

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
НАВЧАЛЬНО–НАУКОВИЙ ІНСТИТУТ ІНФОРМАЦІЙНИХ ТЕХНОЛОГІЙ
Кафедра інженерії програмного забезпечення

Пояснювальна записка

до бакалаврської роботи
на ступінь вищої освіти бакалавр
на тему: «РОЗРОБКА КІБЕРСПОРТИВНОЇ ТУРНІРНОЇ WEB-
СИСТЕМИ НА ОСНОВІ MVC НА МОВІ ПРОГРАМУВАННЯ PHP»

Виконав: студент 5 курсу, групи ППЗ–52 спеціальності
121 Інженерія програмного забезпечення
(шифр і назва спеціальності)

_____ Пімкін В. Є.
(прізвище та ініціали)

Керівник _____ Залива В. В.
(прізвище та ініціали)

Рецензент _____
(прізвище та ініціали)

ДЕРЖАВНИЙ УНІВЕРСИТЕТ ТЕЛЕКОМУНІКАЦІЙ
Навчально-науковий інститут інформаційних технологій

Кафедра Інженерії програмного забезпечення

Ступінь вищої освіти - «Бакалавр»

Напрямок підготовки – 121 – Інженерія програмного забезпечення

ЗАТВЕРДЖУЮ

Завідувач кафедри

Інженерії програмного забезпечення

Негоденко О. В.

“ ___ ” _____ 2021 року

З А В Д А Н Н Я
НА БАКАЛАВРСЬКУ РОБОТУ СТУДЕНТУ

Пімкін Владислав Євгенійович

(прізвище, ім'я, по батькові)

1. Тема роботи: «Розробка кіберспортивної турнірної web-системи на основі mvc на мові програмування php»

Керівник роботи: асистент кафедри ІІЗ, аспірант Залива В. В.

(прізвище, ім'я, по батькові, науковий ступінь, вчене звання)

затверджені наказом вищого навчального закладу від 12 березня 2021 року, №65.

2. Строк подання студентом роботи 01 червня 2021 року.

3. Вихідні дані до роботи: мова програмування PHP, мова програмування для створення динамічного WEB - сайту

4. Зміст розрахунково-пояснювальної записки (перелік питань, які потрібно розробити)

4.1 Постановка задачі

4.2 Огляд аналогів і програмних засобів розробки WEB - сайтів

4.3 Проектування архітектури та розробка WEB - сайту

- 5. Перелік графічного матеріалу
- 5.1 Титульний слайд. _____
- 5.2 Об'єкт, предмет та мета дослідження. _____
- 5.3 Актуальність вибраної теми. _____
- 5.4 Автоматизація роботи фітнес-клубу. _____
- 5.5 Системи аналоги. _____
- 5.6 Постановка задачі. _____
- 5.7 Програмні засоби для вирішення поставленої задачі. _____
- 5.8 UML діаграма прецедентів. _____
- 5.9 Алгоритм роботи програми . _____
- 5.10 UML діаграма компонентів. _____
- 5.11 Схема бази даних. _____
- 5.12 Результати роботи ПЗ. _____
- 5.13 Висновки. _____

6. Дата видачі завдання _____ 19 квітня 2021 року _____

КАЛЕНДАРНИЙ ПЛАН

№ з/п	Назва етапів бакалаврської роботи	Строк виконання етапів роботи	Примітка
1	Підбір науково-технічної літератури	19.04.2021	Виконано
2	Вимоги до встановленого додатку	23.04.2021	Виконано
3	Оцінка якості тестування до систем	29.04.2021	Виконано
4	Концепція та архітектура додатку	02.05.2021	Виконано
5	Вступ, висновки, реферат	05.05.2021	Виконано
6	Розробка презентаційних матеріалів	06.05.2021	Виконано
7	Попередній захист роботи	11.05.2021	
8	Здача роботи	01.06.2021	

Студент _____ Пімкін В.Є.
(підпис) (прізвище та ініціали)

Керівник роботи _____ Залива В. В.
(підпис) (прізвище та ініціали)

РЕФЕРАТ

Пояснювальна записка до дипломної роботи: 55 с., 17 рис., 9 табл., 2 додаток, 20 джерел.

Об'єкт дослідження. Робота онлайн-змагань, як вид спортивної активності.

Предмет дослідження. Можливість автоматизації роботи онлайн-змагань та функції автоматизованих систем, які задовольняють вимоги до автоматизації роботи онлайн-змагань.

Метою роботи є розробка динамічного веб сайту на основі MVC, який надасть змогу користувачам слідкувати за новинами кіберспорту, приймати участь в турнірах та створювати власні, знаходити інших гравців та створювати власні команди.

Актуальність роботи полягає у тому, що кіберспортивна індустрія продовжує розвиватися і з кожним роком її прибуток на ринку збільшується, як в світі, так і, зокрема, в Україні. Таким чином, ідея створення WEB – сайту для створення турнірних ігор є перспективним напрямком.

Для досягнення даної мети мають бути вирішені наступні **завдання**:

- 1) здійснити постановку задачі;
- 2) виконати огляд аналогів і програмних засобів розробки динамічних WEB - сайтів;
- 3) спроектувати додаток;
- 4) реалізувати і скласти документацію (керівництво користувача) на додаток.

Стислий опис результатів дослідження: реалізовано веб-додаток та базу даних, які пов'язані у систему.

Ключові слова: КІБЕРСПОРТ, CS:GO, ТУРНІР, ЗМАГАННЯ, МАРІО, PHP, MVC.

ЗМІСТ

<i>Вступ</i>	16
<i>1. Загальна частина</i>	10
1.1 Постановка задачі	10
1.2 Дослідження і аналіз об'єкту програмування.....	11
1.3 Використані програмні засоби.....	16
1.4 Вимоги до апаратного та програмного забезпечення.....	18
1.4.1 Загальні відомості.....	18
1.4.2 Функціональне призначення.....	18
1.4.3 Технічні засоби, що використовуються	21
1.4.4 Вхідні та вихідні дані.....	21
<i>2 Практична частина</i>	22
2.1 Створення та налагодження програми.....	22
2.1.1 Розробка інтерфейсу	22
2.2 Опис програми та її алгоритмів	23
2.2.1 Опис компонентів системи	23
2.2.2 Аналіз ситуацій використання.....	26
2.2.3 Складання алгоритму	28
2.3 Інструкція програміста.....	30
2.3.1 Призначення і умови застосування	30
2.2.3 Діаграма взаємодії програмної системи.....	32
2.4 Інструкція оператора	34

<i>3. Організаційно - економічна частина</i>	<i>40</i>
3.1 Оцінка вартості програмного продукту	40
3.1.1 Розрахунок часу	40
3.1.2 Розрахунок заробітної плати виконавця робіт із створення програмного продукту	44
3.1.3 Розрахунок нарахувань на заробітну плату (єдиного соціального внеску).....	46
3.1.4 Розрахунок витрат на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту	46
3.2 Розрахунок собівартості програмного продукту	47
3.3 Розрахунок ціни програмного продукту	48
3.4 Зведена таблиця показників.....	48
<i>4. Безпека життєдіяльності</i>	<i>50</i>
4.1 Вимоги до приміщень і організації робочих місць	50
4.2 Вимоги охорони праці при експлуатації комп'ютерних систем.....	55
<i>Висновки</i>	<i>58</i>

ВСТУП

Стрімкий розвиток мережі Інтернет, застосування новітніх технологій і комунікацій у комерційній діяльності і повсякденному житті зумовило виникнення нових економічних явищ, таких як електронна комерція та інші види WEB платформ.

Одними із таких є кіберспортивні платформи. Кіберспорт – змагання з відеоігор – це вид спортивної активності, в якій гравці розвивають і тренують як розумові, так і фізичні навички. Молоді гравці формують команди та змагаються за призові фонди, що можуть досягати величезних сум. Річний дохід гравців при цьому може складати сотні тисяч доларів, враховуючи спонсорство та призові виграші, і тим паче призові фонди зі змагань. Розвиток технологій, таких як стрімінгові сервіси, дозволяє масовий перегляд змагань, і з більшим залученням спонсорів, інвестицій та сучасних цифрових можливостей, кіберспорт стає дедалі більш привабливим.

Метою даної дипломної роботи є створення web-сайту, з проведення онлайн-турнірів, який дозволить кожному, хто має доступ до інтернету та бажання увійти у світ турнірних змагань, знайти собі команду або зібрати свою для змагання з іншими.

Існують три основні функції, які, по можливості, повинен виконувати сайт:

1. Іміджева функція. Вона відповідає за формування образу власника сайту серед інтернет-користувачів. Головну роль при цьому відіграє оформлення ресурсу.

2. Маркетингова функція. Допомогає продавати товар або ж послуги, представлені на сайті.

3. Інформаційна функція. Полягає в тому, що інтернет-ресурс повинен надати відвідувачам максимум регулярно оновлюваної тематичної інформації про компанію та її діяльність.

1. ЗАГАЛЬНА ЧАСТИНА

1.1 Постановка задачі

Турнірний веб-сервіс призначений для полегшення створення турнірних змагань з різних дисциплін. Даний веб-сайт надаватиме змогу не тільки приєднатись до вже запланованого турніру, але й створити власний, що надає змогу розвивати проект у маркетинговому напрямку.

Метою дипломної роботи є розробка сайту для швидкого перегляду та вибору турніру, створення власного, підбору потрібних гравців до команди та перегляду інформації по світовим кіберспортивним змаганням.

Розроблюваний сайт, повинен відповідати таким вимогам:

- надавати потенційному користувачеві інформацію про вже створені турніри;
- наявність системи підбору гравців;
- надавати Потенційному користувачеві всю необхідну інформацію для створення власного турніру;
- мати зручну і швидко систему замовлення, реєстрації та пошуку потрібної інформації;
- можливість додавання, видалення і редагування інформації про турніри;
- можливість перегляду історії турнірів.

Також на сайті повинна бути розміщена повна інформація про компанію та всі необхідні контактні дані. Візуальний ряд повинен бути приємний для користування і відповідати стилістиці компанії, а структура сайту зручна і зрозуміла для користувача. Повинна бути передбачена оптимізація програмного коду в ході розробки сайту з метою подальшого просування сайту в пошукових системах.

Кіберспортивні ігри набувають все більш і більш широкого розмаху. І відповідно потрібні різні програмні засоби, які дозволяють зробити їх максимально ефективними.

Переваги систем онлайн-турнірів:

Інформація про послуги обертається швидше. Фактично, ви отримуєте додатковий комунікаційний канал, відкритий 24/7.

Ви доступніші для клієнта географічно і в часі, до того ж він отримує новий засіб пошуку і роботи з інформацією.

Швидше обертається внутрішня інформація (документи, службова переписка, прийняття та затвердження рішень і ін.). Розширена можливість для контролю виконання. Прискорюються всі бізнес-процеси за рахунок доступності та швидкості передачі інформації.

Інтернет пропонує нові сервіси для клієнтів - наприклад, відстеження просування замовлення. Наявність додаткових сервісів створює конкурентну перевагу і призводить до вас нових відвідувачів.

Оскільки Інтернет є технологічною оболонкою, він дозволяє збирати важливу інформацію про ваших клієнтів. Крім того, він дозволяє використовувати маркетинговий інструментарій - опитування, розсилки і т.п. оперативно і без додаткових витрат.

Таким чином, основними перевагами онлайн-турнірів є доступність для великої кількості клієнтів, географічно і в часі, широкі можливості реклами і маркетингу, економія власного часу користувача.

1.2 Дослідження і аналіз об'єкту програмування

Спортивний турнір - спортивне змагання з ігрових видів спорту з великим числом учасників (гравців або команд).

Турніри можуть проводитися як короткі заходи в певному місці, або як довгострокова система матчів, що триває протягом тривалого часу (сезон) і проводиться з роз'їздами команд.

Кіберспорт (також відомий як «комп'ютерний спорт» або «електронний спорт», англ. Esports) - командне або індивідуальне змагання на основі відеоігор.

Усі кіберспортівні дисципліни діляться на кілька основних класів, що розрізняються властивостями просторів, моделей, ігровий завданням і що розвиваються ігровими навичками кіберспортсменів: шутери від першої особи, стратегії в реальному часі, спортивні симулятори, автосимулятори, авіасимулятори, файтинг, командні рольові ігри з елементами тактико-стратегічної гри і т. д.

Розігруються призові фонди можуть досягати декількох мільйонів доларів США. Турнір по Dota 2 The International кілька разів бив рекорди по виплатах: так, в 2017 було розіграно \$ 25 млн, в 2018 - \$ 26 млн, в 2019 - \$ 34 млн. У 2020 чемпіонат був перенесений через пандемію коронавірусу .

Ігри турнірів транслюються в прямому ефірі в інтернеті, збираючи багатомільйонну аудиторію. Наприклад, за фіналом The International 2015 року, згідно з даними з TrackDota.com, спостерігало понад 4,6 млн глядачів, а в 2020 році глядачі провели на трансляціях матчів першого дивізіону російського турніру EPIC League понад 21 млн годин.

Історія кіберспорту почалася з гри Doom 2, яка мала режим мережевої гри через локальну обчислювальну мережу. Завдяки популярності гри Quake, в 1997 році в США з'явилася перша ліга кіберспортсменів - Cyberathlete Professional League.

2015 року світова аудиторія кіберспорту становила 226 мільйонів людей і планувалося, що 2019 року вона досягне 427 мільйонів.

До того як з'явилися офіційні кіберспортівні ліги, мали місце невеликі змагання в іграх, де проводився підрахунок очок. (Spacewar !, Space Invaders, Asteroids! І т. д.)

У 1996 році з'являється EVO (спочатку мав назву Battle by the Bay), який проводить турнір по грі Street Fighter II.

Пізніше, в 1997 році була заснована CPL, яка зробила перший турнір в дисципліні Quake. Але першою грою з можливістю кооперативної сутички була Doom 2.

З 2018 року розвитком кіберспорту в Україні займається Ukrainian esports Federation (Федерація кіберспорту України) — всеукраїнська громадська організація, створена для популяризації і розвитку комп'ютерного спорту в Україні., що здійснює свою діяльність на всій території України через 23 відокремлені підрозділи.

Діяльність Федерації націлена на розвиток кіберспорту на території України та залучення до цього спортивного напрямку якомога більше молоді, та всіх, кого може зацікавити цей молодий для України вид спортивної активності.

За час свого існування Федерацією було проведено більше 250 турнірів, як національних так і міжнародних, в яких прийняли участь більше 35 000 гравців, а також реалізовано низку соціальних та освітніх проектів.

7 вересня 2020 року стало знаковою датою - завдяки Федерації Кіберспорту України UESF кіберспорт було офіційно визнано видом спорту в Україні.

У кіберспорті не можна використовувати будь-які ігри. Наприклад, для змагань не підійдуть гри, які містять в собі елемент випадковості, що переважає над фактором вміння гри.

Таким чином в категорію кіберспортивних дисциплін часто потрапляють гри жанрів MOBA або RTS.

Ось декілька прикладів основних кібердисциплін:

Бойова арена - розрахована на багато користувачів онлайн
бойова арена - дисципліна комп'ютерного спорту, в якій 2 команди гравців борються один з одним на мапі особливого виду з метою знищення головної будівлі команди суперника. Кожен гравець управляє одним зі списку доступних героїв, що відрізняються

характеристиками і здібностями. Протягом матчу герої можуть ставати сильнішими, отримувати нові здібності і спорядження;

Змагальні головоломки - спортивна дисципліна комп'ютерного спорту, що представляє собою рішення логічних задач учасниками змагань, результат якого залежить від швидкості логічного мислення і кмітливості, що обумовлюють вибір тієї чи іншої стратегії гри для досягнення перемоги;

Спортивний симулятор (симулятор спортивної гри) - спортивна дисципліна комп'ютерного спорту, відтворює за допомогою відеоігри спортивну гру на арені за правилами виду спорту, визнаного в установленому порядку;

Стратегія в реальному часі - спортивна дисципліна комп'ютерного спорту, в якій протиборчі сторони учасників змагань на арені в реальному часі позиціонують і маневрують ігровими персонажами, для захисту районів карти і / або знищення активів своїх суперників. В ході гри можуть створюватися додаткові ігрові персонажі і поліпшуватися властивості вже наявних;

Технічний симулятор - спортивна дисципліна комп'ютерного спорту, в якій учасники змагань, імітуючи фізична поведінка і управління технічними засобами, досягають перемоги відповідно до Технічних правил дисципліни. В результаті змагальної діяльності учасник змагань отримує навички управління реальними технічними засобами (наприклад: танковий симулятор, авіаційний симулятор, автомобільний симулятор);

Розглянемо декілька кіберспортивних сайтів:

Сайт «cspl.ru», який зображено на рисунку 1.1.

З лівої частини ми можемо спостерігати меню для входу, а також можливість змінити дисципліну, переглянути список турнірів або власну команду. Посередині розташований список турнірів(їх назва та правила). Правіше від списку є фільтр для більш гнучкого підбору потрібного вам турніру. Даний сайт має простий інтерфейс, який надає змогу швидко орієнтуватися у його структурі та зробити роботу з ним якомога простіше та швидше. Окрім цього, даний сайт має рейтингову систему гравців, які приймають участь у турнірах.

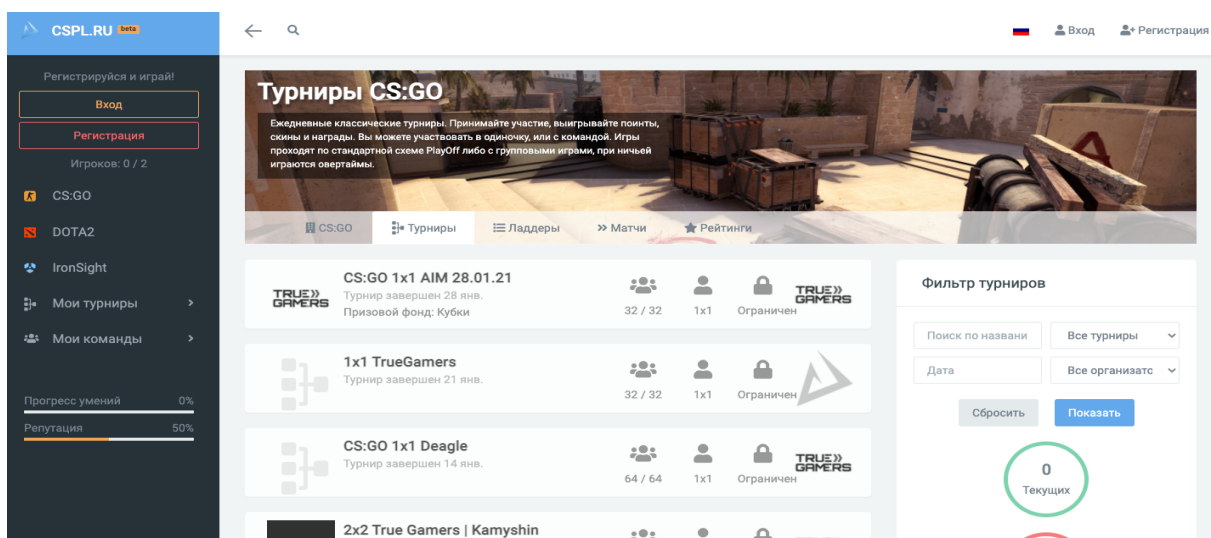


Рисунок 1.1 - сайт «cspl.ru»

Сайт «csgo.starladder.com», який зображено на рисунку 1.2.

Зверху розташоване головне меню, на головній сторінці доступна інформація по останнім новинам кіберспорту, список проведених та майбутніх матчів, їх призові. Є можливість переглянути відеозаписи завершених матчів.

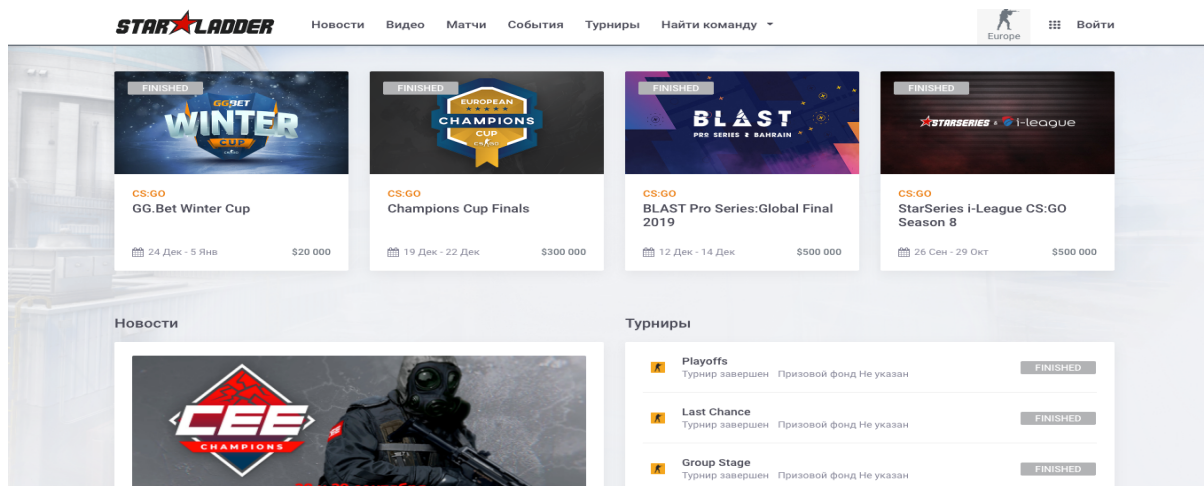


Рисунок 2.2 - сайт «csgo.starladder.com»

Для того, щоб сайт видавав інформацію залежно від запитів, надавав можливості пошуку тощо - на сервер встановлюється підтримка скриптів

(наприклад PHP, Perl). У більш комплексному варіанті, програма інтернет-сервіс — це система управління вмістом сайту, яка вже має підтримку скриптів, надає можливість в онлайн-режимі (головним чином через інтернет) і в межах наявного асортименту надавати пропонований сервіс.

В результаті дослідження предметної області, розгляду систем-аналогів та можливостей веб-програмування, було прийнято рішення розробити Web-орієнтовану систему по проведенню кіберспортивних матчів, турнірів, можливістю реєстрації користувачів, для створення власних турнірів і можливістю створення та пошуку команди для колективної гри.

1.3 Використані програмні засоби

PHP — скриптова мова програмування, була створена для генерації HTML-сторінок на стороні веб-сервера. PHP є однією з найпоширеніших мов, що використовуються у сфері веб-розробок (разом із Java, .NET, Perl, Python, Ruby). PHP підтримується переважною більшістю хостинг-провайдерів. PHP — проект відкритого програмного забезпечення.

PHP інтерпретується веб-сервером у HTML-код, який передається на сторону клієнта. На відміну від скриптової мови JavaScript, користувач не бачить PHP-коду, бо браузер отримує готовий html-код. Це є перевага з точки зору безпеки, але погіршує інтерактивність сторінок. Але ніхто не забороняє використовувати PHP для генерування JavaScript-кодів, які виконуються вже на стороні клієнта.

PHP — мова, код якої можна вбудовувати безпосередньо html-код сторінок, які у свою чергу, будуть коректно оброблені PHP-інтерпретатором. Обробник PHP просто починає виконувати код після відкриваючого тегу (`<?php`) і продовжує виконання до того моменту, поки не зустрине закриваючий тег (`?>`).

Велика різноманітність функцій PHP дає можливість уникати написання багаторядкових функцій, призначених для користувача, як це відбувається в C або Pascal.

HTML — стандартна мова розмітки веб-сторінок в Інтернеті. Більшість веб-сторінок створюються за допомогою мови HTML (або XHTML). Документ HTML оброблюється браузером та відтворюється на екрані у звичному для людини вигляді.

HTML є похідною мовою від SGML, успадкувавши від неї визначення типу документа та ідеологію структурної розмітки тексту.

Попри те, що HTML — штучна комп'ютерна мова, вона не є мовою програмування.

HTML разом із каскадними таблицями стилів та вбудованими скриптами — це три основні технології побудови веб-сторінок.

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для опису сторінок, написаних мовами розмітки даних.

Найчастіше CSS використовують для візуальної презентації сторінок, написаних HTML та XHTML, але формат CSS може застосовуватися до інших видів XML-документів.

CSS (каскадна або блочна верстка) прийшла на заміну табличній верстці веб-сторінок. Головна перевага блочної верстки — розділення змісту сторінки (даних) та їхньої візуальної презентації.

PhpStorm — комерційне крос-платформове інтегроване середовище розробки для PHP, яке розробляється компанією JetBrains на основі платформи IntelliJ IDEA.

PhpStorm являє собою інтелектуальний редактор для PHP, HTML і JavaScript з можливостями аналізу коду на льоту, запобігання помилок у сирцевому коді і автоматизованими засобами рефакторинга для PHP і JavaScript. Автодоповнення коду в PhpStorm підтримує специфікацію PHP 5.3/5.4/5.5/5.6/7.0/7.1 (сучасні і традиційні проекти), включаючи генератори, співпрограми, простори імен, замикання, типажі і синтаксис коротких масивів.

Присутній повноцінний SQL-редактор з можливістю редагування отриманих результатів запитів.

PhpStorm розроблений на основі платформи IntelliJ IDEA, написаної на Java. Користувачі можуть розширити функціональність середовища розробки за рахунок установки плагінів, розроблених для платформи IntelliJ, або написавши власні плагіни.

Веб-сервер (Open Server). Сервер, що приймає HTTP-запити від клієнтів, зазвичай веб-браузерів, видає їм HTTP-відповіді, зазвичай разом з HTML-сторінкою, зображенням, файлом, медіа-потокком або іншими даними. Веб-сервер — основа Всесвітньої павутини.

Веб-сервером називають як програмне забезпечення, що виконує функції веб-сервера, так і комп'ютер, на якому це програмне забезпечення працює.

Клієнти дістаються веб-сервера за URL-адресою потрібної їм веб-сторінки або іншого ресурсу.

1.4 Вимоги до апаратного та програмного забезпечення

1.4.1 Загальні відомості

Для коректної роботи Web-орієнтованої системи «Tournament» необхідна стабільна робота операційної системи типу Windows (XP/Vista/Seven), та встановлений браузер (Google Chrome, Safari, Internet Explorer, Edge, Mozilla Firefox, Opera та ін.)

1.4.2 Функціональне призначення

Мета дипломної роботи – створення Інтернет-сайту для користувачів.

Інтернет сайт – має бути реалізована на такій мові, яка легко пов'язує два типи файлів. Для Інтернет сторінок найбільше підходить мова PHP. Для вироблення вимог до проекту, необхідно уявляти собі потреби майбутніх відвідувачів даного сайту і те, яким чином сайт повинен задовольняти ці потреби.

Можна виділити два основних моменти в проведенні об'єктно-орієнтованого аналізу. Перший – це коротке, загальне формулювання вимог до програми з точки зору кінцевого користувача. Другий момент – аналіз ситуацій використання, де розглядаються всі можливі цілі й дії користувача при використанні програми і те, яким чином вона повинна реагувати на кожну з можливих ситуацій.

Вимоги до створення Інтернет-сайту такі: створити сторінку користувача на якій розмістити основні новини зі світу кіберспортивних турнірів і їх детальний опис, зробити сторінку зі списком активних турнірів, зробити власний кабінет користувача та налаштування його для заповнення потрібними даними, які потребуватимуться під час реєстрації на турнір, налаштувати сторінку на роботу.

В задачу побудови сайту входить створення реляційної бази даних у середовищі MySQL Server, створення сайту мовою html, з використанням технології css та мовами php.

Система повинна отримувати на вхід наступну інформацію від адміністратора:

- контактну інформацію;
- наповнення інформацією сайту новинами;
- підтвердження інформації про сворені турніри;
- підтвердження інформації про склад команд.

Вихідні дані, що подаються адміністратору:

- відображення контактної інформації;
- відображення списку команд;
- відображення списку турнірів;
- відображення інформаційних повідомлень.

Система повинна отримувати на вхід наступну інформацію від користувача:

- основну інформацію для заповнення профілю.

Вихідні дані, що подаються користувачу:

- відображення контактної інформації;
- відображення списку команд;

- відображення списку турнірів;
- відображення інформаційних повідомлень.

Користувачі системи:

- користувачі інтернет мережі.

Стежити за відповідністю текстів на сторінках сайту запитаю потенційних клієнтів.

За допомогою пошуку статистики запитів по регіонах можна подивитися популярність пошукового запиту в місті чи регіоні.

Вибрати кілька ключових фраз, якими можна охарактеризувати послуги які надає сайт. Сторінка повинна давати кращу в інтернеті відповідь на питання користувача.

Вибір колірної гами грає важливу роль при створенні сайту. Від неї безпосередньо залежить його сприйняття відвідувачами: і сприйняття сайту в цілому, і психологічний і фізіологічний стан людини, і зручність читання. Колір також впливає на настрій і самопочуття.

Тому при створенні сайту дуже важливо враховувати:

- особливості фірмового стилю, на підставі якого розробляється сайт;
- фізіологічні і психологічні особливості сприйняття колірної гами людьми.

Оформлення сайту має відповідати його тематиці. Крім відповідності тематиці сайту, колірна гамма повинна бути побудована так, щоб не порушувати основні закономірності впливу кольорів на психологічну реакцію людини, повинна відповідати стійким гармонійним поєднанням кольорів. Звичайно, сприйняття кольорів в цілому суб'єктивно, залежить від зовнішніх обставин. Існують деякі закономірності, які варто враховувати при розробці web-сайтів.

Вибираючи колірне рішення, важливо не забувати наступне:

- кольори повинні відповідати фірмовому стилю компанії;
- комбінація кольорів залежить від цільової аудиторії даного сайту;
- кольори повинні гармоніювати між собою.

Важливим аспектом також являє колір тексту. Він не обов'язково повинен бути чорним, але повинен гармоніювати з основною колірною гамою і бути добре читаним.

Головний критерій - гармонійне поєднання кольорів.

1.4.3 Технічні засоби, що використовуються

Для запуску програми необхідним є використання на ПК ОС Microsoft Windows XP–Microsoft Windows 8.

Мінімальні системні вимоги:

- ОЗП 32 МБ;
- Відеокарта voodoo або atiradeon 9200;
- Наявність миші або Touch Pad.
- Наявність доступу до Ethernet
- Наявність браузера (Opera, Google Chrome, Explore, телефонні браузер – Opera mini).

Програма призначена для роботи в середовищі Windows 9x та вище і вимагає для інсталяції 9 Мбайт вільного простору на вашому жорсткому диску.

1.4.4 Вхідні та вихідні дані

Запуск мого Інтернет проекту здійснюється при завантаженні файлу www/tournament, після запуску вашого веб браузера Internet Explorer або ін.

Вхідні дані це шлях до ресурсу, який користувач хоче відкрити. Шлях може бути як локальний (ресурс знаходиться на комп'ютері користувача) так і глобальний (ресурс знаходиться в мережі, при цьому користувач повинен мати доступ до мережі). Це введенні користувачем дані про інформацію якого його цікавить. Збереження інформації по кожному клієнту сайту.

Вихідними даними є результати запиту, що виконується після введення ключа і обирання критерію при пошуку турнірів а також вихідними даними є отриманні дані при фільтруванні даних.

2. ПРАКТИЧНА ЧАСТИНА

2.1 Створення та налагодження програми

Для створення програми були використані такі основні технології MVC.

MVC. Модель-вигляд-контролер — архітектурний шаблон, який використовується під час проектування та розробки програмного забезпечення.

Цей шаблон передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування. Застосовується для відокремлення даних (моделі) від інтерфейсу користувача (вигляду) так, щоб зміни інтерфейсу користувача мінімально впливали на роботу з даними, а зміни в моделі даних могли здійснюватися без змін інтерфейсу користувача.

Мета шаблону — гнучкий дизайн програмного забезпечення, який повинен полегшувати подальші зміни чи розширення програм, а також надавати можливість повторного використання окремих компонентів програми. Крім того використання цього шаблону у великих системах сприяє впорядкованості їхньої структури і робить їх більш зрозумілими за рахунок зменшення складності.

2.1.1 Розробка інтерфейсу

Титульна сторінка Web-орієнтованої турнірної системи сервіс «Tournament» максимально інформативно і в стислому об'ємі відображає необхідну користувачеві інформацію про сайт. На головній сторінці є логотип веб-сайту, основне меню сайту (для навігації по його структурі), форма аутентифікації (входу зареєстрованих користувачів), реєстраційне посилання (реєстрація нових клієнтів).

Сайт забезпечує основні функції. Каталог дозволяє зробити вибір, використовуючи стандартні форми:

- тип турніру;
- рівень турніру.

Також існує можливість можливість реєстрації постійних клієнтів.

2.2 Опис програми та її алгоритмів

2.2.1 Опис компонентів системи

Розроблений сайт являє собою повноцінний інтернет - сайт, який створено засобами середовища та використовує реляційну базу даних MySQL

Схематично основні функції системи представлені на діаграмі прецедентів у термінах мови UML .

Функції системи представлені у вигляді діаграми прецедентів, на якій виділено користувача та ядро.

Функції які може виконувати користувач зображені на рисунку 2.1.

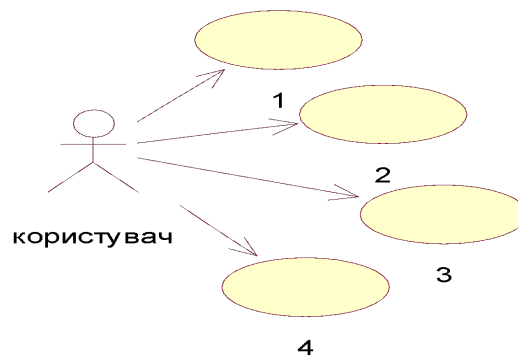


Рисунок 2.1 – Функції користувача

1. Переглядати списку турнірів.
2. Переходити на інформаційні сторінки про турнір.
3. Реєструватися на сайті.
4. Залишати заявки.

Функції, які виконує ядро зображені на рисунку 2.2.

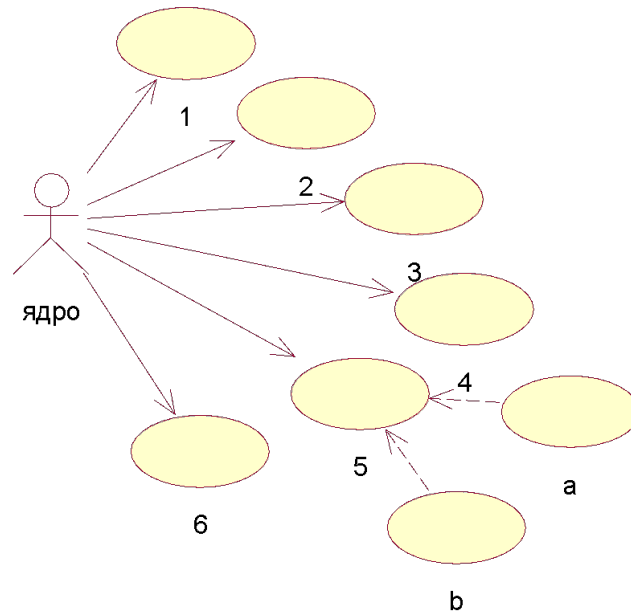


Рисунок 2.2 – Функції ядра

1. Запускає сайт та вимикає його.
2. Дає можливість переходу на інші сторінки через силки.
3. Можливість реєстрації.
4. Зберігає зареєстрованих користувачів.
5. Авторизація.
 - a. Ввод логіна .
 - b. Ввод пароля.
6. Дає їм можливість залишати коментарі в свою чергу зберігає їх.

Розроблений сайт являє собою повноцінний інтернет - сервіс, який створено засобами середовища та використовує реляційну базу даних MySQL

Схематично основні функції системи представлені на діаграмі прецедентів у термінах мови UML.

На діаграмі прецедентів представлено 3 виконавця.

Виконавець 1 це звичайний користувач, який переглядає сайт.

Виконавець 2 це зареєстрований користувач, який переглядає сайт та робить замовлення.

Виконавець 3 це інтернет-сервіс, він вносить зміни до основної інформації сайту, редагує списки, додає та видаляє користувачів, вносить зміни до інформації.

Діаграма прецедентів зображена на рисунку 2.3.

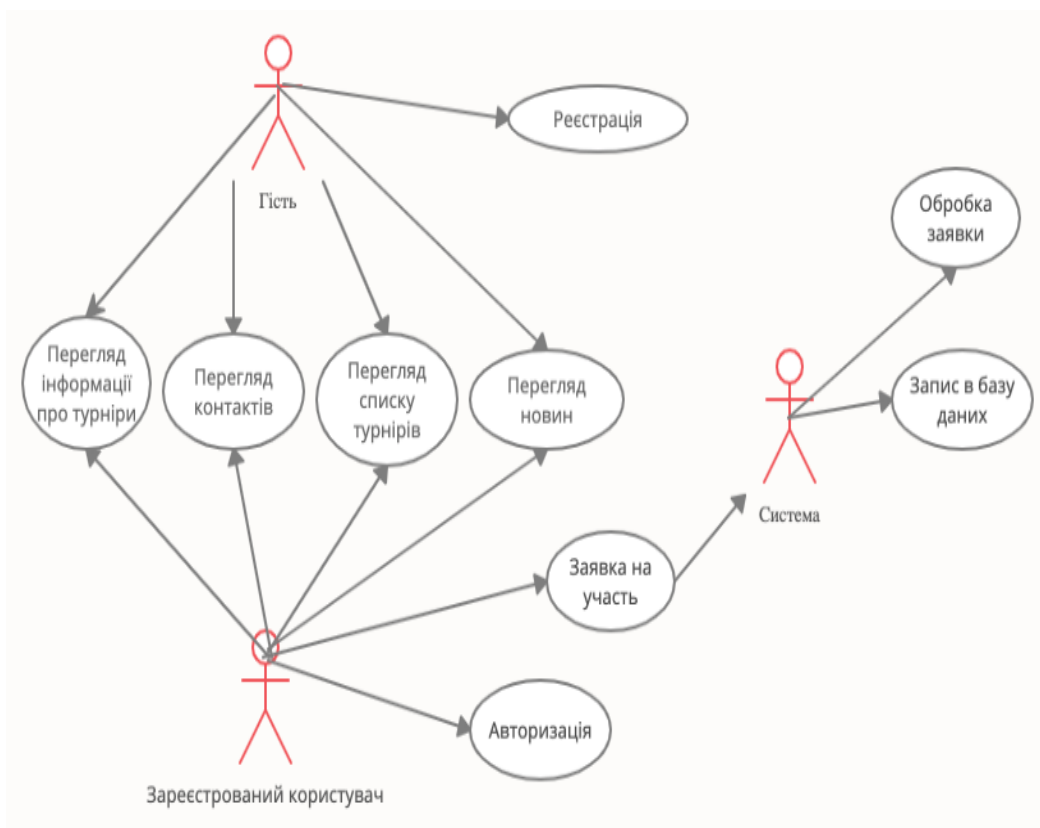


Рисунок 2.3 – UML – діаграма прецедентів

Потоки прецеденту Авторизація(вхід в систему).

Основний потік:

- 1 Користувач обирає увійти в систему
- 2 Прецедент виводить форму для входу.
- 3 Користувач вводить логін та пароль, якщо ввів невірний один з параметрів, то виникає помилка E1.

4 Прецедент робить перевірку чи зареєстрований такий користувач, якщо ні, то виникає альтернативний потік А1.

5 Прецедент дає доступ до системи.

Альтернативний потік А1:

1 Прецедент пропонує зареєструватись.

2 Користувач реєструється.

3 Прецедент робить перевірку чи зареєстрований такий користувач, якщо ні, то виникає альтернативний потік А1.

4 Прецедент дає доступ до системи.

Потік помилки Е1:

1 Користувач ввів невірний один з параметрів.

2 Прецедент виводить повідомлення про помилку.

3 Користувач повинен знову ввести дані.

2.2.2 Аналіз ситуацій використання

Користувач може подивитися Інтернет сторінку в текстовому форматі, а також в Інтернет – браузері.

Ситуація виникнення помилок при створенні Інтернет - сторінки: теоретично такої проблеми виникнути не може, швидше за все, це проблема знання HTML вона буде пов'язана з особливостями програмного забезпечення, а точніше – Інтернет - браузера, в якому можна подивитися сторінку.

Аналіз ситуацій використання допомагає краще зрозуміти, як повинна виглядати сторінка, і дозволяє вибрати найоптимальніші рішення під час програмування.

При розробці даного програмного сайту була створена реляційна база даних, модель цієї бази даних показана на рисунку 2.4.

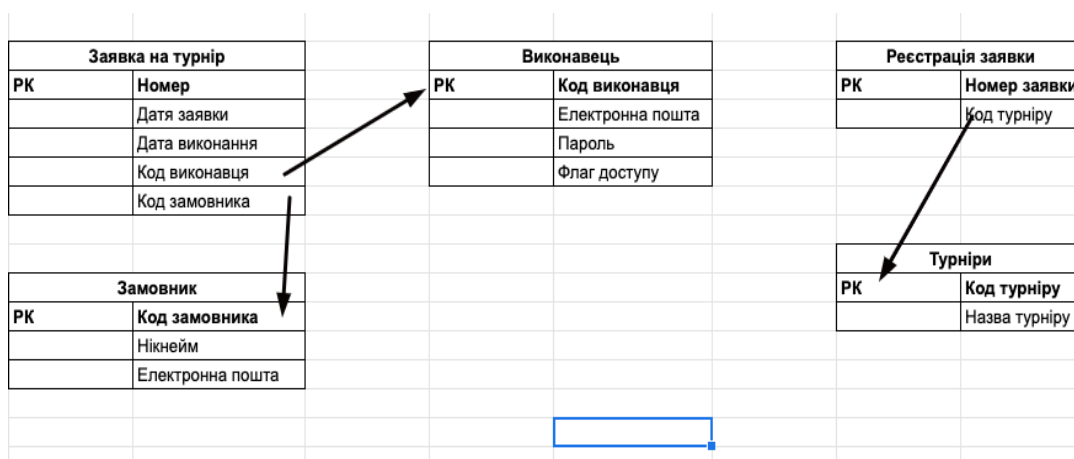


Рисунок 2.4 – Концептуальна модель бази даних

Таблиця 2.1 - «request», у ній зберігається інформація про дату замовлення, код виконавця, тощо

Назва	Опис	Тип	Розмір	Обмеження
Id	Код	integer		primary key
Date of request	Дата заявки	integer		primary key;
Date of implementation	Дата виконання	integer		primary key
code artist	Код виконавця	integer		primary key
customer numb	Код замовника	integer		primary key

Таблиця 2.2 - «Client» у ній зберігається інформація про клієнта

Назва	Опис	Тип	Розмір	Обмеження
Id	Код	integer		primary key
Nickname	Нікнейм клієнта	Текстовий	50	
Mail address	Електронна пошта	Текстовий	50	

Таблиця 2.3 - «Performer», у ній зберігається прізвище, пароль виконавця.

Назва	Опис	Тип	Розмір	Обмеження
Id	Код	integer		primary key
Mail address	Електронна пошта	Текстовий	50	
Password	Пароль	Текстовий	50	
Flag access	Флаг доступу	Текстовий	50	

Також оскільки веб-застосування містить форму авторизації, то у створену базу даних автоматично додаються таблиці, які необхідні для зберігання логіну та паролю користувачів, та вказання відповідних прав доступу кожного з користувачів.

2.2.3 Складання алгоритму

Для візуального зображення алгоритмів часто використовують блок-схеми.

Основними властивостями алгоритму є:

Дискретність (подільність) - алгоритм повинен представляти процес рішення задачі як послідовне виконання простих (чи раніше визначених) кроків.

Визначеність - кожне правило алгоритму повинен бути чітким і однозначним.

Результативність - алгоритм повинен приводити до рішення задачі за кінцеве число кроків.

Масовість - алгоритм рішення задачі розробляється в загальному вигляді, тобто, він повинен бути застосований для деякого класу задач, що розрізняються тільки вихідними даними.

Щоб алгоритм виконав своє призначення, його необхідно будувати за визначеними правилами.

Перше правило – при побудові алгоритму необхідно задати множину об'єктів, з якими буде працювати алгоритм. Алгоритм починає роботу з деяким набором даних, що називаються вхідними, і в результаті своєї роботи видає дані,

що називаються вихідними. Таким чином, алгоритм перетворює вхідні дані у вихідні.

Друге правило – для роботи алгоритму потрібна пам'ять. У пам'яті розміщуються вхідні дані, проміжні дані і вихідні дані, що є результатом роботи алгоритму.

Третє правило – дискретність. Алгоритм будується з окремих кроків (дій, операцій, команд).

Четверте правило – детермінованість. Після кожного кроку необхідно вказувати, який крок виконується наступним, або давати команду зупинки.

П'яте правило – результативність. Алгоритм повинен завершувати свою роботу після закінченого числа кроків.

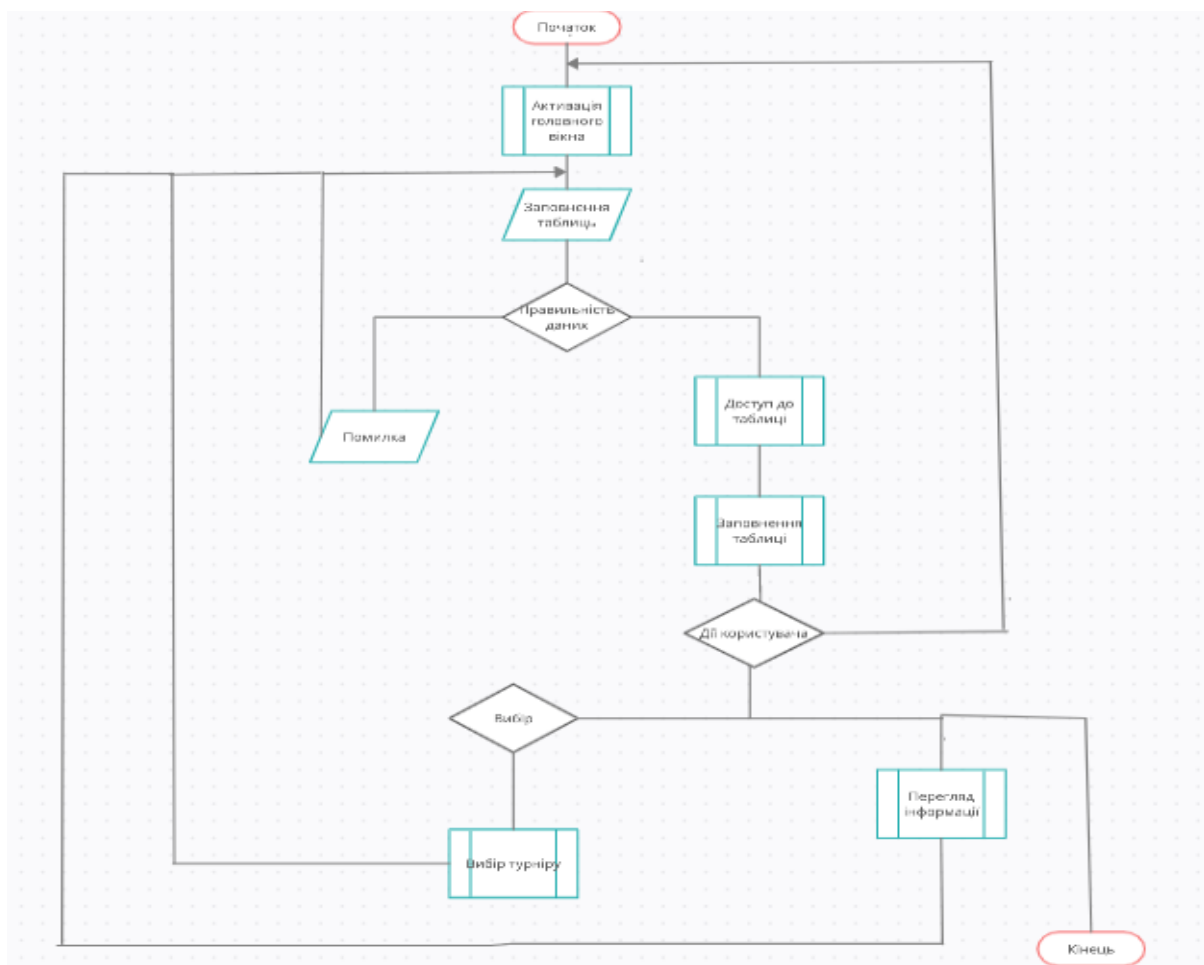


Рисунок 2.5 – Алгоритм можливостей користувача сайтом

Інтернет каталог доволі складний та відлагоджений механізм, робота якого полягає в реалізації ідей людини котра ним керує.

2.3 Інструкція програміста

2.3.1 Призначення і умови застосування

Програмний продукт призначений для продажу різних сортів кави за допомогою з'єднання «Internet»

Назва програми: Tournament

Автор: Пімкін Владислав

Мова програмування HTML, php

Файли, які використовує програма: Tournament

Розмір файлу Tournament8,68 МБ 24.05.2018

Проект написаний мовою HTML та php з використанням баз даних Msq1 5.1.

Web-орієнтована система турнірний сервіс «Tournament» допоможе на сторінках сайту знайти різноманітну інформацію, яка буде цікавою і корисною для користувача сайту.

В розділі:

- новини можна знайти потрібну інформацію про останні новини кіберспорту;

- інформація можна знайти повну інформацію про призначення сайту та його можливості;

- турніри можна знайти список різноманітних турнірів;

Створений сайт зручний і функціональний, тому відвідувачі можуть на сайті легко знайти те, що йому потрібно. Перевага цього сайту - величезна аудиторія, можливість розміщувати необмежену кількість пропозицій, своєчасно реагувати на зміни на ринку, враховувати потреби клієнтів та інші.

Для того щоб побачити як поводить себе користувач та ядро при вході в систему була створена діаграма діяльності, яка зображена на рисунку 2.5.

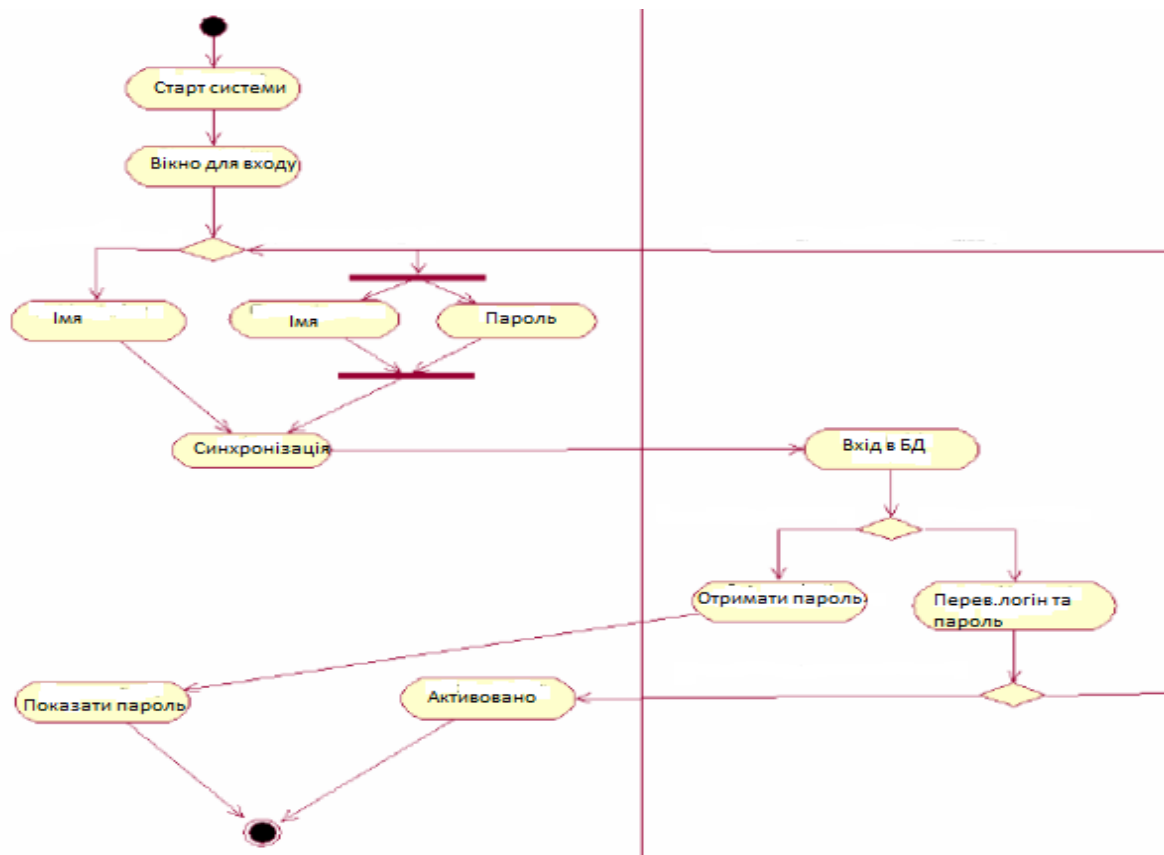


Рисунок 2.6 – діаграма послідовності

Опис діаграми

Після старту системи буде показано діалогове вікно (Вікно для входу). Після цього можливо два варіанти виконання системи:

1. Вхід в систему: виконується у випадку виконання умови ввод = «логін». Після цього будуть введені ім'я та пароль користувача, а потім вони синхронізуються. Після цього на сервері виконується підключення до БД системи – дія Вхід в БД та виконується перевірка імені та паролю. У разі успішної перевірки буде показано привітання користувачеві («Активовано») і система перейде в кінцевий стан, у разі неуспішної буде виконано перехід до активності («Вікно для входу»).

2. Відновлення паролю: виконується у випадку виконання умови введення = «відновлення». Після цього система отримує ім'я користувача, на сервері виконується підключення до бази даних та пошук паролю користувача. Після завершення пошуку паролю виконується активність «Показати пароль» та система переходить до кінцевого стану.

2.2.3 Діаграма взаємодії програмної системи

Діаграма взаємодії програмної системи. Контекст взаємодії – конкретна операція (вхід у систему та вибір послуги). У даному прикладі виділено 3 ролі: користувач, система, база даних.

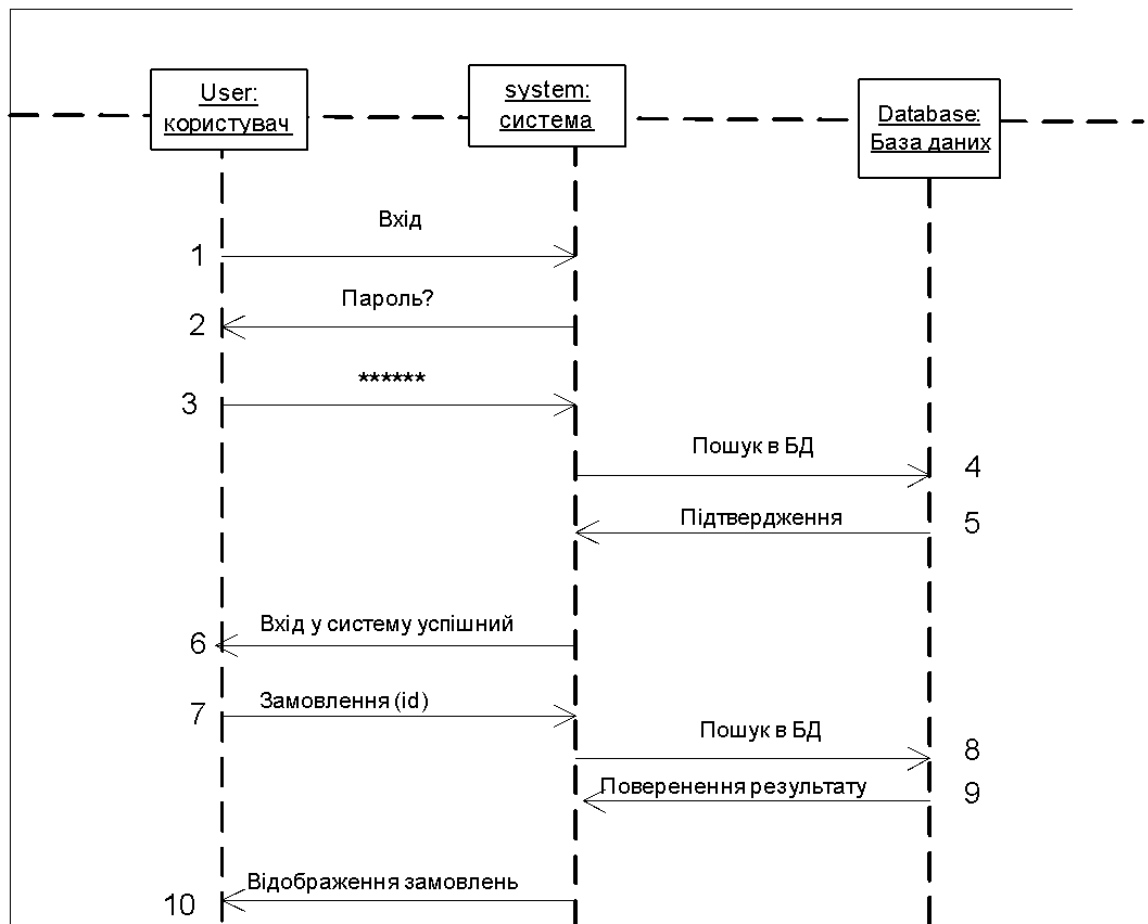


Рисунок 2.7 – Діаграма взаємодії програмної системи

- 1 Користувач надсилає запит на вхід у систему та надає свій логін.
- 2 Система потребує введення паролю.
- 3 Користувач вводить пароль.
- 4 Система шукає інформацію про користувача (логін та пароль) у базі даних.
- 5 База даних підтверджує, що такий користувач існує у системі.
- 6 Система підтверджує, що введений пароль є правильним і інформує користувача про успішний вхід у систему.
- 7 Користувач надсилає запит на відображення замовлень, вказуючи параметр, за яким відбувається пошук (id – код замовлень).
- 8 Система шукає необхідну інформацію у базі даних.
- 9 База даних повертає результат системі.
- 10 Система відображає задану послугу.

Також було створено діаграму пакетів, яка зображена на рисунку 2.8.

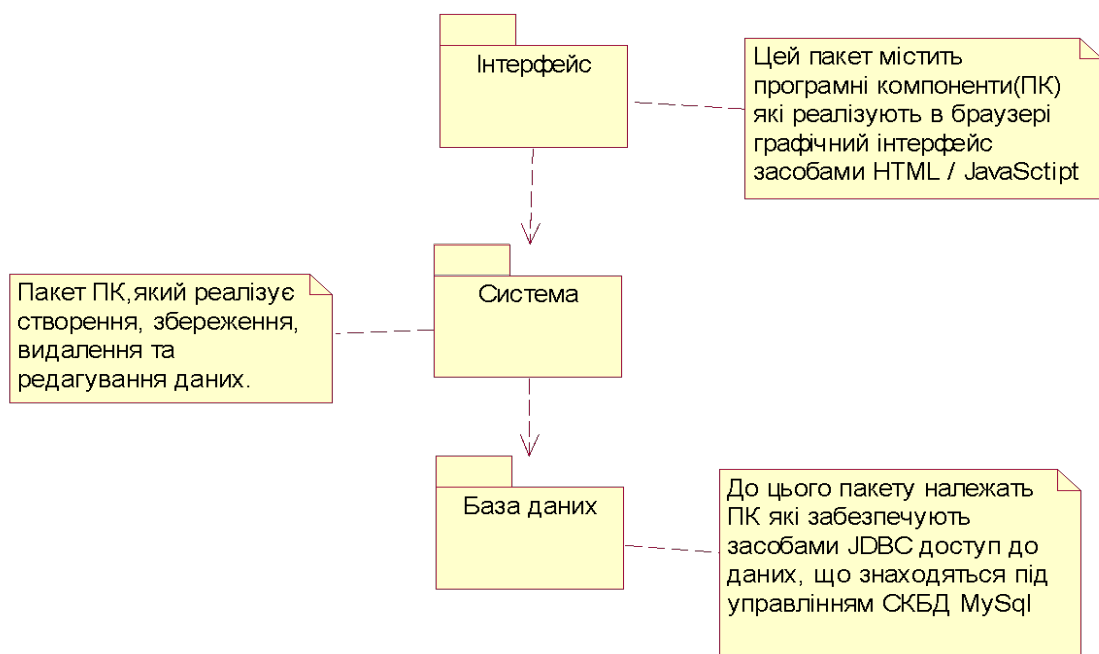


Рисунок 2.8 – діаграма пакетів

Пакет Інтерфейс – пакет який містить програмні компоненти які реалізують в браузері графічний інтерфейс засобами HTML / JavaScript.

Пакет Система - пакет який реалізує створення, збереження, видалення та редагування даних.

Пакет База даних – пакет до якого належать програмні компоненти які забезпечують засобами JDBC доступ до даних, що знаходяться під управлінням СКБД MySql.

2.4 Інструкція оператора

Для можливості користування веб-сайтом на ПК повинен бути підключений інтернет та будь-який браузер (Chrome, Yandex, Firefox і т. д.). Також комп'ютер повинен відповідати системним вимогам.

Файли проекту:

- assets – папка, в якій знаходяться класи, які відповідають за підключення необхідних ресурсів до вибраних веб-сторінок
- config – папка, в якій знаходяться найважливіші файли проекту. Вони відповідають за поведінку сайту у цілому. Наприклад: налаштування бази даних, основні компоненти проекту;
- controllers – папка, в якій знаходяться класи контролери;
- mail – папка, в якій знаходяться файли, які відповідають за email-повідомлення;
- models – папка, в якій знаходяться моделі проекту;
- modules – папка, в якій знаходяться окремі модулі проекту;
- tests – папка, в якій знаходяться тести;
- vendor – папка, в якій знаходяться усі бібліотеки чи окремі компоненти від яких залежить сам проект;
- views – папка, в якій знаходяться усі представлення проекту;

- web – папка, в якій знаходяться необхідні ресурси для коректного відображення проекту
- .htaccess – файл, який встановлює папка web за замовчуванням;

Щоб запустити веб-сайт у локальному режимі, необхідно використати локальний сервер (Open Server, Denwer і т. д.). Він дає можливість запускати сайти написані на PHP у звичайному режимі.

Після установки локального серверу та підключення веб-сайту до нього, необхідно зайти по вказаному адресу у браузері, який є ім'ям папки із сайтом. Після чого відобразиться головна сторінка сайту.

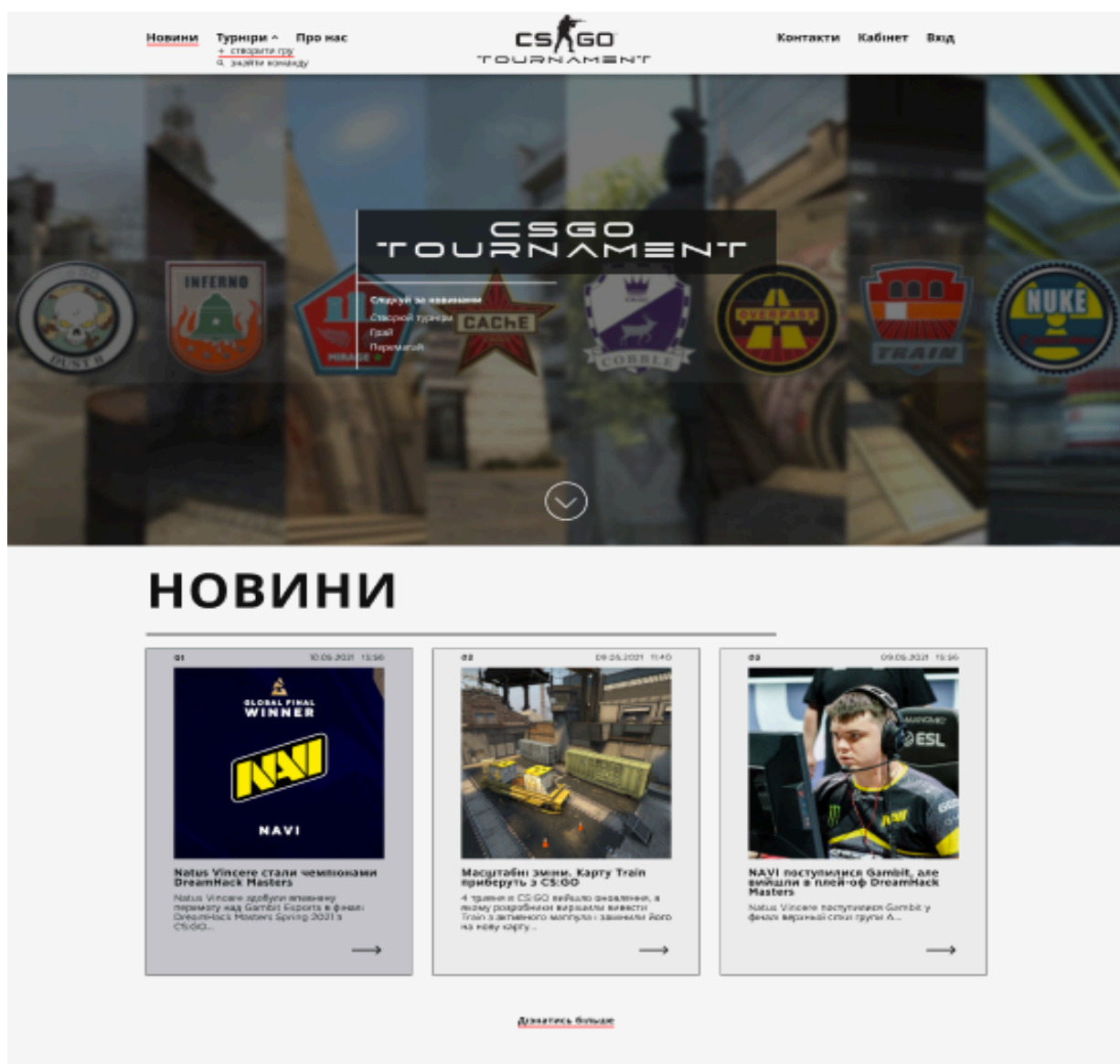


Рисунок 2.10 – стартова сторінка веб-сайту «Tournament»

У верхній частині сайту знаходиться навігаційне меню, яке веде на окремо створену сторінку.

Натиснувши на пункт “Новини” ми перейдемо на сторінку з лістингом усіх новин (див. рис. 2.11).

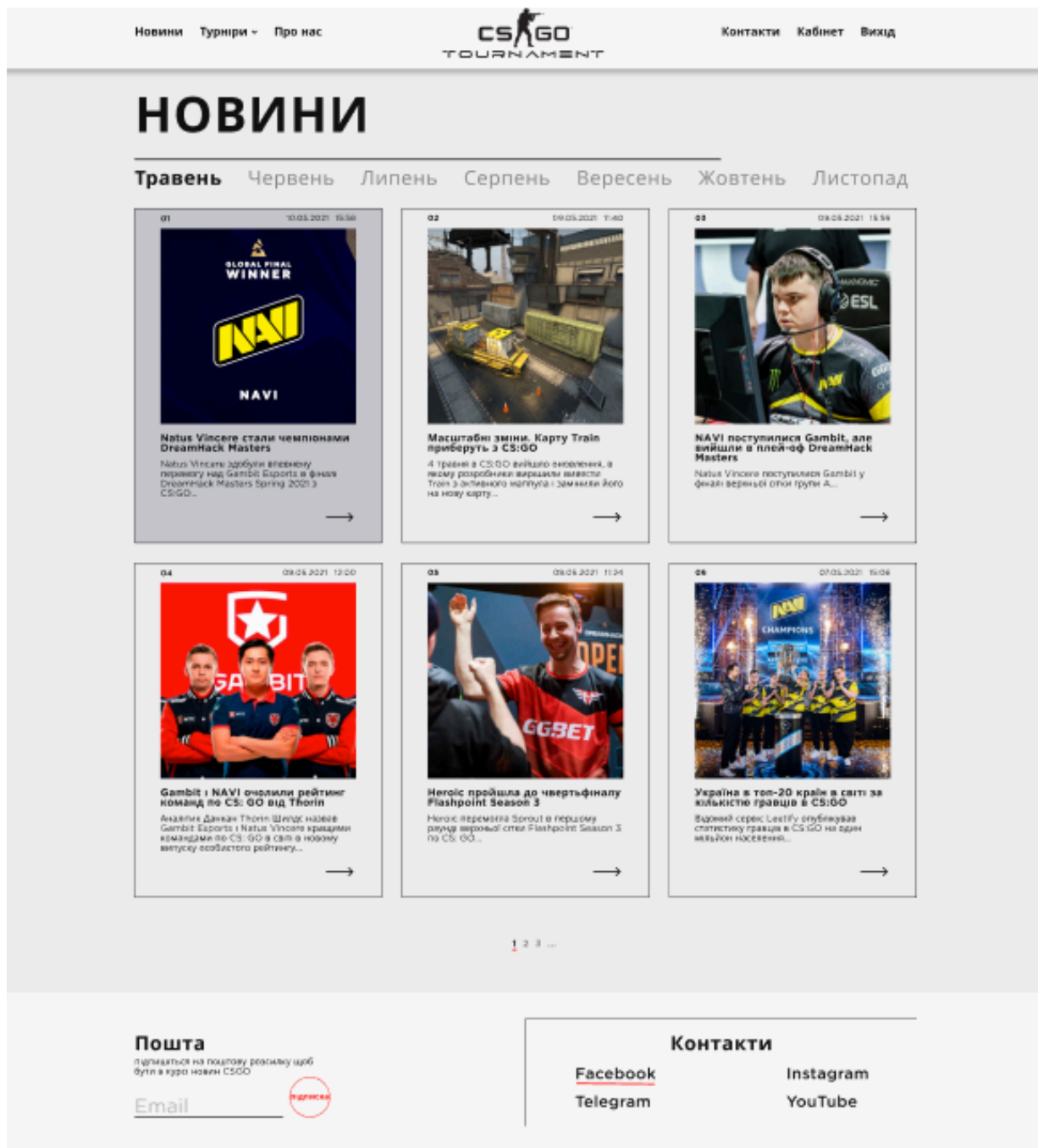


Рисунок 2.11 – сторінка «Новини»

На даній сторінці знаходиться список новин, натиснувши на які, ми перейдемо на сторінку з повним описом новини.

Одним із найважливіших пунктів є пункт “Турніри” перейшовши на який, користувач перейде на сторінку зі списком створених турнірів (див. рис. 2.12).

ТУРНІРИ

PGL Major Stockholm 2021
 ⌚ початок о 22:31 по Києву
 24 команди
 5 x 5 формат
 Реєстрація відкрита
 Реєстрація до 10.05.2021
 Призовий фонд: \$2,000,000 →

Funspark ULTI 2021 S1
 ⌚ ...
 32 команди
 5 x 5 формат
 Реєстрація завершена
 Турнір завершено 24.04.2021
 Призовий фонд: \$45,000 →

5x5 Турнір Б03
 ⌚ ...
 16 команди
 5 x 5 формат
 Реєстрація завершена
 Турнір завершено 23.04.2021
 Призовий фонд: \$20,000 →

[Дивіться більше](#)

Пошта
 підпишіться на пошту розсилку щоб бути в курсі новин CSGO
 Email [Підписки](#)

Контакти
[Facebook](#) [Instagram](#)
[Telegram](#) [YouTube](#)

Рисунок 2.12 – сторінка «Турніри»

На даній сторінці можна обрати турнір та ознайомитись з коротким його описом.

Перейшовши на сторінку “Кабінет” користувач потрапить до форми авторизації, якщо він зареєстрований, то після авторизації відбудеться перехід до

особистого кабінету. У особистому кабінеті користувач може бачити повну інформацію про свій профіль, редагувати його, та перевірити в яких турнірах він приймає участь, та запрошення на турніри (див. рис. 2.13 - 2.14).

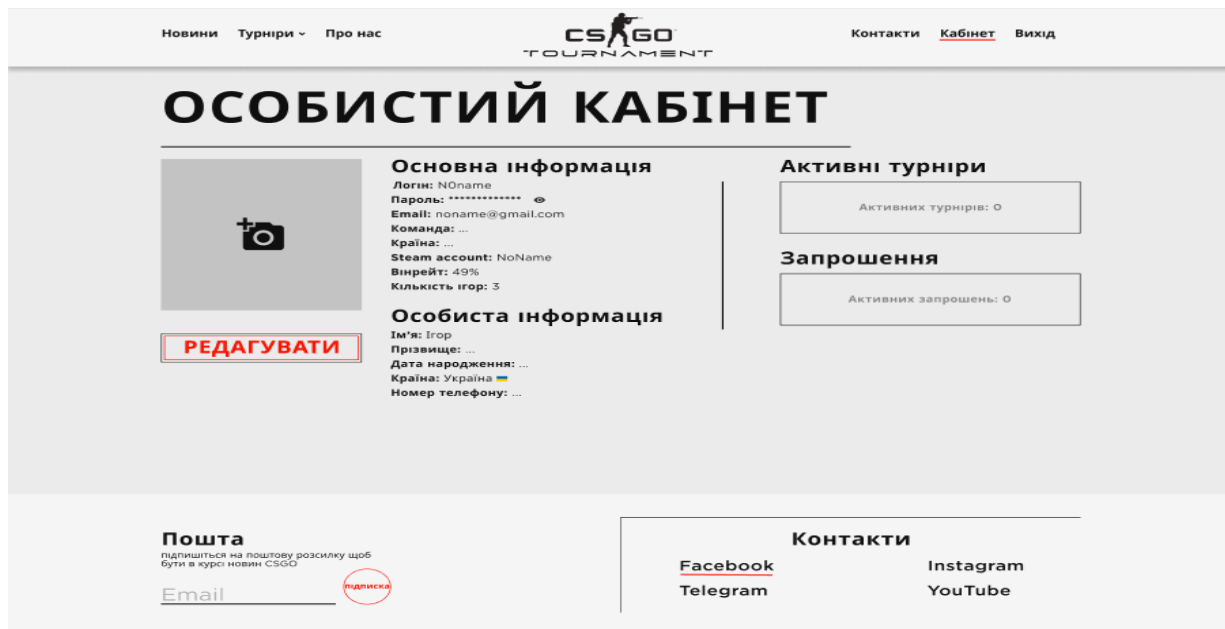


Рисунок 2.13 –особистий кабінет (приклад 1)

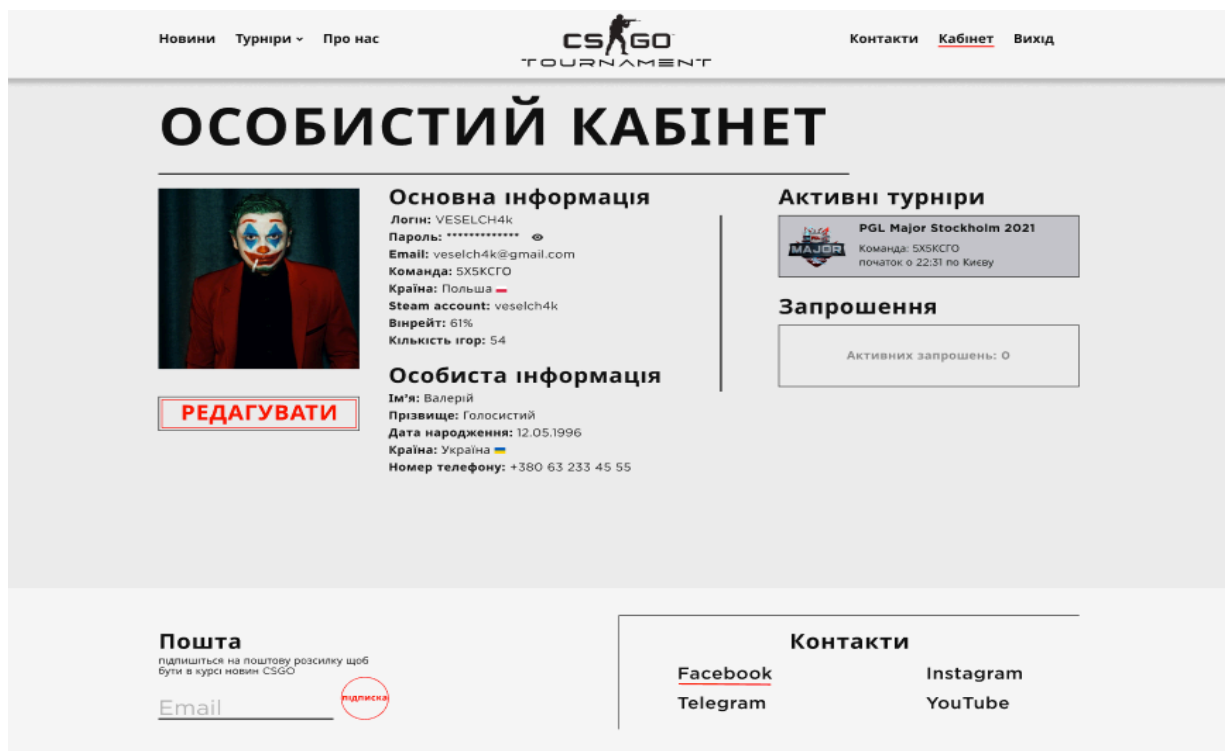


Рисунок 2.14 –особистий кабінет (приклад 1)

Натиснувши на пункт “Про нас” ми перейдемо на сторінку з інформацією про даний сайт (див. рис. 2.15)



Рисунок 2.15 – сторінка «Про нас»

Цю сторінку називають по-різному: Who we are / About us / Our team / Company і т.п. Але суть її залишається тією ж: ви допомагаєте користувачеві дізнатися вас краще, познайомитися і побачити, хто ви.

Сторінка "Про нас" часто є однією з найбільш популярних сторінок сайту і грає істотну роль в його просуванні. Ось кейс з життя нашого клієнта.

На його сайті сторінку «Про нас» була на 4-му місці за відвідуваністю, показник відмов і тривалість сесії були в нормі. Ця сторінка грає важливу роль у популяризації сайту.

Сторінка «Про компанії» або «Про нас» часто має на увазі «за замовчуванням»: начебто це ж очевидно, навіщо вона потрібна. З таким підходом сторінка виходить безликої: до тексту можна підставити назву іншої компанії чи бренду - сенс не зміниться.

3. ОРГАНІЗАЦІЙНО - ЕКОНОМІЧНА ЧАСТИНА

3.1 Оцінка вартості програмного продукту

Розробка сайту вимагає певних інтелектуальних і трудових витрат, а також обов'язкового використання комп'ютерної техніки, що визначає особливості розрахунку собівартості програмного продукту.

Оцінка вартості програмування розглядається як оцінка затрат коштів і часу, необхідних для виконання етапів проекту.

Вартість програмного продукту включає:

- собівартість програмного продукту;
- плановий прибуток.

3.1.1 Розрахунок часу

Загальний час на створення програми складається з різних компонентів. Структура загального часу на створення програмного продукту представлена в таблиці 3.1.

Таблиця 3.1 – Структура загального часу на створення програмного продукту.

№ етапу	Позначення часу даного етапу	Зміст етапу
1	$T_{ПО}$	Підготовка опису завдання
2	$T_{О}$	Опис завдання
3	$T_{А}$	Розробка алгоритмів
4	$T_{БС}$	Розробка блок-схеми алгоритмів
5	$T_{П}$	Проектування інтерфейсу
6	$T_{Н}$	Написання програми (кодування)
7	$T_{ВТ}$	Відладка і тестування програми

8	Т _д	Оформлення документації, інструкції користувачеві, записки пояснення
---	----------------	--

Час розраховується в людино-годинах, причому Т_{по} та Т_д береться по фактично відпрацьованому часу, а час останніх етапів визначається розрахунковий по умовному числу команд Q.

Умовне число команд Q визначається за формулою

$$Q = q \cdot z, \quad (3.1)$$

де q - коефіцієнт, що враховує умовне число команд залежно від типу завдання. Значення коефіцієнта q вибирається з таблиці 3.2.

Значення коефіцієнта z вибирається з таблиці 3.3.

Таблиця 3.2 – Значення коефіцієнта q.

Тип завдання	Значення коефіцієнта q
Завдання обліку	1400...1500
Завдання оперативного управління	1500...1700
Завдання планування	3000...3500
Багатоваріантні завдання	4500...5000
Комплексні завдання	5000...5500

Для даного завдання коефіцієнт q приймається = 1610

Програмні продукти за мірою новизни можуть бути віднесені до однієї з чотирьох груп:

- група А - розробка принципово нових завдань;
- група Б - розробка оригінальних програм;
- група В - розробка програм з використанням типових рішень;
- група Г - разове типове завдання.

Для даного завдання міра новизни: Б

За складністю програмні продукти можуть бути віднесені до однієї з трьох груп:

- 1 - алгоритми оптимізації і моделювання систем;
- 2 - завдання обліку, звітності і статистики;
- 3 - стандартні алгоритми.

Дане завдання може бути віднесено до другої групи складності високого рівня.

Коефіцієнт (z) визначається з таблиці 3.3 в залежності від складності і міри новизни.

Таблиця 3.3 – Значення коефіцієнта z .

Мова програмування	Група складності	Міра новизни			
		А	Б	В	Г
Високого рівня	1	1,38	1,26	1,15	0,69
	2	1,30	1,19	1,08	0,65
	3	1,20	1,10	1,00	0,60
Низького рівня	1	1,58	1,45	1,32	0,79
	2	1,49	1,37	1,24	0,74
	3	1,38	1,26	1,15	0,69

Для даного завдання коефіцієнт $z = 1,19$.

За формулою 3.1 визначається умовне число команд Q .

$$Q = 1610 * 1,19 = 1915,90 \text{ чол / годин.}$$

Визначається час, витрачений на кожен етап створення програмного продукту:

- $T_{\text{ПО}}$ (час на підготовку опису завдання), береться по факту і для даного завдання.

$$T_{\text{ПО}} = 40 \text{ чол / годин.}$$

- T_0 (час на опис завдання) визначається за формулою

$$T_0 = Q \cdot V / (50 \cdot K),$$

де V - коефіцієнт обліку змін завдання. Коефіцієнт V залежно від складності завдання і числа змін вибирається в інтервалі від 1,2 до 1,5. Для даного завдання $V = 1,4$

K - коефіцієнт, що враховує кваліфікацію програміста. Значення коефіцієнта вибирається з таблиці 3.4.

Таблиця 3.4 – Значення коефіцієнта K .

Стаж програміста	Значення коефіцієнта K
до 2-х років	0,8
від 2 до 3 років	1,0
від 3 до 5 років	1,1...1,2
від 5 до 10 років	1,2...1,3
понад 10 років	1,3...1,5

В даному випадку коефіцієнт $K = 1,2$.

За формулою 3.2 підраховується час на опис завдання.

$$T_0 = 1915,90 \cdot 1,4 / (50 \cdot 1,2) = 44,70 \text{ [чол. / годин]}.$$

- T_A (час на розробку алгоритмів) розраховується за формулою

$$T_A = Q / (50 \cdot K),$$

За формулою 3.3 підраховується час на розробку алгоритмів.

$$T_A = 1915,90 / (50 \cdot 1,2) = 31,0 \text{ [чол. / годин]}.$$

- T_{BC} (час на розробку блок-схеми) визначається аналогічно T_A за формулою 3.3 і складає

$$T_{BC} = 1915,90 / (50 \cdot 1,2) = 31,0 \text{ [чол. / годин]}.$$

- T_H (час написання програми на мові програмування) визначається за формулою

$$T_H = Q \cdot 1,5 / (50 \cdot K),$$

За формулою 3.4 підраховується час написання програми на мові програмування.

$$T_H = 1915,90 \cdot 1,5 / (50 \cdot 1,2) = 47,8 \text{ [чол. / годин]}.$$

- T_{III} (час на проектування інтерфейсу) визначається за формулою

$$T_{III} = Q / 50,$$

За формулою 3.5 підраховується час на проектування інтерфейсу програми.

$$T_{III} = 1915,90 / 50 = 38,0 \text{ [чол. / годин]}.$$

- T_{BT} (час відладки і тестування програми) визначається за формулою

$$T_{BT} = Q \cdot 4,2 / (50 \cdot K),$$

За формулою 3.6 підраховується час відладки і тестування програми.

$$T_{BT} = 1915,90 \cdot 4,2 / (50 \cdot 1,2) = 134,0 \text{ [чол. / годин]}.$$

- T_D (час на оформлення документації), береться по факту і для даного завдання

$$T_D = 40 \text{ чол / годин.}$$

Підраховується загальний час на створення програмного продукту за формулою

$$T = T_{ПО} + T_O + T_A + T_{BC} + T_H + T_{III} + T_{BT} + T_D$$

$$T = 40 + 44,70 + 31,0 + 31,0 + 47,8 + 38,0 + 134,0 + 40 = 406,5 \text{ [чол. / годин]}.$$

3.1.2 Розрахунок заробітної плати виконавця робіт із створення програмного продукту

Основна ЗП визначається за формулою

$$Z_{Посн.} = (ЗП_{1р} \cdot K_T \cdot T) / (Ч_r \cdot tr.д.) \cdot (1 + П/100),$$

де $ЗП_{1р}$ - місячна зарплата 1-го розряду, грн.;

Кт - тарифний коефіцієнт, відповідний розряду тарифної сітки, за яким працює виконавець;

Т - загальний час на створення програмного продукту, чол. / годин;

Чр - кількість робочих днів в місяць;

тр.д. - тривалість робочого дня в годинах;

П - відсоток премії.

Для даного завдання:

- ЗП 1р = 450,00 грн.;

- виконавець має 14 розряд;

- для 14 розряду тарифний коефіцієнт Кт = 3,36;

- загальний час створення програмного продукту Т=406,5 чол. / годин;

- кількість робочих днів Чр = 21;

- тривалість робочого дня тр.д. = 8 годин;

- відсоток премії П=15%.

Таким чином, визначаємо основну заробітну плату виконавця робіт із створення програмного продукту

$$\text{ЗП осн.} = (450,00 * 3,36 * 406,5) / (21 * 8) * (1 + 15/100) = 4207,27 \text{ [грн]}.$$

Додаткова заробітна плата береться у розмірі 15 % від основної і розраховується за формулою

$$\text{ЗП дод.} = \text{ЗПосн.} \cdot 15 / 100,$$

$$\text{ЗП дод.} = 4207,27 * 15 / 100 = 631,09 \text{ [грн]}.$$

Загальна заробітна плата розраховується за формулою

$$\text{ЗПзагальна} = \text{ЗПосн.} + \text{ЗПдод.},$$

$$\text{ЗПзагальна} = 4207,27 + 631,09 = 4838,36 \text{ [грн]}.$$

3.1.3 Розрахунок нарахувань на заробітну плату (єдиного соціального внеску)

Єдиний соціальний внесок нараховується на заробітну плату за формулою 3.11 і складає 22 % від загальної заробітної плати

$$\text{ЄСВ} = \text{ЗП}_{\text{загальна}} \cdot 22 / 100,$$

$$\text{ЄСВ} = 4838,36 \cdot 22 / 100 = 1064,43 \text{ [грн]}.$$

3.1.4 Розрахунок витрат на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту

Витрати на збереження та експлуатацію ПЕОМ визначаються у розмірі 130% від основної заробітної плати працівників, що забезпечують функціонування ПЕОМ.

До таких витрат відносяться витрати на:

- основну заробітну плату працівників, що забезпечують функціонування ПЕОМ (інженер-електронник; системний програміст; оператор);
- основна ЗП адміністративного і допоміжного персоналу;
- амортизаційні відрахування;
- витрати на електричну енергію;
- витрати на матеріали (до їх числа входять диски, картриджі і папір для принтерів та інше);
- витрати на профілактику ПЕОМ з периферією;
- витрати на опалення виробничих площ;
- витрати на обслуговування виробничих площ;
- інші виробничі витрати.

Розрахунок витрат на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту за формулою

$$\text{В зб. експл.} = \text{ЗП}_{\text{осн.}} \cdot 130 / 100,$$

$$\text{В зб. експл.} = 4207,27 \cdot 130 / 100 = 5469,45 \text{ [грн]}.$$

3.2 Розрахунок собівартості програмного продукту

У собівартість програмного продукту входять наступні елементи:

- основна заробітна плата виконавця робіт із створення програмного продукту;
- додаткова заробітна плата виконавця робіт із створення програмного продукту;
- нарахування на заробітну плату (єдиний соціальний внесок);
- витрати на збереження та експлуатацію ПЕОМ, що відносяться до даного програмного продукту;
- інші витрати.

Інші витрати складають 10% від суми перших чотирьох елементів і розраховуються за формулою

$$I.вит. = (ЗП осн. + ЗП дод. + ЄСВ + В зб. експл.) \cdot 10 / 100$$

$$I.вит. = (4207,27 + 631,09 + 1064,43 + 5469,45) \cdot 10 / 100 = 1137,22 \text{ [грн].}$$

Визначається собівартість програмного продукту за формулою

$$Сп.п. = ЗП осн. + ЗП дод. + ЄСВ + В зб.експл. + I.вит.$$

$$Сп.п. = 4207,27 + 631,09 + 1064,43 + 5469,45 + 1137,22 = 12509,46 \text{ [грн].}$$

Структура собівартості програмного продукту відображається в таблиці 3.5.

Таблиця 3.5 – Структура собівартості програмного продукту

Елементи структури	Сума грн.	Питома вага %
1 Основна заробітна плата виконавця	4207,27	34
2 Додаткова заробітна плата виконавця	631,09	5
3 Нарухування на заробітну плату (ЄСВ)	1064,43	9
4 Витрати збереження та експлуатацію ПП	5469,45	43

5 Інші витрати	1137,22	9
Собівартість програмного продукту	12509,46	100

3.3 Розрахунок ціни програмного продукту

Ціна програмного продукту складається з декількох компонентів і розраховуються за формулою

$$Ц = \text{Сп.п.} + П + \text{ПДВ},$$

де Сп.п.- собівартість програмного продукту, грн.;

П - плановий прибуток, грн.;

ПДВ - податок на додану вартість, грн.

П – прибуток складає 40% від повної собівартості програмного продукту і розраховуються за формулою

$$П = \text{Сп.п.} \cdot 40 / 100$$

$$П = 12509,46 \cdot 40 / 100 = 5003,78 \text{ [грн]}.$$

ПДВ - податок на додану вартість, який береться у розмірі 20% від суми собівартості і прибутку розраховуються за формулою

$$\text{ПДВ} = (\text{Сп.п.} + П) \cdot 20 / 100,$$

$$\text{ПДВ} = (12509,46 + 5003,78) \cdot 20 / 100 = 3502,64 \text{ [грн]}.$$

За формулою 3.15 визначається ціна програмного продукту.

$$Ц = 12509,46 + 5003,78 + 3502,64 = 21015,88 \text{ [грн]}.$$

3.4 Зведена таблиця показників

Результати розрахунків відображаються в підсумковій таблиці 3.6.

Таблиця 3.6 – Результати розрахунків

Найменування показника	Сума,
------------------------	-------

	грн.
Собівартість програмного продукту	12509,46
Прибуток	5003,78
ПДВ	3502,64
Ціна програмного продукту	21015,88

Розроблений сайт має всі базові функції наявних аналогів на світових ринках, а також доповнює деякі з них. При цьому розроблений сайт має конкурентноспроможну ціну. В порівнянні з популярними сайтами має нижчу ціну і заслуговує уваги від користувачів мережі. Результати розрахунку показані в діаграмі.

Собівартість програмного продукту відображається у вигляді діаграми (Рисунок 3.1).

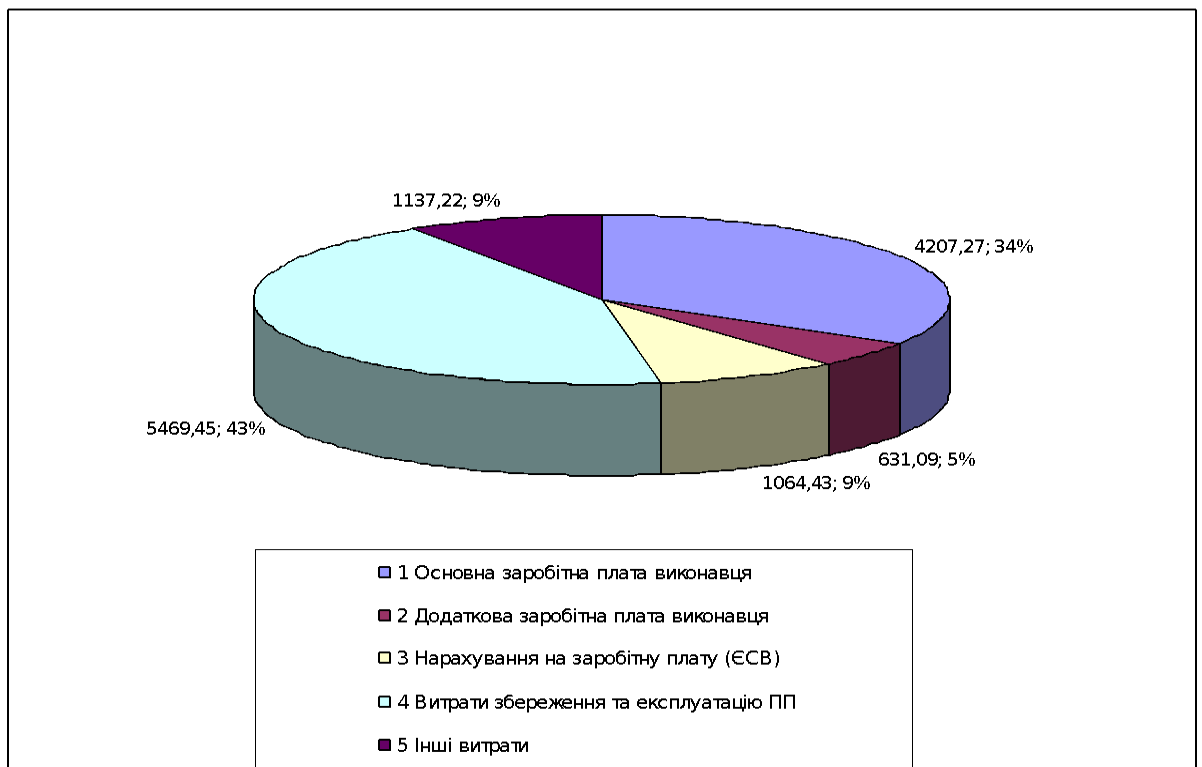


Рисунок 3.1- Діаграма собівартості web-орієнтованої системи
турнірний сервіс «Tournament»

4. БЕЗПЕКА ЖИТТЄДІЯЛЬНОСТІ

4.1 Вимоги до приміщень і організації робочих місць

Загальні вимоги:

Облаштування робочих місць, обладнаних відео терміналами, повинно забезпечувати:

- належні умови освітлення приміщення і робочого місця, відсутність відблисків;
- оптимальні параметри мікроклімату (температура, відносна вологість, швидкість руху, рівень іонізації повітря);
- належні ергономічні характеристики основних елементів робочого місця;
- а також враховувати такі небезпечні та шкідливі фактори:
 - наявність шуму та вібрації;
 - м'яке рентгенівське випромінювання;
 - електромагнітне випромінювання;
 - ультрафіолетове та ультрачервоне випромінювання;
 - електростатичне поле між екраном та оператором;
 - наявність пилу, озону, оксидів азоту й аероіонізації.

Будівлі та приміщення, в яких експлуатують ЕОМ та виконують їх обслуговування, налагодження і ремонт, повинні відповідати вимогам:

НіП 2.09.02-85 „Виробничі будівлі”;

СНіп 2.09.04-87 „Адміністративні і побутові будівлі”;

Правил безпечної експлуатації електроустановок споживачів, затверджених наказом Держнаглядохоронпраці 09.01.98 №4, зареєстрованих у Мін'юсті України 10.02.98 № 93/2533 (ПБЕ);

СНіП 2.01.02-85 „Протипожежні норми”;

Правил пожежної безпеки в Україні, затверджених наказом Управління Державної пожежної охорони МВС України від 14.06.95 № 400, зареєстрованих в Міністерстві юстиції України 14.07.95 за № 219/755;

СНіП 2.08.02-89 „Громадські будівлі і споруди” з доповненнями, затвердженими наказом Держкоммістобудування України від 29.12.94 № 106;

ДСанПіН 3.3.3.-007-98 „Державні санітарні правила і норми роботи з візуальними дисплеями терміналами електронно-обчислювальних машин”, затверджених МОЗ України 10.12.98, а також вимогами нормативно-технічної та експлуатаційної документації заводу-виробника ЕОМ, чинних санітарних норм, санітарних норм і правил, правил у сфері охорони праці.

Для всіх споруд і приміщень, в яких експлуатуються відеотермінали та ЕОМ, повинна бути визначена категорія з вибухопожежної та пожежної безпеки відповідно до ОНТП 24-86 „Определение категорий помещений и зданий по взрывопожарной и пожарной опасности”, затверджених МВС СРСР 27.02.86, та клас зони згідно з ПВЕ. Відповідні позначення повинні бути нанесені на входні двері приміщення.

Будівлі і ті їх частини, в яких розташовуються ЕОМ, повинні мати не нижче II ступеня вогнестійкості. Приміщення для обслуговування, ремонту та налагодження ЕОМ повинні належати за пожежовибухобезпекою до категорії В відповідно до ОНТП 24-86, а за класом приміщення – до П-Па за ПВЕ. Якщо відповідно до СНіП 2.09.02-85 ці приміщення повинні бути відокремленими від приміщень іншого призначення протипожежними стінами, то межа їх вогнестійкості визначається відповідно до СНіП 2.01.02 -85.

Неприпустимим є розташування приміщень категорій А іБ (ОНТП 24-86), а також виробництва з мокрими технологічними процесами поряд з приміщеннями, де розташовуються ЕОМ, виконується їх обслуговування, налагодження і ремонт, а також над такими приміщеннями або під ними. Виробничі приміщення, в яких розташовані ЕОМ, не повинні межувати з приміщеннями, де рівні шуму та вібрації перевищують норму (механічні цехи, майстерні тощо).

Робочі місця з відеотерміналами або персональними ЕОМ у приміщеннях з джерелами шкідливих виробничих факторів повинні розміщуватися в ізольованих кабінах з обладнаним повітрообміном. Стіни кабін виготовляються з негорючих матеріалів. Дозволяється виготовляти їх зі скла та металевих конструкцій. У кабіні мусить бути оглядове вікно (вікна). Висота оглядового вікна мусить бути не менше 1,5 м, а відстань від підлоги не більше 0,8 м.

Відповідно до ДСанПіН 3.3.3-007-98 „Державні санітарні правила і норми роботи з візуальними дисплейними терміналами електронно-обчислювальних машин”, затверджених МОЗ України 10.12.98, є неприпустимим розташування приміщень для роботи з відеотерміналами та ЕОМ у підвалах та цокольних поверхах.

Площу приміщень, в яких розташовують відеотермінали, визначають згідно з чинними нормативними документами з розрахунку на одне робоче місце, обладнане відеотерміналом: площа – не менше 6,0 м², обсяг – не менше 20,0 м³, з урахуванням максимальної кількості осіб, які одночасно працюють у зміні.

Стіни, стеля, підлога приміщень, де розміщені ЕОМ, повинні виготовлятися з матеріалів, дозволених для оздоблення приміщень органами державного санітарно-епідеміологічного нагляду.

Обслуговування, налагодження і ремонт ЕОМ, вузлів та блоків ЕОМ слід виконувати в окремому приміщенні (майстерні).

При цьому робочі місця електромеханіків повинні бути оснащені спеціальним обладнанням та захисними засобами відповідно до вимог до організації робочого місця з обслуговування, ремонту та налагодження ЕОМ.

У приміщеннях для обслуговування, ремонту та налагодження ЕОМ слід передбачити можливість вологого очищення поверхонь комунікації та опалювальних приладів.

Підлога всієї зони обслуговування, ремонту та налагодження ЕОМ, вузлів та блоків ЕОМ має бути вкрита діелектричними килимками, термін використання

яких після їх випробування на електричну міцність не закінчився, або викладена ізолюваними підстилками (шириною не менше ніж 0,75-0,8 м) для ніг.

Приміщення комп'ютерних класів (залів), в яких проводиться навчання на ЕОМ, крім зазначених у пункті 1.2, повинні мати суміжне приміщення (лаборантську) площею не менше 18 м² з двома входами: в учбове приміщення та в коридор (на сходову клітку).

Заземлені конструкції, що знаходяться в приміщеннях (батареї опалення, водопровідні труби, кабелі із заземленим відкритим екраном тощо), мають бути надійно захищені діелектричними щитками або сітками від випадкового дотику.

У приміщеннях з ЕОМ слід щоденно проводити вологе прибирання.

У приміщеннях з ЕОМ повинні бути медичні аптечки першої допомоги.

Приміщення, в яких розміщуються ЕОМ типу ЄС, СМ та інші великі ЕОМ загального призначення, обладнуються системою автоматичної пожежної сигналізації та засобами пожежогасіння відповідно до вимог Переліку однотипних за призначенням об'єктів, які підлягають обладнанню автоматичними установками пожежогасіння та пожежної сигналізації, СНіП 2.04.09-84, СН 512-78, Правил пожежної безпеки в Україні та вимог нормативно-технічної та експлуатаційної документації заводу-виробника.

Вимоги до освітлення

Природне світло повинно проникати через бічні світлопрорізи, зорієнтовані, як правило, на північ чи північний схід, і забезпечувати коефіцієнт природної освітленості (КПО) не нижче 1,5%. Розрахунки КПО проводяться відповідно до СНіП II-4-79.

При виробничій потребі дозволяється експлуатувати ЕОМ у приміщеннях без природного освітлення за узгодженням з органами державного нагляду за охороною праці ц органами та установами санітарно-епідеміологічної служби.

Вікна приміщень з відеотерміналами повинні мати регульовальні пристрої для відкривання, а також жалюзі, штори, зовнішні козирки тощо.

Штучне освітлення приміщення з робочими місцями, обладнаними відеотерміналами ЕОМ загального та персонального користування, має бути обладнане системою загального рівномірного освітлення. У виробничих та адміністративно-громадських приміщеннях, де переважають роботи з документами, допускається вживати систему комбінованого освітлення (додатково до загального освітлення встановлюються світильники місцевого освітлення).

При розташуванні відеотерміналів ЕОМ за периметром приміщення лінії світильників штучного освітлення повинні розміщуватися локально над робочими місцями.

Як джерело світла при штучному освітленні повинні застосовуватися, як правило, люмінесцентні лампи типу ЛБ. При обладнанні відбивного освітлення у виробничих та адміністративно-громадських приміщеннях можуть застосовуватися металогалогенові лампи потужності до 250 Вт. Допускається у світильниках місцевого освітлення застосовувати лампи розжарення.

Яскравість світильників загального освітлення в зоні кутів випромінювання від 50° до 90° відносно вертикалі в подовжній і поперечній площинах повинна складати не більше 200 кд/м^2 , а захисний кут світильників повинен бути не більшим за 40° .

Коефіцієнт запасу (Кз) відповідно до СНиП II-4-79 для освітлювальної установки загального освітлення слід приймати рівним 1,4.

Коефіцієнт пульсації повинен не перевищувати 5% і забезпечуватися застосуванням газорозрядних ламп у світильниках загального і місцевого освітлення.

При відсутності світильників або розташовані поруч світильники загального освітлення необхідно підключати до різних фаз трифазної мережі.

Рівень освітленості на робочому столі в зоні розташування документів має бути в межах 300-500 лк. У разі неможливості забезпечити даний рівень освітленості системою загального освітлення допускається застосування

світильників місцевого освітлення, але при цьому не повинно бути відблисків на поверхні екрану та збільшення освітленості екрану більше ніж 300 лк.

4.2 Вимоги охорони праці при експлуатації комп'ютерних систем

Користувачі комп'ютерних систем повинні слідкувати за тим, щоб відеотермінали, комп'ютерних систем, периферійні пристрої комп'ютерних систем та устаткування для обслуговування, ремонту та налагодження комп'ютерних систем були справними і випробуваними відповідно до чинних нормативних документів.

Щоденно перед початком роботи необхідно проводити очищення екрану відеотерміналу від пилу та інших забруднень.

Після закінчення роботи відеотермінал та персональна комп'ютерних систем повинні бути відключені від електричної мережі.

У разі виникнення аварійної ситуації необхідно негайно відключити відеотермінал та комп'ютерні системи від електричної мережі.

Вплив шуму.

Відомо, що шум він несприятливо діє на слуховий аналізатор та інші органи та системи організму людини. Визначальне значення щодо такої дії має інтенсивність шуму, його частотний склад, тривалість щоденного впливу, індивідуальні особливості людини, а також специфіка виробничої діяльності. Ті види діяльності, у яких поєднується напружена розумова робота та інтенсивне використання комп'ютера (редагування тексту, верстка оригіналу, "запуск" та відлагодження програм тощо) характеризується відчутним впливом навіть незначних рівнів шуму. Цей вплив виражається у зниженні розумової працездатності, швидкій втомлюваності, послабленні уваги, появі головного болю та ін.

На комп'ютеризованих робочих місцях основними джерелами шуму є вентилятори системного блоку, накопичувачі, принтери ударної дії. Для зниження

рівнів шуму на робочих місцях рекомендується розмістити друкувальні пристрої ударної дії (матричні, шрифтові принтери тощо) в іншому приміщенні, або огородити їх звукоізолюючими екранами.

Під час виконання робіт з ВДТ і ПК у виробничих приміщеннях значення характеристик вібрації на робочих місцях не повинні перевищувати допустимих значень, визначених СН 3044-84 та ГОСТ 12.1.012-90.

Для зниження вібрації обладнання, пристрої, пристосування необхідно встановлювати на спеціальні амортизуючі прокладки, передбачені нормативними документами.

Головним небезпечним фактором безпеки при експлуатації та ремонті пристрою є електробезпека.

Електрика — сукупність явищ, зумовлених існуванням, рухом взаємодією електрична заряджених тіл або часток. Електричний струм — це упорядкований (спрямований) рух електрична заряджених часток. Струм у металах зумовлений наявністю вільних електронів, у електролітах — іонів. Звичайно силою, яка викликає такий рух, є сила з боку електричного поля усередині провідника, яке визначається електричною напругою на кінцях провідника.

Наявність електричного струму в провідниках призводить до їх нагрівання, зміни хімічного складу, створення магнітного поля.

Електричні прилади, установки, обладнання, з яким людина має справу, становлять для неї велику потенційну небезпеку, яка посилюється тим, що органи чуття людини не можуть на відстані виявити наявність електричної напруги, як, наприклад, теплову, світлову чи механічну енергію. Тому захисна реакція організму виявляється тільки після безпосереднього потрапляння під дію електричного струму. Другою особливістю дії електричного струму на організм людини є те, що струм, проходячи через людину, діє не тільки в місцях контактів і на шляху протікання через організм, а й викликає рефлекторні порушення нормальної діяльності окремих органів (серцево-судинної системи, системи дихання). Третя особливість — це можливість одержання електротравм без

безпосереднього контакту із струмопровідними частинами — при переміщенні по землі поблизу ушкодженої електроустановки (у випадку замикання на землю), ураження через електричну дугу.

Електричний струм, проходячи через тіло людини, зумовлює перетворення поглинутої організмом електричної енергії в інші види і спричиняє термічну, електролітичну, механічну і біологічну дію. Найбільш складною є біологічна дія, яка притаманна тільки живим організмам.

Статистика свідчить, що більше половини всіх електротравм становлять опіки. Вони важко піддаються лікуванню, тому що глибоко проникають у тканини організму. В електроустановках напругою до 1 кВ найчастіше спостерігаються опіки контактного виду при дотиканні тіла до струмопровідних частин.

Опіки можливі при проходженні через тіло людини струму більше 1 А. Тільки при великому струмі тканини, які уражаються, нагріваються до температури 60—70°C і вище, при якій згортається білок і з'являються опіки.

В електроустановках напругою вище 1 кВ опіки можуть виникнути при випадковому наближенні частин тіла людини до струмопровідних частин на небезпечну відстань; при цьому збільшується напруга електричного поля і внаслідок ударної іонізації діелектрика (повітряного проміжку) опір цього проміжку зменшується, його «пробиває» електричний розряд — електрична дуга, температура якої досягає приблизно 4000°C. Електричний струм протікає через дугу і тіло людини. За такої високої температури і великої кількості тепла, яка виділяється при проходженні струму через тіло, потерпілий одержує тяжкі опіки, його м'язи скорочуються, дуга і ланцюг струму розриваються.

Майже у всіх випадках включення людини в електричний ланцюг на її тілі і в місцях дотикання спостерігаються «електричні знаки» сіро-жовтого кольору круглої або овальної форми.

При опіках від впливу електричної дуги можлива металізація шкіри частками металу дугової плазми. Уражена ділянка шкіри стає твердою, набуває кольору солей металу, які потрапили в шкіру.

ВИСНОВКИ

Головна мета даної дипломної роботи розробити турнірну WEB-систему виконана.

Даний Web-сайт дозволяє потенційним клієнтам ознайомитися з різними новинами, вибрати та створювати турніри. Візуальне подання даних значно полегшує пошук необхідної інформації. Інтерфейс інтерактивного довідника дуже простий і зручний у пошуку.

На розробленому веб-сайті можна дізнатися про останні новини кіберспорту, ознайомитися зі списком вже створених турнірів, прийняти в них участь або створити власний.

Дизайн сайту вибраний не випадково, він виконаний в яскравих кольорах і є максимально простим та зрозумілим для користувача.

При розробці веб-сайта були проаналізовані сучасні веб-технології, що дозволяють створювати інтерактивні веб-сторінки.

Розроблений сайт задовольняє всім вимогам, поставленим на етапі постановки завдання.

Як подальше вдосконалення веб-сайта представляється можливим розробка модулів доступу. Так само можливе доопрацювання інтерфейсу сайту з метою подальшого підвищення його інформативності, привабливості і зручності.

ПЕРЕЛІК УМОВНИХ ПОЗНАЧЕНЬ, СИМВОЛІВ, ОДИНИЦЬ,
СКОРОЧЕНЬ, ТЕРМІНІВ

XHTML	–	англ. Extensible Hypertext Markup Language розширювана мова розмітки гіпертексту.
MySQL	–	вільна система керування реляційними базами даних
PHP	–	PHP: Hypertext Preprocessor — популярна скриптова мова.
SQL	–	англ.. Structured Query Language — мова структурних запитів.
UML	–	англ.. Unified Modeling Language — уніфікована мова моделювання.
XHTML	–	англ.. Extensible Hypertext Markup Language — розширена мова розмітки гіпертексту.
XML	–	англ. EXtensible Markup Language — розширена мова розмітки.
БД	–	база даних.
Грн	–	грошова одиниця, гривні.
ЕОМ	–	електронна обчислювальна машина — загальна назва для обчислювальних машин.
ПЗ	–	Пояснювальна записка.

СПИСОК ВИКОРИСТАНИХ ДЖЕРЕЛ

1. Закон України Про оплату праці.
2. Закон України Про підприємництво.
3. Закон України Про оподаткування прибутку підприємств.
4. ГОСТ 19.001-77 ЄСПД. Загальні положення.
5. ГОСТ 19.002-80 ЄСПД. Схеми алгоритмів і програм. Правила виконання.
6. ГОСТ 19.003-80 ЄСПД. Схеми алгоритмів і програм. Позначення умовні графічні.
7. ГОСТ 19.105-78 ЄСПД. Загальні вимоги до програмних документів.
8. ГОСТ 19.401-78 ЄСПД. Текст програми. Вимоги до змісту і оформлення.
9. ГОСТ 19.402-78 ЄСПД. Опис програми.
10. ГОСТ 19.504-79 ЄСПД. Інструкція програміста.
11. ГОСТ 19.505-79 ЄСПД. Інструкція оператора.
12. ГОСТ 19.506-79 ЄСПД. Опис мови. Вимоги до змісту і оформлення
13. ДСТУ – 2293 – 93. Система стандартів безпеки праці. Охорона праці. Терміни та визначення.
14. СН 4088-86 Мікроклімат виробничих приміщень.
15. А.Матросов. HTML 4.
16. М.В. Кузнецов и И.В. Симдянов Учебник PHP
17. Р. Шелдон, Д. Мойе, MySQL: базовый курс - Beginning MySQL. М.: Диалектика, 2007.
18. М. Кузнецов, И. Симдянов, MySQL на примерах. — Спб.: БХВ Петербург, 2008.
19. П.Дюбуа, MySQL 3-е изд. — М.: Вильямс, 2006
20. К. Дари, Э. Баланеску, PHP и MySQL: создание интернет-магазина. М.: Вильямс, 2010.

ДОДАТОК А

Файл Index.php

```
ini_set('display_errors',1);
error_reporting(E_ALL);

session_start();

//Подкл. файловой системы
define('ROOT', dirname(__FILE__));
require_once (ROOT. '/components/Router.php');
require_once (ROOT. '/components/Db.php');

//Вызов Router

$router= new Router();
$router->run();
```

Файл AdminBase.php

```
<?php
include_once ROOT . '/models/user.php';

abstract class AdminBase
{
    public static function checkAdmin()
    {
        $userId = user::checkLogged();

        $user = user::getUserById($userId);
```

```
if($user['role'] == 'admin'){
    return true;
}
die('Access denied');
}
}
```

Файл Db.php

```
<?php
```

```
class Db
{
    public static function getConnection()
    {
        $paramsPath = ROOT . '/config/db_params.php';
        $params = include($paramsPath);

        $dsn = "mysql:host={$params['host']};dbname={$params['dbname']}";
        $db = new PDO($dsn, $params['user'],$params['password']);
        $db->exec("set names utf8");
        return $db;
    }
}
```

Файл Router.php

```
<?php
```

```
class Router
```

```
{
    private $routes;

    public function __construct()
    {
        $routesPath = ROOT.'./config/routes.php';
        $this->routes = include ($routesPath);
    }

    private function getURI()
    {
        if (!empty($_SERVER['REQUEST_URI'])) {
            return trim($_SERVER['REQUEST_URI'], '/');
        }
    }

    /**
     *
     */
    public function run()
    {
        $uri = $this->getURI();

        foreach($this->routes as $uriPattern => $path){

            if (preg_match("~$uriPattern~", $uri)){

                $internalRoute = preg_replace("~$uriPattern~",$path,$uri);
```

```

//Identify controller, action, parameters
$segments = explode('/', $internalRoute);
$controllerName = array_shift($segments).'Controller';
$controllerName = ucfirst($controllerName);

$actionName = 'action'.ucfirst(array_shift($segments));

$parameters = $segments;

//Connect a controller class file
$controllerFile = ROOT . '/controllers/' .
    $controllerName . '.php';

if (file_exists($controllerFile)){
    include_once($controllerFile);
}

//Create object, call the method
$controllerObject = new $controllerName;

$result = call_user_func_array(array($controllerObject, $actionName),
$parameters);

if ($result != null){
    break;
}
}

```



```
}  
}  
  
}
```

Файл AdminController.php

```
<?php  
include_once ROOT . '/components/AdminBase.php';  
  
class AdminController extends AdminBase  
{  
    public function actionIndex()  
    {  
        self::checkAdmin();  
  
        require_once (ROOT . '/views/admin/index.php');  
    }  
}
```

Файл UserController.php

```
<?php  
include_once ROOT . '/models/user.php';  
  
class UserController  
{  
    public static function actionRegister()  
    {  
        $name = "  
        $email = "  
        $password = "
```

```
$password2 = "";
$result = false;

if (isset($_POST['submit'])) {
    $name = $_POST['name'];
    $email = $_POST['email'];
    $password = $_POST['password'];
    $password2 = $_POST['password2'];

    $errors = false;

    if (user::checkName($name)) {
    } else {
        $errors[] = 'Логін не може бути менше 2х символів';
    }

    if (user::checkEmail($email)) {
    } else {
        $errors[] = 'Невірний email';
    }

    if (user::checkPassword($password)) {
    } else {
        $errors[] = 'Невірний пароль ';
    }

    if (user::checkPassword2($password, $password2)) {
    } else {
        $errors[] = 'Паролі не співпадають';
    }
}
```

```
        if (user::checkEmailExists($email)){
            $errors[] = 'Такий email вже створений';
        }

        if ($errors == false){
            $result = user::register($name, $email, $password, $password2);
        }
    }

    require_once (ROOT . '/views/user/register.php');

    return true;
}

public function actionLogin()
{
    $email = '';
    $password = '';

    if (isset($_POST['submit'])){
        $email = $_POST['email'];
        $password = $_POST['password'];

        $errors = false;

        if(!user::checkEmail($email)){
            $errors[] = 'Невірний email';
        }
    }
}
```

```
    }  
    if(!user::checkPassword($password)){  
        $errors[] = 'Невірний пароль';  
    }  
  
    $userId = user::checkUserData($email, $password);  
  
    if ($userId == false){  
        $errors[] = 'Неверные данные';  
    }else{  
        user::auth($userId);  
  
        header("Location: /cabinet/");  
    }  
}  
  
require_once (ROOT . '/views/user/login.php');  
  
return true;  
}  
  
public function actionLogout()  
{  
    unset($_SESSION["user"]);  
    header("Location: /");  
}
```

```
}
```

Файл user.php

```
<?php
```

```
class user
```

```
{
```

```
    public static function register($name, $email, $password, $password2)
```

```
    {
```

```
        $db = Db::getConnection();
```

```
        $sql = 'INSERT INTO user (name, email, password, password2) '
            . 'VALUES (:name, :email, :password, :password2)';
```

```
        $result = $db->prepare($sql);
```

```
        $result->bindParam(':name', $name, PDO::PARAM_STR);
```

```
        $result->bindParam(':email', $email, PDO::PARAM_STR);
```

```
        $result->bindParam(':password', $password, PDO::PARAM_STR);
```

```
        $result->bindParam(':password2', $password2, PDO::PARAM_STR);
```

```
        return $result->execute();
```

```
    }
```

```
    public static function checkName($name)
```

```
    {
```

```
        if (strlen($name)>=2){
```

```
            return true;
```

```
        }
```

```
        return false;
```

```
}  
public static function checkPhone($phone)  
{  
    if (strlen($phone) >= 10) {  
        return true;  
    }  
    return false;  
}
```

```
public static function checkEmail($email)  
{  
    if (filter_var($email, FILTER_VALIDATE_EMAIL)){  
        return true;  
    }  
    return false;  
}
```

```
public static function checkPassword($password)  
{  
    if (strlen($password)>=6){  
        return true;  
    }  
    return false;  
}
```

```
public static function checkPassword2($password,$password2)  
{  
    if ($password == $password2){  
        return true;  
    }
```

```
    }  
    return false;  
}
```

```
public static function checkEmailExists($email)
```

```
{  
    $db = Db::getConnection();  
  
    $sql = 'SELECT COUNT(*) FROM user WHERE email = :email';  
  
    $result = $db->prepare($sql);  
    $result->bindParam(':email', $email, PDO::PARAM_STR);  
    $result->execute();  
  
    if ($result->fetchColumn())  
        return true;  
    return false;  
}
```

```
public static function checkUserData($email, $password)
```

```
{  
    $db = Db::getConnection();  
  
    $sql = 'SELECT * FROM user WHERE email = :email and PASSWORD =  
:password';
```

```
$result = $db->prepare($sql);
$result->bindParam(':email',$email, PDO::PARAM_INT);
$result->bindParam(':password', $password, PDO::PARAM_INT);
$result->execute();

$user = $result->fetch();
if ($user){
    return $user['id'];
}
return false;
}

public static function auth($userId)
{
    $_SESSION['user'] = $userId;
}

public static function checkLogged()
{
    if(isset($_SESSION['user'])){
        return $_SESSION['user'];
    }

    header("Location: /user/login");
}
}
```



```
public static function isGuest()
{
    if (isset($_SESSION['user'])) {
        return false;
    }
    return true;
}
```

```
public static function getUserById($id)
{
    if ($id) {
        $db = Db::getConnection();
        $sql = 'SELECT * FROM user WHERE id = :id';

        $result = $db->prepare($sql);
        $result->bindParam(':id', $id, PDO::PARAM_INT);
        $result->setFetchMode(PDO::FETCH_ASSOC);
        $result->execute();

        return $result->fetch();
    }
}
```

```
public static function edit($id, $name, $password)
{
    $db = Db::getConnection();
    $sql = "UPDATE user
```

```
SET name = :name, password = :password  
WHERE id = :id";
```

```
$result = $db->prepare($sql);  
$result->bindParam(':id',$id, PDO::PARAM_INT);  
$result->bindParam(':name',$name, PDO::PARAM_STR);  
$result->bindParam(':password', $password, PDO::PARAM_STR);  
return $result->execute();  
}  
  
}
```

ДОДАТОК Б

РОЗРОБКА КІБЕРСПОРТИВНОЇ ТУРНІРНОЇ WEB-СИСТЕМИ НА ОСНОВІ MVC НА МОВІ ПРОГРАМУВАННЯ PHP

Виконав: Студент гр. ППЗ-52
Пімкін В.Є.

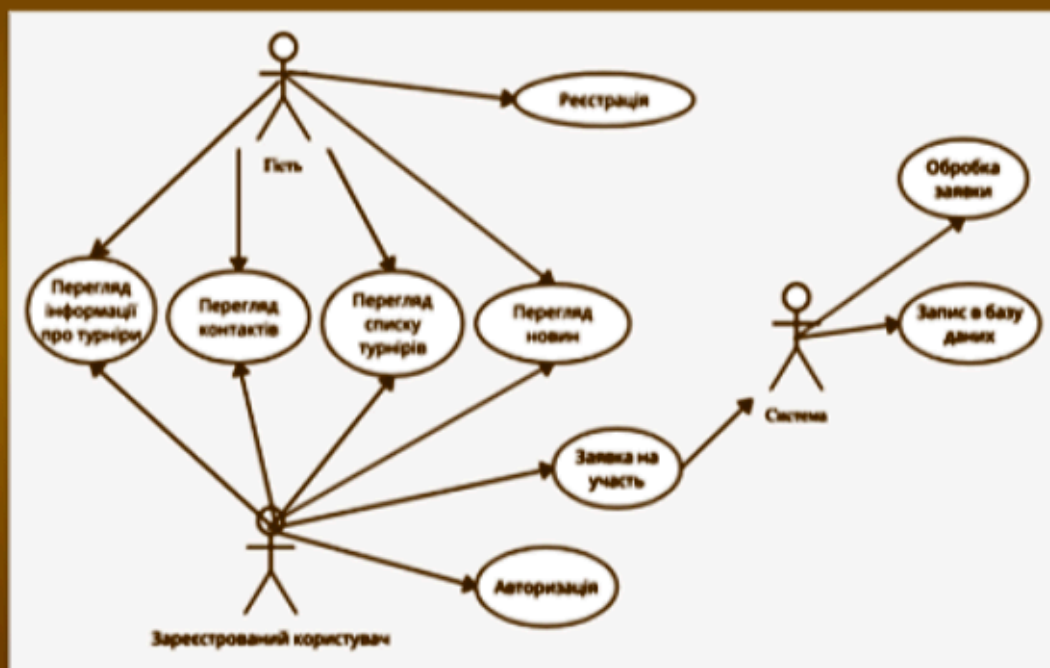
Вступ

- Стрімкий розвиток мережі Інтернет, застосування новітніх технологій і комунікацій у комерційній діяльності і повсякденному житті зумовило виникнення нових економічних явищ, таких як електронна комерція та інші види WEB платформ.
- Одним із таких є кіберспортивні платформи. Кіберспорт – змагання з відеоігор – це вид спортивної активності, в якій гравці розвивають і тренують як розумові, так і фізичні навички. Молоді гравці формують команди та змагаються за призові фонди, що можуть досягати величезних сум. Річний дохід гравців при цьому може складати сотні тисяч доларів, враховуючи спонсорство та призові виграші, і тим паче призові фонди зі змагань. Розвиток технологій, таких як стрімінгові сервіси, дозволяє масовий перегляд змагань, і з більшим залученням спонсорів, інвестицій та сучасних цифрових можливостей, кіберспорт стає дедалі більш привабливим.
-

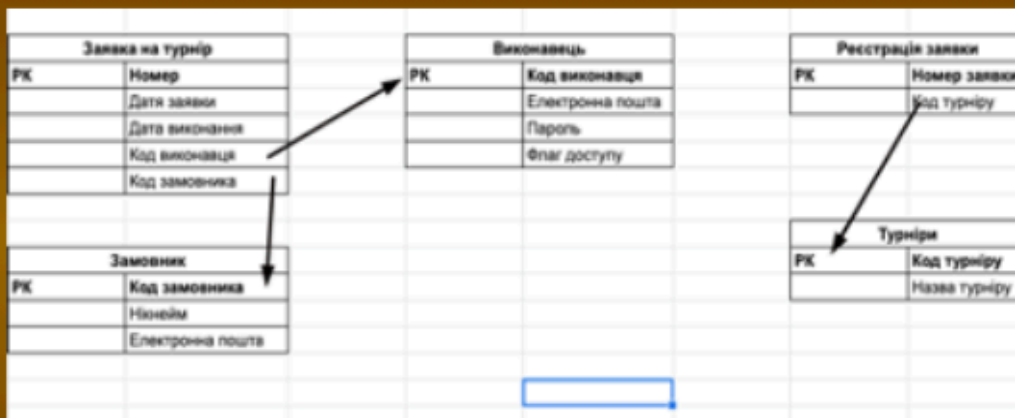
Постановка задачі

- Турнірний веб-сервіс призначений для полегшення створення турнірних змагань з різних дисциплін. Даний веб-сайт надаватиме змогу не тільки приєднатись до вже запланованого турніру, але й створити власний, що надає змогу розвивати проект у маркетинговому напрямку.
- Метою дипломної роботи є розробка сайту для швидкого перегляду та вибору турніру, створення власного, підбору потрібних гравців до команди та перегляду інформації по світовим кіберспортивним змаганням.
- Розроблюваний сайт, повинен відповідати таким вимогам:
 - надавати потенційному користувачеві інформацію про вже створені турніри;
 - наявність системи підбору гравців;
 - надавати Потенційному користувачеві всю необхідну інформацію для створення власного турніру;
 - мати зручну і швидку систему замовлення, реєстрації та пошуку потрібної інформації;
 - можливість додавання, видалення і редагування інформації про турніри;
 - можливість перегляду історії турнірів.

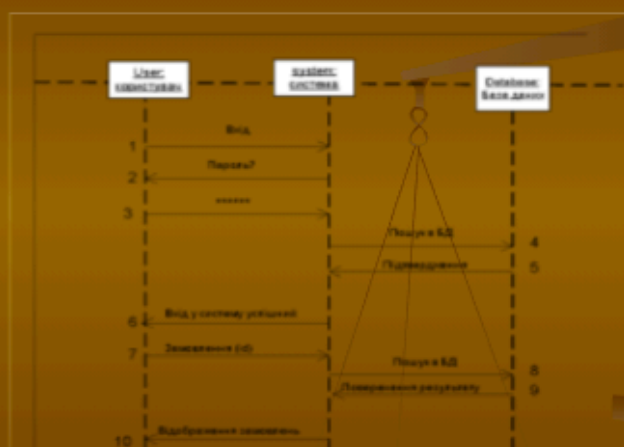
Функціональні можливості системи (UML діаграма прецедентів)



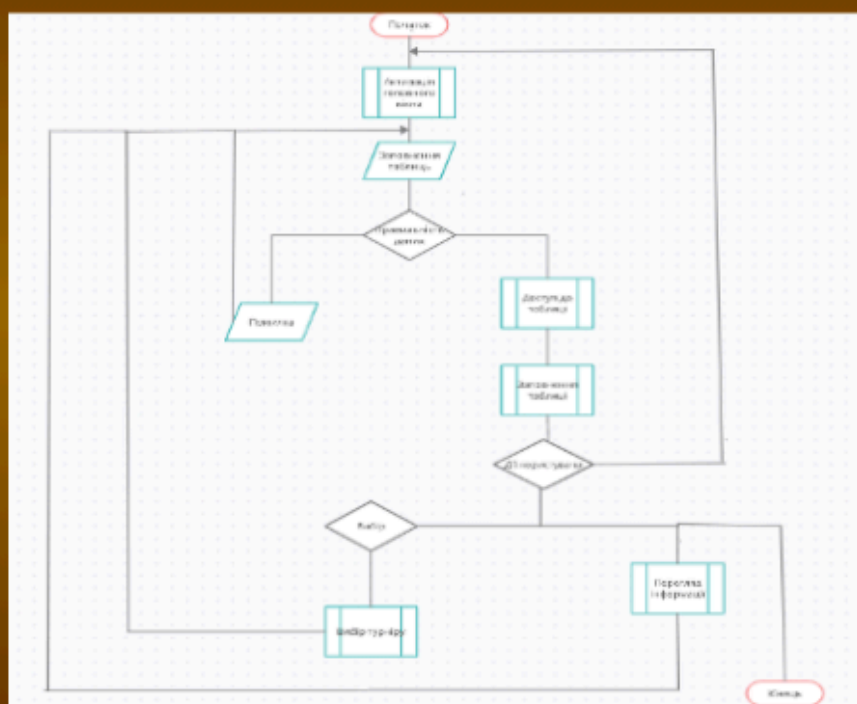
Структура бази даних



Діаграма взаємодії програмної системи



Блок-схема алгоритму можливостей користувача сайтом



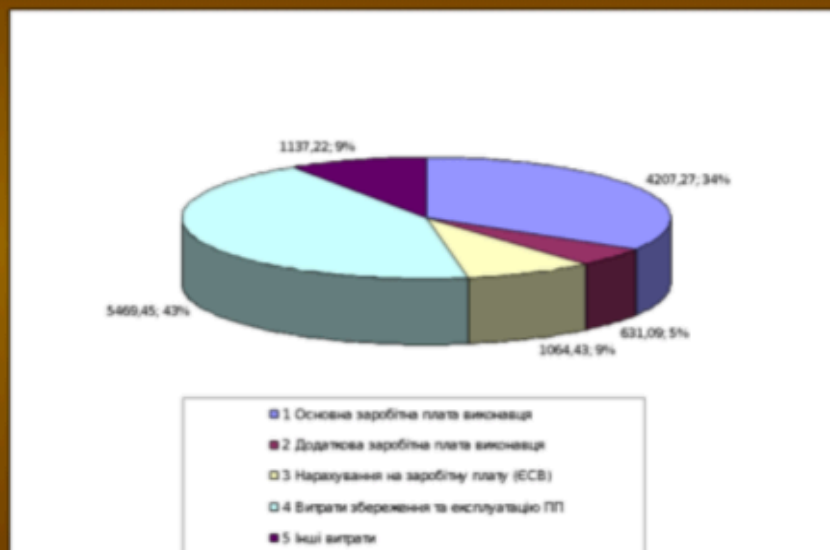
Використані технології

Для коректної роботи програмного продукту необхідна стабільна робота операційної системи типу Windows (XP/Vista/Seven).

При виявленні помилок в програмному продукті або при його вдосконаленні, необхідно встановити програму PHPStorm або звичайний блокнот.

Задля продуктивної роботи бажано встановити програмні компоненти MS Office 2007-2010

Діаграма собівартості програмного продукту



Висновок

- Головна мета дипломної роботи розробити турнірну WEB-систему виконана.
- Даний web-сайт дозволяє потенційним клієнтам ознайомитися з різними новинами, вибирати та створювати турніри. Візуальне подання даних значно полегшує пошук необхідної інформації. Інтерфейс інтерактивного довідника дуже простий і зручний для пошуку.
- При розробці веб-сайту були проаналізовані сучасні веб-технології, що дозволяють створювати інтерактивні веб-сторінки.
- Розроблений сайт задовольняє всім вимогам, поставленим на етапі постановки завдання.
- Як подальше вдосконалення веб-сайта представляється можливим розробка модулів доступу. Так само можливе доопрацювання інтерфейсу сайту з метою подальшого підвищення його інформативності, привабливості і зручності.