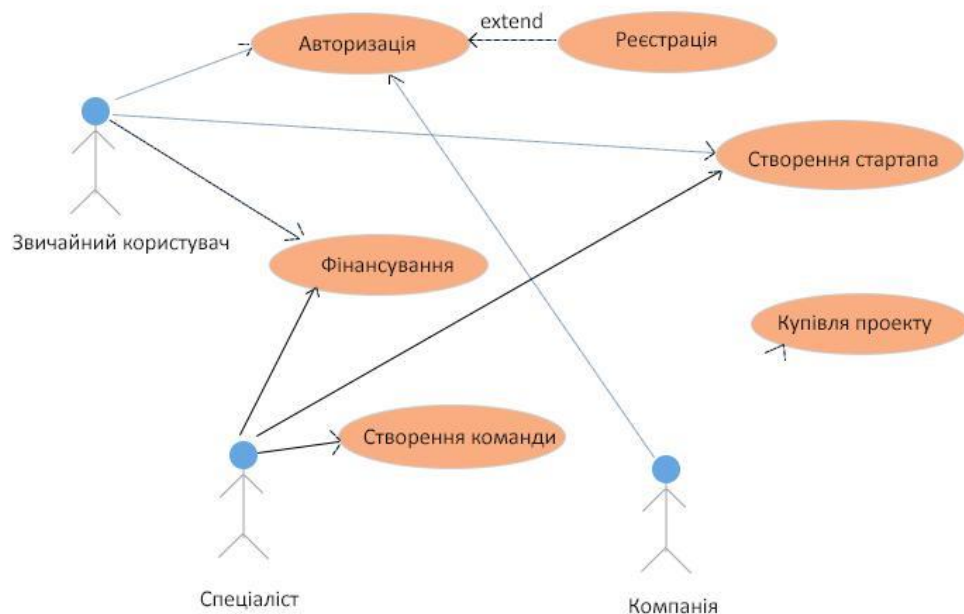


Рисунок 1.20 - Діаграма варіантів використання

Таблиця 1.10 - Варіант використання авторизації

Контекст використання	Перенаправлення на головну сторінку користувача що авторизувався
Дійові особи	Користувач, його інтересом є вхід у систему та подальше використання її в своїх цілях, а саме (створенні стартапа, пошуку та ін.)
Передумова	Має бути сторінка авторизації з формою для введення даних, при коректному вводі вхід у систему
Тригер	-
Сценарій	-

Постумова .	Після реєстрації відкривається сторінка користувача
-------------	---

The image shows a login interface within a rectangular frame. At the top center is a button labeled "Вхід". Below it are two input fields: the first is labeled "Введіть ваш e-mail" and the second is labeled "Введіть пароль". Below the password field is a link labeled "забули пароль?". To the right of this link is a button labeled "Війти".

Рисунок 1.21 - Розкадровка вхід в систему

Для використання системи управління стартапами користувач повинен зареєструватися. Для реєстрації необхідно заповнити наступні поля:

- поле email – потрібно ввести e-mail, який був введений при реєстрації;
- поле пароль - потрібно ввести пароль, який був введений при реєстрації;
- кнопка ввійти – після нажаття на кнопку, якщо введені дані були вірні, нас перенаправляє на сторінку сайту, якщо ні, то або не правильно були введені дані, або користувач з такими даними не існує.

Таблиця 1.11 - Варіант використання “створення стартапу”

Контекст використання	Перенаправлення на головну сторінку користувача що авторизувався
Дійові особи .	Користувач, спеціаліст – можуть створюють проект.

Передумова	Має бути сторінка авторизації з формою для введення даних, при коректному вводі вхід у систему
Тригер	
Сценарій	
Постумова	Перенаправлення на сторінку створеного проекту

Створення стартапа

Назва

Короткий опис

Категорія

Бюджет

Фінансування ☐ Да ☐ Ні

Команда

Рисунок 1.22 - Розкадровка створення стартапу

При створенні нового проекту користувач повинен заповнити всі необхідні поля. Після створення проекту, користувач може розпочати розробку та розвиток даного проекту.

Для створення нового проекту потрібно заповнити наступні поля:

- поле “Назва” – назва проекту, має бути унікальною;
- короткий опис - в цьому полі описують основну діяльність проекту і головну його мету;
- категорія – в даному полі потрібно вказати до якої категорії буде відноситися проект (веб – додаток, додаток на android тощо) ;

- бюджет – в даному полі користувач вводить доступний бюджет на розробку даного проекту;
- фінансування – в цьому полі є два варіанта вибору, так або ні. Коли користувач вибрав так, то даний проект зможе фінансуватися користувачами та компаніями, коли користувач вибрав ні, тоді дана можливість пропадає;
- команда – в даному полі користувач вказує, які користувачі йому потрібні (php програмісти, дизайнери тощо) ;
- кнопка вийти – після нажаття на кнопку, якщо введені дані були вірні, користувача перенаправляє на нашу сторінку, якщо ні, то або не правильно були введені дані, або користувач з такими даними не існує.

Висновок до розділу.

1. Основна діяльність об'єкту управління стартапами – створення, пошук та реалізація проектів різного типу.
2. Основними бізнес – процесами є:
 - процес формування календарного плану;
 - процес управління стартапом;
 - процес фінансування стартапу;
 - процес управління зустрічами;
3. Аналогами програмної системи є:
 - Indiegogo
 - Teamfinding
 - Sturtup fellows

2 ПРОЕКТУВАННЯ

2.1 Розробка архітектури програмної системи.

Веб сервіс буде використовувати 3 основні компоненти: веб браузер, веб сервер та базу даних. На рисунку 23 зображен діаграму компонентів, яка відображає залежності між компонентами програмного забезпечення.

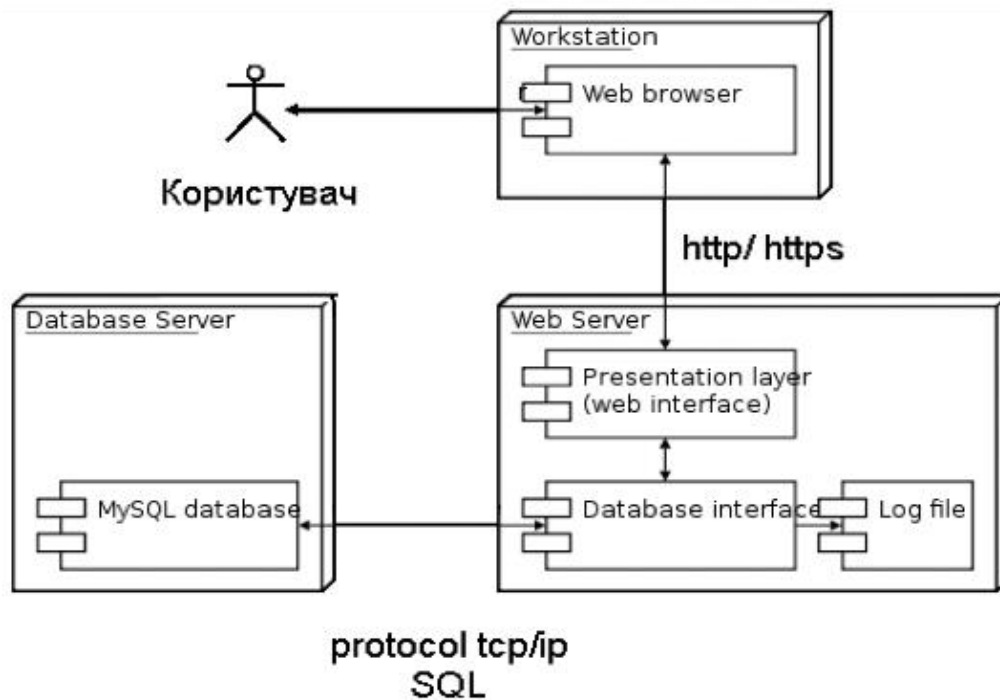


Рисунок 2.1 – Діаграма компонентів розроблюваної системи

Статичне представлення структурної моделі зображають у вигляді діаграми класів. Структурна модель додатку є досить складною. Система матиме велику кількість класів, що забезпечуватимуть можливість її функціонування. Більшість модулів системи відповідатимуть за інтерфейс або стосуватимуться особливостей налагодження роботи. Тому їх відображення на даному етапі є не потрібним, оскільки воно зробить діаграму класів занадто громіздкою та важкою для читання і сприйняття. Отже, на рисунку 2.2

діаграма відображена лише тими класами, які забезпечуватимуть виконання логіки програми.

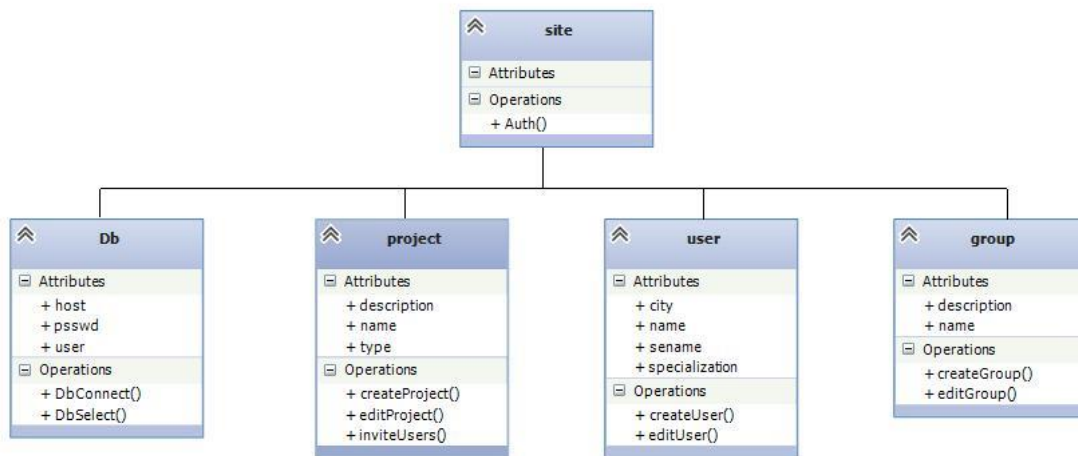


Рисунок 2.2 - Діаграма класів

На рисунку 2.3 зображена діаграма послідовності для варіанту використання «Редагування користувачів», яка відображає взаємодію об'єктів впорядкованих за часом. Можливість маніпулювання даними користувачів буде доступна лише адміністратору.

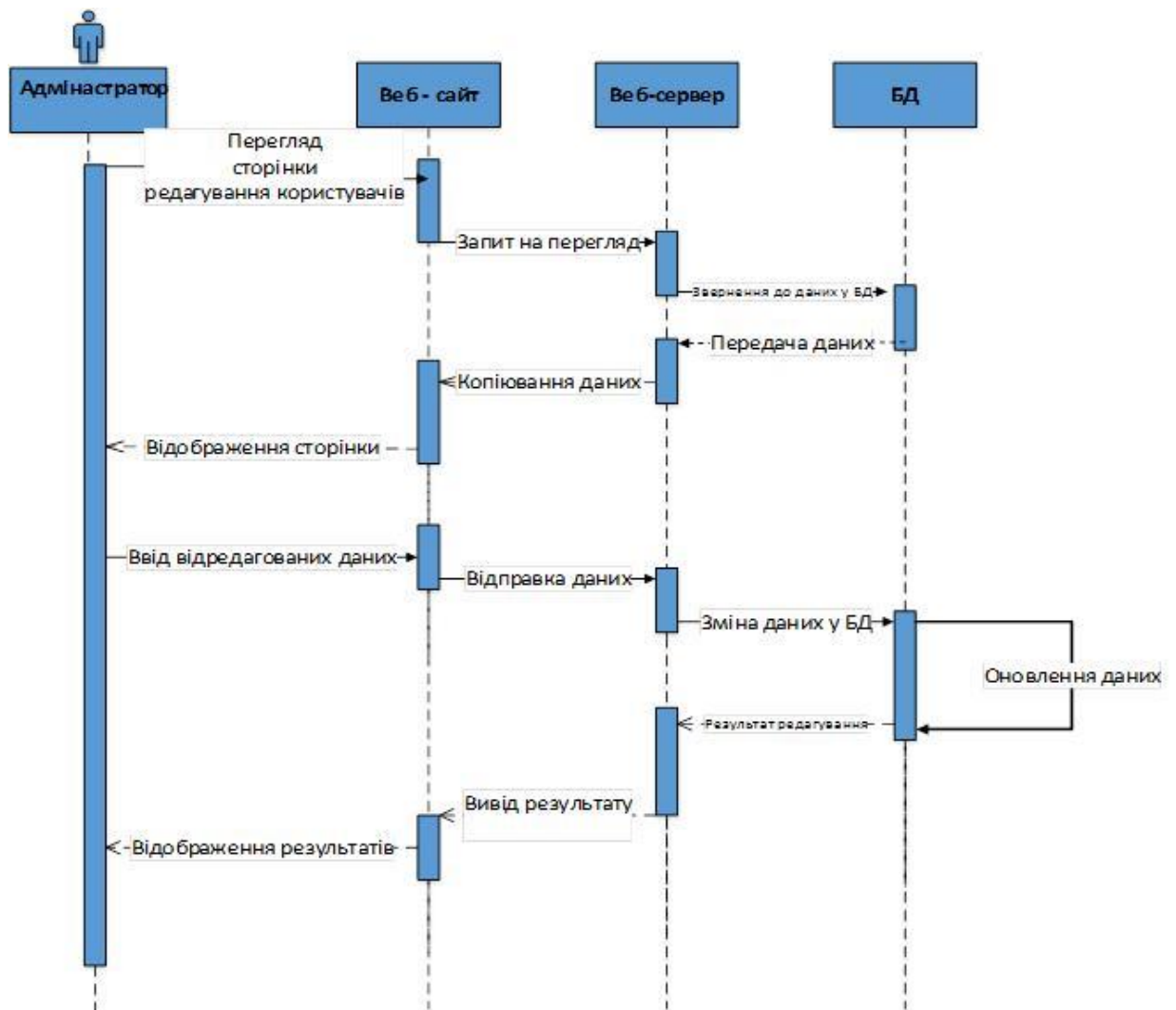


Рисунок 2.3 - Діаграма послідовності (процесу редагування користувачів)

На рисунку 2.4 зображена діаграма послідовності для варіанту використання «Додавання проекту», яка відображає взаємодію об'єктів впорядкованих за часом.

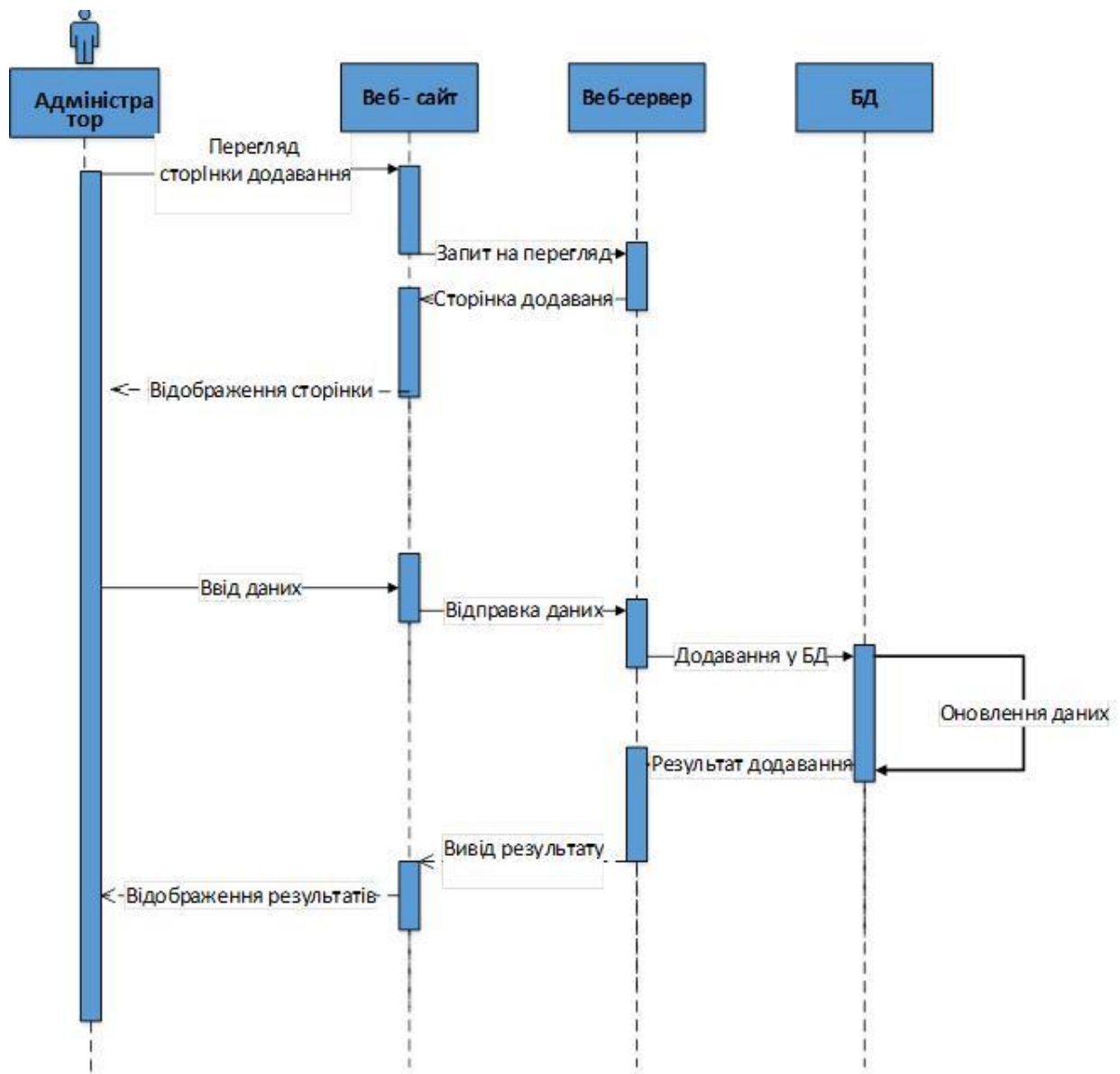


Рисунок 2.4 - Діаграма послідовності “Процесу додавання нового проекту”

Як видно з рисунку, найдовшим процесом буде відображення результатів після внесення змін, оскільки він містить найбільше етапів.

Для моделювання поведінки об'єктів системи при переході з одного стану в інший використовують діаграми станів. Діаграма станів для функції «Редагування» зображено на рисунку 2.5

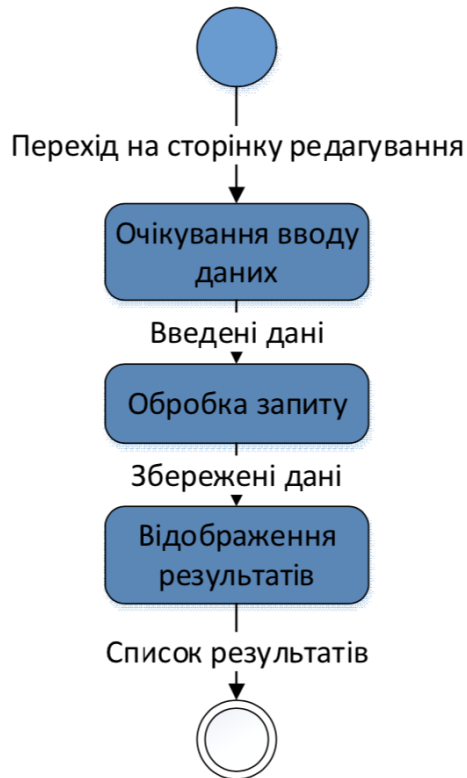


Рисунок 2.5 - Діаграма станів процесу “Редагування”

Як видно з рисунку, після переходу на сторінку редагування, система очікує вводу даних. Коли користувач здійснить дану операцію, введені дані обробляються, після чого зберігаються у БД. В подальшому дані оновлюються і відображаються користувачу.

На рисунку 2.6 зображено діаграму станів процесу “Видалення”



Рисунок 2.6- Діаграма станів процесу “Видалення”

Як видно з рисунку, після переходу на сторінку видалення, обраний кортеж обробляється системою. Під час виконання запиту запис видаляється.

На рисунку 2.7 зображено діаграму станів процесу “Додавання студента”

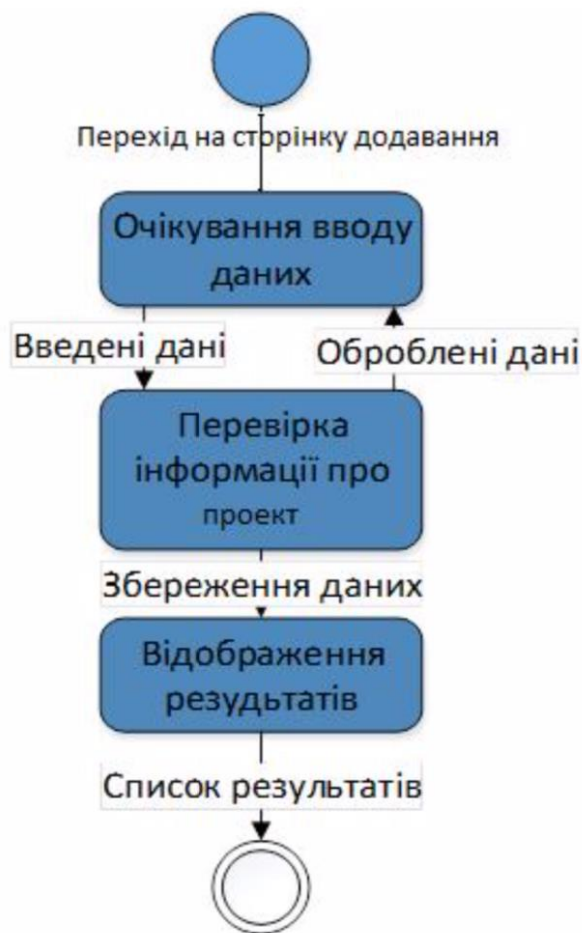


Рисунок 2.7 - Діаграма станів процесу “Додавання студента”

Як видно з рисунку, після переходу на сторінку додавання, система очікує вводу даних. Якщо введено назву проекту, ідентичний тому, який вже є у БД, система повідомить про те, що запис про проект уже присутній і запропонує перевірити правильність вводу. Якщо ж всі поля заповнені

коректно дані успішно будуть внесені у БД та відобразиться повідомлення, про успішно доданий запис.

2.2 Проектування структури бази даних

Для нормальної роботи системи, потрібно забезпечити збереження великого обсягу даних, доступ до яких буде можливий лише користувачам цього програмного продукту. Це стало можливим за допомогою баз даних та засобами їх управління, зокрема системою управління базами даних (СУБД)MySQL, яка використовується в системі.

Для реалізації програмного продукту було обрано клієнт-серверну архітектуру. У цій архітектурі запити розподілені між постачальниками послуг (серверами) та споживачами (користувачами). Фізично клієнт і сервер – це програмне забезпечення. Зазвичай вони між собою взаємодіють, за допомогою мережі, через мережеві протоколи і знаходяться на різних обчислювальних машинах. Програми – сервера, очікують від клієнтських програм запитів і надають їм свої ресурси, у вигляді даних або ж сервісних функцій.

На рисунку 2.8 наведено елементів та зв'язки між ними, які будуть використовуватись на етапі експлуатації системи.

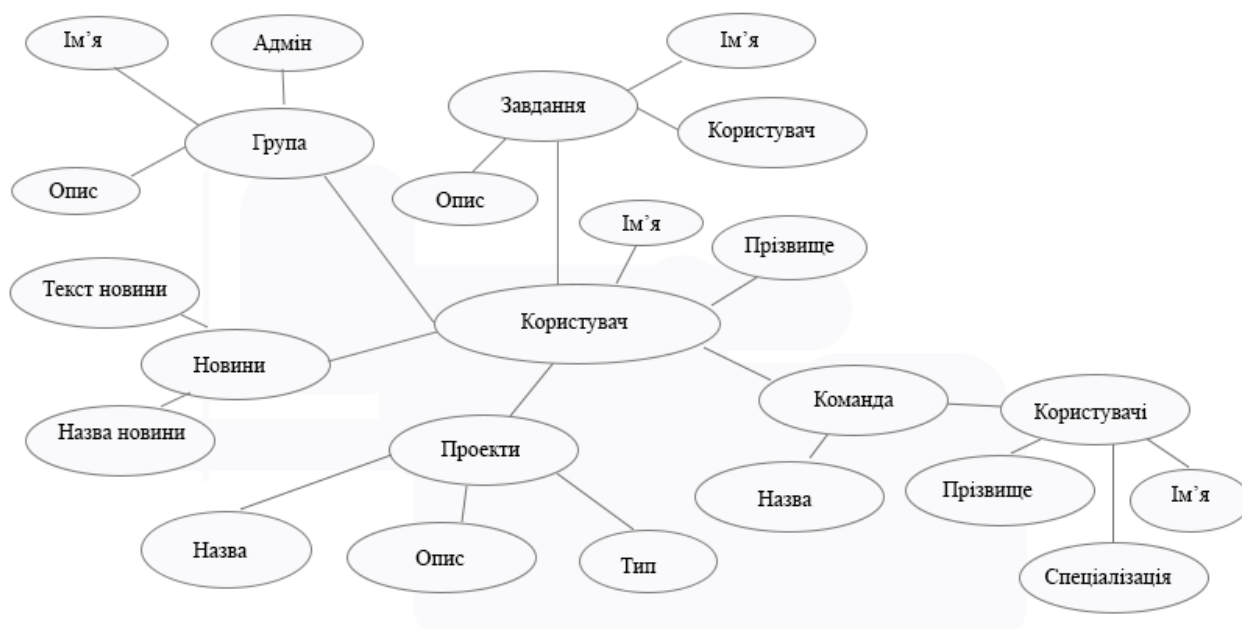


Рисунок 2.8 - Діаграма елементів і зв'язків

При високорівневому (концептуальному) проектуванні баз даних використовують ERD-модель. Під час проектування БД відбувається перетворення ERD-моделі в конкретну схему бази даних на основі обраної моделі даних. На рисунку 2.9 зображено модель ERD для розроблюваної системи.

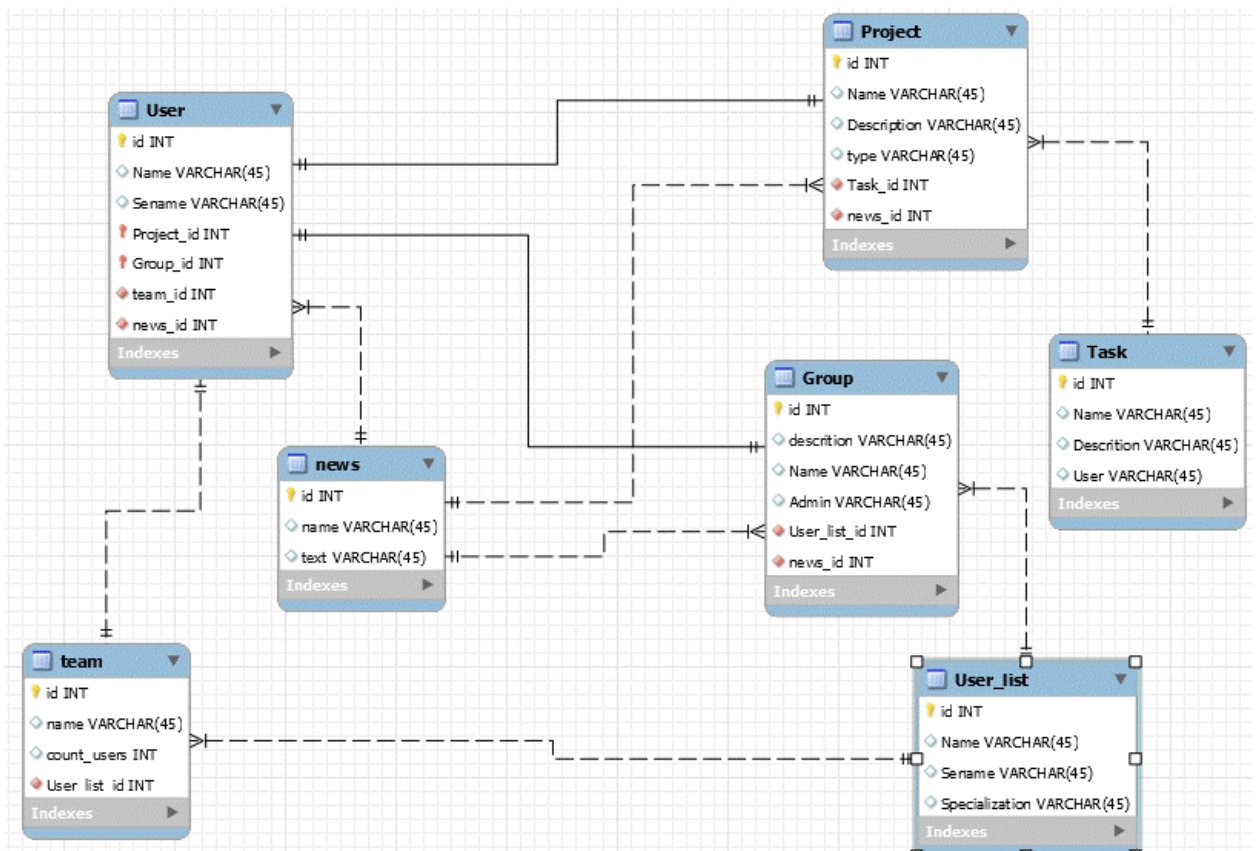


Рисунок 2.9 - Модель сутність-зв'язок (ERD)

Усі відношення автоматично індексуються, по первинному ключі. Завдяки чому значно пришвидшується процес пошуку даних у БД. DDL опис бази даних наведено нижче

```
SET @OLD_UNIQUE_CHECKS=@@UNIQUE_CHECKS,  
UNIQUE_CHECKS=0; SET  
@OLD_FOREIGN_KEY_CHECKS=@@FOREIGN_KEY_CHECKS,  
FOREIGN_KEY_CHECKS=0;  
SET @OLD_SQL_MODE=@@SQL_MODE,  
SQL_MODE='TRADITIONAL,ALLOW_INVALID_DATES';
```

Висновок до розділу.

В ході проектування програмної системи:

1. Було спроектовано архітектуру програмного продукту, розроблена діаграма класів, послідовності, діаграма станів.
2. Вибрана СУБД, спроектована діаграма елементів і зв'язків, ERD діаграма.

РОЗДІЛ 3. ПРОГРАМНА РЕАЛІЗАЦІЯ

3.1 Обґрунтування вибору мови програмування

Для розробки web-орієнтованої системи управління стартапами, була використана мова php.

PHP (Hypertext PreProcessor) - мова програмування, виконувана на стороні веб-сервера для розробки динамічних сайтів. Дана мова програмування є дуже гнучкою і потужною, тому використовується в проектах різного масштабу і є дуже популярним інструментом для розробки web-додатків.

Основну роль у виборі мови програмування зіграли такі аспекти:

- вільне програмне забезпечення;
- простий у використанні;
- підтримується великою кількістю користувачів і розробників;
- розвинена підтримка баз даних;
- велика кількість бібліотек і розширень;
- може використовуватися в ізольованому середовищі;
- може бути розгорнутий майже на будь-якому сервері;
- легко працювати на різних платформах та ОС.

3.1.1 Організація інтерфейсу з користувачем

При організації інтерфейсу з користувачем головна увага приділялася простоті і зручності, щоб людина, яка вперше зайшла на сайт, могла відразу розібратися і розпочала працювати з програмною системою.

Інтерфейс головної сторінки користувача містить в собі:

- “шапку” сайту:
 - а) назва сайту;
 - б) меню;
 - в) форма пошуку;

г) іконку користувача;

д) кнопку “Створити”

- блок з головною інформацією про користувача:

а) фото користувача;

б) інформація (Ім’я, прізвище, країна, місто, рік народження, мови, де навчався, де працює чи працював) ;

в) статус (коротка інформація, яку користувач хоче донести до інших, які будуть переглядати його сторінку) ;

г) дві кнопки (“Написати” – для обміну повідомленнями між користувачами, “Запросити” – для запрошення користувача у проект) ;

- блок з власними проектами (проекти, які створив користувач) ;

- блок з проектами у яких приймає участь;

- моя команда (учасники команди, в які перебуває сам користувач);

- контактна інформація (e-mail, телефон, скайп, сторінка в фейсбуці, сторінка в контакті).

- інтерфейс головної сторінки користувача має два різних вигляди:

- як бачить свою сторінку сам користувач;

- як бачать її інші користувачі.

На сторінці, яку бачать інші користувачі знаходиться основна інформація про користувача, контактні дані, проекти в яких користувач приймає участь і власні проекти, які створив він. Також, на сторінці користувача знаходиться дві кнопки: “Написати”, “Запросити”.

Скориставшись кнопкою “Написати”, ми можемо зв'язатися з користувачем, кнопка “Запросити” для того щоб ви могли запропонувати користувачеві прийняти участь у вашому проекті.

Інтерфейс користувача, як бачить його сам користувач містить таку саму інформацію, але відрізняється тим що замість кнопок: “Написати ”,

“Запросити”, кнопки: Проект “створити новий проект”, ”Команда” (створити команду).

Інтерфейс проекту, має містити в собі дані, які будуть зрозумілі всім хто зайде на сторінку проекту, щоб користувач відразу міг зрозуміти чому і для чого був створений цей проект, хто бере участь в розробці, кого шукає даний стартап, можливо саме такого спеціаліста не вистачає проекту, яким є переглядач проекту.

Інтерфейс сторінки проекту містить в собі:

- “шапку” сайту:
 - а) назва сайту;
 - б) меню;
 - в) форма пошуку;
 - г) іконку користувача;
 - д) кнопку “Створити”
- блок з інформацією про проект:
 - а) Опис головної ідеї проекту;
 - б) чи потребує проект фінансування ;
 - в) які спеціалісти потрібні для розробки проекту;
- блок зустрічей (якщо зустріч створена) ;
- блок з календарним планом;
- блок зі списком учасників, які приймають участь у розробці проекту.

Інтерфейс сторінки проекту, містить в собі список всіх створених стартапів користувачами, це зроблено для зручного перегляду проектів і щоб кожен спеціаліст міг вибрати зі списку проект, який йому найбільш до вподоби.

Всі проекти динамічно завантажуються при прокрутці сторінки користувачем, що є дуже зручно для швидкого перегляду списку.

Інтерфейс списку проектів містить:

- список всіх проектів:

а) зображення проекту;

б) назву.

- кнопку “Завантажити ще” (на випадок, якщо сторінка динамічно не загрузилася).

Інтерфейс сторінки списку команд, містить в собі список команд. Кожен користувач може перейти на сторінку команди і подивитися на її учасників, проекти які вона розробила.

Всі команди динамічно завантажуються при прокрутці сторінки користувачем, що є дуже зручно для швидкого перегляду.

Інтерфейс списку проектів містить:

- список всіх команд:

а) зображення команди;

б) назву.

в) список двох проектів команди.

- кнопку “Завантажити ще” (на випадок, якщо сторінка динамічно не загрузилася).

3.1.2 Програмна реалізація проекту

При створенні системи управління стартапами використовувалась мова програмування PHP та об'єктно-орієнтований підхід.

Об'єктно – орієнтований підхід - це метод програмування, в якому програми представляються у вигляді сукупності об'єктів, які взаємодіють між собою.

Об'єкт є екземпляром класу, а клас – це спосіб опису сутності, який визначає стан і поведінку, а також правила для взаємодії з цією сутністю. Клас складається з властивостей та методів.

В даній програмній системі було розроблено такі класи:

- Register;
- User;

- Db;
- Project.

За допомогою класу Register відбувається реєстрація нових користувачів, в даному класі основну роль виконують наступні методи:

- GetId().

```
public function getId($email){
    $db = new Db();
    $db->connect();
    $sql = "SELECT id,email,password FROM users WHERE
    email='$email' limit 1"; $res = mysql_query($sql);
    $val = mysql_fetch_assoc($res);
    return $val['id'];
}
```

- ChekData().

```
public function
checkData($user_name,$last_name,$email,$password,$repeat_
password){
    if(empty($user_name) || empty($last_name) ||
    empty($email) || empty($password)||
    empty($repeat_password) ){
        echo("Не всі дані заповнено");
    }else{
        if($password === $repeat_password){
            $password = md5($password);
            $db = new Db();
            $db->connect();
            $sql = 'INSERT INTO `users` (`user_name`, `last_name`,
            `email`, `password`) VALUES ("'.$user_name.'",
            "'.$last_name.'", "'.$email.'", "'.$password.'")';
            $db->write_data($sql);
```

```

header('Location: http://dev.pro/reg_info.php');
}else{
echo('Паролі не співпадають');
}
}
}

```

- Login().

```

public function login($email,$password){
if(empty($email) || empty($password)){
echo("Не всі поля заповнені");
}else{
$db = new Db();
$db->connect();
$user = new Register();
$email = $user->getEmail();
$sql = "SELECT id,email,password FROM users WHERE
email='$email' limit 1"; $res = mysql_query($sql);
$val = mysql_fetch_assoc($res);
if($val['password'] === md5($password)){
$_SESSION['id'] = $val['id'];
header('Location: http://dev.pro');
}else{
echo('Не вірно введені дані');
}
}
}

```

- WriteInfoUser();

```

public function
writeInfoUser($country,$city,$language,$school,$job){

```

```

if(empty($country) || empty($city) || empty($language) ||
empty($school) || empty($job)){
echo("Не всі поля заповнені");
}else{
$db = new Db();
$email = $_SESSION['email'];
$db->connect();
$sql = "SELECT id FROM users WHERE email='$email'"; $res
= mysql_query($sql);
$val = mysql_fetch_assoc($res);
$id = $val['id'];
$sql = "UPDATE users SET country = '$country', city = '$city',
language = '$language', school = '$school', job = '$job' WHERE
id='$id'";
$db->write_data($sql);
}
}

```

Ці методи виконують перевірку, запис, та збереження даних, які вводить користувач а форму при реєстрації.

Клас User несе в собі методи за допомогою яких створюється персональна сторінка користувачів усіх трьох типів, а саме спеціаліст, компанія, звичайний користувач.

Клас DB розроблений для роботи з базою даних, а саме підключенням та вибором потрібної бази даних і маніпуляціями з даними різного роду, таких як запис, видалення, редагування.

Клас Project виконує функцію створення нового проекту користувачем.

Містить такі методи:

- CreateProject();

```

public function CcreateProject($title,$description,$id_creator
,$ava,$id){ if(empty($title) || empty($description) ||
empty($id_creator) || empty($ava) ){
echo("Не всі дані заповнено");

}else{
$db = new Db();
$db->connect();
$sql = 'INSERT INTO `projects` (`title`, `description`,
`id_creator`, `ava`) VALUES ("'.$title.'", "'.$description."',
"'.$id_creator.'", "'.$ava.'")';
$db->write_data($sql);
header('Location: http://dev.pro/project.php?id=$id');
}
}

```

- UpdateProject();

```

public function UpdateProject($title,$description,$id_creator
,$ava,$id){
if(empty($title) || empty($description) || empty($id_creator) ||
empty($ava) ){
echo("Не всі дані заповнено");
}else{
$db = new Db();
$email = $_SESSION['email'];
$db->connect();
$sql = "SELECT id FROM users WHERE email='$email'"; $res
= mysql_query($sql);
$val = mysql_fetch_assoc($res);
$id = $val['id'];

```

```
$sql = "UPDATE projects SET title = '$title, description=
'$description, ava = '$ava, WHERE id='$id'";
$db->write_data($sql);}
```

Ці методи відповідають за створення нового проекту, редагування даних, запрошення нових учасників, створення календарного плану та зустрічей.

3.2 Програмна реалізація бази даних

Створення бази даних програмної системи відбувається в СУБД MySQL. Phpmyadmin – веб-додаток написаний на php і є інтерфейсом для адміністрування СУБД MySQL, дозволяє через браузер здійснювати адміністрування сервера MySQL, запускати SQL команд і переглядати вміст таблиць і баз даних!

Для створення таблиць ми будемо використовувати SQL запити.

Створення таблиці відбувається в phpmyadmin. Створити нову таблицю можна 2 способами:

- за допомогою SQL запита (див. рис. 3.1)

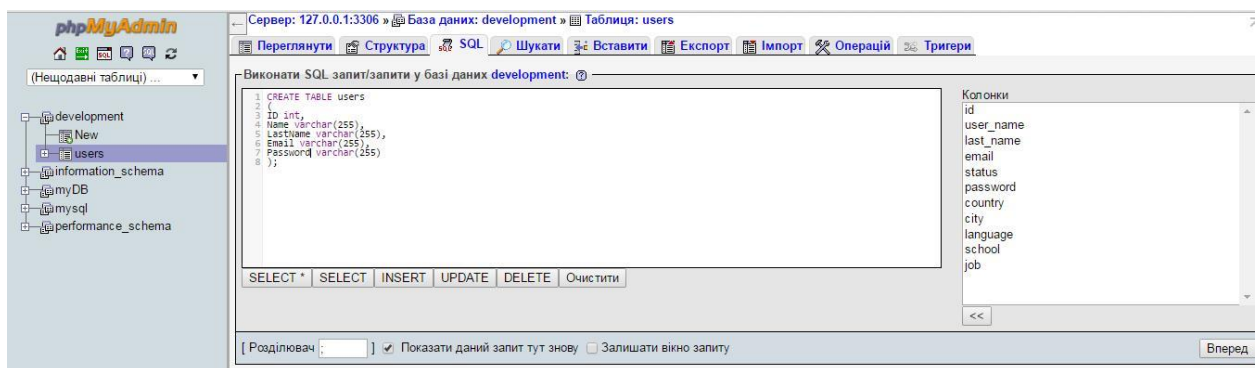


Рисунок 3.1 - Створення таблиці за допомогою SQL запиту

- за допомогою форми (див. рис. 3.2)

Рисунок 3.2 Створення таблиці за допомогою форми

В результаті ми отримаємо таблицю з заданими полями (дивитися рис. 3.3).

id	user_name	last_name	email	status	password	country	city	language	school	job
29	2020	2020	2020@gmail.com	0	1fa5d3708a5ad9daaa944a8de2cd24f6	США	США	Українська	ЕУ	Фріланс

Рисунок 3.3 Створена таблиця

Після створення нової таблиці ми можемо робити різні операції над нею, такі як додавання записів, видалення, редагування, видалення таблиць, редагування таблиці.

Є чотири основні SQL запити, без яких була б неможлива робота з базами даних:

- додавання нового запису;
- видалення нового запису;
- редагування запису;
- вибір даних з бази даних.

Для додавання нового запису виконують такий SQL запит - \$sql = 'INSERT INTO `users` (`user\_name`, `last\_name`, `email`, `password`) VALUES (".\$user_name.", " ".\$last_name.", " ".\$email.", " ".\$password.)";

Видалення даних відбувається за допомогою команди – “DELETE FROM users WHERE name=\$name;”;

Редагування вже існуючих даних в базі даних відбувається за допомогою запиту - \$sql = "UPDATE users SET country = '\$country', city = '\$city', language = '\$language', school = '\$school', job = '\$job' WHERE id='\$id'";

Вибірка виконується за допомогою команди - \$sql = SELECT *`users
WHERE name = \$name;

Висновок до розділу.

В ході програмної реалізації було:

1. Організовано інтерфейс користувача
2. Розроблена програмна реалізація проекту, а саме опис класів, методів.
3. Програмна реалізація баз даних, вибрана система управління базами даних, варіанти SQL запитів

4. ТЕСТУВАННЯ ТА ДОСЛІДНА ЕКСПЛУАТАЦІЯ

4.1 Тестування програмного продукту

Тестування – це процес дослідження, що використовується для виміру якості програмного продукту і має на меті:

- показати розробникам і замовникам відповідність програми до вимог;
- визначити моменти, в яких поведінка програми не відповідає специфікації.

Як показує практика, програм без помилок не буває і найчастіше за все винуватцями помилок є самі програмісти. Тому програміст, повинен писати не лише ефективні програми, а знаходити помилки в цих програмах. Тому тестування веб-орієнтованої системи управління стартапами є не лише доцільним, а й необхідним аспектом при розробці даної системи.

Під час проведення тестування програмної системи були використані такі види тестування:

- функціональне тестування;
- тестування сумісності;
- тестування продуктивності;
- тестування GUI;
- тестування безпеки;

Також було застосовано як автоматизоване тестування так і ручне. Для автоматизованого тестування використовувався інструмент Selenium.

Selenium – це інструмент для автоматичного тестування Web – додатків. На рисунку 4.1 зображено тестування системи реєстрації на сайті за допомогою Selenium.

Таблиця 4.1 - Функціональні тестові випадки

id	Модуль	Назва	Кроки	Очікуваний результат	Статус
1	Register	Реєстрація	1. Перейти на сторінку реєстрації 2.Заповнити форму 3.Натиснути кнопку “Реєстрація”	Перенаправлення на сторінку заповнення інформації про користувача	Виконано
2	Register	Заповнення даних про користувача	1. Заповнити форму 2. Натиснути на кнопку “Завершити реєстрацію”	Перенаправлення на сторінку користувача	Виконано
3	Project	Створення нового проекту	1.Натиснути на кнопку “Створити” в шапці сторінки 2.Заповнити форму 3. Натиснути на кнопку “Створити”	Створення нового проекту та добавлення на сторінку користувача	Виконано
4	Project	Запрошення в проект	1.Натиснути на сторінці проекту на кнопку “Запросити” 2.Вибрати зі списку тих, кого потрібно запросити	Відправитися запрошення на участь у проекті	Виконано
5	Project	Запрошення в проект через сторінку користувача	1.Зайти на сторінку користувача, якого хочемо запросити 2.Натиснути на кнопку “Запросити” 3.Вибрати зі списку проект, в який хочемо запросити	Відправитися запрошення на участь у проекті	Виконано
6	Project	Вигнання з проекту	1.Зайти на сторінку проекту 2.Відкрити список учасників 3.Натиснути на кнопку “Вигнати з проекту ” на проти того	Учасник пропаде зі списку учасників проекту	Виконано

			користувача, яий нам потрібен		
7	Project	Розприділ ен ня ролей в проекті	1.Відкрити список учасників 2. Натиснути на кнопку “Посада” 3.Вибрати зі списку посаду 4. Натиснути “Призначити”	Учасник отримає посаду в команді	Виконано
8	Project	Звільнення з посади	1.Відкрити список учасників 2. Натиснути на кнопку “посада” 3. Натиснути на кнопку “Звільнити”	Учасника буде звільнено з посади	Виконано
9	Project	Зміна посади	1.Відкрити список учасників 2. Натиснути на кнопку “Посада” 3.Вибрати зі списку іншу посаду 4. Натиснути “Призначити”	Змінена посада	Виконано
10	Profile	Контактна інформація	1.Натиснути на посилання “Змінити контактні дані” 2.Заповнити дані 3.Натиснути “Зберегти”	Дані будуть змінені	Виконано
11	Profile	Основна інформація	1.Натиснути в головному блоці на іконку “налаштування” 2. Змінити дані на потрібні 3. натиснути “зберегти”	Інформація буде змінена	Виконано

4.2.2 Тестові випадки тестування сумісності

Під час розробки тестів сумісності було спроектовано 8 тестових випадків.

Таблиця 4.2 - Тестові випадки тестування сумісності

id	Назва	Очікуваний результат	Результат	Статус
1	Реєстрація	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано
2	Сторінка користувача	Вигляд, у всіх браузерах однаковий	В ІЕ 8 версії поїхав блок з контактами	Помилка
3	Сторінка проекту	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано
4	Створення проекту	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано
5	Сторінка користувачів	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано
6	Сторінка команд	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано
7	Сторінка компаній	Вигляд, у всіх браузерах однаковий розробки	Вигляд, у всіх браузерах однаковий розробки	Виконано
8	Сторінка Учасників проекту	Вигляд, у всіх браузерах однаковий	Вигляд, у всіх браузерах однаковий	Виконано

4.2.3 Тестові випадки тестування продуктивності

Під час розробки тестів продуктивності було спроектовано 8 тестових випадків.

Таблиця 4.4 - Тестові випадки тестування продуктивності

id	Назва	Очікуваний результат	Результат	Статус
1	Завантаження сторінки користувача	Швидкість завантаження не більше 1 сек	0,542	Виконано
2	Завантаження сторінки проекту	Швидкість завантаження не більше 1 сек	0,443	Виконано
3	Завантаження сторінки команд	Швидкість завантаження не більше 1 сек	0,574	Виконано
4	Завантаження сторінки компаній	Швидкість завантаження не більше 1 сек	0,612	Виконано
5	Завантаження сторінки користувачів	Швидкість завантаження не більше 1 сек	0,489	Виконано
6	Завантаження сторінки реєстрацій	Швидкість завантаження не більше 1 сек	0,375	Виконано
7	Створення нового проекту	Швидкість завантаження не більше 1 сек	0,468	Виконано
8	Завантаження сторінки учасників проекту	Швидкість завантаження не більше 1 сек	0,398	Виконано

4.2.4 Тестові випадки тестування GUI

Під час розробки тестів GUI було спроектовано 9 тестових випадків.

Таблиця 4.6 - Функціональні тестові випадки

id	Назва	Кроки	Очікуваний результат	Статус
1	Форма реєстрації	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
2	Сторінка користувача	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
3	Сторінка проекту	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
4	Форма створення проекту	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
5	Сторінка управління учасниками проекту	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
6	Сторінка команд	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
7	Сторінка компаній	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
8	Сторінка команд	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
9	Сторінка редагування даних	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано

10	Форми	Вигляд, спроектований на початку розробки	Вигляд, спроектований на початку розробки	Виконано
11	Кнопки	Вигляд, спроектований на початку розробки	В ІЕ 7.0 в кнопок відсутні заокруглення	Помилка

4.2.5 Тестові випадки тестування безпеки

Під час розробки тестів безпеки було спроектовано 6 тестових випадків.

Таблиця 4.7 - Тестові випадки тестування безпеки

id	Назва	Очікуваний результат	Результат	Статус
1	Перевірка на запис в формі SQL запиту	Всі SQL запити будуть видалені з форми і не записані в базу даних	Всі SQL запити видалені з форми і не записані в базу даних	Виконано
2	Перевірка на запис в формі проекту SQL запиту	Всі SQL запити будуть видалені з форми і не записані в базу даних	Всі SQL запити видалені з форми і не записані в базу даних	Виконано
3	Перевірка на цілісність інформації користувача	Доступ до інформації користувача лише за його дозволом	Доступ до інформації користувача лише за його дозволом	Виконано
4	Перевірка на шифрування паролів користувачів	Паролі зашифровані за допомогою функції md5 і записані в базу даних у зашифрованому вигляді.	Паролі зашифровані і записані в базу даних	Виконано
5	Перевірка на дешифрування паролів	При авторизації введений пароль користувачем шифрується і порівнюється з	При авторизації введений пароль користувачем шифрується і порівнюється з	Виконано

		записаним шифром в базі даних	записаним шифром в базі даних	
6	Перевірка на доступ до закритих розділів	При переході на закриті сторінки, користувача має перенаправляти на його сторінку.	При переході на закриті сторінки, користувача направляє на його сторінку. .	Виконано

4.3 Функціональне тестування

Функціональне тестування – це найбільш трудомісткий етап тестування ПЗ. Суть даного тестування полягає у перевірці всього функціоналу, в даному випадку ми будемо перевіряти наступний функціонал:

- перевірка всіх обов’язкових функцій
- тестування всіх користувацьких форм (реєстрації, авторизації, створення нового стартапу);
- перевірка роботи пошуку;
- перевірка всіх гіперпосилань, та пошук не робочих посилань;
- перевірка завантаження файлів на сервер;

Результат виконання функціонального тестування.

Підсумок тестування:

10 тестів з 11 функціональних тестів наведених в таблиці 4.1 Пройшли успішно, отже функціональне тестування розглядається як частково успішне – 97% тестових випадків пройшли.

Відомі дефекти:

Дефект No1: Коли користувач вводить дані в форму реєстрації, то може ввести декілька разів один і той самий логін.

Опис: Відповідно, до тестового випадку “Перевірка форми”, система повинна при повторному введенні одного і того ж логіна інформувати, що користувач з таким логіном вже є.

Подолання дефекту: Даний дефект можна виправити, добавивши в class Register умову, яка буде перевіряти введений користувачем логін на існування ідентичного у базі даних.

4.4 Тестування сумісності

Тестування сумісності – це вид тестування, метою якого є перевірка роботи програми в певному оточенні. Суть даного тестування полягає у перевірці роботи програми в різних середовищах, в даному випадку у різних браузерах. Це дасть змогу з’ясувати необхідні зміни, щоб в подальшому система працювала коректно у всіх браузерах.

Перевірку на сумісність будемо проводити у таких браузерах:

- Google Chrome;
- Opera;
- Mozilla Firefox;
- Internet Explorer (не нижче 7 версії);
- Safari;
- UCBrowser (для Android);
- Safari (для IOS);
- UCBrowser (для Windows phone).

Результат виконання функціонального тестування. Підсумок тестування:

7 тести з 8 функціональних тестів наведених в таблиці 4.2 Пройшли успішно, отже тестування на сумісність розглядається як частково успішне – 97% тестових випадків пройшли.

Відомі дефекти:

Дефект No1: В Internet Explorer 7.0 при перегляді головної сторінки користувача, блок з контактною інформацією з'їхав в бік.

Опис: Система повинна при перегляді сторінок в різних браузерах виглядати однаково з мінімальним відхиленням у показниках.

Подолання дефекту: Даний дефект можна виправити, змінивши у файлі стилів позиціонування блоку з fixed на absolute.

4.5 Тестування продуктивності

Тестування продуктивності – це тестування призначене для перевірки на скільки швидко працює програма, або її частина під навантаженням.

Головним критерієм продуктивності веб-додатку є:

- час відповіді сервера;
- час відображення.

При перевірці продуктивності програмного продукту ми будемо використовувати такі види тестування продуктивності:

- навантажувальне тестування;
- стресове тестування;
- тестування стабільності;
- конфігураційне тестування.

Результат виконання тестування продуктивності.

Підсумок тестування:

10 тестів з 10 тестів продуктивності наведених в таблиці 1. Проїшли успішно. Час відповіді сервера – 0,35 сек. , час з'єднання з сервером – 0,14 сек, отже спираючись на результати, можна сказати, що тестування продуктивності пройшло успішно.

4.6 Тестування GUI

Тестування GUI – це перевірка графічного інтерфейсу користувача на відповідність до програмного додатку, як він виглядає, чи виконаний в

єдиному стилі. Під час тестування GUI було проведено інтерфейс користувача на:

- відповідність стандартам графічних інтерфейсів;
- тестування з різними розширеннями екранів;
- тестування в обмежених умовах.

Підсумок тестування:

7 тестів з 8 тестів GUI пройшли успішно , отже тестування розглядається як частково успішне – 97% тестових випадків пройшли.

Відомі дефекти:

Дефект No1: В Internet Explorer заокруглення кнопок та блоків не відображається.

Опис: Заокруглення елементів інтерфейсу повинне виглядати скрізь однаково.

Подолання дефекту: Даний дефект можна виправити, добавивши в стилі строчку, що містить наступний код – `behavior: url(border-radius.htc)`.

4.7 Тестування безпеки

Тестування безпеки – це тестування в якому перевіряється вразливість програмного додатку, та проводиться аналіз ризиків, пов'язаних з вірусами та атаками хакерів. Під час даного тестування було перевірено програмну систему на:

- конфіденційність;
- цілісність;
- ідентифікація;
- доступність;
- безвідмовність.

Результат виконання функціонального тестування.

Підсумок тестування:

10 тестів з 11 тестів пройшли успішно, отже функціональне тестування розглядається як успішне – 100% тестів пройшли

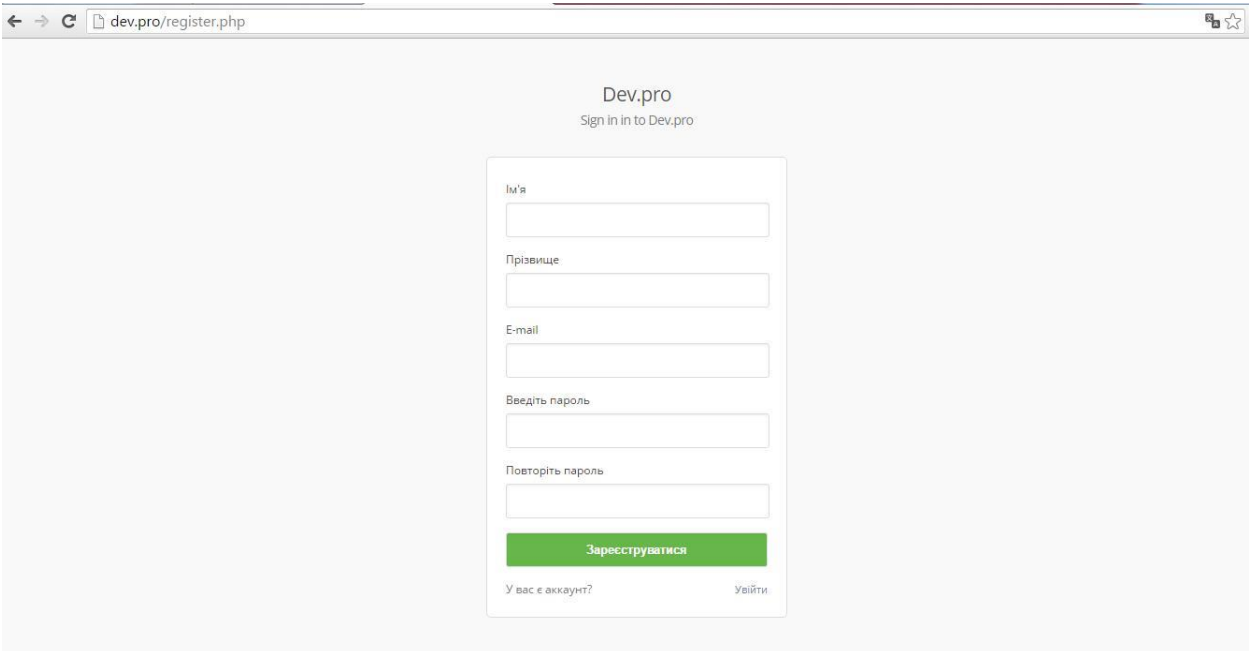
Умови тестування, які визнавалися успішними були наступні:

- розробка тестів:

1. Всізапланованітестовівипадкирозроблено;
2. Покриття тестами програмних вимог досягає 100%;
3. Покриття тестами варіантів використання досягає 100%;
4. тестування:
5. Всірозробленітестовівипадкивиконано;
6. Виконано тестування продуктивності, вимоги продуктивності задоволено;
7. Всі внутрішні дефекти виправлені і виправлення підтверджено.
8. Всі наведені умови задоволено, проект вважається успішним.

4.8 Інструкція користувача

Для того що розпочати роботу з програмною системою нам необхідно зареєструватися, для цього потрібно перейти в браузері по посиланні <http://127.0.0.1/register.php> . В результаті ми побачимо форму реєстрації (див. рис. 4.2), в якій потрібно заповнити всі поля.



The screenshot shows a web browser window with the address bar displaying 'dev.pro/register.php'. The page content is a registration form for 'Dev.pro'. At the top, it says 'Dev.pro' and 'Sign in to Dev.pro'. The form consists of several input fields: 'Ім'я' (Name), 'Прізвище' (Surname), 'E-mail', 'Введіть пароль' (Enter password), and 'Повторіть пароль' (Repeat password). Below these fields is a green button labeled 'Зареєструватися' (Register). At the bottom of the form, there are two links: 'У вас є акаунт?' (Do you have an account?) and 'Увійти' (Login).

Рисунок 4.2 – Форма реєстрації

Після заповнення форми потрібно натиснути на кнопку “Зареєструватися”. Якщо всі дані були вірно заповнені, то нас направить на сторінку заповнення даних про користувача (див. рис. 4.3).

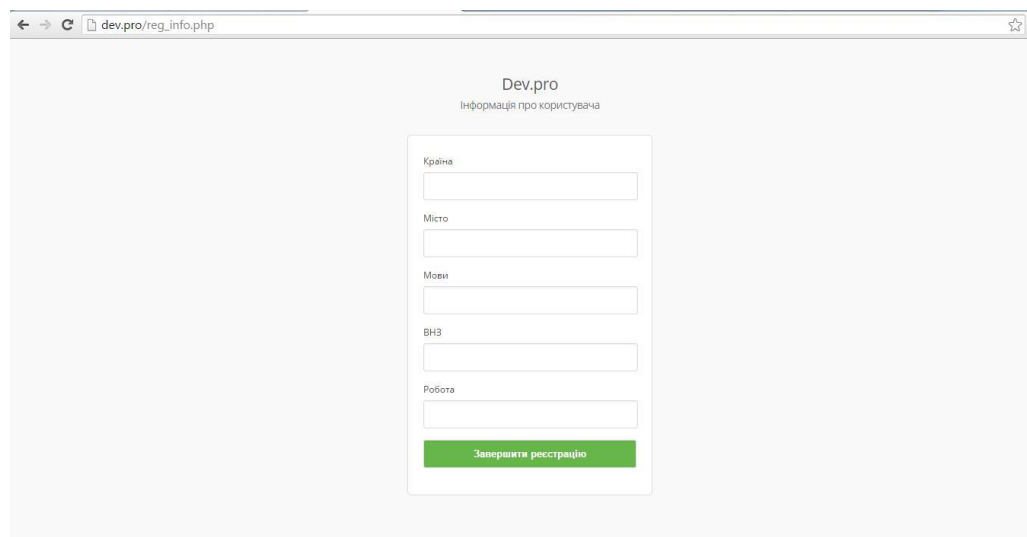


Рисунок 4.3 – Інформація про користувача

Після завершення реєстрації, система направить нас на сторінку користувача (див. рисунок 4.4).

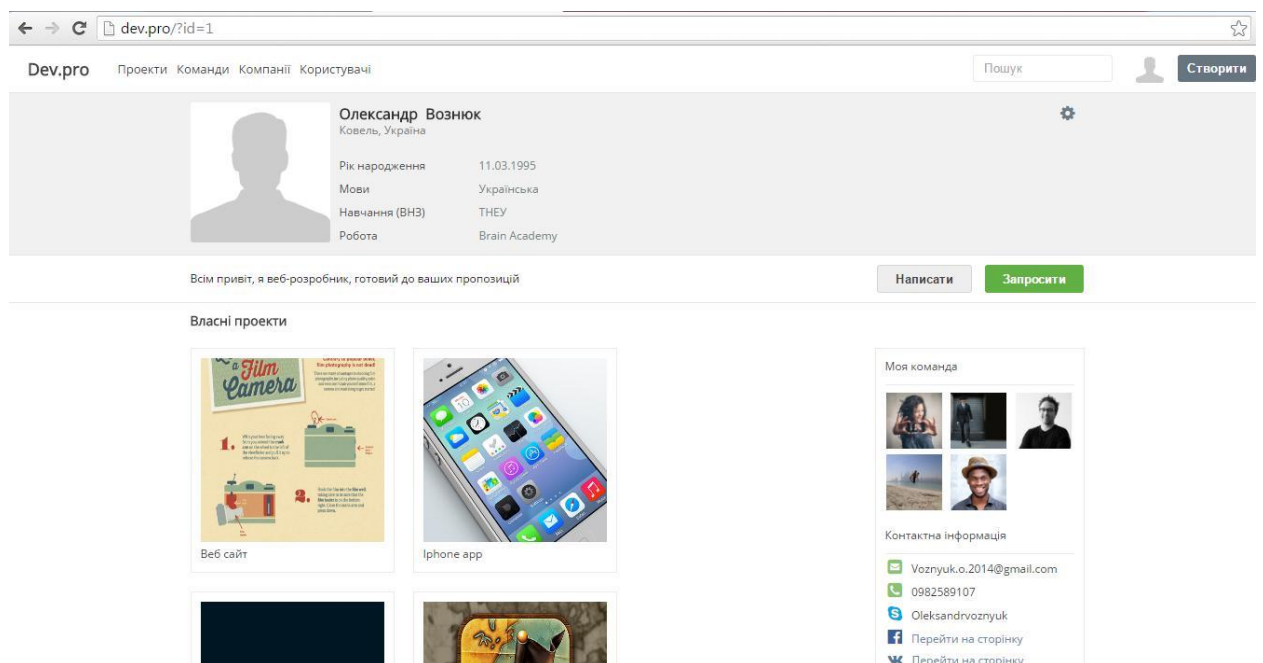
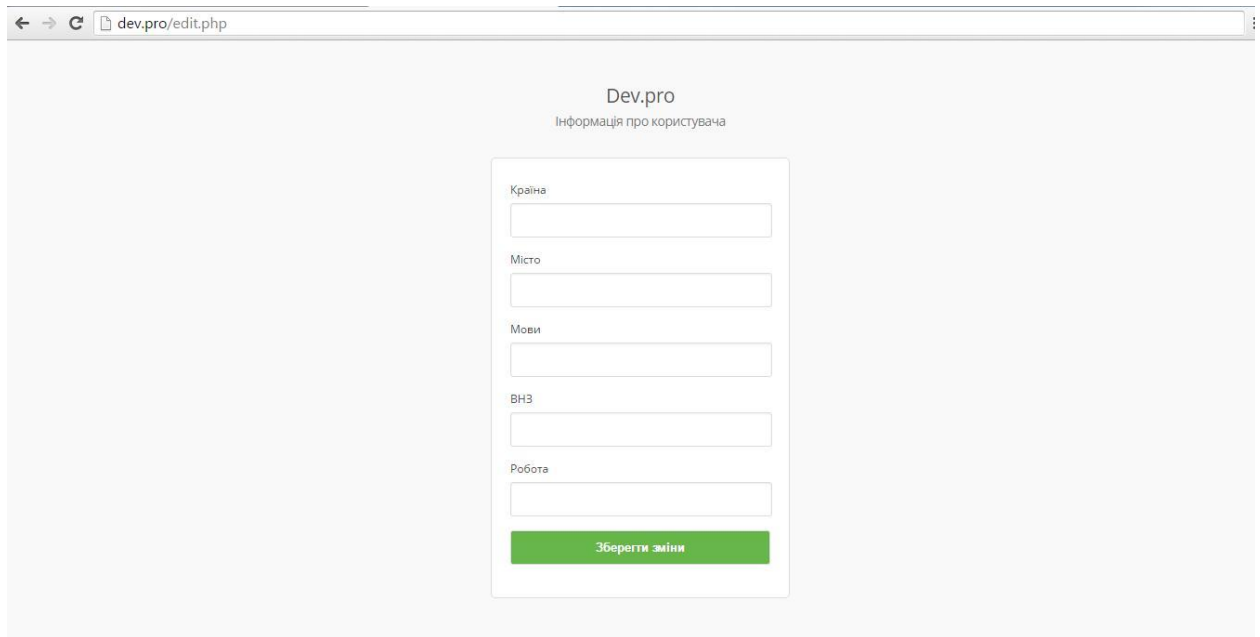


Рисунок 4.4 – Сторінка користувача

Для того щоб редагувати дані користувача потрібно натиснути на іконку “налаштування” з правої сторони в блоці про користувача. Після чого нас направить на сторінку редагування даних (див. ри. 4.5).



dev.pro
Інформація про користувача

Країна

Місто

Мови

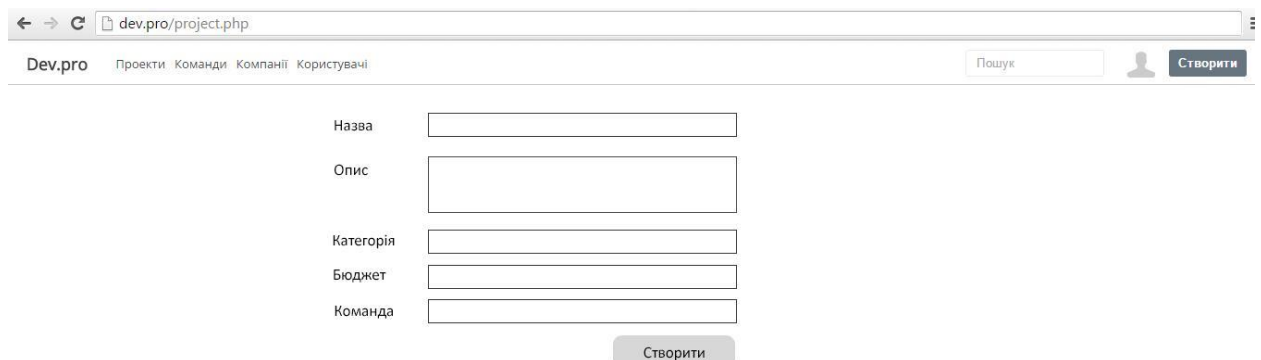
ВНЗ

Робота

Зберегти зміни

Рисунок 4.5 – Редагування даних

Для того щоб створити новий проект, потрібно в “шапці” сторінки натиснути на кнопку “Створити”. Після чого на направить на сторінку створення нового проекту (див. рис. 4.6)



dev.pro Проекти Команди Компанії Користувачі Пошук  Створити

Назва

Опис

Категорія

Бюджет

Команда

Створити

Рисунок 4.6 – Створення нового проекту

Після коректного заповнення форми користувача направить на сторінку створеного проекту (див. рис. 4.7).

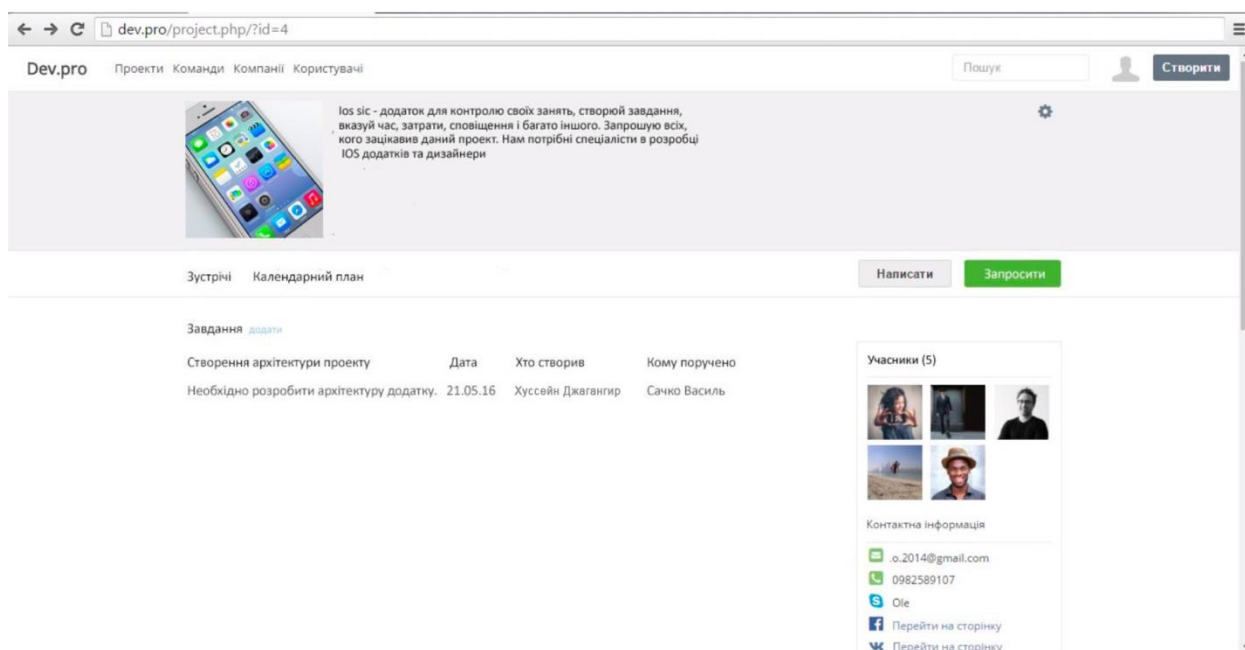


Рисунок 4.7 – Проект

В кінці ми отримаємо створений проект з яким можна працювати, запрошувати людей, створювати задачі, зустрічі.

Для перегляду всіх зареєстрованих користувачів необхідно перейти за посиланням <http://localhost/users.php> (див. рис. 4.8).

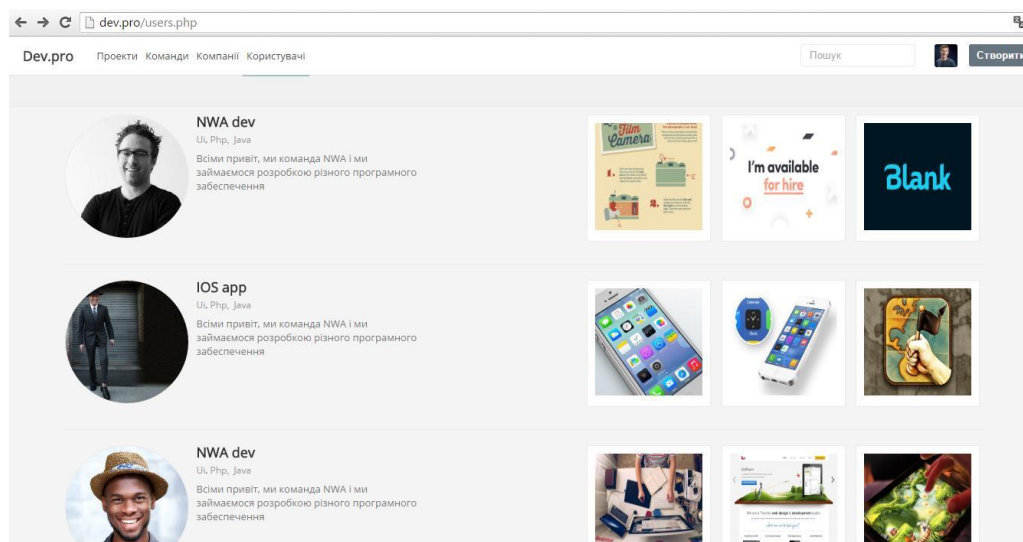


Рисунок 4.8 – Список зареєстрованих користувачів

Щоб переглянути всі створені користувачами проекти необхідно перейти

за посиланням <http://localhost/projects.php> (див. рис. 4.9).

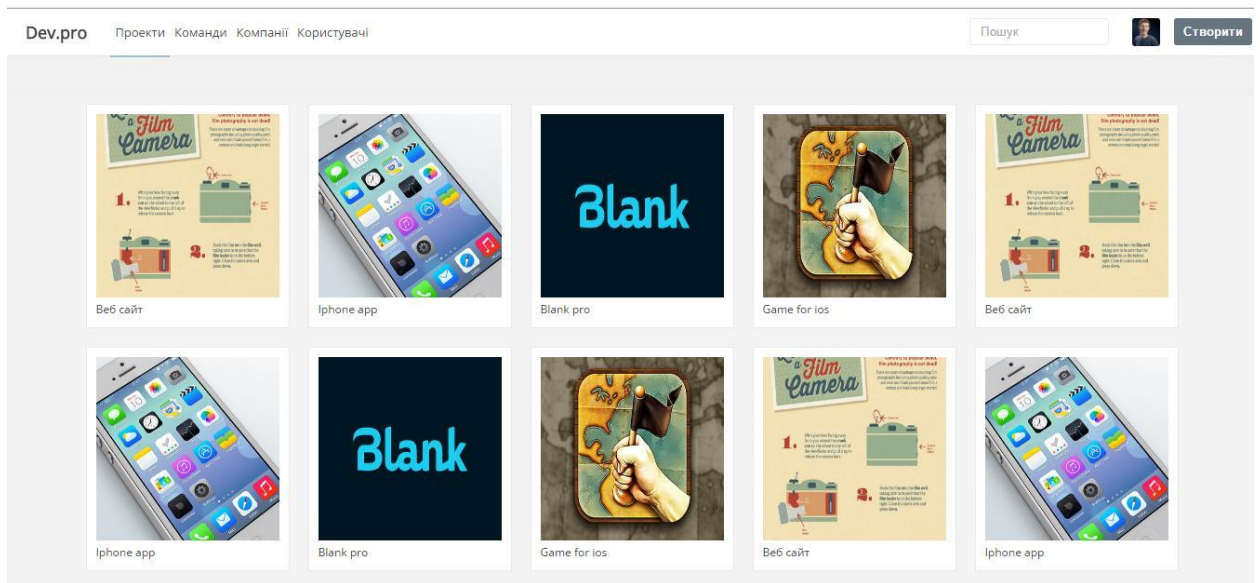


Рисунок 4.9 – Список проектів

Висновок до розділу.

Провівши тестування програмної системи можна сказати, що це є невід’ємною частиною при розробці програмних продуктів. Було проведено функціональне тестування, де ми перевірили всі функції, що виконує програм. Тестування сумісності – перевірили на сумісність з усіма браузерами. Тестування продуктивності – перевірили як швидко і ефективно працює наша програма. Тестування GUI – перевірили зручність користувацького інтерфейсу. Тестування безпеки – перевірили чи є вразливим наш програмний продукт.

В результаті всіх тестів було виявлені і ліквідовані дефекти в роботі програмної системи

5 СТАРТАП-ПРОЕКТ

Стартап проект характеризується завжди новими ідеями, яка може бути побудована на вже існуючих концептах, проте привнести на ринок щось принципово нове, що відокремить її від інших. Також, крім актуальності і новаторської складової, не менш значущим при створенні ідеї стартап проекту є аналіз конкуренції на ринку та комерційна складова. Для успішної реалізації стартапу необхідно обирати або нішу з високим потенціалом, або фінансово активну. Проте, беручи до уваги масштаби ринку програмного забезпечення знайти порожню нішу зараз майже неможливо, а створити систему яка за своїми якісними характеристиками буде домінувати на ринку й випередити конкурентів цілком реально. Вищенаведені аргументи приводять розробників програмного забезпечення до концепції, коли вони беруть за базу вже готову систему, вирішують її недоліки, та додають якісь додатковий функціонал, який і відокремлює їх на ринку. Саме ці переваги, які реалізують стартап-проекти, й вважають новаторською діяльністю розробника. Для створення принципово нових систем необхідні великі фінансові втручання, а зазвичай в стартап-проектах фінанси дуже обмежені.

10.1 Опис ідеї проекту

Техніко-економічними характеристики стартап-проекту наступні:

- кросплатформеність додатку;
- надання користувачу гнучкої системи опитування;
- віддалений доступ до системи;
- зручний інтерфейс для швидкої взаємодії;
- велика кількість параметрів сортування результатів;
- розбиття отриманих відгуків за регіоном ;
- альтернативні шляхи використання для різних типів замовників;
- аналітичні можливості системи;
- підтримка будь-якої кількості мов.

Таблиця 10.1 – Опис ідеї стартап-проекту

Зміст ідеї	Напрямки використання	Вигоди для користувача
	1. Автоматизований збір відгуків користувачів системи або компаній	Створення принципово нового програмного забезпечення для аналізу якості роботи компанії, з метою покращення взаємодії користувач-система, та прогресування на ринку
	2. Збереження отриманої інформації	Такий підхід щодо розвитку компанії має сенс як на довгостроковій, так і швидкій перспективі. Для цього необхідно додати функцію зберігання відгуків, для подальшого аналізу прогресу компанії
	3. Сортування отриманих відгуків по регіонам, віковим параметрами	Основною функцією бідь якого якого програмного забезпечення для збору й аналізу інформації містить у собі інструменти швидкої та зручної її обробки, сортуванню, для спрощення роботи аналітиків.
	4. Аналіз отриманої інформації за встановленими параметрами	Аналітична функція системи представляє собою найважливіший елемент системи, заради якого вона і створюється. Кінцевий користувач хоче отримати систему, не лише ту, яка буде займатись збором інформації, а й її обробкою.

Для комплексного аналізу перспектив стартап проекту, необхідно ретельно дослідити вже існуючі на ринку аналогічні системи, та виявити їх недоліки та переваги. Спираючись на переваги конкурентів, та додавання до системи власних якісних характеристик, дасть змогу вивести стартап-проект на ринок, та зайняти власну нишу. Для аналізу ринку розглянемо системи кількох напрямів: візуально привабливий інструмент, який фокусується надосвіді користувача GetFeedback, Get Satisfaction - це платформа спільноти, яка допомагає отримувати відгуки та ділитися ними з іншими клієнтами, платформа опитування клієнтів, яка легко інтегрується з маркетинговим додатком - Survicate

Система, що є метою магістерської дисертації, містить у собі переваги, що були виявлені в аналогів, та окрім цього матиме власну низку переваг, яка виведе її на якісно новий рівень на ринку. Система була розроблена кроссплатформеною на етапі проектування, задля збільшення кількості можливих клієнтів. Ретельне дослідження ринку встановило: сильні, слабкі й нейтральні сторони систем. Аналіз техніко-економічних характеристик конкурентів дав змогу сформулювати ті переваги за які користувачі цінують системи конкурентів й реалізувати їх у власній. Порівняння стартап-проекту з конкурентами зображено на таблиці 10.2.

Таблиця 10.2 – Визначення сильних, слабких та нейтральних характеристик ідеї проекту.

№	Техніко- економічні характеристики ідеї	Потенційні товари/концепції конкурентів				W (Слабка сторона)	N (Нейтр альна сторон а)	S (Сильн а сторон а)
		Стартап	t Feeb Back	Satis- faction	Survi- cate			

1.	Кросплатформеність додатку	Має	Має	Має	Має	-	-	+
2.	Далений доступ до системи	Має	Немає	Немає	Немає	+	-	-
3.	Декілька каналів зв'язку	Має	Має	Має	Має	-	-	+
4.	Ціна	Має	Немає	Має	Має	-	-	+
5.	Можливість порівняння статистики по регіонах	Має	Має	Немає	Немає	-	+	-
6.	Можливість автоматизації процесу опитування	Має	Має	Немає	Немає	-	-	+
7.	Процент відповіді на опросник	Має	Має	Немає	Немає	-	+	-
8.	Зручний й інтуїтивно зрозумілий інтерфейс	Має	Немає	Має	Має	-	+	-
9.	Простота інтегрування до інших CRM систем	Має	Немає	Немає	Немає	+	-	-

10.2 Технологічний аудит ідеї проекту

Реалізація будь якої системи не можлива без технологій, на яких вона побудована. Для реалізації проекту на всіх етапах необхідно обрати технології з використанням яких, розробник зможе реалізувати весь заявлений функціонал. З цієї причини, потрібно розглянути функціональні вимоги до стартап проекту та технології, що потрібні для її реалізації. У таблиці 10.3 розглянуті доступні технології та виявлено можливість здійснення проекту. .

Таблиця 10.3 – Технологічна здійсненність ідеї проекту.

№	Ідея проекту	Технології її реалізації	Наявність технологій	Доступність технологій
1.	Авторизація користувача	Власна реалізація	Наявна	Доступна
2.	Збереження даних користувача	MsSQL	Наявна	Доступна
3.	Реалізація бізнес-логіки та користувацького інтерфейсу	Веб-фреймворк Angular, мова програмування JavaScript Та надбудова над нею TypeScript з строгою типізацією	Наявна	Доступна
5.	Серверна частина та зв'язок до неї	ASP.NET Core 2, проткол HTTP	Наявна	Доступна
Обрана технологія реалізації ідеї проекту: є можливою.				

Згідно з результатами дослідженням, було встановлено, що система може бути реалізована, а кожна з необхідних технологій є доступною.

10.3 Аналіз ринкових можливостей запуску стартап-проекту

Перш ніж випускати систему на ринок, команда стартап-проекту має дослідити ринкові загрози, провести аналіз долей на ринку, визначити перспективну аудиторію, що є потенційними користувачами системи, проаналізувати потреби користувачів, можливості для масштабування та загальний фінансовий оборот ринку. Отже, був проведений аналіз потенційного ринку результати якого наведені нище у таблиці 10.4.

Після визначення вимог клієнтів необхідно провести огляд ринкового середовища, розглянути фактори, які сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають

Отримані дані свідчать, що ринок є привабливим для нових стартап проектів, а середня норма рентабельності, що напряду пов'язана з майбутнім доходом проекту, залежить лише від якості наданих вами функцій. Після завершення аналізу ринку, потрібно перейти до дослідження учасників цього ринку, або інше кажучи “клієнтської бази”. Характеристика наведена в таблиці 10.5.

Таблиця 10.4 – Повна характеристика ринку для впровадження стартап-проекту

№	Показники ринку (найменування)	Характеристика
1.	Кількість головних гравців, од	8 (розробники, аналітики, крупні компанії, маркетологи, менеджери по продажам, служба підтримки, фінансові аналітики, спеціалісти з бізнес-досліджень)
2.	Загальний обсяг продаж, грн/ум. од.	Вимірюється мільйонами долларів. Чітку сумму встановити неможливо, тому що усі великі компанії проводять маркетингові заходи, а потім збирають відгуки користувачів щодо їх якості
3.	Динаміка ринку (якісна ціна)	Динамічно зростає з поширенням інтернет-маркетингу, соціальних мереж та загальної автоматизації всіх процесів у світі
4.	Наявність обмежень для входу	Для запуску повноцінної системи, необхідно зареєструвати спочатку фізичну особу підприємця бажано третьої групи, щоб була змога співпрацювати з юридичними лицами
5.	Специфічні вимоги до стандартизації та сертифікації	Відповідно до закону економічної діяльності, стартап проект має мати офіційно зареєстровану одиницю. Та має сплачувати податки з отриманого доходу
6.	Середня норма рентабельності в галузі (або по ринку), %	Норма рентабельності залежить від якості опитування, та в загальному вигляді становить від 5 до 100 гривень за оброблену анкету.

Таблиця 10.5 – Характеристика потенційних клієнтів стартап-проекту

Потреба, що формує ринок	Цільова аудиторія (цільові сегменти ринку)	Відмінності у поведінці різних потенційних цільових груп клієнтів	Вимоги споживачів до товару
Потреба в якісному сучасному програмному забезпеченні, яке надає не лише функції розсилання форм з опросом, а й збором, зберіганням, сортуванням та аналізом отриманих від користувачів даних	Розробники, аналітики, крупні компанії, маркетингологи, менеджери по продажам, служба підтримки, фінансові аналітики, спеціалісти з бізнес-досліджень.	В загальному випадку користувачем може стати будь хто, хто зацікавлений у покращенні якості обслуговування. Програмне забезпечення може працювати з будь-якою базою користувачів, та аналізувати якість: освіти, товарів, послуг, медицини тощо	- висока якість програмного продукту - легкість у використанні - інтуїтивно зрозумілий інтерфейс - приємний дизайн - безкоштовне використання - альтернативні варіанти використання системи - робота з будь-яким списком клієнтів улюбій сфері

Після визначення вимог клієнтів необхідно провести огляд ринкового середовища, розглянути фактори, які сприяють ринковому впровадженню проекту, та факторів, що йому перешкоджають (табл. 10.6, табл. 10.7).

Таблица 10.6 – Факторы загрозов стартап-проекту

№	Фактор	Зміст загрози	Можлива реакція компанії
1.	Складності узгодження діяльності з законом	Для ведення економічної діяльності необхідна реєстрація фірми або часної особи підприємця	Основними клієнтами ринку є крупні компанії, що також є юридичними лицями, та вони замовляють послуги сторонніх компаній офіційно, а для цього стартап має бути офіційно оформлений
2.	Поява або модернізація нових технологій	Світ ІТ не стоїть на місці і усі технології, які в ньому існують постійно покращуються, поширюються, пришвидшуються.	Необхідність в постійній доробці системи, розширення її функціоналу, перехід на нові технології
3.	Поява нових гравців на ринку	Сучасний ринок програмного забезпечення постійно зростає, це пов'язано з динамічним зростанням впливу сучасних технологій	Необхідно постійно аналізувати недоліки власної системи, та намагатись її покращувати, щоб у інших не виникало бажання конкурувати з вами на ринку.
4.	Зміна інструментів доступу до стороннього API	Система має можливість інтегрування її до стороннього CRM, що виступає великою перевагою перед аналогами.	Адаптація до нового API, або взаємодія з іншим
5.	Зміна законодавств а у сфері азартних ігор в Україні	Як і будь якій іншій сфері діяльності, зміна законодавства вносить свої корективи в роботу підприємства	Знаходження нових шляхів використання та взаємодії

6.	Оновлення Пз	З часом, будь яке програмне забезпечення застаріває і його потрібно модернізувати.	Необхідність підтримки системи, або створення нової, що буде побудована на актуальних технологіях.
----	-----------------	--	--

Таблиця 10.7 – Фактори можливостей стартап-проекту

№	Фактор	Зміст можливості	Можлива реакція компанії
1.	Заохочення нових клієнтів з усього світу	Річ у тому, що ринок букмекерських контор є перенасиченим, проте компанії, що їми володіють інвестують велику кількість коштів в модернізування власних систем і заохочення нових клієнтів	Покращення фінансових можливостей системи, розширення функціоналу, збільшення кількості працівників, покращення технічного обладнання
2.	Зацікавленість у продукті нових користувачів	Збільшення кількості користувачів сприяє зростанню системи в цілому, тому що кожен користувач є потенційним істочником монетизації системи	Масштабування програмного продукту, знаходження спонсорських контрактів, збільшення конкуренції між користувачами
3.	Інтеграція програмного продукту у інші CRM системи	Інтеграція стартапу у великий проект з великими фінансовими спроможностями	Адаптація програмного продукту до можливостей іншої системи. Велика капіталізація системи, можливість масштабування, залучення нових клієнтів

Отже, нами були сформовані фактори загроз і можливостей. Більшість з яких пов'язані з змінами на ринку програмного забезпечення або зростанням конкуренції у сфері функціонування системи, як наслідок, система має

адаптуватись під зміни на ринку й розширюватись задля збереження позицій. Однак, незважаючи на рівень загроз якими може бути підвержена система, але вони не є критичними й система может легко адаптуватися під ці зміни. Позитивні можливості, які виникнуть при розвинені системи перекривають недоліки як і фінансової, так і інших точок зору. Звідси випливає, що проект має потенціал розвитку в ринковому середовищі й виникає необхідність аналізу існуючої конкуренції (табл. 10.8).

Таблиця 10.8 – Ступеневий аналіз конкуренції на ринку

Особливості конкурентного середовища	В чому проявляється дана характеристика	Вплив діяльності підприємства
Тип конкуренції: вільна конкуренція	Конкурентами є велика кількість програмних продуктів різного напрямку з великими капіталами	Залучення зацікавлених осіб у сфері з метою покращення маркетингових акцій та взаємодії з користувачами
За рівнем конкурентної боротьби: міжнародна	Ринок охоплює всі верстви користувачів з усього світу, бо система є багатомовною	Забезпечення можливості використання системою мови загального користування: англійською
За галузевою ознакою: внутрішньо-галузева	Всі конкуренти взаємодіють в галузі програмного забезпечення	Фактори ризику й переваги впливають в певній мірі на всіх гравців ринку
Конкуренція за видами товарів: товарно-видова	Система надає користувачам товар у вигляді інформації якою вони користуються	В загальному випадку користувачем може стати будь-хто, хто зацікавлений у покращенні якості обслуговування. Програмне забезпечення може працювати з будь-якою базою користувачів, та аналізувати якість: освіти, товарів, послуг, медицини тощо

За характером конкурентних переваг: не цінова	Товар має якісні переваги серед аналогів й надає загальний функціонал у одній системі, що спрощує зворотній зв'язок між міжнародними компаніями та їх клієнтами	Усі ресурси програми направлені у наданні користувачу повного функціоналу з зменшенням витрат часу
За інтенсивністю: не марочна	Товар не є принципово новим на ринку, проте має ряд відмінностей які його характеризують унікальним	Основною метою є розповсюдження інформації про новий продукт та його переваги для швидкого старту на ринку.

Ступеневий аналізу конкуренції на ринку показав, що програмні продукти, що вже існують на ринку й мають певну кількість користувачів містять в собі великий рід недоліків й при виході на ринок нового принципово нового інструменту аналізу з великими аналітичними можливостями, можна буде витіснити з ніші інші старі програми. Розглянемо більш детально аналіз умов конкуренції в галузі (табл. 10.9).

Таблиця 10.9 – Аналіз конкуренції в галузі за М. Портером

Складові аналізу	Прямі конкуренти в галузі	Потенційні конкуренти	Постачальники	Клієнти	Товари-замінники
	Крупні системи, що реалізують зворотній зв'язок між компаніями	Конкурентом є будь які системи, що збирають інформацію з користувачів	Платні програми для розробки програмного забезпечення, хостінгові	Усі користувачі з покупною спроможністю	Не беручи до уваги конкурентів, товарів-замінників на даний

	та користувачами	будь якої ніші.	системи, системи продажу доменів та інші		момент не існує
Висновки	На ринку існує дуже висока конкуренція, проте її учасники мають системи, що далекі від ідеалу. За цієї причини є можливість виходу на ринок	Велика конкуренція свідчить про те, що користувачів й спрос у ніші постійно зростає	Є певна низка залежностей від постачальників послуг, проте їх використання не має завадити постійній роботі системи	Головна умова до системи - це швидка дія, гнучкість, надання можливості багатомовного перекладу, інтуїтивно зрозумілий інтерфейс та аналітичні функції системи	Відсутність товарів-замінників свідчить про те, що система буде надійно протягом ближайшого часу

Згідно результатів аналізу за методом М. Портером можна зробити наступний висновок: на даний момент на ринку вже функціонує велика кількість програмних рішень, що забезпечують користувачів системами збору даних з користувачів, але ринок такого ПЗ постійно зростає з збільшенням користувацької бази.

Обґрунтування факторів конкурентоспроможності описано в таблиці 10.10.

Таблиця 10.10 – Обґрунтування факторів конкурентоспроможності

№	Фактор конкурентоспроможності	Обґрунтування факторів
1.	Зручність використання	Спроектований за всіма канонами сучасної розробки інтерфейс буде інтуїтивно зрозумілий користувачу
2.	Комплексний підхід	Поєднання всіх функцій, які надають аналогічні системи по окремі у одному сервісі, робить розробляємо програмне забезпечення більш зручним й інформативним
3.	Можливість розширення продукту	Архітектура системи побудована таким чином, що може бути розширеною новими модулями для збільшення функціоналу
4.	Аналітична складова	Велика кількість функцій системи, які стосуються обробки інформації, її сортуванню, створення вибірок.

Для отримання переваги перед конкурентами на ринку були обрані основні фактори на яких буде побудовано позиціювання розробляемого програмного продукту на ринку. Найважливішими серед них можна виділити легкість інтеграції і можливість розширення. Виходячи з визначених вище факторів конкурентоспроможності було проведено аналіз сильних та слабких сторін стартап-проекту, результати якого можна переглянути нище (табл. 10.11).

Таблиця 10.11 – Порівняльний аналіз сильних і слабких сторін стартап-проекту

Фактор конкурентоспроможності	Бали 1-20	Рейтинг товарів-конкурентів порівняно з проектом						
		-3	-2	-1	0	+1	+2	+3
Зручність використання	6			Feedback	Satisfaction	Survicate		
Комплексний підхід	10		Survicate		Feedback		Satisfaction	
Можливість розширення продукту	5		Satisfaction	Feedback			Survicate	
Аналітична складова	12			Satisfaction	Feedback		Survicate	

Для повноти аналізу, необхідно складання SWOT-аналізу на основі ринкових можливостей та загроз, а також сильних і слабких сторін (табл 10.12).

Таблиця 10.12 – SWOT-аналіз стартап-проекту

Сильні сторони: - широкий функціонал системи; - зручний і детальний користувацький інтерфейс; - безплатформенність системи; - можливість автоматизації процесу опитування;	Слабкі сторони: - можливість автоматизації процесу опитування; - один з найнижчих процентів відповіді на опитування; - відсутність інтегрування з іншими CRM та соціальними мережами.
--	--

сокий процент відповіді на опитування.	
Можливості: - можливість отримати власну клієнтуру завдяки новим перевагам системи; - знаходження інвесторів та масштабування компанії; - витіснення конкурентів з ринку; - постійна модернізація системи з появою нових технологій.	Загрози: - відсутність прагнення до покращення взаємодії з користувачем з боку потенційних клієнтів; - фінансовий тиск з сторони компаній вже існуючих на ринку ПЗ; - зміна доступу до CRM яке використовує системи; - проблема постійної технічної підтримки системи і її поліпшення, розширення.

По завершенню SWOT-аналізу, були отримано інформацію щодо створення альтернативної ринковій поведінки для виведення стартап-проекту на ринок (табл. 10.13). На основі SWOT-аналіз були досліджені якісні характеристики продуктів на ринку, та проведена оцінка і порівняння ключових сторін проекту у межах ринкового середовища.

Таблиця 10.13 – Альтернативи ринкового впровадження

Альтернатива (орієнтовний комплекс заходів) ринкової поведінки	Ймовірність отримання ресурсів	Строки реалізації
Загарбник	Значна	Максимум рік
Наступник	Суттєва	Максимум рік

Спираючись на весь аналіз, який був наведений у попередніх розділах були обрані дві стратегії поведінки продукту на ринку: загарбник і наступник. Наступник підходить оскільки основний функціоналу системи може бути

інтегрований у сторонню CRM, тобто спочатку може займає меншу нішу. Альтернативною стратегією є загарбник, який орієнтований на захоплення ринку від конкурентів, у нашому випадку вона побудована на тому, що система запозичує чужий функціонал та надає до нього додаткові аналітичні функції.

10.4 Розроблення ринкової стратегії проекту

Для успішного запуску проекту на ринок, потрібно розробити комплекс заходів для зростання, збільшення обсягів продажів, клієнтської бази, в загальному це називається: ринкова стратегія. Почати слід визначенні стратегію залучення користувачів до системи (табл. 10.14).

Таблиця 10.14 – Вибір цільових груп потенційних споживачів

Опис профілю цільової групи потенційних клієнтів	Готовність споживачів сприйняти продукт	Орієнтований попит в межах цільової групи (сегменту)	Інтенсивність конкуренції в сегменті	Простота входу в сегмент
Користувачі-споживач	Висока	Високий	Висока	Середня
Аналітики, маркетингологі	Середня	Середній	Низька	Середня
Які цільові групи обрано: всі користувачі споживач				

За результатом попереднього дослідження були визначені стратегія охоплення ринку. Наступний крок розробки ринкової стратегії полягає в визначенні базової стратегії розвитку (табл. 10.15).

Таблиця 10.15 – Визначення базової стратегії розвитку

Обрана альтернатива розвитку проекту	Стратегія охоплення ринку	Ключові конкурентоспроможні позиції відповідно до обраної альтернативи	Базова стратегія розвитку
Наступник	Концепція полягає у поступовому розширенні функціоналу системи та збільшення обсягів покриття ринку	Надання користувачам потужного аналітичного інструмента для дослідження якості товарів, послуг тощо	Стратегія спеціалізації

Отже, оберемо стратегію конкурентної поведінки (табл. 10.16).

Таблиця 10.16 – Визначення базової стратегії конкурентної поведінки

Чи є проект «першопрохідцем» на ринку?	Чи буде компанія шукати нових споживачів, або забирати існуючих у конкурентів?	Чи буде компанія копіювати основні характеристики товару конкурента, і які?	Стратегія конкурентної поведінки
Система є модернізацією вже існуючого аналогу	В першу чергу, система може стати розширенням вже існуючих систем, наприклад CRM	Дублювання таких суттєвих характеристик в проектуванні і реалізації системи як: - зручний інтерфейс; - кросс-платформеність додатка;	Стратегія виклику лідера.

		- підтримка декількох мов; - функціональна насиченість.	
--	--	---	--

ВИСНОВОК

Магістерська робота направлена на аналіз та створення веб-сервусу для керування стартапами.

Основний інтерес був націлен на створенні цікавих проектів, пошуку робітників та пошуку інвестора для свого проекту.

Були отримані наступні результати:

1. Аналіз який був проведений системи керування стартапами, де ми розглянули актуальність програми. Також були розглянуті аналоги, які є найбільш схожими на дану систему, проведений їхній детальний аналіз і порівняння.

2. Розробленні класи, архітектрута та структура бази даних додатку.

3. Вибрана мова програмування для створення програмного продукту. Створили інтерфейс користувача, запрограмували методи та класи, які виконували всю логіку сервісу та створили БД для керування даними користувачів.

4. Проведене тестування веб-сервісу на навантаження, безпеку, функціональність, GUI.

В результаті виконання дипломної роботи , були вдосконалені знання у проектуванні, тестуванні та розробці програмних систем.

Під час розробки було застосовані на практиці знання в:

- Мові програмування PHP;
- мові програмуванні Javascript;
- СУБД MySQL;
- локальному сервері Open Server;

В кінцевому результаті була отримана веб-орієнтована система управління стартапами, яка дасть змогу усім бажаючим створити власний стартап, розвивати його, залучати в його нових людей, фінансування, ділитися власними розробками, ідеями та надихати інших користувачів на створення цікавих та корисних сервісів.

ПЕРЕЛІК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Поздняев А.С., Власов А.И. Развитие информационно-телекоммуникационного сектора экономики при гармонизации мировой финансовой системы им. Н.Э. Баумана. серия: Приборостроение. 2010. No 1. - С. 121-125.
2. Корсаков С.Н. Начертание нового способа исследования при помощи машин, сравнивающих идеи / Пер. с франц. под ред. А.С. Михайлова. – М.: МИФИ, 2009, 44 с.
3. Novak J. D., Canas A.J. The Theory Underlying Concept Maps and How to Construct Them. //Technical Report IHMC CmapTools 2006-01 Rev 01-2008, Florida Institute for Human and Machine Cognition, 2008.
4. Буч Г., Рамбо Дж., Джекобсон А. Язык UML. Руководство пользователя.-М.:ДМК ПРЕСС; СПб.:Питер,2004.-429 с.
5. Герман О.В. Введение в теорию экспертных систем и обработку знаний. Учебное посо-бие. Мн.: ДизайнПро, 1995. - 456 с.
6. Сотник С. Л. Конспект лекций по курсу «Основы проектирования систем искусственного интеллекта», 1997-1998.
7. 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.
8. 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.
9. Кормен Т., Лейзерсон И. Ч., Ривест Р. Л., ШтайнК. Алгоритмы: построение и анализ = INTRODUCTION TO ALGORITHMS — 2-е изд. — М.: «Вильямс», 2006. — С. 1296. — ISBN 0-07-013151-1.
- 10.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

11.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.

12.Титтел Э., НоублДж. HTML, XHTML и CSS для чайников, 7-е издание = HTML, XHTML & CSS For Dummies, 7th Edition— М.: «Диалектика», 2011. — 400 с. — ISBN 978-5-8459-1752-2.

13.Zakas N., McPeak J., Fawcett J. Professional Ajax — 2nd ed. — Wrox, 2007. — 624 p. — (Programmer to Programmer). — ISBN 0470109491.

14.Lindley C. jQuery Cookbook. Solutions & Examples for jQuery Developers. O'Reilly Media, 2009. – 478 с.

15.Джон К. Вандик , Мэт Вестгейт. Pro Drupal 7 Development: Third Edition / Todd Tomlinson . John K. VanDyk - Apress, 2010 .

16.Стивен Хольцнер . PHP в примерах. / Стивен Хольцнер . М.: 000 «Бином-Пресс», 2007 г. Пер. с англ. 352 с

17.Ларри Ульман. Ульман Л. Основы программирования на PHP:/Ларри Ульман. Пер. с англ. -М.: ДМК Пресс, 2001. -288 с.: ил. (Самоучитель).

18.Александр Мазуркевич. МВ PHP: настольная книга программиста /Александр Мазуркевич, Дмитрий Еловой. — Мн.: Новое знание, 2003. — 480 с.: ил

19.Томсон Лаура. Разработка Web-приложений на PHP и MySQL: Пер. с англ. /Лаура Томсон, Люк Вел.

20.JavaScript. Подробное руководство. Дэвид Флэнаган.