

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»

Факультет комп'ютерних наук і технологій

(повне найменування інституту, назва факультету)

Кафедра програмних засобів

(повне найменування кафедри)

Пояснювальна записка

до дипломного проекту (роботи)

магістр

(ступінь вищої освіти)

на тему ДОСЛІДЖЕННЯ ТА РОЗРОБКА ЗАСТОСУНКУ ДЛЯ
АВТОМАТИЧНОГО ФОРМУВАННЯ ТУРНІРНИХ КОМАНД
ЗА РЕЙТИНГОМ ГРАВЦІВ
RESEARCH AND DEVELOPMENT OF AN APPLICATION
FOR AUTOMATIC FORMATION OF TOURNAMENT TEAMS
BASED ON PLAYER RATINGS

Виконав(ла): студент(ка) 2 курсу, групи КНТ-122м

Спеціальності 121 Інженерія програмного

(код і найменування спеціальності)

забезпечення

Освітня програма (спеціалізація)

Інженерія програмного забезпечення

КОТОВ О.О.

(ПРИЗВИЩЕ та ініціали)

Керівник ФЕДОРОНЧАК Т.В.

(ПРИЗВИЩЕ та ініціали)

Рецензент ШИТІКОВА О.В.

(ПРИЗВИЩЕ та ініціали)

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
Національний університет «Запорізька політехніка»
 (повне найменування закладу вищої освіти)

Факультет ФКНТ
 Кафедра програмних засобів
 Ступінь вищої освіти магістр
 Спеціальність 121 Інженерія програмного забезпечення
 (код і найменування)
 Освітня програма (спеціалізація) Інженерія програмного забезпечення
 (назва освітньої програми (спеціалізації))

ЗАТВЕРДЖУЮ

Завідувач кафедри ПЗ, д.т.н, проф.
Сергій СУББОТІН
 “ ” 2023 року

З А В Д А Н Н Я

НА ДИПЛОМНИЙ ПРОЄКТ (РОБОТУ) СТУДЕНТА(КИ)

КОТОВА Олександра Олександровича
 (ПРІЗВИЩЕ, ім'я, по батькові)

1. Тема проєкту (роботи) Дослідження та розробка за стосунку для автоматичного формування турнірних команд за рейтингом гравців
Research and Development of an Application for Automatic Formation of Tournament Teams Based on Player Ratings

керівник проєкту (роботи) к.т.н., доцент, ФЕДОРОНЧАК Тетяна Василівна,
 (науковий ступінь, вчене звання, ПРІЗВИЩЕ, ім'я, по батькові)

затверджені наказом закладу вищої освіти від “14” листопада 2023 року
 № 437

2. Строк подання студентом проєкту (роботи) 29 листопада 2023 року

3. Вихідні дані до проєкту (роботи) рекомендована література, технічне завдання

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити) 1. Аналіз проблеми та постановка завдань дослідження. 2. Матеріали і методи. 3. Розробка архітектури програми. 4. Основні рішення щодо реалізації компонентів системи. 5. Експлуатація, тестування та експериментальне дослідження програми. 6. Додатки

5. Перелік графічного матеріалу (з точним зазначенням обов'язкових креслень)
Слайди презентації

6. Консультанти розділів проєкту (роботи)

Розділ	ПРІЗВИЩЕ, ініціали та посада консультанта	Підпис, дата	
		завдання видав	прийняв виконане завдання
1-4 Основна частина	ФЕДОРОНЧАК Т.В., доцент		
Нормоконтролер	ДЕЙНЕГА Л.Ю., ст. викладач		

7. Дата видачі завдання « 14 » листопада 2023 року.

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів дипломного проєкту (роботи)	Строк виконання етапів проєкту (роботи)	Примітка
1	Постановка завдання роботи.	1 тиждень	Завдання, ТЗ
2	Аналіз предметної області.	1 тиждень	Розділ 1
3	Розробка та удосконалення методів, моделей й алгоритмів вирішення задачі.	2 тиждень	Розділ 2
4	Розробка архітектури програми.	2 тиждень	Розділ 3
5	Розробка програми.	3-5 тижні	Розділи 3,4
6	Тестування та експериментальне дослідження програмного забезпечення.	6 тиждень	Розділ 5
7	Оформлення пояснювальної записки та документів до неї.	7-8 тижні	Додатки
8	Нормоконтроль та рецензування.	9 тиждень	
9	Захист роботи.	10 тиждень	

Студент(ка)

(підпис) Олександр КОТОВ
(Ім'я ПРІЗВИЩЕ)

Керівник проєкту (роботи)

(підпис) Тетяна ФЕДОРОНЧАК
(Ім'я ПРІЗВИЩЕ)

РЕФЕРАТ

Пояснювальна записка до дипломної кваліфікаційної роботи магістра:
96 с., 30 рис., 3 табл., 4 дод., 21 джерело.

КІБЕРСПОРТ, ТУРНІР, МІКС КАП, ГРАВЕЦЬ, ВЕБЗАСТОСУНОК,
ГЕЙМІНГ, РОЗРОБКА, ПРОГРАМУВАННЯ.

Об'єкт дослідження – процес програмної реалізації застосунку для автоматичного формування турнірних команд на основі рейтингу гравців.

Предмет дослідження – методи реалізації застосунку для автоматичного формування турнірних команд на основі рейтингу гравців.

Мета роботи – програмна реалізація застосунку для автоматичного формування турнірних команд за рейтингом гравців.

Матеріали, методи та технічні засоби: об'єктно-орієнтоване програмування, мова програмування PHP, система керування вмістом WordPress, система керування базами даних MySQL.

Результати. Створено програмний застосунок, який прискорює формування турнірних команд за рівнем гри.

Висновки. Розроблено систему для автоматичного формування турнірних команд за рейтингом гравців.

Галузь використання – користувачі мережі Інтернет.

ABSTRACT

Explanatory note to the diploma qualifying work of the bachelor: 96 pages, 30 figures, 3 tables, 4 appendixes, 21 sources.

ESPORTS, TOURNAMENT, MIX CUP, PLAYER, WEB APPLICATION, GAMING, DEVELOPMENT, PROGRAMMING.

The object of research is the process of software implementation of the application for the automatic formation of tournament teams based on the player rating.

The subject of research is the implementation methods of the application for the automatic formation of tournament teams based on the rating of the players.

The purpose of the work is to develop an application for automatic formation of tournament teams based on the rating of players.

Materials, methods, and technical tools: Object-Oriented Programming, PHP programming language, WordPress content management system, MySQL database management system.

Results. A software application has been created that accelerates the formation of tournament teams according to the level of play.

Conclusions. A system has been developed for the automatic formation of tournament teams based on player ratings.

Field of usage: users of the Internet network.

ЗМІСТ

	С.
Перелік скорочень та умовних познач.....	8
Вступ.....	9
1 Аналіз проблеми та постановка завдань дослідження	10
1.1 Контекст та актуальність.....	10
1.2 Мета та об'єкт дослідження.....	11
1.3 Огляд існуючих рішень	11
1.3.1 Платформа «Epulze»	12
1.3.2 Платформа «Faceit».....	13
1.3.3 Сервіс «Challonge»	14
1.4 Порівняння існуючих аналогів	15
1.5 Технічний Аспект та інновації	17
1.6 Очікувані результати	18
1.7 Висновки за розділом 1	19
2 Матеріали і методи.....	20
2.1 Завдання та цілі дослідження	20
2.2 Теоретичні засади алгоритмів	22
2.3 Методи оцінки ефективності	23
2.4 Ключові Показники Продуктивності.....	25
3 Розробка архітектури програми	27
3.1 Вибір інструментарію розробки	27
3.1.1 Вибір мови програмування	27
3.1.2 Система керування контентом.....	31
3.1.3 Обґрунтування вибору CMS для розробки вебресурсу	32
3.2 Структурна схема розробки	33
3.3 Висновки за розділом 3	35
4 Основні рішення щодо реалізації компонентів системи	36
4.1 Проектування моделей системи	36
4.2 Моделювання системи з точки зору прецедентів.....	40

4.3	Фізичне проектування	41
4.4	Проектування інтерфейсу	45
4.5	Висновки за розділом 4	47
5	Експлуатація, тестування та експериментальне дослідження програми	49
5.1	Опис застосунку	49
5.2	Умови виконання програми	49
5.3	Інструкція по експлуатації програми	50
5.3.1	Характеристики програми	50
5.3.2	Звернення до програми	51
5.3.3	Вхідні й вихідні дані	51
5.3.4	Виконання програми	51
5.3.5	Розширення функціональності вебзастосунку за допомогою модулів	52
5.4	Методика та результати тестування	53
5.4.1	Аналіз реалізованих функціональних вимог	53
5.4.2	Ергономічність вебзастосунку	53
5.4.3	Тестування безпеки вебзастосунку	55
5.4.4	Тестування стабільності	55
	Висновки	57
	Перелік джерел посилання	58
	Додаток А Технічне завдання	61
	Додаток Б Опис програми	66
	Додаток В Текст програми	71
	Додаток Д Слайди презентації	89

ПЕРЕЛІК СКОРОЧЕНЬ ТА УМОВНИХ ПОЗНАК

Кіберспорт – комп’ютерний спорт;

Мікс кап (mіx cup) – турнір зі змішаними командами;

Мікс (mіx) – мішанка, змішувати;

Кап (cup) – кубок, турнір;

CMS– системи управління контентом;

URL – uniform resource locator;

UML – unified modeling language;

БД – база даних;

ОЗУ – оперативна пам’ять;

ОС – операційна система;

ПЗ – програмний засіб;

ПК – персональний комп’ютер;

IBM – international business machines.

ВСТУП

Сьогодні кіберспорт є однією з найпопулярніших сфер розваг у світі. Одним із важливих аспектів кіберспорту є проведення турнірів. Турніри дають можливість гравцям змагатися один з одним за призові гроші та славу. Існують два основних формати проведення турнірів: командний та індивідуальний. Формат індивідуальний, також відомий як "мікс кап", є популярним вибором для кіберспортивних турнірів, оскільки він дозволяє гравцям будь-якого рівня майстерності брати участь у змаганнях. Однак, у цьому форматі виникає проблема балансу сил між командами.

Звідси випливає мета даної роботи – спроектувати і розробити вебзастосунок для автоматичного формування турнірних команд за рейтингом гравців.

Об'єктом дослідження даного розрахункового завдання є проектування і розробка вебзастосунку «Cup Mixer», що дозволить гравцям швидко реєструватись на турнір, який автоматично сформує команду, що економить багато часу та дозволяє приймати участь в турнірах гравцям без команди.

Також під час виконання роботи розглянуто засоби для проектування та розробки програмного забезпечення.

Актуальність розробки такого застосунку пояснюється тим, що зараз йде активне зростання попиту турнірів такого формату, але на просторах інтернету складно знайти зручну платформу для проведення турнірів такого формату.

Існують багато аналогічних платформ, але кожна має свої переваги та недоліки, а також жодна не може запропонувати проведення турнірів для гравців без команди. Саме тому було вирішено розробити новий застосунок.

1 АНАЛІЗ ПРОБЛЕМИ ТА ПОСТАНОВКА ЗАВДАНЬ ДОСЛІДЖЕННЯ

Перший розділ роботи присвячено висвітленню питань організації та проведення кіберспортивних турнірів у форматі “мікс кап”. В розділі буде розглянуто основні питання контексту та актуальності проблеми, визначені мета та об’єкт дослідження, досліджено існуючі аналогічні рішення, технічний аспект та інновації, а також розглянуті методологія та підходи для вирішення поставленої мети.

1.1 Контекст та актуальність

Дивлячись на швидкі темпи розвитку кіберспортивних турнірів, особливо в таких іграх як Dota 2 та Counter-Strike, існує актуальна проблема для більшості гравців – вони, не маючи своєї сформованої команди, часто відчують важкість участі в турнірах. Зазвичай, причиною цього є низький рівень гри. Існуючі турнірні платформи в основному орієнтовані на готові колективи, залишаючи гравців, які не мають стабільних команд або можливості їх створення, в стані відокремленості [1]-[3].

У цьому контексті стає доречним створення спеціалізованої платформи, яка надає можливість гравцям індивідуально реєструватись для участі в турнірах, а потім автоматично формує команди на основі їхнього рівня гри. Такий підхід розв'язує проблему відсутності стабільних команд у гравців та відкриває нові можливості для участі в турнірах навіть тим, хто раніше відчував себе відчуженим від турнірної атмосфери [1]-[3].

Отже, створення моєї платформи, яка базується на ідеї "мікс капів" для турнірів з Dota 2 або Counter-Strike, має значущий потенціал зробити кіберспорт більш доступним та відкритим для широкого кола гравців, незалежно від їхнього досвіду та командного статусу [1]-[3].

1.2 Мета та об'єкт дослідження

Метою цього дослідження є розробка, впровадження та оцінка ефективності інноваційної турнірної платформи для гравців Dota 2 та Counter-Strike, яка дозволяє індивідуальним учасникам реєструватися та автоматично формувати балансовані "мікс капи" на основі їхнього рівня гри.

Об'єктом дослідження є процес та алгоритмічні аспекти автоматичного формування турнірних команд за рейтингом гравців.

Завдання дослідження включають наступне:

- розробити алгоритм автоматичного формування "мікс капів" на основі рівня гри та інших критеріїв;
- реалізувати та впровадити турнірну платформу, яка дозволяє гравцям індивідуально реєструватись та брати участь в "мікс кап" турнірах;
- оцінити ефективність та прийняття платформи серед цільової аудиторії;
- забезпечити ретельний аналіз реакції користувачів та здобуття відгуків для подальшого вдосконалення системи.

Це дослідження спрямоване на заповнення прогалини в існуючих турнірних платформах та вдосконалення досвіду гри для гравців Dota 2, які дотримуються індивідуального підходу та необхідності балансу в утворених командах [1]-[3].

1.3 Огляд існуючих рішень

На сучасному ринку існує низка турнірних платформ, спрямованих на кіберспорт у Dota 2. Багато з них орієнтовані на створені вже команди та колективи, пропонуючи турніри, де гравці забезпечують власний склад. Однак, для індивідуальних учасників, або тих, хто не має стабільних команд, такі платформи можуть виявитися менш доступними та привабливими.

Для розробки конкурентоспроможної платформи слід на початку проаналізувати існуючі аналоги та виявити їх переваги або недоліки.

1.3.1 Платформа «Epulze»

Epulze (рис. 1.1). Epulze – це онлайн-платформа для кіберспортивних змагань. Основна ідея проекту полягає у створенні зручного середовища для організації турнірів в форматі 1x1 [4].

Основні характеристики та аспекти Epulze можуть включати:

- турніри – Epulze надає можливість організації та участі в онлайн-турнірах різних кіберспортивних ігор. Гравці можуть приєднуватися до турнірів у форматі 1x1 [4];
- різноманіття дисциплін – платформа підтримує різні кіберспортивні дисципліни, такі як Dota 2, Counter-Strike, League of Legends, і багато інших;
- призові фонди та винагороди – учасники турнірів мають можливість вигравати призові фонди та отримувати інші винагороди, що можуть включати предмети, грошові винагороди, або інші призи [4];
- гнучкість реєстрації – гравці можуть зареєструватися на турніри як індивідуальні учасники або як команди в залежності від конкретних вимог турніру [4];
- зручний інтерфейс – платформа пропонує зручний інтерфейс для реєстрації, управління турнірами та спілкування з іншими гравцями [4].

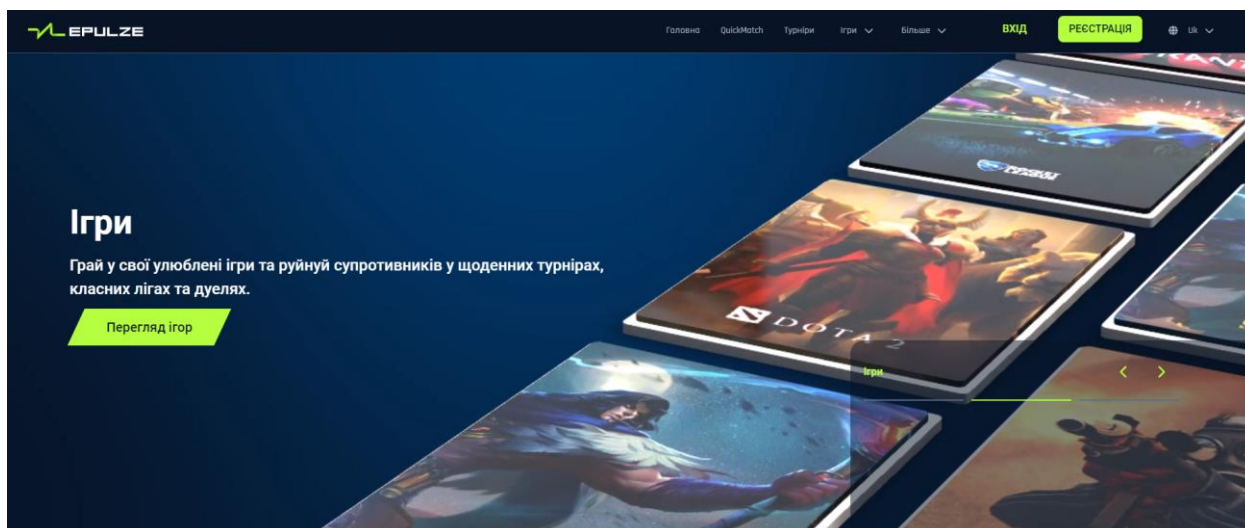


Рисунок 1.1 – Головна сторінка платформи Epulze [4]

1.3.2 Платформа «Faceit»

Faceit (рис. 1.2). FACEIT є онлайн-платформою для кіберспортивних турнірів і матчмейкінгу, яка підтримує гравців у різних кіберспортивних іграх [5].

Основні аспекти та функції FACEIT включають:

- матчмейкінг та рейтингові системи – гравці можуть приєднатися до матчів або турнірів та грати проти інших гравців на аналогічному рівні. Рейтингова система визначає рівень гравців і враховує їхні досягнення [5];
- турніри та ліги – FACEIT проводить регулярні турніри та ліги для різних ігор. Гравці можуть приєднатися до цих турнірів для випробування своїх сил і конкурування за призи [5];
- партнерства з розробниками ігор – FACEIT має партнерства з розробниками ігор, що дозволяє їм інтегрувати свою технологію в платформу для покращення інтерфейсу та геймплею [5];
- спільнота та спілкування – FACEIT надає можливість гравцям взаємодіяти, обговорювати гри та спілкуватися через вбудований чат та форуми [5];

- профілі та статистика гравців – кожен гравець має власний профіль, на якому відображаються його досягнення, статистика гри та інші важливі дані;
- античіти – в системі FACEIT є механізм античиту, який допомагає виявляти та запобігати використанню читів та інших заборонених програм [5].

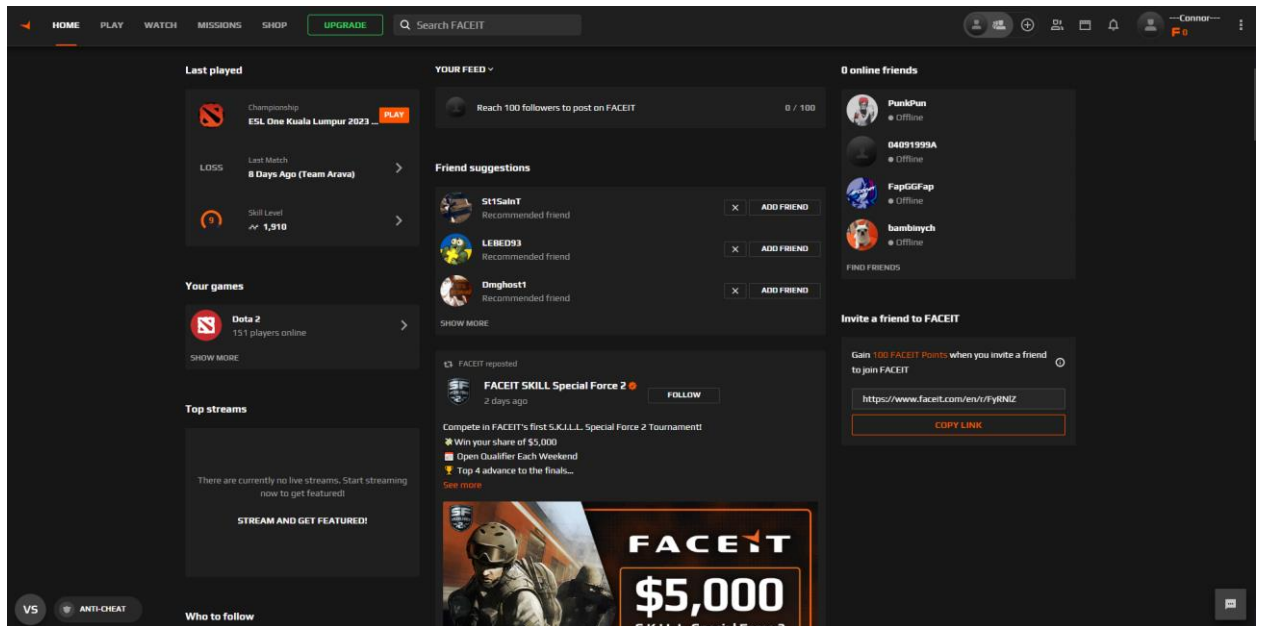


Рисунок 1.2 – Приклад головної сторінки Faceit [5]

1.3.3 Сервіс «Challonge»

Challonge (рис. 1.3). Challonge є платформою для створення турнірів та ліг у світі електронних спортивних ігор та інших видів змагань. Нижче подано деталізацію основних функцій цього проекту:

- створення турнірів та ліг – Challonge надає користувачам можливість легко створювати різні типи турнірів, включаючи одноразові турніри, ліги зі збором балів, турніри з подвійним видаленням, тощо [6];
- параметри турніру – адміністратори турніру можуть налаштовувати різні параметри, такі як формат турніру, правила гри, час та дата проведення, карти чи режими гри, інші обмеження та налаштування [6];

- реєстрація та участь – гравці можуть зареєструватися на турнір через платформу та приймати участь в змаганнях відповідно до встановлених правил та обмежень [6];
- статистика та результати – платформа веде статистику проведених турнірів, включаючи результати кожного матчу, забиті та пропущені очки, переможців тощо [6];
- гнучкість редагування – адміністратори турніру можуть редагувати параметри турніру або результати матчів в реальному часі, щоб врахувати будь-які зміни чи внесені помилки [6].

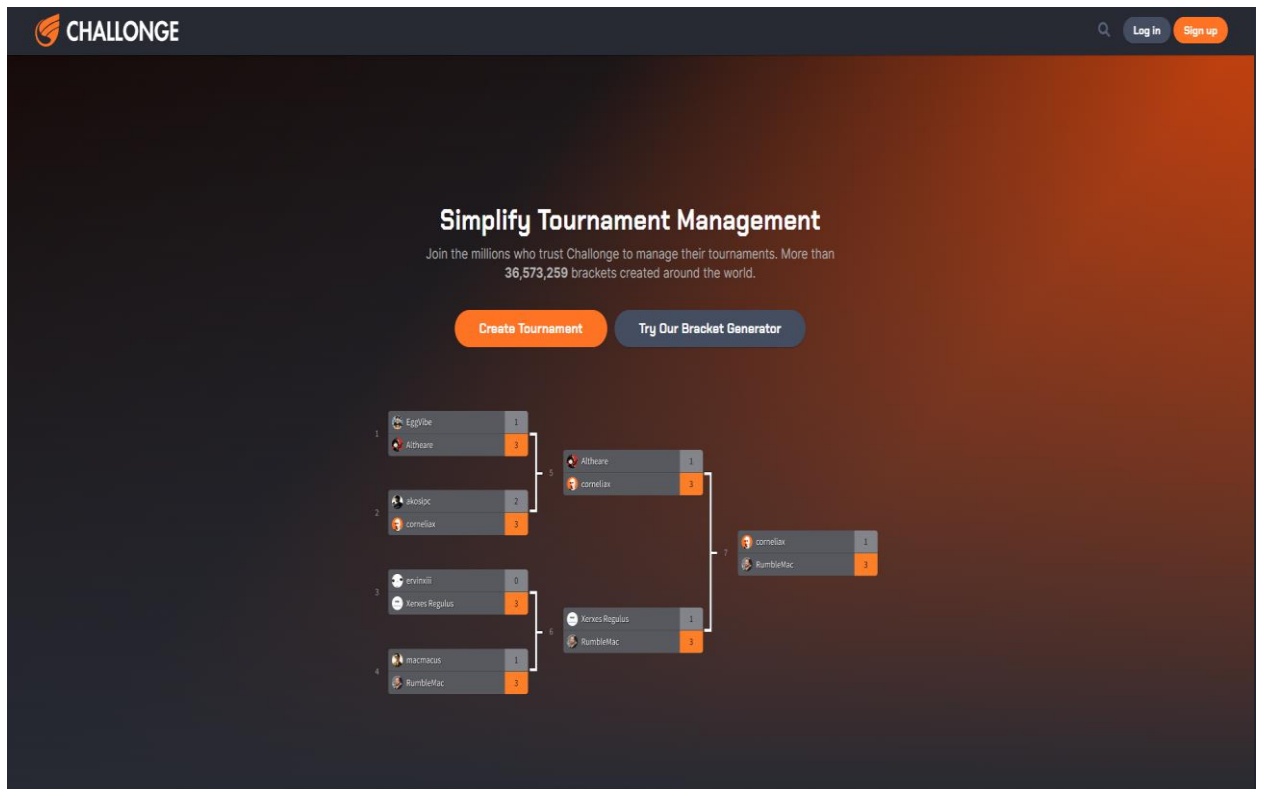


Рисунок 1.3 – Головна сторінка сервісу «Challonge» [6]

1.4 Порівняння існуючих аналогів

Проведемо порівняння за певними критеріями, хід якого представимо у вигляді таблиці (табл. 1.1). Серед критеріїв виділимо наступні:

- формат турнірів – наявність на платформі можливості проведення турнірів у форматі 5x5, 1x1, “мікс капу”;
- баланс команд – яким чином відбувається балансування команд;
- статистика та результати – які можливості надані для ведення статистики та відстеження результатів турнірів;
- реєстрація та участь – як просто гравці можуть реєструватися та приймати участь у турнірах.

З результатів порівняння можна зробити висновок, що більшість існуючих платформ має достатньо широкий функціонал, для проведення турнірів як в форматі ліг, в форматі матчів 5x5, так і в форматі 1x1. Враховуючи, що данні платформи є найпопулярнішими серед користувачів у сфері кіберспорту, але кожна з них має свої переваги та недоліки.

На платформах Faceit та Challonge відсутня можливість проведення “мікс капів”, але Faceit в свою чергу пропонує різноманітні формати турнірів, включаючи одноразові, ліги та змагання зі збором балів. На платформі Epulze є можливість участі лише у в форматі “QuickMatch” (формат має на увазі проведення одного матчу 5x5 для 10 індивідуально зареєстрованих користувачів з випадковим змішуванням, який не враховує рейтинг гравців), або в форматі 1x1.

Для балансування команд на платформі Faceit використовується власна рейтингова система, яка дозволяє балансувати між собою тільки готові колективи. На платформі Challonge баланс здійснюється за рахунок ручного налаштування адміністратором турніру.

Faceit має розширений обсяг статистики, який охоплює різні ігри та показники гравців. Challonge та Epulze надає базову інформацію, але на платформі Challonge адміністратор турніру має певний контроль над визначенням статистики.

Всі турнірні платформи мають інтуїтивно зручний інтерфейс підтримують зручну реєстрацію та участь в змаганнях. Платформи Faceit та Epulze пропонує управління турнірами та спілкування з іншими гравцями.

Таблиця 1.1 – Порівняння існуючих аналогів

Критерій порівняння	Epulze	Faceit	Challonge	Застосунок, що розробляється
Формат турнірів	Підтримує тільки формат 1x1	Підтримує турніри в форматі 5x5	Підтримує турніри в форматі 5x5	Підтримує турніри в форматі 5x5 в форматі “мікс кап”
Баланс команд	Випадковий	Присутній тільки для готових команд	Присутній тільки для готових команд	Балансує команди за середнім рівнем гри
Статистика та результати	Базова інформація	Розширений обсяг статистики	Базова інформація (можливість редагування адміністратором)	Відсутня
Реєстрація та участь	Проста реєстрація та участь в турнірах	Проста реєстрація та участь в турнірах (можливість управління турнірами)	Проста реєстрація та участь в турнірах (можливість управління турнірами)	Проста реєстрація та участь в турнірах

Тому можна зробити висновок, що враховуючи велику кількість турнірних платформ з різним функціоналом та можливостями, кожна з них має свої переваги та недоліки. Тому актуальною задачею залишається розробка нового застосунку, який виправить відсутність турнірів в форматі “мікс кап” та дати можливість одинарним гравцям покинутись в змагальну атмосферу.

1.5 Технічний Аспект та інновації

На підставі концепції розробки турнірної платформи для мікс кап турнірів на базі WordPress, важливим є ретельний огляд технічних аспектів та

інновацій, які впроваджуються для забезпечення ефективності та унікальності застосування.

Засновуючись на WordPress, платформа використовує потужний фреймворк для забезпечення легкості управління контентом та швидкої розробки.

Підхід включає в себе використання API WordPress та додаткових плагінів для створення зручної системи реєстрації гравців на турніри. Це забезпечує ефективне управління списком учасників та прискорює процес підготовки до турнірів. Щоб мінімізувати час організації турнірів, інтегровано додатковий функціонал в адміністративну панель, який дозволяє швидко та ефективно редагувати параметри турнірів. Використання асинхронних технік та оптимізації запитів спрощує процес планування та проведення турнірів.

На платформі впроваджена система зміни мови та локалізації, яка дозволяє гравцям з різних країн користуватися платформою на рідній мові. Використання динамічної зміни контенту робить інтерфейс максимально доступним та зручним.

Головною інновацією є впровадження адаптивного алгоритму для формування команд. Цей алгоритм враховує рівень гравців за їх індивідуальним рейтингом, забезпечуючи баланс та конкурентоспроможність команд на кожному етапі турніру.

1.6 Очікувані результати

Розробка турнірної платформи з інтегрованим функціоналом для швидкого формування команд у мікс кап турнірах визначається як амбіційна ініціатива. Очікується, що такий підхід до технічного аспекту та інновацій принесе ряд значущих результатів.

Забезпечення гравців легкістю реєстрації, швидким доступом до турнірів та зручним формуванням команд підвищить рівень задоволення гравців та приверне нових користувачів.

Мінімізація часу організації турнірів через панель адміністратора дозволить адміністраторам швидше реагувати на зміни, додавати та редагувати турніри, що веде до покращення ефективності організаційних процесів.

Впровадження алгоритму для формування команд гарантує більший баланс та конкурентоспроможність, що робить турніри більш привабливими для широкого кола гравців.

1.7 Висновки за розділом 1

В першому розділі відбулося ознайомлення із проблемами сучасних турнірних платформ. Було детально розглянуто проблему відсутності турнірів в форматі “мікс кап” для кіберспортивних ігор, таких як Dota 2 або Counter-Strike. Це обмежує можливості для індивідуальних гравців, які не мають власних команд. На основі виявлених недоліків було створено відповідну турнірну платформу, яка є доступною для кожного гравця.

2 МАТЕРІАЛИ І МЕТОДИ

2.1 Завдання та цілі дослідження

Проаналізуємо поставлене перед нами завдання для дослідження.

У сучасному геймінговому середовищі існує ряд проблем, пов'язаних із процесом формування команд "мікс кап" турнірів на різних турнірних платформах. Більшість турнірних платформ відображають відділені розділи для гравців і команд, при цьому не надаючи ефективних механізмів автоматичного підбору гравців у команди. Це ускладнює здійснення учасниками пошуку товаришів по команді та може призводити до нерівних за рівнем гри складів. Гравці, які не належать до вже сформованих команд, можуть зіткнутися з труднощами при пошуку або приєднанні до вже існуючих груп. Відсутність зручного механізму для швидкого та ефективного пошуку команд може відповідати за обмежену участь в турнірах. Існуючі платформи часто не забезпечують індивідуалізованого підходу до гравців, що може враховувати їхні унікальні стилі гри та персональні вподобання. Це може впливати на якість взаємодії та командного спілкування. Причиною цьому є те, що більшість турнірних платформ більше орієнтовані на "кіберспортивну" складову, ніж на розважальну [7], [8].

Для вирішення проблем, виявлених у попередньому аналізі, пропонується розробити новий алгоритм формування команд, спрямований на ефективність, баланс та уникнення "нечесності" у "мікс кап" турнірах. Основні характеристики алгоритму включають:

- аналіз рейтингу – алгоритм буде базуватися на аналізі рейтингу кожного гравця;
- формування балансованих груп – алгоритм спрямований на формування балансованих груп з урахуванням індивідуального рейтинга;

- система зміни композиції – алгоритм буде забезпечувати нестандартність складу команд, уникати формування однотипних груп, тим самим сприяючи різноманітності ігрового процесу;
- автоматичне підбирання гравців – система буде автоматично підбирати гравців для формування команд;
- публічна відкритість критеріїв – система надасть публічний доступ до критеріїв, за якими вона формує команди. Це забезпечить прозорість та довіру користувачів до алгоритму [7], [8].

Як працює алгоритм: на першому етапі приймається масив гравців, чого гравці сортуються за рейтингом в зростаючому порядку та розділяються на команди. На другому етапі відбувається оптимізація балансу гравців між командами. Основна його мета полягає в тому, щоб обмінювати гравців між командами так, щоб покращити баланс рейтингів у кожній команді. На цьому етапі для кожної команди визначається поточний середній рейтинг, після чого йде перебір інших команд для порівняння. Для кожної пари команд розглядаються всі можливі обміни гравцями і розраховується новий баланс рейтингів для обох команд після обміну. Далі йде порівняння нового балансу з поточним. Якщо новий баланс поліпшується, здійснюється обмін між командами. Цей цикл продовжується до того моменту, коли жоден обмін не поліпшує баланс. Також в контексті алгоритму виконується функція, яка дозволяє оцінити середній рейтинг кожної команди під час оптимізації балансу гравців. Основна ідея полягає в тому, щоб визначити ефективність кожної команди шляхом розрахунку середнього рейтингу її гравців. Це допомагає визначити, наскільки "сильною" є кожна команда з погляду рейтингу її гравців. Процес оптимізації балансу використовує цю інформацію для прийняття рішень про те, чи слід вносити зміни в розподіл гравців між командами. Це важливо для того, щоб алгоритм міг "виміряти" ефективність команд та визначати, які обміни гравцями можуть покращити баланс. Ці обміни мають спрямовуватися на збільшення середнього рейтингу кожної команди або зменшення відмінностей у рейтингах між командами.

Розглянемо цілі дослідження.

Для гравців, які не мають власних команд, буде впроваджено автоматичну систему підбору, яка самостійно формує балансовані групи на основі індивідуального рейтингу. Це дозволить гравцям без колективного досвіду з легкістю приєднуватися до турнірів.

Процес реєстрації буде оптимізовано для швидкості та зручності. Мінімізуючи кількість обов'язкових полів та забезпечуючи інтуїтивно зрозумілий інтерфейс, це зменшить час та зусилля, необхідні для вступу в турнірний процес.

Буде розглянуто можливість розширення кількості турнірів, призначених спеціально для індивідуальних гравців. Це дозволить створити додаткові можливості для участі та конкуренції для гравців, які не мають власних команд.

Використання плагінів для розширення функціоналу та автоматизації операцій, таких як формування таблиць результатів, автоматичне визначення переможців та видача нагород [7], [8].

2.2 Теоретичні засади алгоритмів

В основі алгоритму лежить рейтингова система, яка присвоює кожному гравцеві числове значення на основі його рівня гри. Це число відображає загальний рівень гравця та його навички [9], [10].

Використання генетичного алгоритму для групування гравців на основі їхніх рейтингів. Мета полягає в створенні команд, середні рейтинги яких мають мінімальну розбіжність, щоб забезпечити баланс і справедливість [9], [10].

Включення в алгоритм системи збалансованості, яка враховує не лише середній рейтинг команд, але й баланс між ролями забезпечує різноманітність у складі кожної команди.

Забезпечення прозорості та визначених правил для гравців шляхом доступу до інформації про те, як рейтинговий алгоритм впливає на формування команд.

Проведення тестування та валідації алгоритму на різних даних та сценаріях турнірів для забезпечення його ефективності та адаптивності.

2.3 Методи оцінки ефективності

В рамках оцінки ефективності будуть використовуватись математичні моделі для розробки алгоритму формування мікс-кап команд для турнірів. Основна ідея полягає в тому, щоб система автоматично визначала оптимальний склад команд, щоб забезпечити баланс та рівень гри.

Ключові компоненти:

- рейтинг гравців – визначення рейтингу кожного гравця на основі їхньої ігрової статистики;
- позиції гравців – врахування позиції гравців в команді;
- варіативність – розробка варіативності алгоритму, яка дозволить різні оптимально збалансовані команди;
- формула балансу – створення математичної формули, яка визначає баланс команд на основі рейтингів та позицій гравців;
- попереднє оцінювання – використання попереднього оцінювання гравців для визначення їхнього потенційного внеску в команду та визначення, як вони впливають на загальний баланс.

Математичні моделі спрямовані на автоматизоване формування команд таким чином, щоб гравці були подібні за рівнем гри, уникати великих відмінностей у навичках та рейтингах. Це допомагає забезпечити адекватність та баланс учасників у мікс кап турнірах.

Експериментальні дослідження будуть використовуватись для оцінки ефективності та реалізації розробленого алгоритму формування мікс кап

команд для турнірів. Ці дослідження базуються на реальних турнірах або симуляціях у контрольованих умовах.

Основні аспекти:

- участь у реальних турнірах – залучення розробленого алгоритму до реальних турнірів для спостереження його дії в реальному середовищі;
- контрольовані симуляції – створення контрольованих сценаріїв для тестування алгоритму в умовах, які можуть виникнути під час турнірів;
- збір даних та статистики – акумулювання даних з реальних турнірів або симуляцій для подальшого аналізу та порівняння з передбачуваними результатами;
- оцінка продуктивності – визначення ключових показників продуктивності, таких як точність формування команд, час виконання алгоритму та баланс учасників;
- коригування та покращення – врахування отриманих даних та результатів для внесення коригувань у алгоритм та його оптимізації.

Експериментальні дослідження дозволяють перевірити ефективність розробленого алгоритму в реальних умовах турнірів або їх симуляціях. Результати цих досліджень служать основою для оцінки його працездатності, а також для подальшого удосконалення та оптимізації алгоритмів формування мікс кап команд.

Зворотний зв'язок від користувачів грає ключову роль у вдосконаленні та адаптації розробленого алгоритму для формування мікс кап команд. Цей етап дослідження базується на зборі думок, вражень та рекомендацій користувачів, які брали участь у турнірах за допомогою запропонованої системи.

Основні складові:

- анкетування та опитування – створення анкет та опитувань для збору думок гравців щодо ефективності алгоритму та їхнього загального досвіду використання;

- фокус-групи – проведення фокус-груп для глибокого аналізу думок та обговорення важливих питань з користувачами;
- оцінка задоволеності – визначення рівня задоволеності користувачів від участі в мікс кап турнірах та ефективності системи формування команд;
- аналіз скарг та пропозицій – врахування конструктивних скарг і пропозицій користувачів для подальшого вдосконалення системи;
- підготовка звіту – спільно зібрали даних і вражень користувачів підготовка докладного звіту, який включає рекомендації та можливості покращення.

Отриманий зворотний зв'язок визначає, наскільки користувачі вважають систему формування мікс кап команд ефективною та зручною. Детальний аналіз цього зворотного зв'язку надає цінні відомості для вдосконалення алгоритму та врахування особливостей, які можуть поліпшити загальний досвід гравців під час турнірів.

2.4 Ключові показники продуктивності

Ключові показники продуктивності в контексті дипломної роботи визначаються для кількісної оцінки ефективності алгоритму формування мікс кап команд та участі у турнірах. Ці показники грають важливу роль у визначенні якості та робочого стану розробленої системи. Основними показниками є:

- час формування команд – вимірюється час, необхідний для системи на вибір та оповіщення гравців про їхню участь в мікс кап турнірі;
- збалансованість складу – оцінюється рівень збалансованості складів мікс кап команд за рівнем гри, рейтингом гравців та іншими факторами;
- рейтингова точність – показник, який визначає, наскільки точно алгоритм враховує рейтинг гравців при формуванні команд;

- ефективність турнірів – вимірює, як система впливає на загальний досвід гравців та ефективність турнірів, зокрема, чесність та конкурентоспроможність;
- сприйняття користувачів – базується на думках та враженнях користувачів щодо взаємодії з системою, комфортності та задоволеності від участі;
- реагування на зміни – визначає, наскільки швидко та ефективно система адаптується до змін у складі гравців, а також на внутрішні та зовнішні фактори турнірів.

Ці ключові показники визначаються з урахуванням особливостей мікс кап турнірів, де об'єднання гравців у команди має враховувати не лише їхній рівень гри, але й забезпечувати рівномірність та конкурентоспроможність учасників. Ці показники становлять критичний аспект оцінки ефективності розробленого алгоритму та його впливу на учасників мікс кап турнірів.

3 РОЗРОБКА АРХІТЕКТУРИ ПРОГРАМИ

В попередніх розділах відбувся аналіз проблем організації та проведення кіберспортивних турнірів у форматі “мікс кап”, а також визначені мета та об’єкт дослідження. Було проведено порівняння існуючих аналогів турнірних платформ. На основі виявлених проблем було сформовано нові вимоги до створюваного за стосунку, що буде реалізовувати функцію змішування гравців по командах в турнірах формату “мікс кап”.

Далі попередньо проведений аналіз дозволить визначитись з питаннями розробки застосунку: вибором технології та принципом реалізації. В цьому розділі буде проведено аналіз з приводу вибору мови програмування для реалізації функціональної частини, програмних комплексів та засобів оформлення ПЗ.

3.1 Вибір інструментарію розробки

3.1.1 Вибір мови програмування

PHP, попередня назва: Personal Home Page Tools — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні вебсервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веброзробок (разом із Java, .NET, JavaScript, Python, Ruby) [11].

PHP є однією з найпоширеніших мов, яку використовують у сфері веброзробки, її підтримує більшість хостинг-провайдерів. PHP інтерпретує вебсервер у HTML-код, який передається на сторону клієнта. На відміну від JavaScript, користувач не бачить PHP-коду, тому що браузер отримує готовий HTML-код. Це є перевагою з погляду безпеки, але погіршує інтерактивність сторінок [12]-[14].

Згідно з опитуванням DOU, PHP для роботи використовують 10,6% українських розробників. PHP — на п'ятому місці в рейтингу мов

програмування серед українців, її популярність на початку 2022 року зросла на 4,2% порівняно з 2021-м [12]-[14].

Якщо говорити про розподіл мов за сферами використання, то PHP послуговуються в роботі близько 18% Back-end розробників та близько 15% Full Stack розробників. Частка PHP як першої мови початківців щороку зростає, у 2022 році вона становить 5,6%. Якщо подивитися на історію мови і перші версії, стає ясно, що мова втілила в собі всі кращі можливості існуючих мов програмування для настільних систем, наприклад, багато функцій нагадують поведінку функцій з мови C або C++... Згідно з деякими джерелами, на розробку PHP вплинули такі мови, як Perl, C, C++, Java, Tcl [12]-[14].

Як працює PHP. Коли ви відкриваєте вебсайт за допомогою браузера, браузер відправляє HTTP-запит до вебсерверу, на якому знаходиться сайт. Вебсервер отримує запит і дає відповідь HTML-документом [12]-[14].

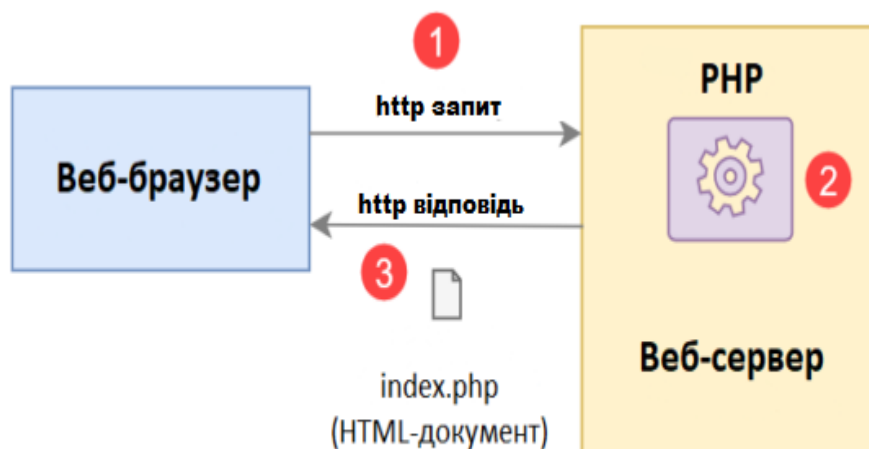


Рисунок 3.1 – Принцип роботи PHP

В цьому прикладі браузер є клієнтом, а вебсервер – сервером. Клієнт запитує сторінку, а сервер оброблює запит. PHP працює на вебсервері, оброблюючи запит і повертаючи HTML-документ [12]-[14].

PHP-файли мають розширення `.php` та можуть мати в собі HTML-теги і JavaScript-код. PHP-код може бути влаштованим в HTML-код або використовуватись з різними фреймворками або системами керування контентом (CMS) [12]-[14].

Переваги PHP:

- безкоштовний і з відкритим вихідним кодом – не потрібно платити ліцензійний збір за використання PHP для розробки програмних продуктів;
- кросплатформність – не залежить від конкретної платформи. PHP працює на більшості операційних системах, включаючи Linux, Windows та MacOS;
- простота – PHP порівняно легкий у вивченні та старті для новачків;
- стабільність – PHP вже багато років;
- має велику кількість бібліотек;
- вбудовані в PHP модулі підключення до баз даних допомагають скоротити кількість проблем та час, необхідний для розробки вебзастосунків;
- підтримка спільноти — PHP має активну онлайн-спільноту, яка допоможе вам, коли ви зіткнетеся з проблемою;
- мова сценаріїв за сервера – це означає, що вам потрібно встановити його тільки на сервер, а клієнти (користувачі), які запитують ресурси з сервера, не повинні мати PHP; достатньо буде лише браузера;
- PHP має вбудовану підтримку для спільної роботи з MySQL, а також може працювати з іншими системами керування базами даних (наприклад, PostgreSQL, Microsoft SQL Server, Oracle Database та ін.);
- PHP підтримується такими популярними вебсерверами, як Nginx, OpenBSD та Apache. Деякі хмарні середовища також підтримують PHP, наприклад, Microsoft Azure та Amazon AWS;

– PHP досить гнучкий, він не обмежується лише обробкою HTML, а має вбудовану підтримку для створення та обробки файлів форматів PDF, GIF, JPEG та PNG [12]-[14].

Проведемо аналіз мов програмування з метою визначення їхньої придатності для реалізації вебзастосунків, і отримані результати представлені у табличній формі (див. таблицю 3.1). Критерії порівняння включають такі аспекти:

- складність синтаксису – розглядає загальну складність синтаксису, яка в подальшому впливає на швидкість розробки, підтримку та розвиток проєкту;
- швидкість та продуктивність – оцінює швидкість та продуктивність роботи серверної частини застосунку;
- програмні бібліотеки фреймворків та CMS – аналізує механізми та конструкторські рішення, які спрощують та прискорюють розробку.
- підтримка – враховує наявність матеріалів та активної спільноти, які допомагають під час розробки.

З аналізу видно, що PHP тривалий час залишається найпопулярнішою мовою програмування для створення вебзастосунків. Це пояснюється великою активною спільнотою та найоптимальнішими сценаріями підтримки на серверах, роблячи її найвигіднішим вибором для веброзробки.

Таблиця 3.1 – Порівняння мов програмування

Критерій	C# ASP.NET	Python	PHP
Складність синтаксису	+	++	+
Швидкість та продуктивність	+	±	++
Програмні бібліотеки фреймворків та CMS	±	±	++
Підтримка	+	++	++

3.1.2 Система керування контентом

Система керування контентом (Content Management System – CMS) – це програмна система, яка дозволяє створювати, редагувати, зберігати у єдиному репозиторії історичні версії цифрового вмісту, такого як документи та мультимедіа, та виконувати їх публікацію на вебсайті. Тобто, така система підтримує редакційний процес контенту [16], [17].

Сьогодні використання CMS дуже поширене. Їх використовують Інтернет видання, державні установи і організації та приватні компанії, а також навчальні заклади. Також перевагою використання CMS є те, що вони дозволяють відокремити технології презентації (HTML, CSS, JavaScript тощо) від керування контентом [16], [17].

Серед розробників сайтів, великою популярністю користується система управління сайтів або движок WordPress (Вордпресс). WordPress – система управління сайтом з відкритим вихідним кодом, реалізована на мові програмування php і використовує стандартну базу даних MySQL. Незважаючи на те, що дана платформа, створювалася в першу чергу для блогів, сучасні можливості WordPress дозволяють створювати сайти будь-якої складності, від корпоративних сайтів до інтернет-магазинів або порталів. Згодом система набрала популярність і під неї стали створювати додаткові плагіни і теми, адаптувавши архітектуру WordPress під більш складні проекти. WordPress заслужив високу оцінку в розробників, клієнтів, пошукових системах завдяки ряду переваг [16], [17].

Простота в експлуатації і використанні, завдяки грамотно розробленій адміністративній панелі. Це дає можливість вносити зміни і додавати матеріали людям без додаткових знань програмування. Також величезна кількість безкоштовних тем з різноманітним дизайном, шаблонів і більше 10 тис. модулів дозволяють вибрати і, в разі необхідності, вдосконалити, змінити, поліпшити розробку або роботу будь-якого вебресурсу [16], [17].

WordPress має дуже зручне навігаційне меню, яке можна легко адаптувати під основні завдання сайту, тим самим покращивши його юзабіліті і сприйняття користувачами. Система постійно оновлюється, і вдосконалюється, а разом з нею і вебсайт власника [16], [17].

Сайт на WordPress відмінно індексується пошуковими системами, що дає можливість покращувати рейтингові показники сайту, при пошуковому просуванні ресурсу в ТОП 10. Дана платформа дає можливість внесення змін вибраного шаблону і створення унікального сайту по дизайну або структурі. За рахунок цього на ринку велика кількість розробників володіє WordPress, відповідно великий вибір підрядника на розробку і нижча вартість робіт [16], [17].

Таким чином, у разі прийняття рішення по розробці власного сайту-візитки, корпоративного сайту, блогу або простого інтернет-магазину при виборі движка варто в першу чергу звернути увагу на WordPress. Використовуючи безкоштовний шаблон WordPress, розробка сайту, що продає забере не багато часу і фінансових витрат, а в результаті клієнт отримає відмінний інструмент для продажу власних товарів або послуг в мережі інтернет [16], [17].

3.1.3 Обґрунтування вибору CMS для розробки вебресурсу

Порівняльна оцінка найпоширеніших CMS для розробки звичайних сайтів наведена у таблиці 3.2. Розглянемо основні критерії порівняння.

Серед критеріїв порівняння будемо розглядати наступні:

- простота використання системи – оцінка легкості використання інтерфейсу CMS, спрощення робочого процесу для користувачів без глибоких технічних знань;
- доступність додаткових модулів – вивчення можливостей для легкого впровадження додаткових модулів, розширень та плагінів для розширення функціональності CMS;

- рівень обслуговування CMS – оцінка якості та швидкості технічної підтримки, наявність оновлень та документації для користувачів;
- безпека – аналіз заходів, які надає CMS для забезпечення безпеки вебресурсу, включаючи механізми захисту від зловмисників та вразливостей системи.

З порівняння наведеного в таблиці можна зробити висновок, що використання WordPress є найбільш оптимальним варіантом. Через її популярність більшість внутрішніх механізмів та процесів максимально оптимізовано під потреби розробників.

Таблиця 3.2 – Порівняння CMS

	WordPress	Joomla!	Drupal
Простота	++	+	±
Доступність модулів	++	+	++
Підтримка	++	++	±
Безпека	+	+	±

3.2 Структурна схема розробки

На рисунку 3.2 наведено структурну схему ПЗ, що розробляється.

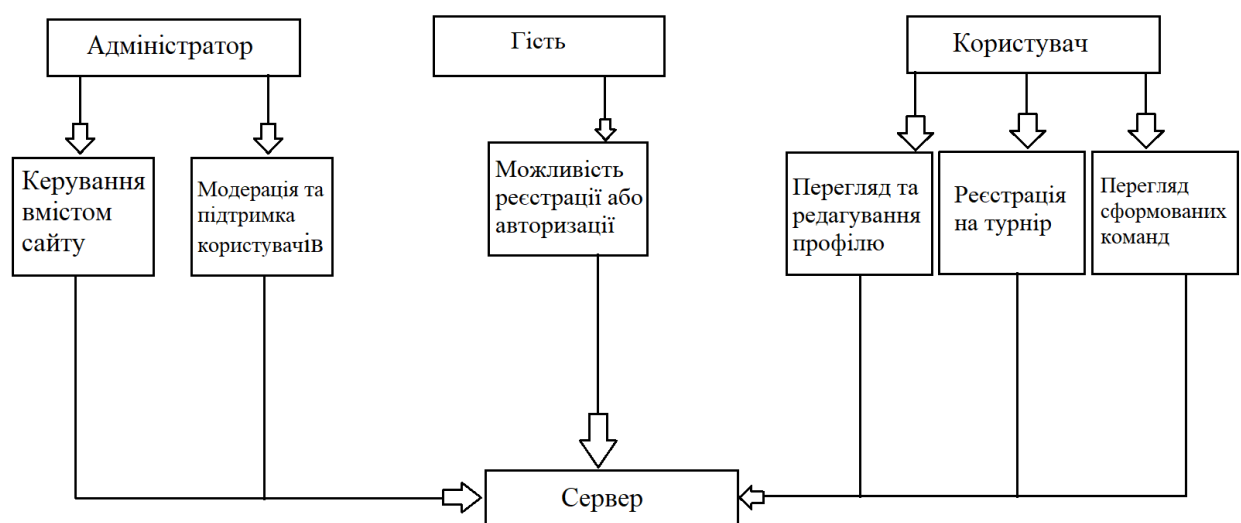


Рисунок 3.2 – Загальна схема системи за обраною темою

WordPress – це проста у встановленні та використанні система керування вмістом з відкритим кодом, яка широко використовується для створення вебсайтів. Сфера застосування – від блогів до складних вебсайтів. Вбудована система тем і плагінів в поєднанні з вдалою архітектурою дозволяє конструювати на основі WordPress практично будь-які вебпроекти [18], [19].

Написана на мові програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License [18], [19].

Структура вебзастосунків, створених на WordPress зображена на рисунку 3.3.

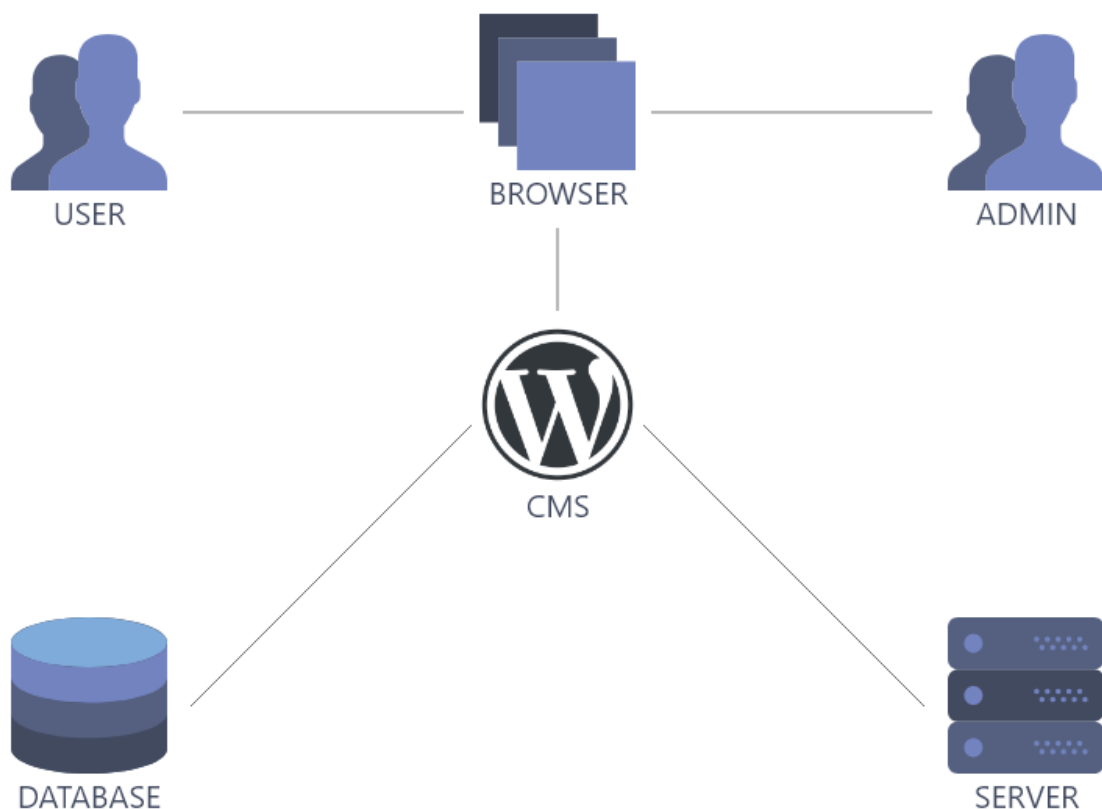


Рисунок 3.3 – Архітектура вебзастосунку

3.3 Висновки за розділом 3

У третьому розділі проведено докладне вивчення загальних технічних аспектів розробки. В результаті аналізу та порівняння різних мов програмування для створення вебзастосунків було вибрано PHP через його численні переваги. Для спрощення та прискорення процесу розробки вирішено використовувати систему управління контентом (CMS), і серед доступних варіантів обрано WordPress за численними перевагами порівняно з іншими CMS.

4 ОСНОВНІ РІШЕННЯ ЩОДО РЕАЛІЗАЦІЇ КОМПОНЕНТІВ СИСТЕМИ

4.1 Проектування моделей системи

Unified Modeling Language (UML) - це мова візуального моделювання загального призначення, яка використовується для уточнення, візуалізації, побудови та документування артефактів програмної системи. Вона фіксує рішення та розуміння про системи, які мають бути створені. Використовується для розуміння, проектування, перегляду, конфігурації, утримання та контролю інформації про такі системи. Призначена для використання з будь-якими методами розробки, етапами життєвого циклу, областями застосування та носіями. Мова моделювання призначена для уніфікації минулого досвіду моделювання та включення поточних найкращих практик з розробки програмного забезпечення у стандартний підхід. UML включає семантичні концепції, нотацію та рекомендації. Вона має статичні, динамічні, середовищні та організаційні частини. Призначена для підтримки візуальних інтерактивних засобів моделювання, що мають генератори коду та засоби для створення звітів. Специфікація UML не визначає стандартного процесу, але призначена для використання з ітеративним процесом розробки. Призначена для підтримки більшості існуючих об'єктно-орієнтованих процесів розробки [20], [21].

UML фіксує інформацію про статичну структуру та динамічну поведінку системи. Система моделюється як набір окремих об'єктів, які взаємодіють для виконання роботи, яка в кінцевому підсумку користується зовнішнім користувачем. Статична структура визначає види об'єктів, важливих для системи та її впровадження, а також відносини між об'єктами. Динамічна поведінка визначає історію об'єктів з часом та взаємодії між об'єктами для досягнення цілей [20], [21].

Моделювання системи з різних, але пов'язаних точок зору дозволяє її розуміти для різних цілей [20], [21].

UML також містить організаційні конструкції для упорядкування моделей в пакети, які дозволяють командам програмного забезпечення розділяти великі системи на робочі частини, розуміти та контролювати залежності між пакетами та керувати версіонуванням модельних одиниць в складному середовищі розробки. Вона містить конструкції для представлення рішень щодо впровадження та організації елементів часу в компоненти [20], [21].

UML - не мова програмування. Інструменти можуть надавати генератори коду з UML для різних мов програмування, а також створювати моделі зворотної інженерії з існуючих програм. UML не є високоформальною мовою, призначеною для доведення теорем. Існують мови такого типу, але вони не є простими для розуміння або використання в більшості випадків. UML - це мова моделювання загального призначення. Для спеціалізованих областей, таких як розташування графічного інтерфейсу користувача, проектування мікросхем VLSI або систем штучного інтелекту на основі правил, може бути відповідним більш спеціалізований інструмент із спеціальною мовою. UML - це дискретна мова моделювання. Вона не призначена для моделювання неперервних систем, таких, як ті, що зустрічаються в інженерії та фізиці. UML призначено для універсального моделювання загального призначення для дискретних систем, таких як програмне забезпечення, вбудоване програмне забезпечення або цифрова логіка [20], [21].

Діаграма класів UML – це графічне позначення, яке використовується для побудови та візуалізації об'єктно-орієнтованих систем [20], [21]. Діаграма класів на UML – це тип діаграми статичної структури, яка описує структуру системи, показуючи:

- класи;
- їх атрибути;

- операції (або методи);
- відносини між об'єктами [20], [21].

Діаграма класів вебзастосунку, що розробляється зображена на рис.

4.1.

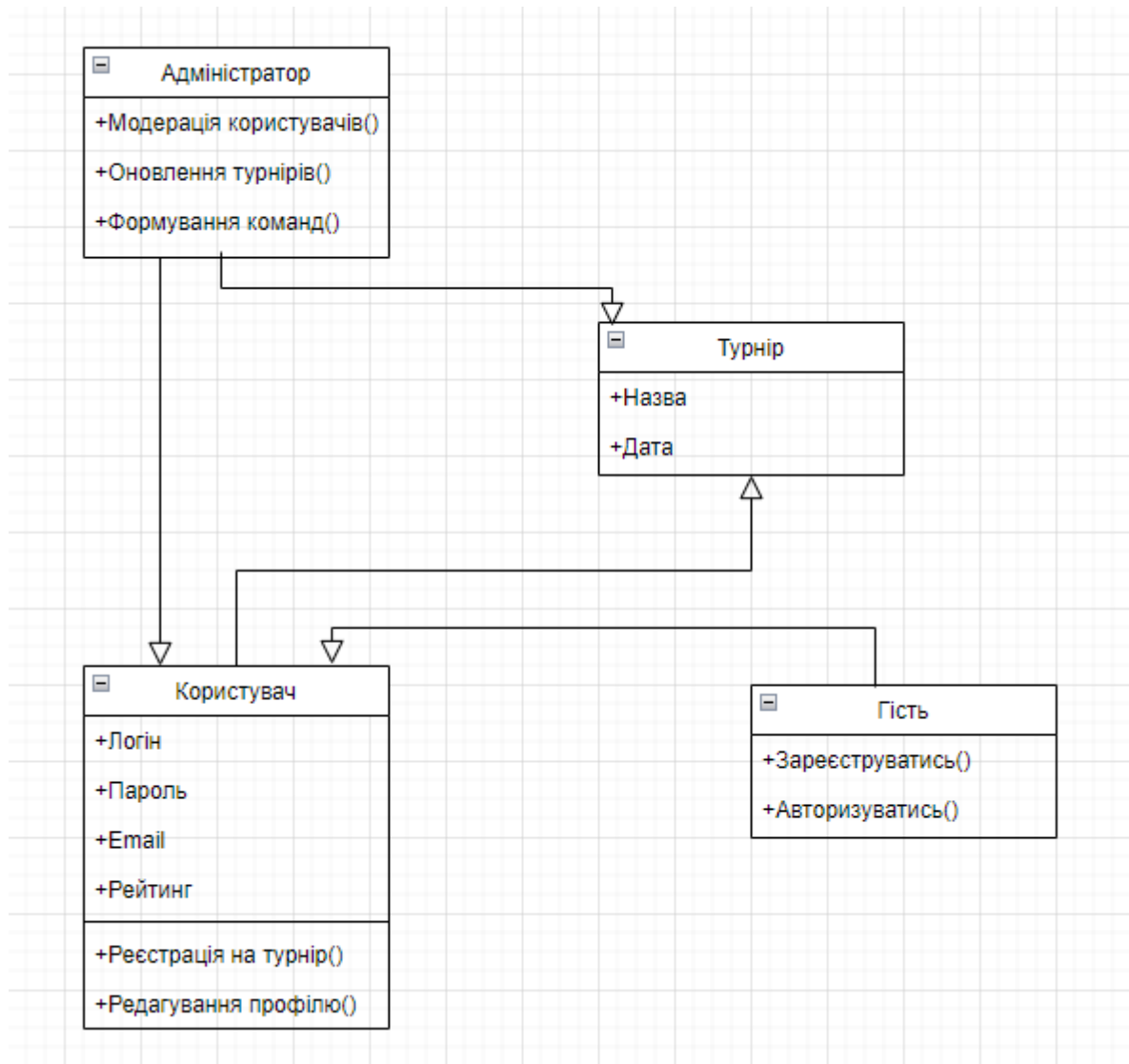


Рисунок 4.1 – Діаграма класів

Діаграми пакетів – це структурні діаграми, які використовуються для показу організації та розташування різних елементів моделі у вигляді пакетів [20], [21]. Пакет – це групування пов'язаних елементів UML, таких як діаграми, документи, класи чи навіть інші пакети. Кожен елемент вкладений в пакунок, який зображений у вигляді папки файлів на діаграмі, а потім

впорядкований ієрархічно в межах діаграми. Діаграми пакетів найчастіше використовуються для наочної організації багат шарової архітектури в будь-якому класифікаторі UML, наприклад, у програмній системі [20], [21].

Діаграма пакетів для вебзастосунку, що розробляється на рис. 4.2.

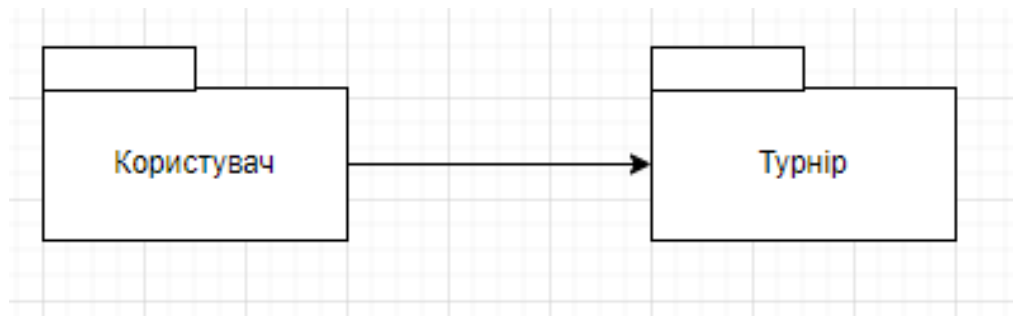


Рисунок 4.2 – Діаграма пакетів

Мета діаграми компонентів – показати взаємозв’язок між різними компонентами в системі. Для цілей UML 2.0 термін «компонент» відноситься до модуля класів, які представляють незалежні системи або підсистеми з можливістю взаємодії з рештою системи [20], [21]. Існує цілий підхід, пов’язаний з розробкою на основі компонентів. У цьому підході діаграми компонентів дозволяють планувальнику ідентифікувати різні компоненти, щоб вся система виконувала те, що вона повинна робити. Частіше, в підході до програмування, діаграма компонентів дозволяє старшому розробнику групувати класи разом на основі спільної мети, щоб розробник та інші могли дивитися на проект розробки програмного забезпечення на високому рівні [20], [21].

Діаграми компонентів для вебзастосунку, що розробляється, зображена на рис. 4.3.

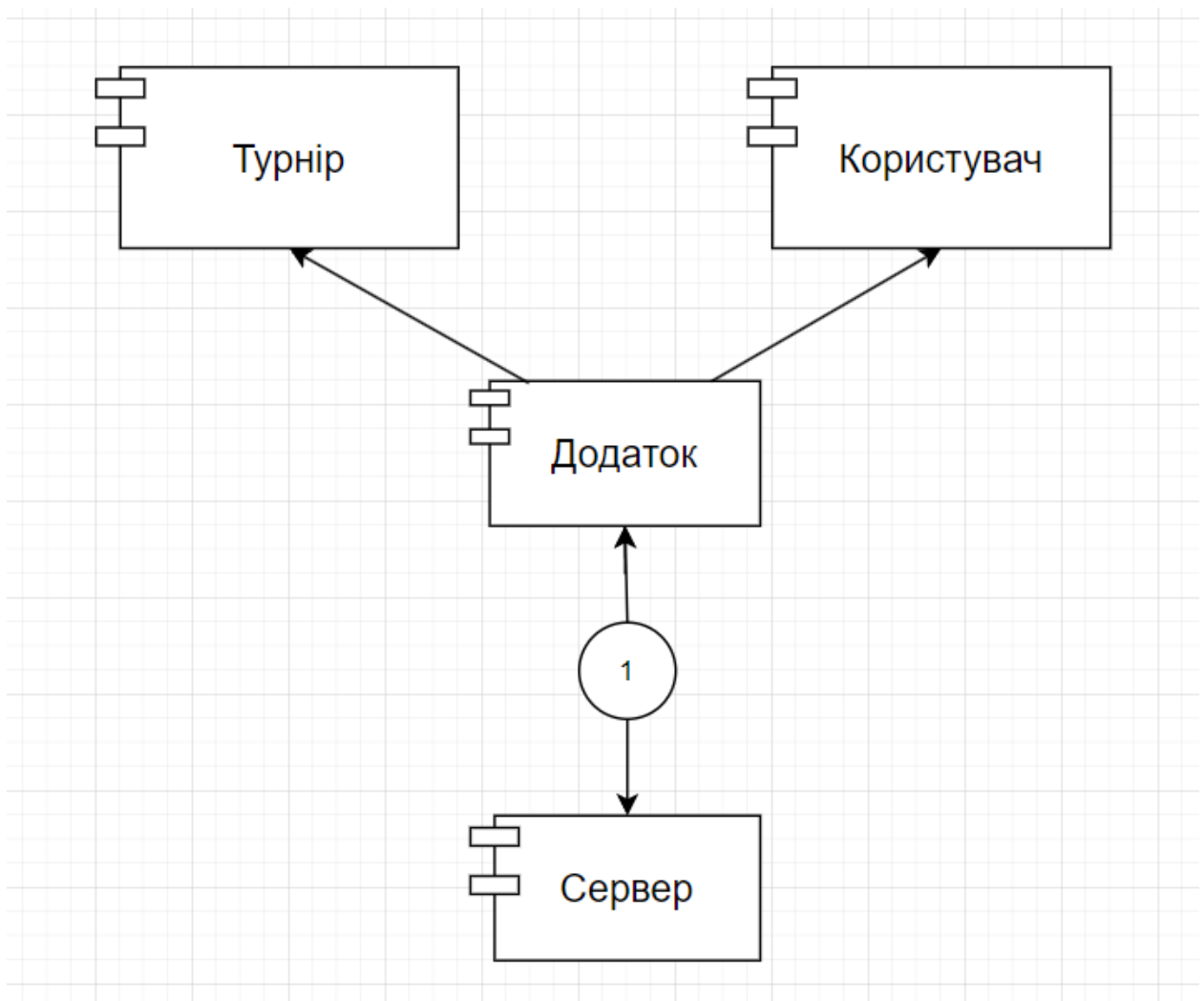


Рисунок 4.3 – Діаграма компонентів

4.2 Моделювання системи з точки зору прецедентів

Діаграма варіантів використання – це графічне зображення можливої взаємодії користувача з системою [20], [21]. Діаграма варіантів використання показує різні варіанти використання та різні типи користувачів, яких має система, і часто супроводжується діаграмами інших типів. Варіанти використання представлені або колами, або еліпсами. Актори часто зображені у вигляді фігурок [20], [21].

Прецедент. Реєстрація користувача на турнір (рис. 4.4).

Основна діюча особа. Гість/Зареєстрований користувач.

Основний позитивний сценарій:

- користувач заходить на сайт;
- користувач натискає на кнопку реєстрації/авторизації;
- вибирає метод і заповнює форму;
- виводиться повідомлення про успішне завантаження;
- можливість реєстрації на турнір.

Альтернативний потік АП1:

- користувач невірно заповнює дані у формі;
- виводиться повідомлення про помилку.



Рисунок 4.4 – Прецедент «Реєстрація користувача на турнір»

4.3 Фізичне проектування

Діаграма активностей (видів діяльності) – один із доступних видів діаграм, що підтримуються Flexberry Designer [22]-[24]. Вона, як і діаграма станів, відбиває динамічні аспекти поведінки системи. По суті, ця діаграма є блок-схемою, яка наочно показує, як потік управління переходить від однієї діяльності до іншої [20], [21].

Діаграми діяльності для вебзастосунку, що розробляється зображені на рис. 4.5–4.6.

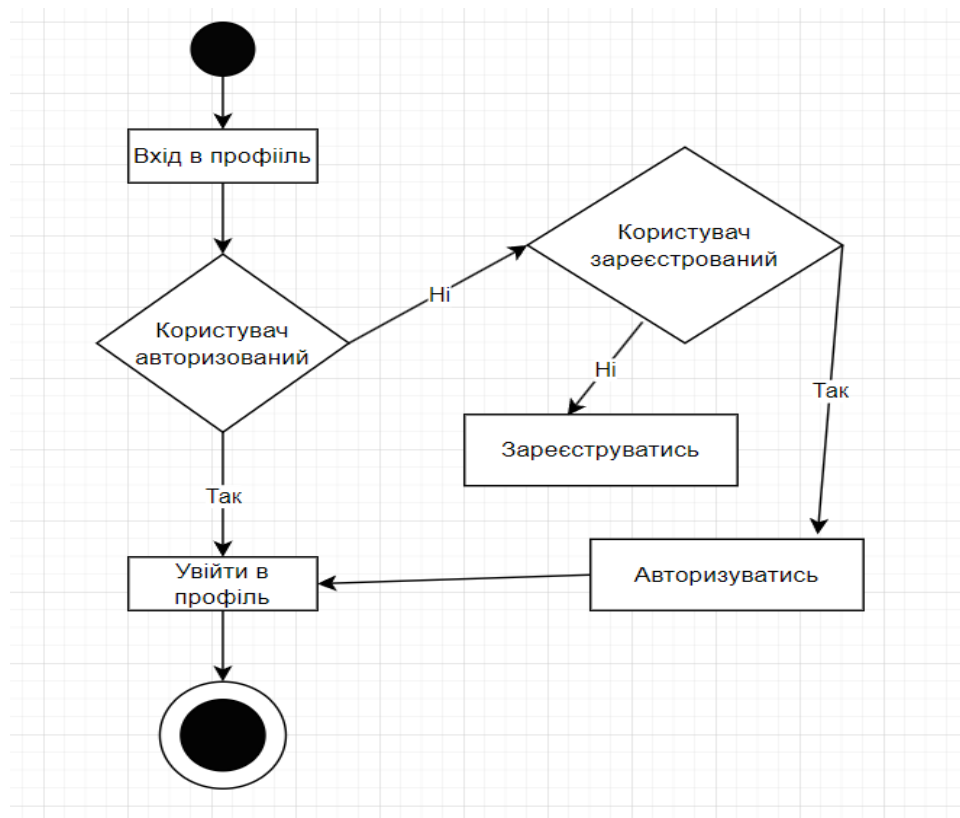


Рисунок 4.5 – Діаграма діяльності: Вхід в профіль користувача

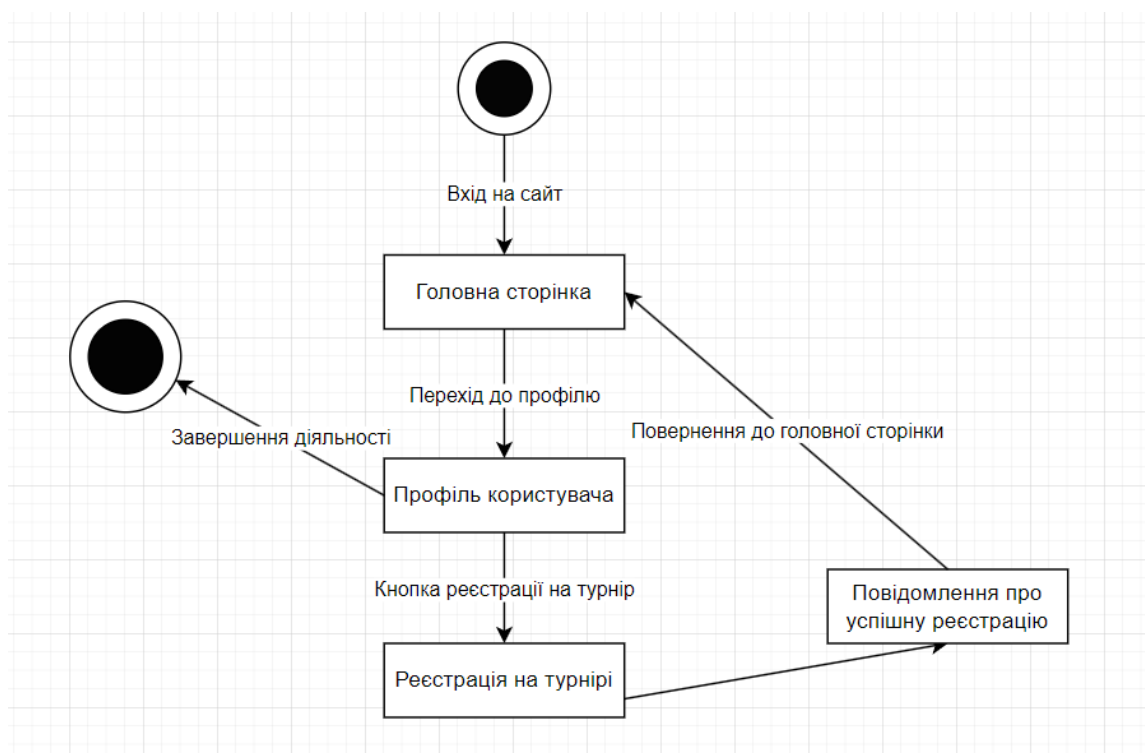


Рисунок 4.6 – Діаграма станів системи

Діаграми кооперації використовуються, щоб показати, як об'єкти взаємодіють, щоб виконувати поведінку конкретного варіанту використання

або частини варіанту використання [20], [21]. Поряд із діаграмами послідовності, кооперація використовується дизайнерами для визначення й уточнення ролей об'єктів, які виконують певний потік подій у варіанті використання [20], [21]. Вони є основним джерелом інформації, яка використовується для визначення обов'язків класу та інтерфейсів [20], [21].

Діаграми кооперації для вебзастосунку, що розробляється зображені на рис. 4.7 та рис. 4.8.

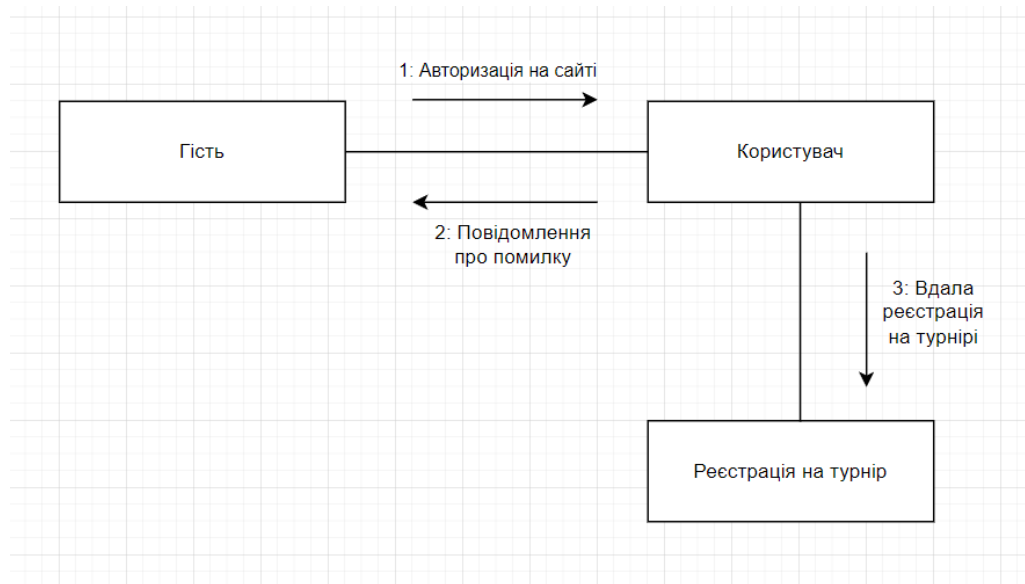


Рисунок 4.7 – Діаграма кооперації. Реєстрація на турнір

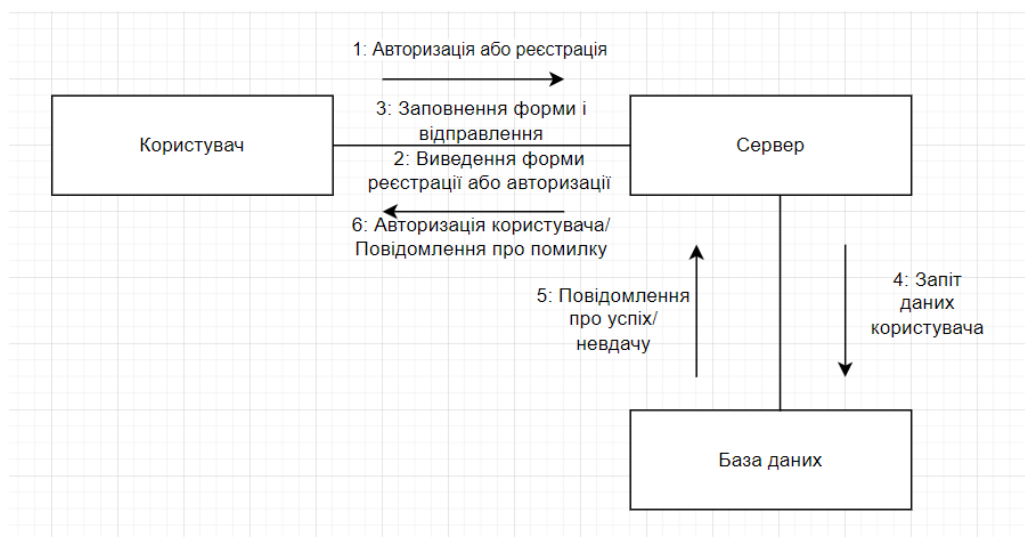


Рисунок 4.8 – Діаграма кооперації. Авторизація користувача

Діаграми послідовностей для вебзастосунку, що розробляється зображені на рис. 4.9 та рис. 4.10.

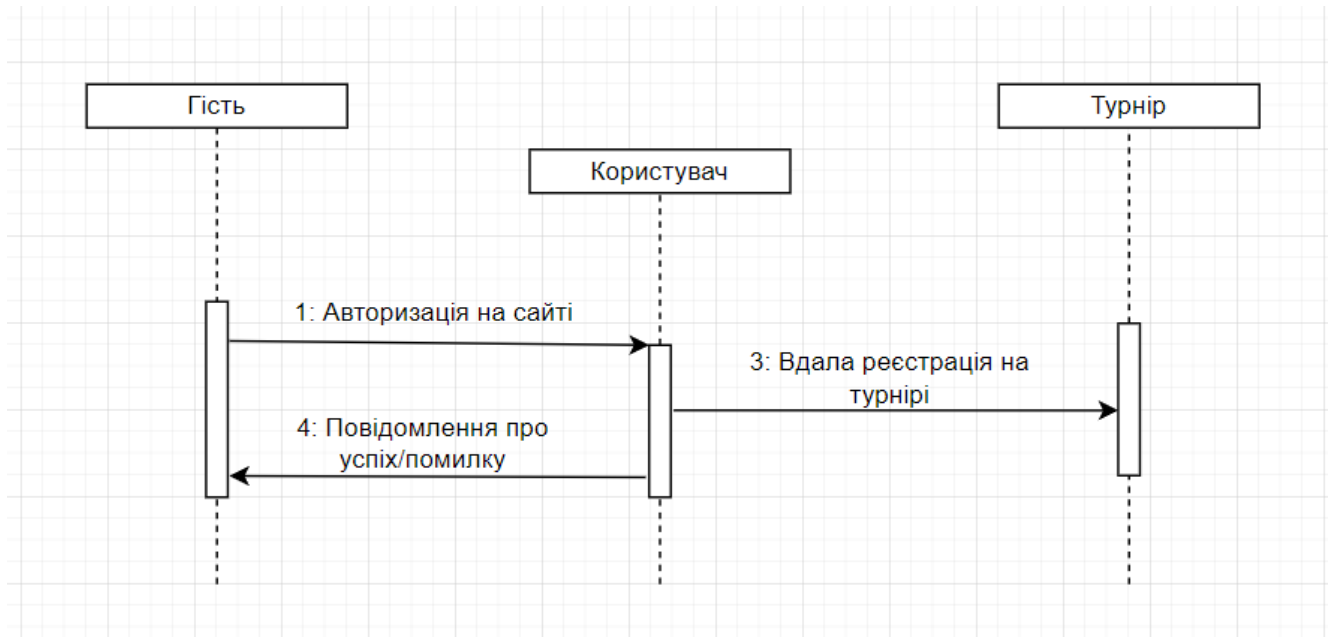


Рисунок 4.9 – Діаграма кооперації. Реєстрація на турнір

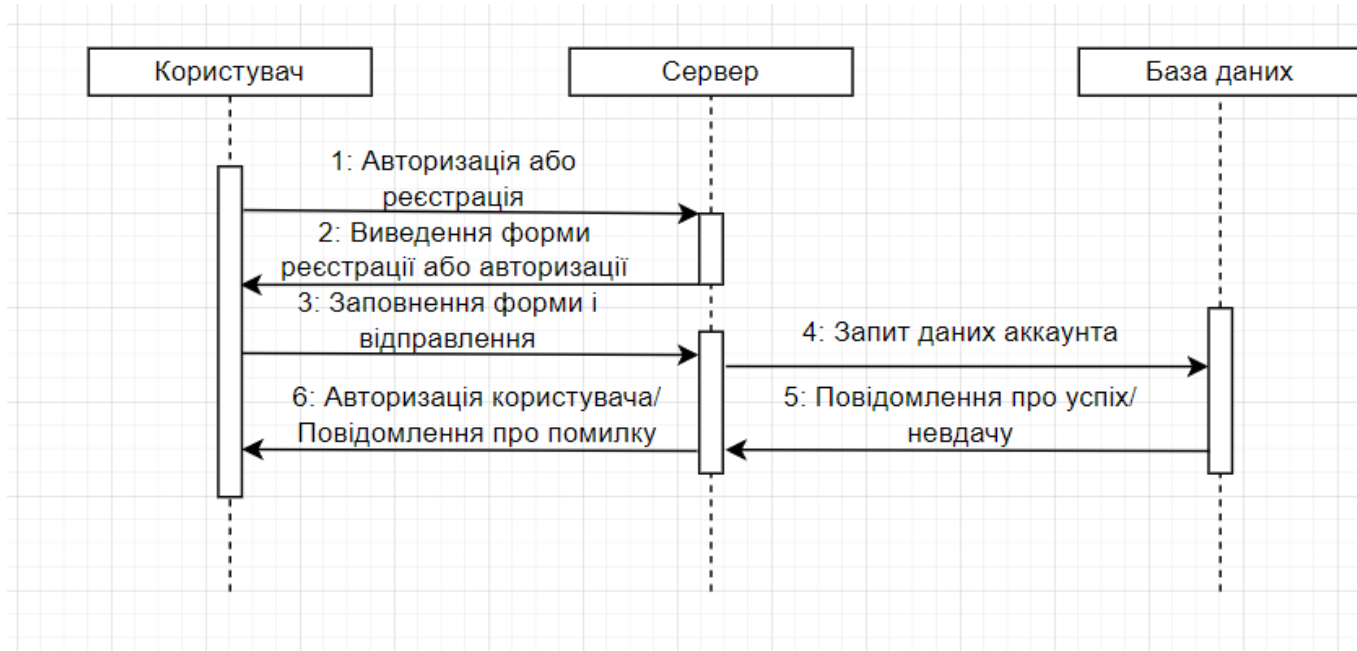


Рисунок 4.10 – Діаграма кооперації. Авторизація користувача

4.4 Проектування інтерфейсу

Спроекований інтерфейс вебзастосунку, враховуючи ергономічні показники інтерфейсу, зображений на рис. 4.11 – 4.17.

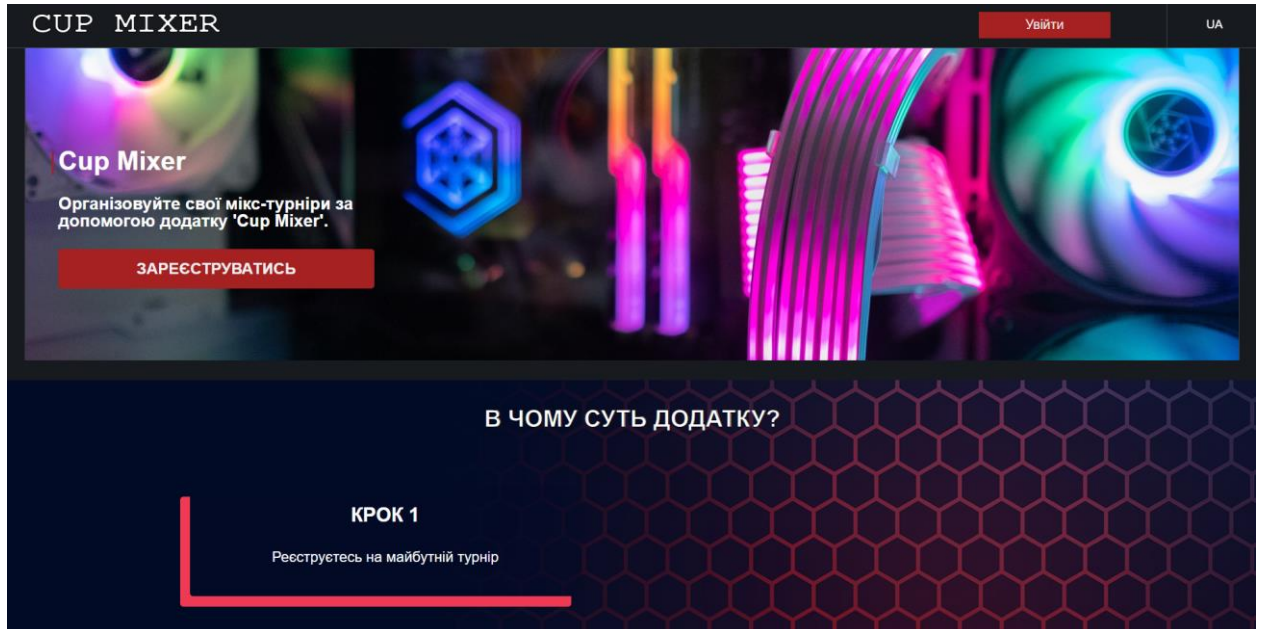


Рисунок 4.11 – Головна сторінка для неавторизованого користувача

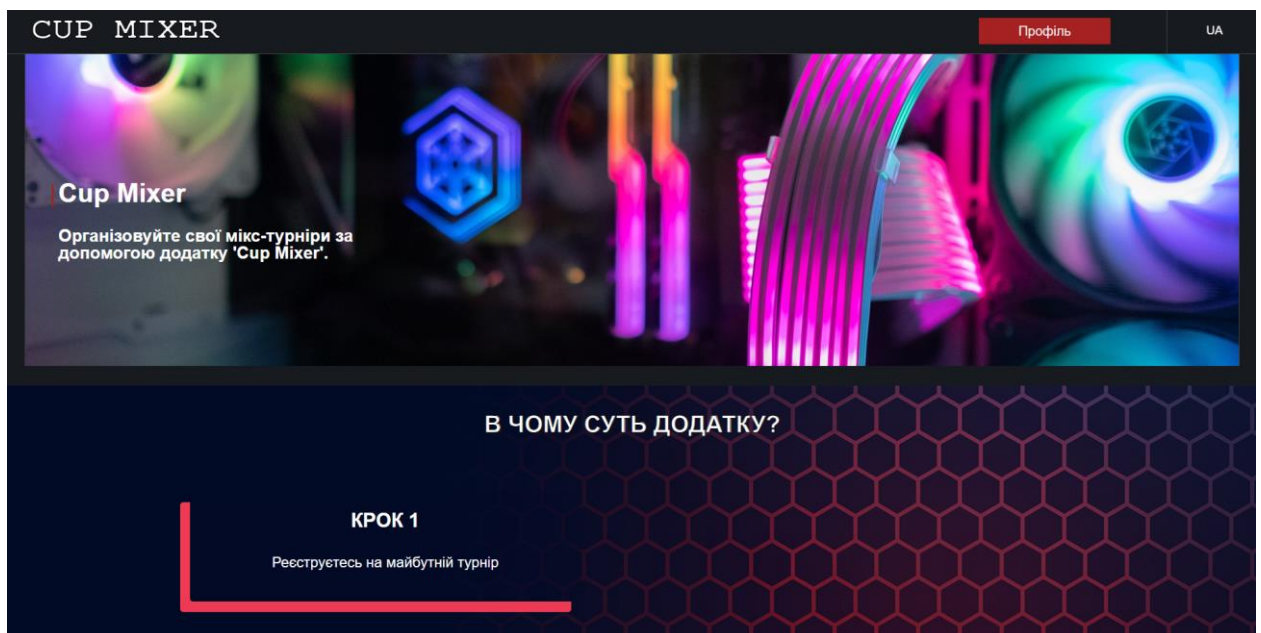


Рисунок 4.12 – Головне меню сайту

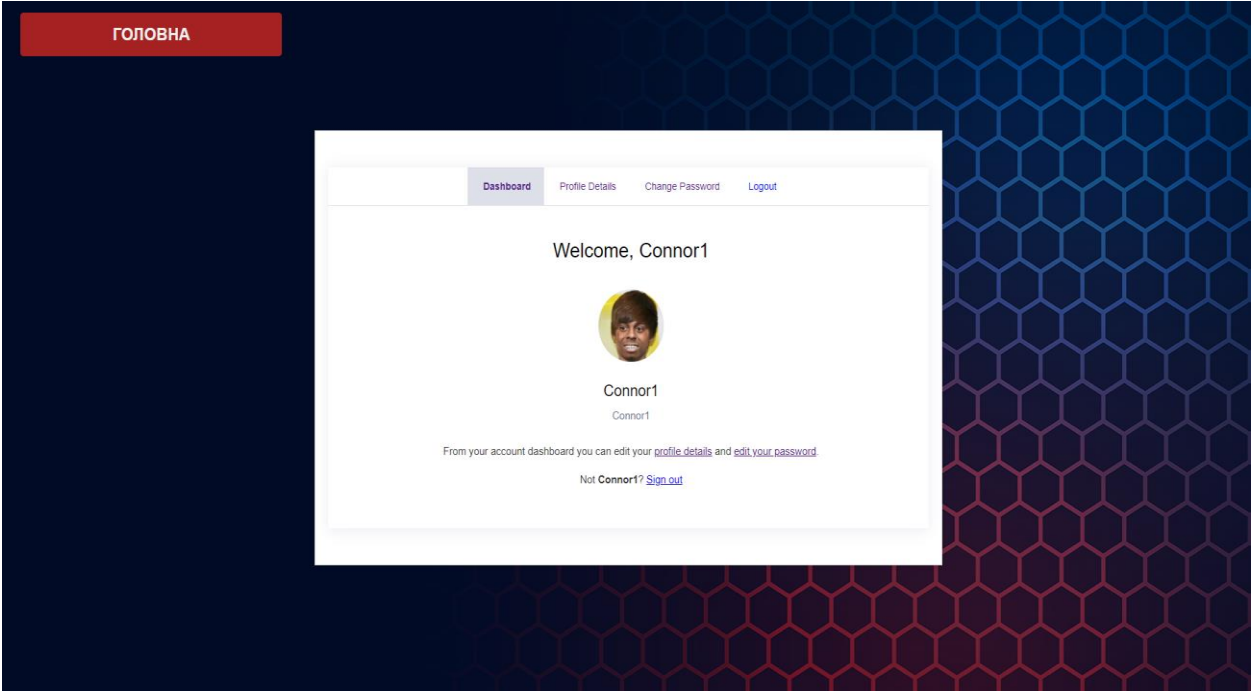


Рисунок 4.13 – Профіль авторизованого користувача

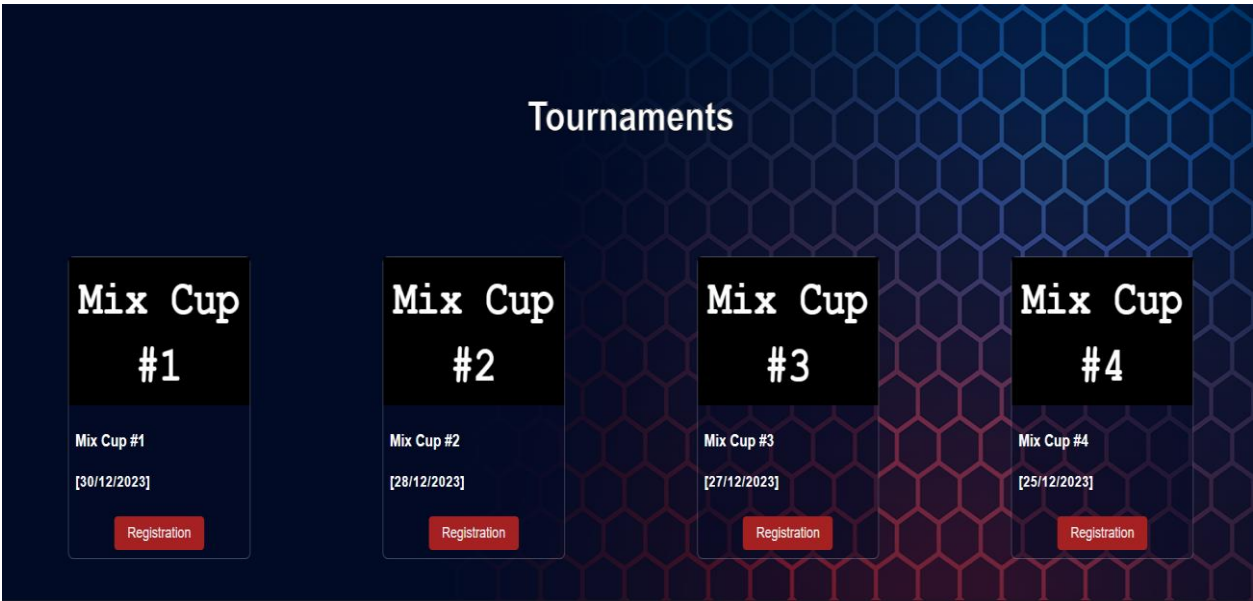


Рисунок 4.14 – Список турнірів на сайті

ГОЛОВНА

Нікнейм або Email *

Пароль *

☐ Запам'ятати мене

[Втратили пароль?](#)

[Ще не учасник? Зареєструватися зараз.](#)

Увійти

Рисунок 4.15 – Форма авторизації користувача

ГОЛОВНА

Нікнейм *

Email *

Пароль *

Підтвердити пароль *

MMR (Ваш рейтинг)

Надіслати

Рисунок 4.16 – Форма реєстрації користувача

4.5 Висновки за розділом 4

Було вивчено різновиди структурних діаграм та їхні особливості в застосуванні. Була розроблена абстрактна модель системи за допомогою мови UML, використовуючи структурні діаграми. Для конкретної теми була розроблена абстрактна модель системи, використовуючи діаграми класів,

пакетів та компонентів, а також діаграми поведінки, такі як прецеденти, діяльності, стани, кооперації та послідовності.

5 ЕКСПЛУАТАЦІЯ, ТЕСТУВАННЯ ТА ЕКСПЕРИМЕНТАЛЬНЕ ДОСЛІДЖЕННЯ ПРОГРАМИ

5.1 Опис застосунку

Вебзастосунок «Cup Mixer» призначений для змішування індивідуально зареєстрованих на турнір гравців змішувати по командам зі збалансованим рівнем гри.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені Open Server, PHP5 та MySQL.

5.2 Умови виконання програми

Мінімальні вимоги до апаратного та програмного забезпечення для оптимального функціонування серверної та клієнтської частини мобільного застосунку включають:

Для серверної частини:

- серверна станція, сумісна з IBM;
- встановлена та налаштована операційна система сервера, яка може бути будь-яким варіантом Unix або Windows Server;
- інтерпретатор PHP5;
- MySQL-сервер для управління користувачами та організацією доступу до бази даних.

Для клієнтської частини:

- персональний комп'ютер, сумісний з IBM;
- процесор із тактовою частотою не менше 1,4 ГГц;
- обсяг оперативної пам'яті не менше 4 ГБ (для безперебійної роботи з новими версіями веббраузерів);
- підключення до мережі Інтернет;
- встановлений браузер.

5.3 Інструкція по експлуатації програми

5.3.1 Характеристики програми

Дане програмне забезпечення являється вебзастосунок, розміщеним на сервері, тобто працює цілодобово, поки сервер запущений та знаходиться в працездатному стані.

Проект має наступну файлову структуру (рис 5.1):

	wp-admin	14.11.2023 22:39	Папка файлів	
	wp-content	24.11.2023 19:55	Папка файлів	
	wp-includes	14.11.2023 22:39	Папка файлів	
	.htaccess	20.11.2023 16:18	Файл HTACCESS	1 КБ
	index	06.02.2020 8:33	Исходный файл ...	1 КБ
	license	01.01.2023 2:06	Текстовий докум...	20 КБ
	readme	05.07.2023 20:41	Chrome HTML Do...	8 КБ
	wp-activate	13.05.2023 0:35	Исходный файл ...	8 КБ
	wp-blog-header	06.02.2020 8:33	Исходный файл ...	1 КБ
	wp-comments-post	14.06.2023 17:11	Исходный файл ...	3 КБ
	wp-config	14.11.2023 22:50	Исходный файл ...	4 КБ
	wp-config-sample	23.02.2023 12:38	Исходный файл ...	3 КБ
	wp-cron	30.05.2023 21:48	Исходный файл ...	6 КБ
	wp-links-opml	26.11.2022 23:01	Исходный файл ...	3 КБ
	wp-load	16.07.2023 15:16	Исходный файл ...	4 КБ
	wp-login	30.09.2023 1:01	Исходный файл ...	50 КБ
	wp-mail	16.09.2023 9:50	Исходный файл ...	9 КБ
	wp-settings	10.10.2023 17:05	Исходный файл ...	26 КБ
	wp-signup	19.06.2023 21:27	Исходный файл ...	34 КБ
	wp-trackback	22.06.2023 17:36	Исходный файл ...	5 КБ
	xmlrpc	30.09.2023 10:39	Исходный файл ...	4 КБ

Рисунок 5.1 – Файлова структура проекту

У кореневому каталозі зберігаються файли конфігурацій та налаштувань [18], [19].

У каталозі wp-admin зберігаються файли, необхідні для функціонування консолі Wordpress, за допомогою якої ми працюємо над проектом. Не підлягає редагуванню [18], [19].

Каталог wp-includes містить ядро CMSi не підлягає будь-якому редагуванню.

Каталог wp-content цікавить нас найбільше, оскільки саме в ньому зберігаються всі медіафайли, теми, плагіни, мовні пакети та інше, які використовуються на сайті. Вміст цього каталогу можна вільно редагувати, не опасаючись пошкодження працездатності CMS [18], [19].

5.3.2Звернення до програми

Для виклику застосунку слід набрати у адресній сторінці браузера «cirmixer.ua». Для керування роботою застосунку використовуються посилання та кнопки [18], [19].

5.3.3Вхідні й вихідні дані

Початковими даними програми є дані про користувача, що пройшов реєстрацію, дані про рівень гри гравців, а також інформація про доступні файли та атрибути гравців. Вихідними даними є: відображення на екрані користувача інформації про створені команди для участі в турнірах "мікс-капів".

5.3.4Виконання програми

Для запуску застосунку потрібно виконати наступні дії:

- запустити встановлений веббраузер;
- у адресному рядку набрати «cirmixer.ua» після чого з'явиться головна сторінка застосунку.

Для приймання участі в турнірі слід:

- запустити вебзастосунок;
- пройти просту реєстрацію, заповнивши відповідну форму;

- вибрати доступний у списку турнір та натиснути кнопку “Реєстрація”;
- очікувати достатньої кількості зареєстрованих гравців для формування команд.

5.3.5 Розширення функціональності вебзастосунку за допомогою модулів

User Registration. Даний плагін потрібен для роботи з акаунтами. Він дозволяє створювати та редагувати користувацькі акаунти, додає сторінку реєстрації та форми входу до свого облікового запису, а також дає можливість користувачам редагувати данні в профілі.

Цей плагін являється одним з ключових при створенні проекту (рис. 5.2).

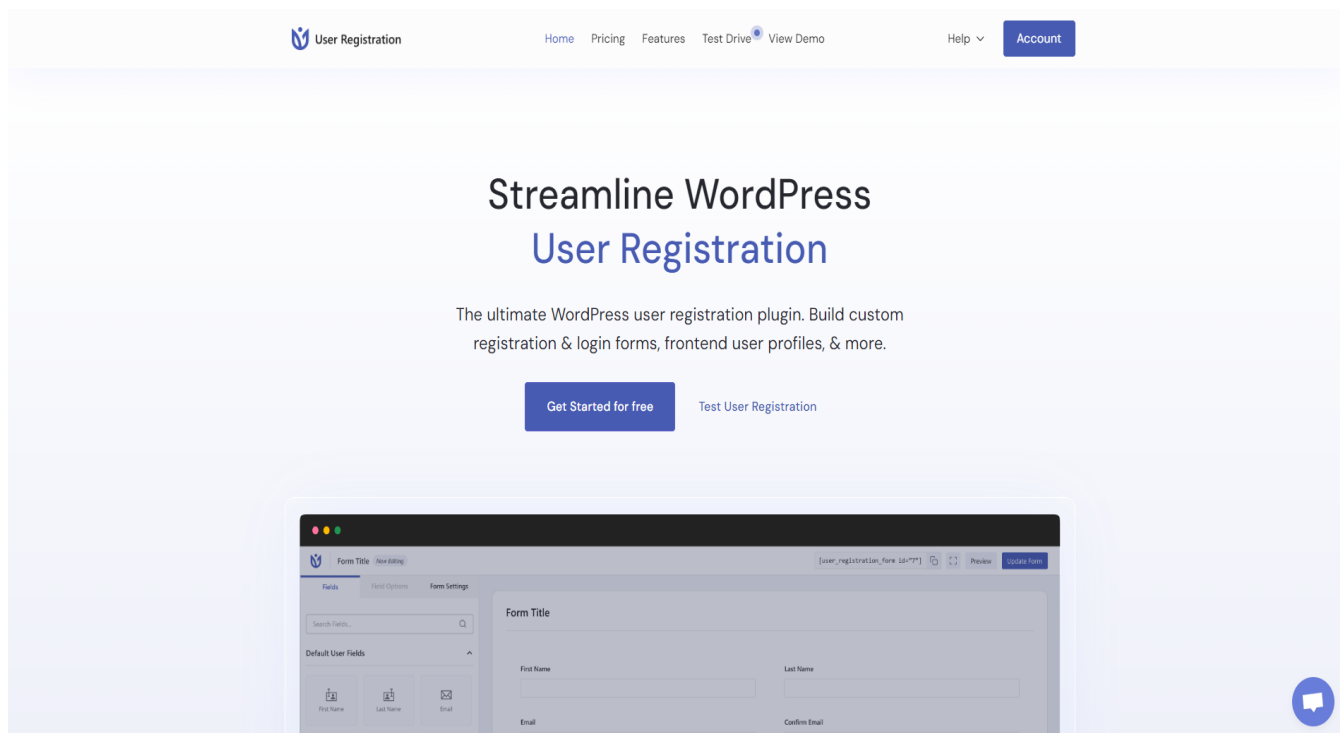


Рисунок 5.2 – User Registration

5.4 Методика та результати тестування

5.4.1 Аналіз реалізованих функціональних вимог

В ході реалізації програмного продукту, було реалізовано наступні функції, передбачені в попередніх планах розробки:

- список турнірів;
- можливість реєстрації для нових користувачів;
- можливість зміни мови на сайті;
- можливість зміни даних в профілі;
- перегляд сформованих команд, готових для участі в турнірі.

5.4.2 Ергономічність вебзастосунку

Існують п'ять основних критеріїв оцінки ергономічності вебзастосунку: лаконічність, чіткість, швидкість, взаємодія, доступність.

Представлений вебзастосунок у достатній мірі відповідає цим критеріям, оскільки являється простим і інтуїтивно зрозумілим.

Приклади основних сторінок застосунку продемонстровані на рис. 5.3 – 5.6.

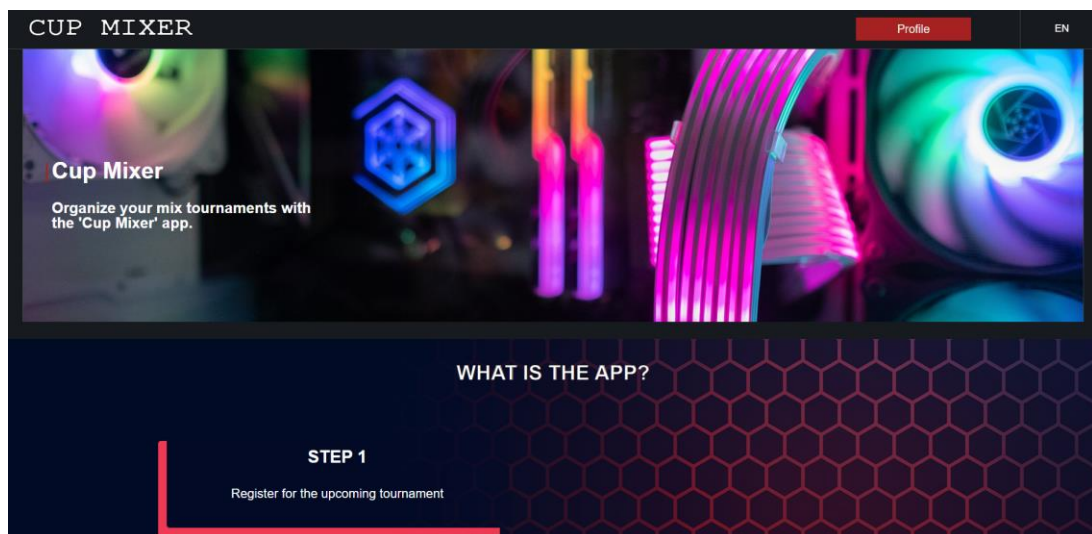


Рисунок 5.3 – Головна сторінка

Сайт поділено на три частини: верхня панель, яка дозволяє користувачам проходити реєстрацію або авторизацію, основна частина, в якій знаходяться опис сайту та список турнірів, та нижня частина, в якій присутні соціальні мережі.

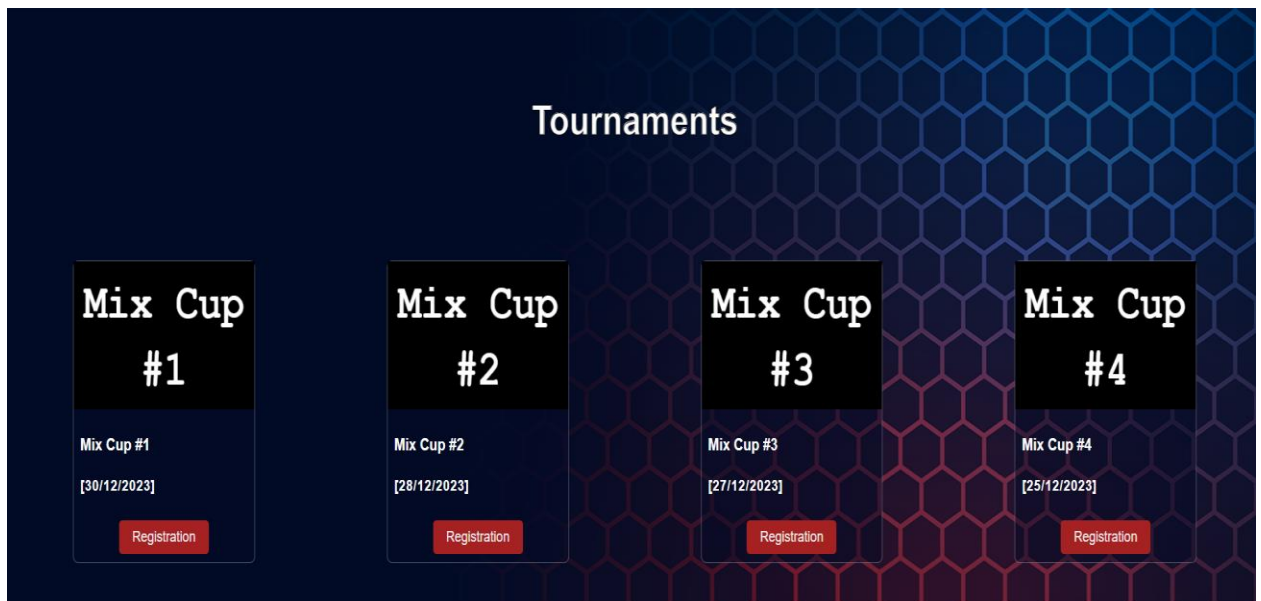
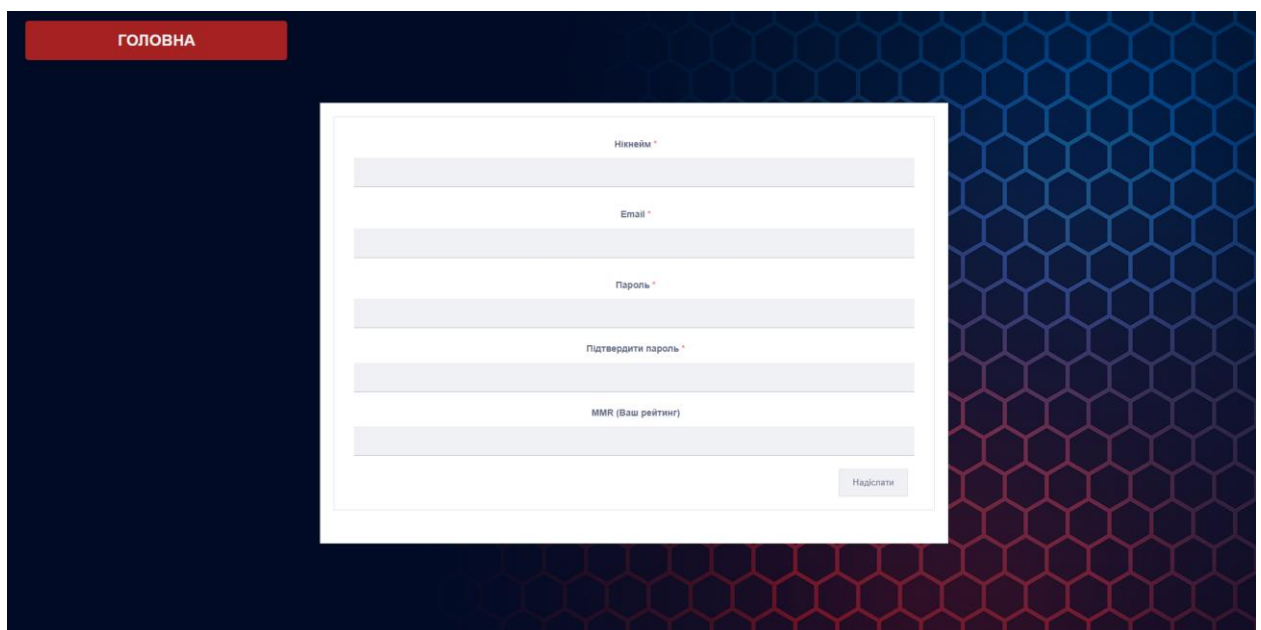


Рисунок 5.4 – Список турнірів



Риунок 5.5 – Форма реєстрації користувача

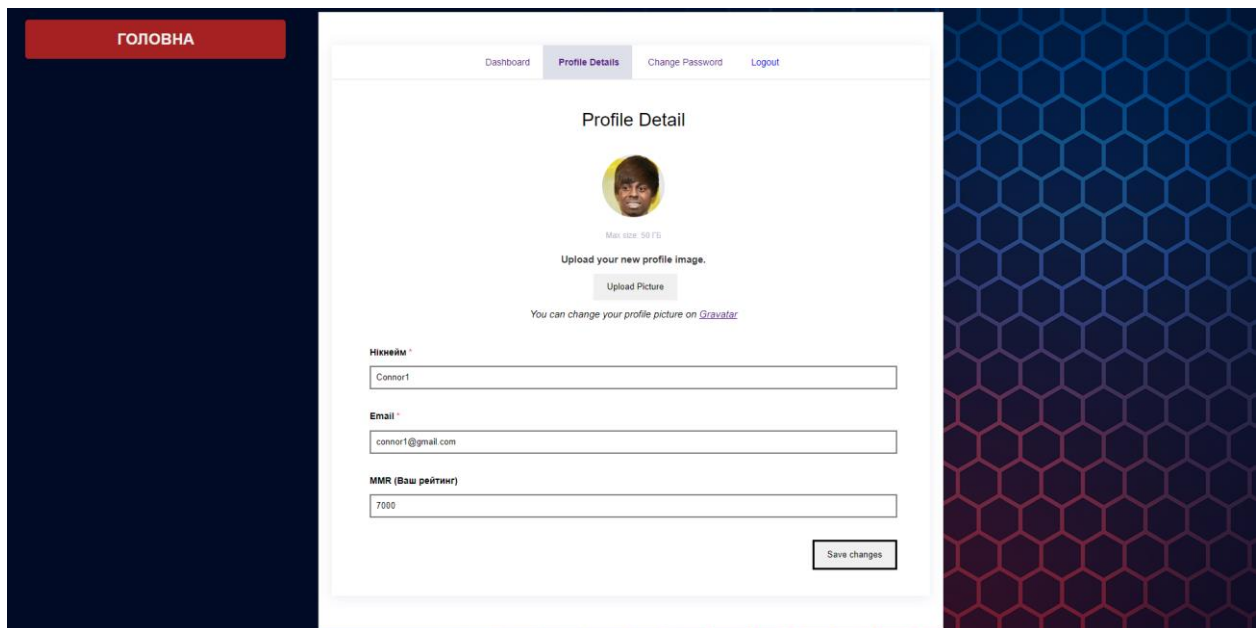


Рисунок 5.6 – Профіль зареєстрованого користувача

5.4.3 Тестування безпеки вебзастосунку

Існують три основні вимоги до безпеки вебзастосунків:

- конфіденційність – забезпечення захисту певних ресурсів від доступу певної групи користувачів. Застосовуючи приклад з консоллю WordPress, можна стверджувати, що лише адміністратори мають доступ до них, що відповідає цьому критерію;
- цілісність – дозволяє змінювати дані сайту лише певній групі осіб (адміністраторам), забезпечуючи неможливість втручання важливих даних для незареєстрованих і зареєстрованих користувачів;
- доступність – забезпечення доступу до ресурсів авторизованому користувачеві, внутрішньому об'єкту або пристрою.

Даний проєкт відповідає всім вищезазначеним критеріям безпеки.

5.4.4 Тестування стабільності

Даний вебзастосунок повністю відповідає всім критеріям стабільності:

- взаємодія між внутрішніми модулями;

- коректне відображення сторінок та їх елементів;
- деструктивне тестування.

ВИСНОВКИ

В ході виконання дипломної кваліфікаційної роботи магістра було розроблено та успішно реалізовано вебзастосунок, спрямований на автоматизоване формування та оптимізацію команд для турнірів з гри Dota 2. Застосування аналізу індивідуальних рейтингів гравців дозволило ефективно створювати балансовані команди, забезпечуючи справедливі та конкурентні умови для учасників турнірів. Для цього було виконано наступні завдання:

- проаналізовано проблемну організацію та проведення кіберспортивних турнірів у форматі “мікс кап”.
- досліджено існуючі аналогічні рішення;
- спроектовано архітектуру вебзастосунку;
- розроблено вебзастосунок;
- проведено тестування.

Розроблена структура вебзастосунку була реалізована із використанням CMS Wordpress.

Були використані модулі розширення функціональності.

Був розроблений сценарій та проведено тестування вебзастосунку з використанням різноманітних напрямів та сценаріїв. Одержані результати свідчать про успішність тестування. В ході виконання завдання була створена система, яка повністю відповідає вимогам, сформульованим на етапі початкового проектування та закріпленим у технічному завданні. Розроблена система є легкою у використанні та виконує всі необхідні функції.

Усі завдання дипломної кваліфікаційної роботи магістра повністю виконано.

ПЕРЕЛІК ДЖЕРЕЛ ПОСИЛАННЯ

1. Імаса Є. В. Кіберспорт: монографія [Текст] / Є. В. Імаса, О. В. Борисова, О. А. Шинкарук. – Київ : Олімпійська література, 2021. – 616 с.
2. Особливості прогнозування результатів матчів у кіберспорті [Електронний ресурс] / М. В.Коробчинський, Л. Б. Чирун, В. А. Висоцька, М. О. Ніч. – 2017. – Режим доступу: http://ric.zntu.edu.ua/article/view/115504/pdf_76.
3. Sergeev S. Cybersport within non-classical ergonomics of immersive and interactive environments [Електронний ресурс] / S. Sergeev, Burmistrov. – 2019. – Режим доступу: <https://iopscience.iop.org/article/10.1088/1755-1315/337/1/012051/pdf>.
4. Платформа Epulze [Електронний ресурс]. – Режим доступу: <https://epulze.com/>.
5. Сервіс Faceit [Електронний ресурс]. – Режим доступу: <https://www.faceit.com/>.
6. Сервіс Challonge [Електронний ресурс]. – Режим доступу: <https://challonge.com/>.
7. Knowledge Mapping and Sustainable Development of eSports Research: A Bibliometric and Visualized Analysis [Електронний ресурс] / C.Weisheng, C. Thomas, N. Sang-Back, S. Ping-Hung. – 2021. – Режим доступу: <https://www.mdpi.com/2071-1050/13/18/10354>.
8. Nalli J. Effect of esports and future development of esports [Електронний ресурс] / Joni Nalli // Business Information Technology Oulu University of Applied Sciences. – 2019. – Режим доступу: https://www.theseus.fi/bitstream/handle/10024/168883/Nalli_Joni.pdf.pdf?sequence=2.
9. Новотарський М. А. Алгоритми та методи обчислень [Електронний ресурс] / М. А. Новотарський. – 2019. – Режим доступу:

https://ela.kpi.ua/bitstream/123456789/27864/1/Alhorytmy_ta_metody_obchislenn.pdf.

10. Коваль В. С. Алгоритми і структури даних [Електронний ресурс] / В. С. Коваль, П. Р. Струбицький. – 2017. – Режим доступу: <http://dspace.wunu.edu.ua/bitstream/316497/41016/1/%D0%BD%D0%B0%D0%B2%D1%87.%D0%BF%D0%BE%D1%81%D1%96%D0%B1.pdf>.

11. Usage statistics of server-side programming languages for websites [Електронний ресурс] – Режим доступу: https://w3techs.com/technologies/overview/programming_language.

12. Meloni J.C. Sams teach yourself PHP, MySQL and Apache: all in one [Текст] / Julie C. Meloni. – 5th ed. – Indianapolis, Indiana : Sams, 2012. – 653 с.

13. Самойленко Д. М. Створення безпечних шаруватих структур мовою PHP/Secure layers construction on PHP [Текст] / Д. М. Самойленко // Вісн. Нац. ун-ту "Львів. політехніка". – 2016. – № 852. – С. 93-98.

14. Як стати PHP-розробником. План дій для початківців [Електронний ресурс] – Режим доступу: <https://dou.ua/lenta/articles/how-to-learn-php/>.

15. Михалюк Н. Основні функції та можливості системи керування вмістом вебсайту WordPress [Текст] / Н. Михалюк // Науковий вісник Східноєвропейського національного університету імені Лесі Українки. Міжнародні відносини. – 2018. – № 5. – С. 80-85.

16. Sabin-Wilson L. WordPress For Dummies [Текст] / Lisa Sabin-Wilson. – Needham : IDG Books, 2014. – 432 с.

17. Bartlett D. WordPress in Easy Steps [Текст] / Darryl Bartlett. – Royal Leamington Spa : In Easy Steps, 2019. – 192 с.

18. Багнюк Н. В. Аналіз та порівняння можливостей розробки вебресурсів засобами PHP-фреймворку Laravel і CMS WordPress [Текст] / Н. В. Багнюк, В. М. Мельник, О. В. Казмірчук // Комп'ютерно-інтегровані технології: освіта, наука, виробництво. – 2017. – № 26. – С. 10-15.

19. Мешечко С. С. Методы и способы защиты CMS WordPress [Текст] / С. С. Мешечко, В. Я. Певнев, В. А. Погорелов // Системи управління, навігації та зв'язку. – 2017. – Вип. 1. – С. 23-25.
20. Буч Г. Введення в UML від творців мови [Текст] / Г. Буч, Д. Рамбо, И. Якобсон. – Москва : ДМК Пресс, 2010. – 496 с.
21. Гудзовата О. О. Використання уніфікованої мови візуального моделювання UML (Unified Modeling Language) як інструменту підтримки проектування інформаційних систем [Текст] / О. О. Гудзовата, В. І. Костирко, І. В. Артищук // Підприємництво і торгівля. – 2019. – Вип. 24. – С. 108-114.

ДОДАТОК А
Технічне завдання

Вступ

Застосунок, що розроблюється, має назву «Cup Mixer». У час активного розвитку кіберспорту є актуальна проблема його доступності для переважної більшості користувачів і тому виникає проблема, що більшість таких гравців не мають можливості брати участь в кіберспортивних турнірах.

Вебзастосунок «Cup Mixer» призначений для швидкого та зручного формування команд зі списку індивідуальних гравців на основі їхнього рейтингу.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені OpenServer, PHP5 та MySQL.

A.1 Підстави для розробки

Розробка виконується на основі завдання на випускню дипломну роботу магістра, яке було визначене за темою «Дослідження та розробка за стосунку для автоматичного формування турнірних команд за рейтингом гравців» і затверджене наказом № 437 від 14 листопада 2023 р. за Національним університетом «Запорізька політехніка».

A.2 Призначення розробки

«Cup Mixer» можна віднести до категорії «Вебсервісів». Він створений для автоматичного змішування гравців по командам на основі їхнього рівня гри з метою зручного доступу до них з будь-якого девайсу, що має доступ до мережі Інтернет.

A.3 Основні вимоги до програми, що розробляється

A.3.1 Вимоги до функціональних характеристик

Система має підтримувати:

- реєстрація користувачів;
- можливість змінити даних користувачем;
- динамічне редагування турнірів;
- автоматичне формування команд з зареєстрованих гравців.

A.3.2 Вимоги до надійності

Система повинна ініціюватися з сервера та гарантувати збереження інформації навіть у випадку виникнення помилок, а також забезпечувати захист від несанкціонованого доступу на всіх етапах збору, обробки та зберігання. Це досягається шляхом регулярного створення резервних копій бази даних і використання системи паролів для керування доступом до інформації.

Для забезпечення конфіденційності особистих даних необхідно, щоб кожен користувач мав власний логін і пароль.

A.3.3 Умови експлуатації

Для користування застосунком, користувач повинен мати девайс з доступом до мережі Інтернет, обладнаний браузером. Для створення власного аккаунта користувач має зареєструватися.

A.3.4 Вимоги до складу та параметрів технічних засобів

Мінімальні системні вимоги та програмне забезпечення, що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна станція;

- встановлена та налагоджена серверна ОС – будь-яка з різновидів Unix або Windows Server;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,4GHz;
- обсяг ОЗУ – від 4GB (для безперебійної роботи нових версій браузерів);
- підключення до мережі Internet;
- встановлений браузер.

A.3.6 Вимоги до маркування та упакування

Перша версія матиме маркування 1.0, наступні версії матимуть маркування 1.1, 2.0 і т.д.

A.3.7 Вимоги до транспортування та збереження

Допускається транспортування програмного виробу в транспортній тарі усіма видами транспорту. При транспортуванні та зберіганні програмного виробу має бути передбачений захист від попадання пилу і атмосферних опадів.

A.3.8 Вимоги до програмної документації

До складу програмної документації повинні входити:

- документ опису вимог;

- технічне завдання;
- опис програми.

A.4 Стадії та етапи розробки

Процес розробки проекту складається з чотирьох етапів:

- аналіз даних. Він включає в себе два під етапи: збір та аналіз даних і складання технічного завдання;
- підготовка до розробки. Він включає в себе такі підетапи: розробка архітектури проекту та підбирання готових компонентів-модулів для системи;
- розробка програмного продукту;
- фінальне тестування системи та передача замовнику готового застосунку з усіма необхідними документами.

A.5 Порядок контролю та приймання системи

Для контролю та приймання системи застосовується широко використовувана методологія керування проектами SCRUM. Згідно з цією методологією, весь проект розбивається на модулі. При завершенні розробки кожного модуля проект представляє собою завершену та повноцінно працюючу систему.

Після повного розроблення та тестування програмного забезпечення його передають замовнику на флеш-носіях. Два архіви з розширенням zip включають в себе наступне: перший містить файли готового проекту та експортовану базу даних, а другий - всю необхідну документацію.

ДОДАТОК Б
Опис програми

Б.1 Загальні відомості

Застосунок, що розроблюється, має назву «Cup Mixer». У час активного розвитку кіберспорту є актуальна проблема його доступності для переважної більшості користувачів і тому виникає проблема, що більшість таких гравців не мають можливості брати участь в кіберспортивних турнірах.

Вебзастосунок «Cup Mixer» призначений для швидкого та зручного формування команд зі списку індивідуальних гравців на основі їхнього рейтингу.

Головною умовою для функціонування системи є наявність сервера, на якому встановлені OpenServer, PHP5 та MySQL.

Б.2 Функціональне призначення

Система має підтримувати:

- реєстрація користувачів;
- можливість змінити даних користувачем;
- динамічне редагування турнірів;
- автоматичне формування команд з зареєстрованих гравців.

Б.3 Опис логічної структури

На рисунку Б.1 наведено структурну схему ПЗ, що розробляється.

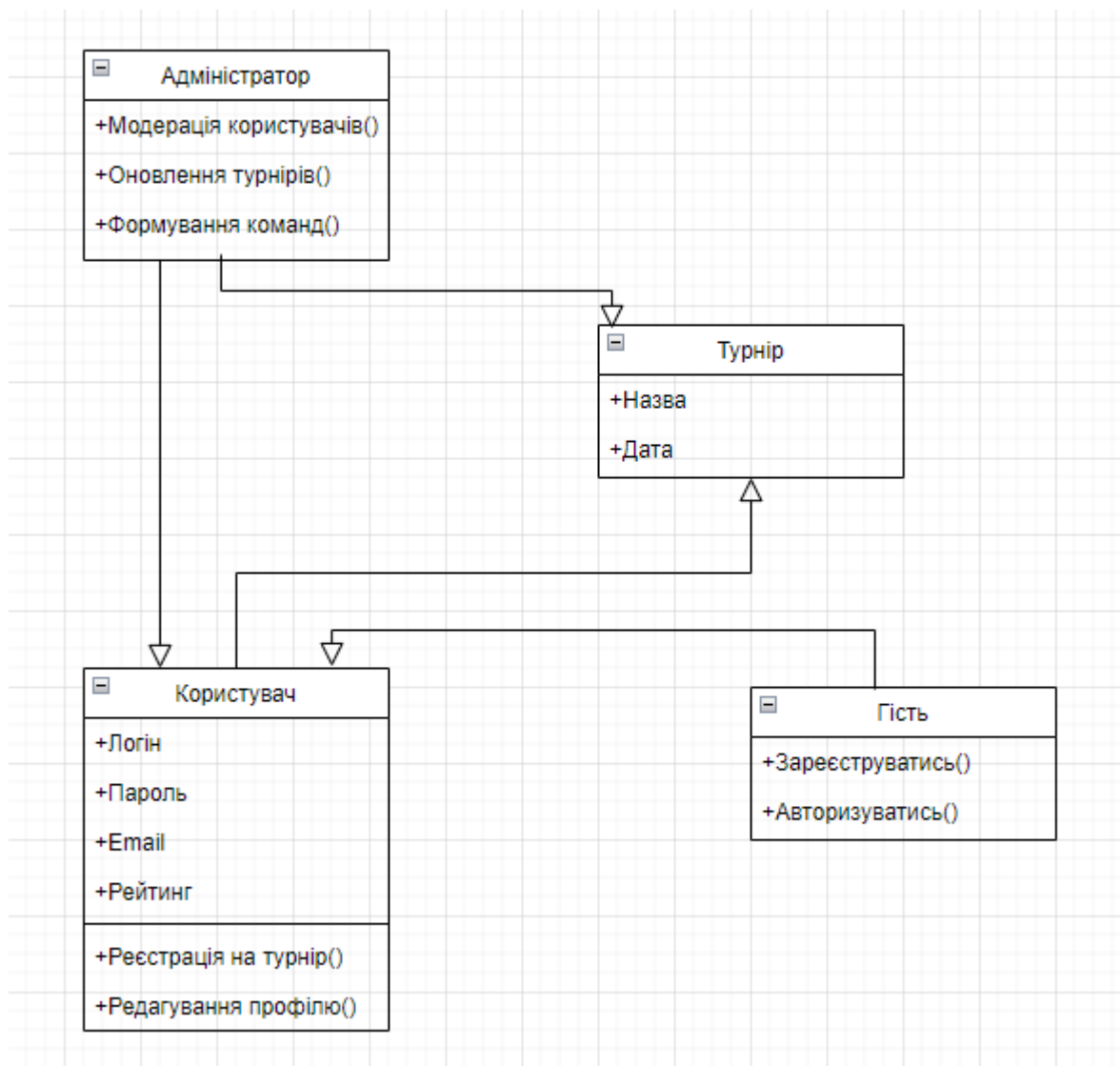


Рисунок Б.1 – Загальна схема системи за обраною темою

WordPress – це система управління вмістом з відкритим вихідним кодом, яка відзначається легкістю встановлення та використання. Широко застосовується для створення різних вебсайтів, від блогів до складних вебпроектів. Вбудована система тем і плагінів, разом із добре продуманою архітектурою, дозволяє конструювати різноманітні вебпроекти на основі WordPress.

Написана на мові програмування PHP з використанням бази даних MySQL. Сирцевий код поширюється на умовах ліцензії GNU General Public License.

Структура вебзастосунків, створених на WordPress зображена на рисунку Б.2.

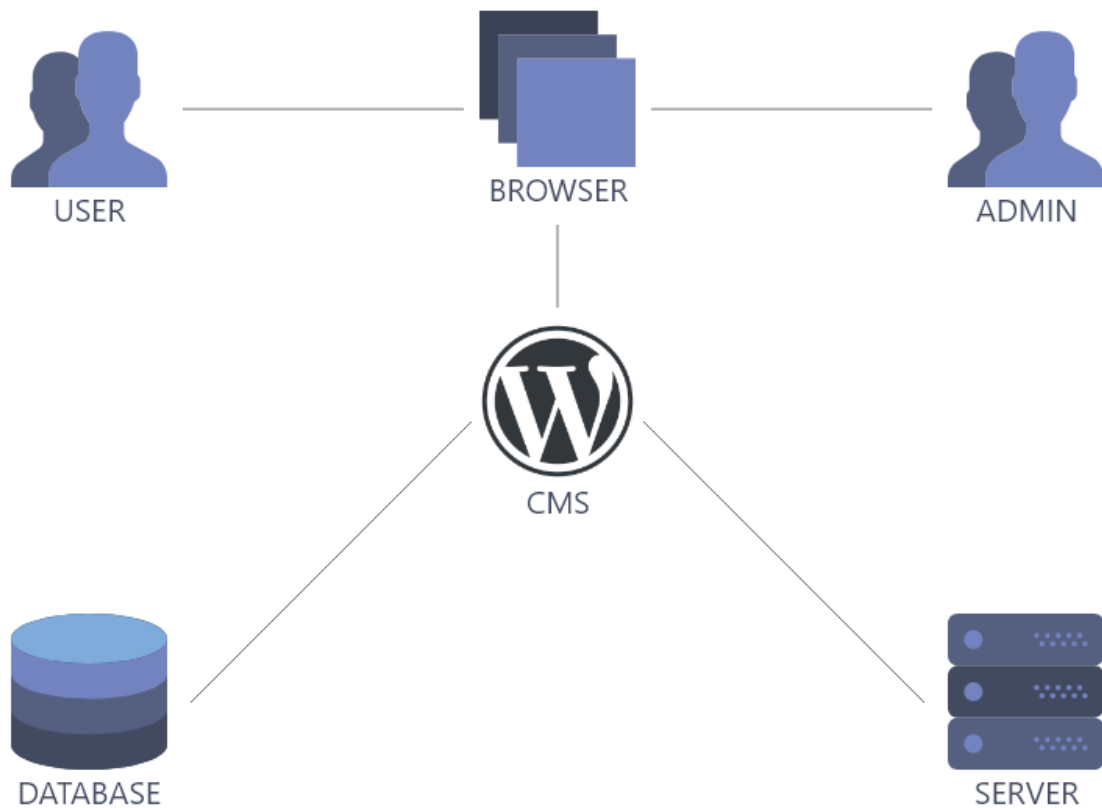


Рисунок Б.2 – Архітектура вебзастосунку

Б.4 Використані технічні засоби

Мінімальні системні вимоги та програмне забезпечення , що необхідні для функціонування серверної частини мобільного застосунку:

- IBM-сумісна серверна станція;
- встановлена та налагоджена серверна ОС – будь-яка з різновидів Unix або Windows Server;
- інтерпретатор PHP5;
- MySQL-сервер для керування користувачами та організації доступу до БД.

Для експлуатації клієнтської частини необхідні такі програмно-технічні засоби:

- IBM-сумісний ПК;
- процесор із тактовою частотою від 1,4GHz;
- обсяг ОЗУ – від 4GB (для безперебійної роботи нових версій браузерів);
- підключення до мережі Internet;
- встановлений браузер.

Б.5 Виклик та завантаження

Для виклику застосунку слід набрати у адресній сторінці браузера «сиртіхер.ua». Для керування роботою застосунку використовуються посилання та кнопки.

Б.6 Вхідні та вихідні дані

Початковими даними програми є дані про користувача, що пройшов реєстрацію, дані про рівень гри гравців, а також інформація про доступні файли та атрибути гравців. Вихідними даними є: відображення на екрані користувача інформації про створені команди для участі в турнірах "мікс-капів".

ДОДАТОК В
Текст програми

index.php:

```

    <?php get_header(); ?>
    <div class="intro">
        <div class="container__inner">
            <div class="left-block">
                </div>
            <div class="main-photo">
                <!---->
                <div class="main-image">
                    <div class="image-caption">
                        <div class="caption-text">
                            <p class="caption-big">Cup Mixer</p>
                            <p class="caption-small">Організуйте свої мікс-турніри за
допомогою додатку "Cup Mixer".</p>
                        </div>
                        <!-- <button class="caption-button" data-target="#registration-
modal">Зареєструватись</button> -->
                    <?php
global $current_user;
get_currentuserinfo();
if (!is_user_logged_in()) {
    $registration_page_url = get_permalink(63);
    echo '<a href="" . $registration_page_url . "" class="caption-
button">Зареєструватись</a>';
} else {
}
?>
        </div>
    </div>

```

```

</div>

<div class="right-block">
</div>

<div class="under-photo">
</div>
</div>

<div class="tournaments-block">
<div class="text-block full-width">
<h2 class="block-title">В чому суть додатку?</h2>
<div class="text-content">
<div class="text-left">
<div class="info-block">
<div class="block-rectangle">
<h3 class="steps_top1">КРОК 1</h3>
<p class="steps_bottom1">Реєструєтесь на майбутній турнір</p>
</div>
</div>
<div class="info-block">
<div class="block-rectangle">
<h3 class="steps_top3">КРОК 3</h3>
<p class="steps_bottom3">Чекаєте, поки додаток автоматично
сформує вам команду</p>
</div>
</div>
</div>
<div class="text-right">
<div class="info-block">
<div class="block-rectangle">
<h3 class="steps_top2">КРОК 2</h3>

```

```

        <p class="steps_bottom2">Під час реєстрації вказуєте свій рейтинг
та контактні данні</p>
    </div>
</div>
</div>
</div>
</div>
    <div class="tournaments-title">
    <h1 class="tournaments__title">Турніри</h1>
    </div>
    <div class="tournaments">
<?php
global $post;
$myposts = get_posts([
    'numberposts' => -1,
]);
if( $myposts ){
    foreach( $myposts as $post ){
        setup_postdata( $post );
    ?>
    <div class="group-1">
        <?php the_post_thumbnail( array(300,180), array('class' => 'img-title-
1') ); ?>
        <div class="tour-1">
            <p class="tour-title11"> <?php the_title(); ?> </p>
            <p class="tour-title1"> <?php the_content(); ?> </p>
            <button class="tour-title111 active">Реєстрація</button>
        </div>
    </div>
<?php

```

```

    }
}
wp_reset_postdata();
?>

</div>

<?php get_footer(); ?>
functions.php:
<?php
function clean_content($content) {
    return wp_strip_all_tags($content, true);
}
add_filter('the_content', 'clean_content');
add_action('wp_enqueue_scripts', function () {
    wp_enqueue_style('style', get_template_directory_uri() . '/assets/css/style.css');
    wp_enqueue_script('jquery');
    wp_enqueue_script('switchlang', get_template_directory_uri() .
'/assets/js/switchlang.js', array(), null, true);
    wp_enqueue_script('modal', get_template_directory_uri() . '/assets/js/modal.js',
array('jquery'), null, true);
});
add_action('wp_ajax_get_user_registration_my_account',
'get_user_registration_my_account');
add_action('wp_ajax_nopriv_get_user_registration_my_account',
'get_user_registration_my_account');
function ur_redirect_after_login( $redirect, $user ) {
    return home_url('/');
}
add_filter( 'user_registration_login_redirect', 'ur_redirect_after_login', 10, 2 );
add_theme_support('post-thumbnails');
add_theme_support('title-tag');

```

```
add_theme_support('custom-logo');
```

```
?>
```

```
modal.js:
```

```
jQuery(function ($) {
```

```
    $(document).on('click', '.nav__link__login, .caption-button', function (e) {
```

```
        e.preventDefault();
```

```
        var targetUrl = $(this).attr('href');
```

```
        if ($(this).hasClass('logged-in')) {
```

```
            window.location.href = targetUrl;
```

```
        } else {
```

```
            var targetModal = $(this).data('target');
```

```
            $(targetModal).show();
```

```
            $('#switch-form').show();
```

```
        }
```

```
    });
```

```
// Додайте обробник подій для кнопки "Реєстрація"
```

```
$('#loginButton').on('click', function (e) {
```

```
    e.preventDefault();
```

```
    var targetUrl = $(this).attr('data-target');
```

```
    window.location.href = targetUrl;
```

```
});
```

```
$('.caption-button').on('click', function (e) {
```

```
    e.preventDefault();
```

```
    var targetUrl = $(this).attr('href');
```

```
    window.location.href = targetUrl;
```

```
});
```

```
})
```

```
page-registration.php:
```

```
<?php
```

```
// Template Name: Registration
error_reporting(E_ALL);
ini_set('display_errors', 1);
?>
<!DOCTYPE html>
<html lang="en">
<head>
<style>
body{
    background-image:url('/wp-content/themes/Cup Mixer/assets/images/main-
photo.png');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
.home-button {
position:absolute;
width: 400px;
height: 60px;
background-color: #a62122;
outline: 0;
border: 0;
color: #fff0f0;
text-decoration: none;
margin-top: 20px;
font-size: 24px;
font-weight: bold;
text-transform: uppercase;
border-radius: 5px;
transition: all 0.4s;
```

```

display: flex;
align-items: center;
justify-content: center;
left: 12%;
transform: translateX(-50%);
}
.home-button:hover {
background: #d12e2f;
color: #fff;
}
.user-registration-form-container{
background: #fff;
}
</style>
<meta charset="<?php bloginfo('charset'); ?>">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Рєєстрація</title>
<?php wp_head(); ?>
</head>
<body>
<a href="http://cupmixer/" class="home-button">Головна</a>
<div class="user-registration-form-container">
<?php echo do_shortcode('[user_registration_form id="23"]'); ?>
</div>
</body>
<?php wp_footer(); ?>
</html>

```

page-login.php:

```
<?php
```

```
// Template Name: Login
error_reporting(E_ALL);
ini_set('display_errors', 1);
?>
<!DOCTYPE html>
<html lang="en">
<head>
<style>
body{
    background-image:url('/wp-content/themes/Cup Mixer/assets/images/main-
photo.png');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
.home-button {
position:absolute;
width: 400px;
height: 60px;
background-color: #a62122;
outline: 0;
border: 0;
color: #fff0f0;
text-decoration: none;
margin-top: 20px;
font-size: 24px;
font-weight: bold;
text-transform: uppercase;
border-radius: 5px;
transition: all 0.4s;
```

```

display: flex;
align-items: center;
justify-content: center;
left: 12%;
transform: translateX(-50%);
}
.home-button:hover {
background: #d12e2f;
color: #fff;
}
.user-login-form-container{
background: #fff;
}
</style>
<meta charset="<?php bloginfo('charset'); ?>">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Авторизація</title>
<?php wp_head(); ?>
</head>

<body>
<a href="http://cupmixer/" class="home-button">Головна</a>
<div class="user-login-form-container">
<?php echo do_shortcode('[user_registration_login
redirect_url="http://cupmixer/"]'); ?>
</div>
</body>
<?php wp_footer(); ?>
</html>

```

page-profile.php:

```
<?php
//Template Name: My Account
?>
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
<style>
body{
    background-image:url('/wp-content/themes/Cup Mixer/assets/images/main-
photo.png');
    background-size: cover;
    background-position: center;
    background-repeat: no-repeat;
}
.home-button {
position:absolute;
width: 400px;
height: 60px;
background-color: #a62122;
outline: 0;
border: 0;
color: #eff0f0;
text-decoration: none;
margin-top: 20px;
font-size: 24px;
font-weight: bold;
text-transform: uppercase;
border-radius: 5px;
transition: all 0.4s;
```

```

display: flex;
align-items: center;
justify-content: center;
left:12%;
transform: translateX(-50%);
}
.home-button:hover {
background: #d12e2f;
color: #fff;
}
.user-registration-form-container{
margin-top:100px;
background:#fff;
}
</style>
<meta charset="<?php bloginfo('charset'); ?>">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Профіль</title>
</head>
<body>
<header class="header-profile">
<?php wp_head(); ?>
</header>
<a href="http://cupmixer/" class="home-button">Головна</a>
<div class="user-registration-form-container">
<?php echo do_shortcode('[user_registration_my_account]'); ?>
</div>
</body>
<?php wp_footer(); ?>
</html>

```

header.php:

```
<!DOCTYPE html>
<html <?php language_attributes(); ?>>
<head>
    <meta charset="<?php bloginfo('charset'); ?>">
    <title>Cup Mixer</title>
    <?php wp_head(); ?>
</head>
<body>
    <header class="header">
        <div class="container">
            <div class="header__inner">
                <div class="header__title">
                    
                </div>
                <div class="login_button">
                    <?php if (is_user_logged_in()) : ?>
                        <a id="profileButton" class="nav__link__profile logged-in" href="<?php echo
get_permalink(25); ?>">Профіль</a>
                    <?php else : ?>
                        <button id="loginButton" class="nav__link__login" href="#" data-
target="<?php echo get_permalink(30); ?>">Реєстрація</button>
                    <?php endif; ?>
                <div class="header__lang">
                    <div class="dropdown">
                        <button id="switchLanguage" class="dropdown-
toggle">UA</button>
                    </div>
```

```

        </div>
    </div>
</div>
</div>
</header>
<?php wp_footer(); ?>
</body>
</html>

```

footer.php:

```

<footer class="bottom">
<div class="bottom__info">
    <div class="bottom-underline">
        <p id="bottom-p" class="bottom_p">Copyright © 2023 Всі права
захищені</p>
    <nav class="bottom__nav">
        <a class="nav__link__bottom" href="#"></a>
        <a class="nav__link__bottom" href="#"></a>
        <a class="nav__link__bottom" href="#"></a>
        <a class="nav__link__bottom" href="#"></a>
    </nav>
</div>

```

```

</div>
</footer>
    <button class="scroll-to-top" onclick="scrollToTop()">Haropy</button>
</div>
</div>
<?php wp_footer(); ?>
    <script>
function scrollToTop() {
    window.scrollTo({ top: 0, behavior: "smooth" });
}
window.addEventListener("scroll", function () {
    var scrollButton = document.querySelector(".scroll-to-top");
    if (document.body.scrollTop > 20 || document.documentElement.scrollTop >
20) {
        scrollButton.style.display = "block";
    } else {
        scrollButton.style.display = "none";
    }
});
</script>
<script>
var adminBarHeight = document.getElementById('wpadminbar') ?
document.getElementById('wpadminbar').offsetHeight : 0;
document.addEventListener('DOMContentLoaded', function () {
    var header = document.querySelector('.header');
    if (header) {
        header.style.marginTop = adminBarHeight + 'px';
    }
});
</script>

```

```

<script>
    var adminBarHeight = document.getElementById('wpadminbar') ?
document.getElementById('wpadminbar').offsetHeight : 0;
    document.addEventListener('DOMContentLoaded', function () {
        var leftBlock = document.querySelector('.left-block');
        var rightBlock = document.querySelector('.right-block');
        if (leftBlock && rightBlock) {
            leftBlock.style.marginTop = adminBarHeight + 'px';
            rightBlock.style.marginTop = adminBarHeight + 'px';
        }
    });
</script>
</body>
</html>

```

switchlang.js:

```

const texts = {
    "en": {
        "nav__link__profile": "Profile",
        "tour-title111": "Registration",
        "tour-title222": "Registration",
        "tour-title333": "Registration",
        "tour-title444": "Registration",
        "bottom_p": "Copyright © 2023 All rights reserved",
        "tournaments__title": "Tournaments",
        "steps_bottom3": "You are waiting for the application to automatically form a
team for you",
        "steps_top3": "STEP 3",
        "steps_bottom2": "During registration, you provide your rating and contact
details",

```

```

"steps_top2": "STEP 2",
"steps_bottom1": "Register for the upcoming tournament",
"steps_top1": "STEP 1",
"block-title": "WHAT IS THE APP?",
"caption-small": "Organize your mix tournaments with the 'Cup Mixer' app.",
"nav__link__login": "Log in",
"dropdown-toggle": "EN",
"caption-button": "Register",
},
"ua": {
  "nav__link__profile": "Профіль",
  "tour-title111": "Реєстрація",
  "tour-title222": "Реєстрація",
  "tour-title333": "Реєстрація",
  "tour-title444": "Реєстрація",
  "bottom_p": "Copyright © 2023 Всі права захищені",
  "tournaments__title": "Турніри",
  "steps_bottom3": "Чекаєте, поки додаток автоматично сформує вам команду",
  "steps_top3": "КРОК 3",
  "steps_bottom2": "Під час реєстрації вказуєте свій рейтинг та контактні данні",
  "steps_top2": "КРОК 2",
  "steps_bottom1": "Реєструєтесь на майбутній турнір",
  "steps_top1": "КРОК 1",
  "block-title": "В ЧОМУ СУТЬ ДОДАТКУ?",
  "caption-small": "Організуйте свої мікс-турніри за допомогою додатку 'Cup Mixer'.",
  "nav__link__login": "Увійти",
  "dropdown-toggle": "UA",

```

```

    "caption-button": "Зареєструватись",
  },
};

function switchLanguage() {
  const currentLanguage = localStorage.getItem("currentLanguage") || "ua";
  const newLanguage = currentLanguage === "ua" ? "en" : "ua";
  localStorage.setItem("currentLanguage", newLanguage);
  updateTexts(newLanguage);
}

function updateTexts(language) {
  console.log("Оновлення тексту з мовою:", language);
  for (const key in texts[language]) {
    const elements = document.querySelectorAll(`.${key}`);
    elements.forEach((element) => {
      element.textContent = texts[language][key];
    });
  }
}

document.addEventListener("DOMContentLoaded", function () {
  const currentLanguage = localStorage.getItem("currentLanguage") || "ua";
  console.log("Поточна мова при завантаженні сторінки:", currentLanguage);
  localStorage.setItem("currentLanguage", currentLanguage);
  updateTexts(currentLanguage);
});

document.querySelector(".dropdown-toggle").addEventListener("click",
switchLanguage);

```

ДОДАТОК Д
Слайди презентації

Міністерство освіти та науки України
Національний університет «Запорізька політехніка»

Дослідження та розробка застосунку для автоматичного формування турнірних команд за рейтингом гравців

Котов Олександр Олександрович
ст. групи КНТ-122м
Федорончак Тетяна Василівна
к.т.н., доцент НУ «Запорізька політехніка»

Рисунок Д.1 – Слайд 1

Об'єкт, предмет та мета роботи

Об'єкт дослідження - процес програмної реалізації застосунку для автоматичного формування турнірних команд на основі рейтингу гравців

Предмет дослідження - методи реалізації застосунку для автоматичного формування турнірних команд на основі рейтингу гравців

Метою роботи є програмна реалізація застосунку для автоматичного формування турнірних команд за рейтингом гравців

Рисунок Д.2 – Слайд 2

Постановка завдання

Метою проекту є реалізація застосунку для автоматичного формування турнірних команд за рейтингом гравців, що дозволить кожному приймати участь у змаганнях не залежно від команди та рівня гри.

Також в алгоритмі має бути передбачена варіативність складів команд та різноманітність ролей гравців.

Рисунок Д.3 – Слайд 3

Порівняння існуючих аналогів

Критерії порівняння	Epulze	Faceit	Challonge	Застосунок, що розробляється
Формат турнірів	Підтримує тільки формат 1x1	Підтримує турніри в форматі 5x5	Підтримує турніри в форматі 5x5	Підтримує турніри 5x5 в форматі "мікс кап"
Баланс команд	Випадковий	Присутній тільки для готових команд	Присутній тільки для готових команд	Балансує команди за середнім рівнем гри
Статистика та результати	Базова інформація	Розширений обсяг статистики	Базова інформація (можливість редагування адміністратором)	Відсутня
Рєстрація та участь	Проста реєстрація та участь в турнірах	Проста реєстрація та участь в турнірах (можливість управління турнірами)	Проста реєстрація та участь в турнірах (можливість управління турнірами)	Просто реєстрація та участь в турнірах

Рисунок Д.4 – Слайд 4

Порівняння мов програмування

Критерій	C# ASP.NET	Python	PHP
Складність синтаксису	+	++	+
Швидкість та продуктивність	+	±	++
Програмні бібліотеки фреймворків та CMS	±	±	++
Підтримка	+	++	++

Рисунок Д.5 – Слайд 5

Порівняння CMS

	WordPress	Joomla!	Drupal
Простота	++	+	±
Доступність модулів	++	+	++
Підтримка	++	++	±
Безпека	+	+	±

Рисунок Д.6 – Слайд 6

Загальна схема системи

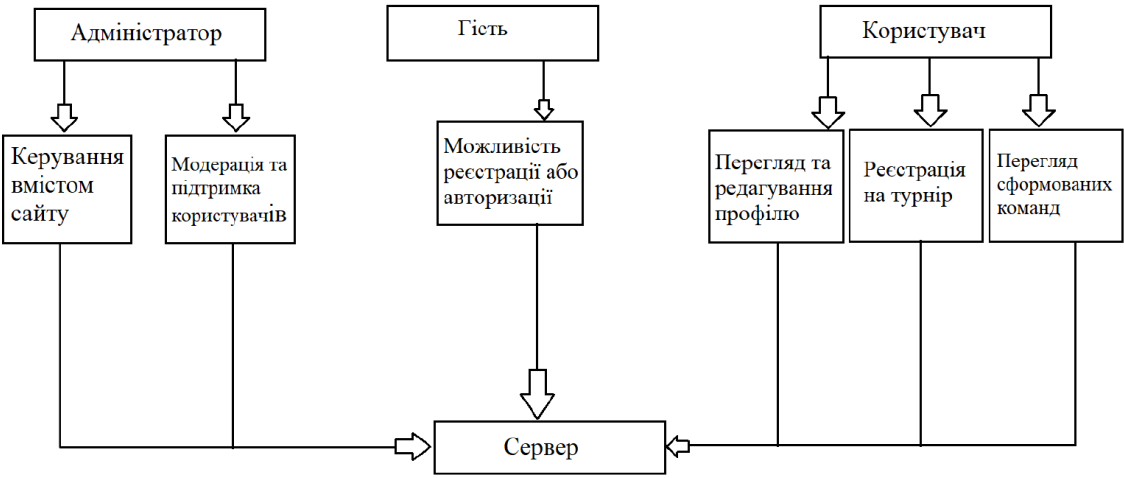


Рисунок Д.7 – Слайд 7

Проектування моделей системи

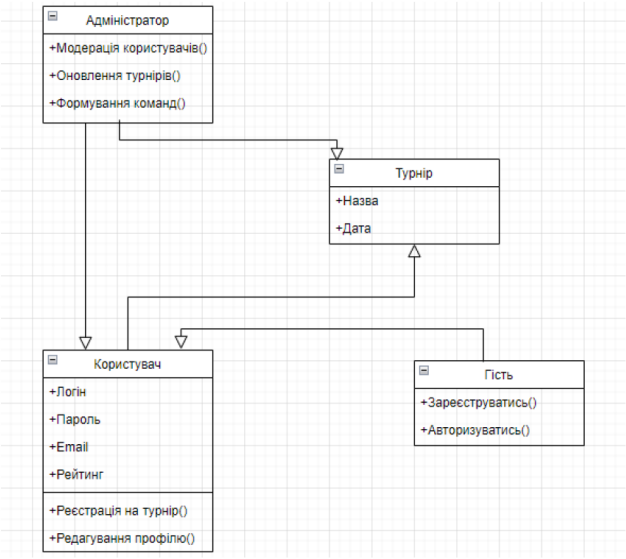


Рисунок Д.8 – Слайд 8

Логічне проектування взаємодії



Рисунок Д.9 – Слайд 9

Алгоритм сортування команд

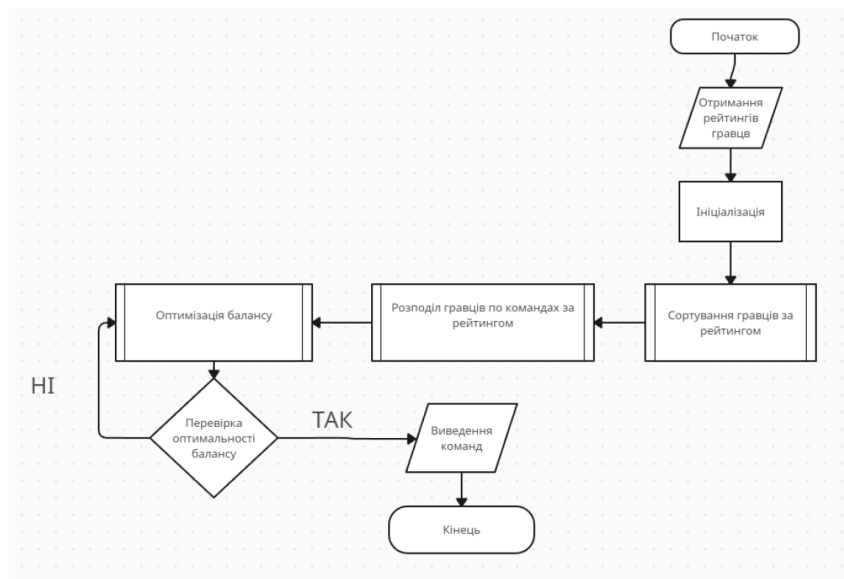


Рисунок Д.10 – Слайд 10

Демонстрація застосунку

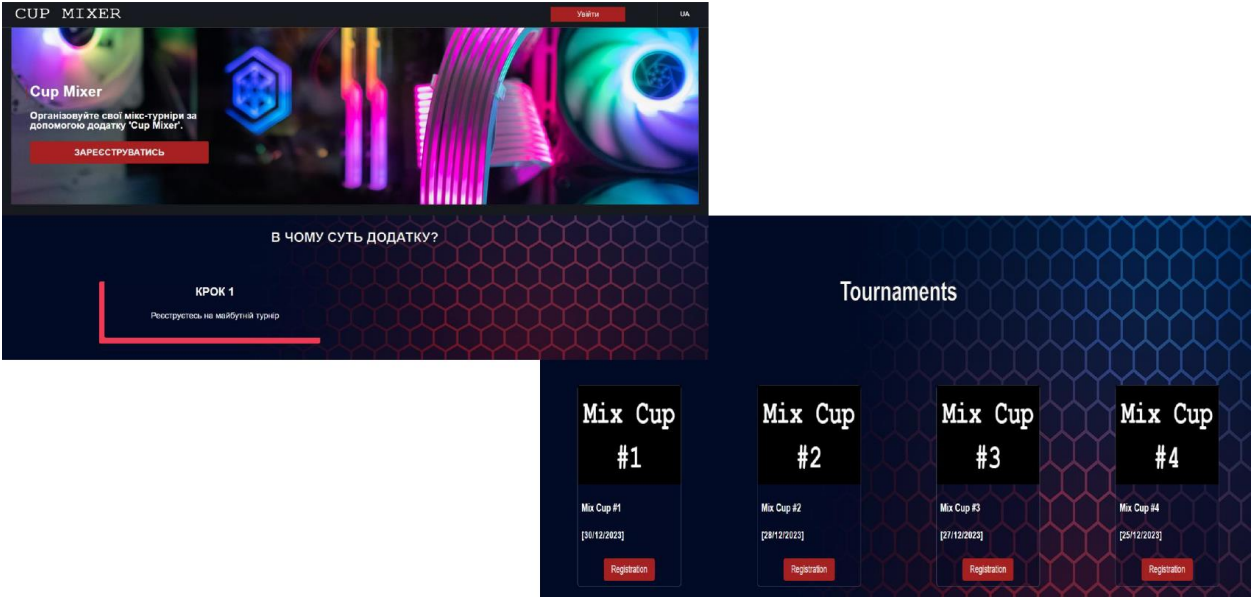


Рисунок Д.11 – Слайд 11

Демонстрація застосунку

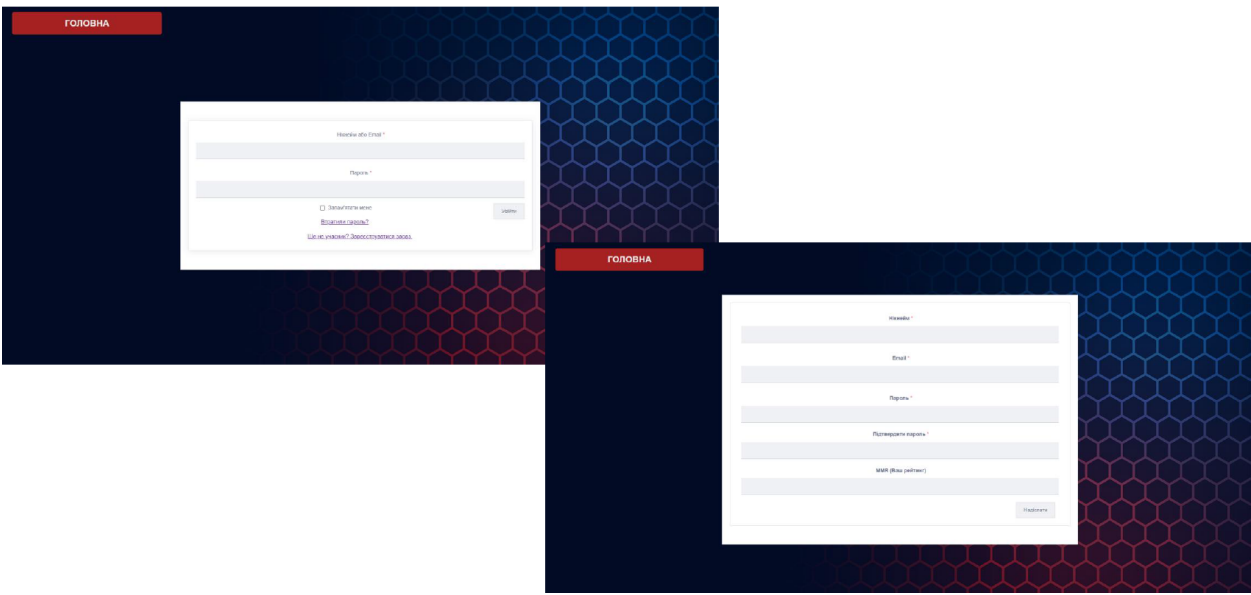


Рисунок Д.12 – Слайд 12

ВИСНОВКИ

Під час роботи було виконано наступні завдання:

- проаналізовано проблеми, пов'язані з процесом формування команд для турнірів формату "мікс кап";
- досліджено існуючі аналоги;
- спроектовано архітектуру вебзастосунку;
- впроваджено алгоритм для автоматичного створення збалансованих команд;
- розроблено вебзастосунок;
- проведено тестування.

В результаті виконання завдання було отримано застосунок, який відповідає всім вимогам, представленим в технічному завданні, зручний для використання та виконує всі необхідні функції. Усі завдання дипломної роботи магістра повністю виконано

Рисунок Д.13 – Слайд 13



Дякую за увагу

Рисунок Д.14 – Слайд 14