

РОЗДІЛ 1
ЗАПИТИ ЗАЦІКАВЛЕНИХ ОСІБ

Вступ

У цьому документі описані запити зацікавлених осіб по відношенню до розроблюваної системи «Організація вуличних змагань WorkOut», в якості яких виступають: організатори цих змагань, судді, учасники та гості(глядачі).

1.1. Мета

Метою документа є визначення основних вимог щодо організації даних спортивних змагань, а саме таких аспектів, як забезпечення інформацією усіх зацікавлених осіб, їх взаємодії за посередництвом системи.

1.2. Контекст

Перелік вимог, перерахованих у цьому документі, є основою технічного завдання для розробки системи для супроводу вуличних спортивних змагань, виставлення балів (оцінок) судьями, забезпечення інформацією учасників та гостей.

1.3. Короткий огляд продукту

Систему «Організація вуличних змагань WorkOut» можна умовно розділити на дві частини програмного забезпечення:

Сайт, який використовують зацікавлені особи в якості інтерфейсу доступу до усіх даних, що пов'язані із змаганнями, інструменту для роботи з ними;

База даних на головному сервері, що відображатиметься на сайті у вигляді турнірної таблиці.

1.4. Ділові правила і приписи

1.4.1. Призначення системи організації вуличних змагань

Система призначена для забезпечення взаємодії між базою даних та сайтом змагань, а також встановлення формальних правил доступу до даних та можливостей роботи з ними для потенційних користувачів системою.

Така організація зберігання даних про учасників, місце, час змагань та іншої інформації виключає можливість втрати інформації або порушення її конфіденційності.

Підп. і дата	
Взаєм. інв. №	
Інв. № дубл.	
Підп. і дата	
Інв. № підп	

									Арк.
									3
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа. »				

Також, використовуючи вищезазначену систему, можна отримати інформацію щодо будь-якого учасника та подальшу його роль у змаганнях.

Отже, завдання системи «Організація вуличних змагань WorkOut» - заміна її потенційного паперового аналогу більш сучасною, доступною і зручною у використанні системою.

1.4.2. Політика взаємодії із зацікавленими особами

При використанні системи «Організація вуличних змагань WorkOut» всі зацікавлені особи мають можливість використовувати програмне забезпечення системи відповідно до їх рівня доступу. Умови доступу до цієї системи встановлює політика конфіденційності.

Організатори мають повний доступ до системи, можуть вносити корективи щодо учасників, місця і часу проведення змагань, нагородження переможців і т.д., за виключенням оцінювання учасників.

Судді мають доступ до всієї інформації про змагання, але вносити зміни мають право лише щодо оцінок.

Учасники реєструються у змаганнях, мають повний доступ до загальної інформації та до власного кабінету.

Гості мають доступ лише до загальної інформації.

1.5. Сценарії

1.5.1. Сценарій реєстрації у змаганнях

Назва: сценарій реєстрації у змаганнях.

Учасники: гість, організатор.

Передумови: гість не зареєстрований у системі.

Результат: зареєстровано нового учасника змагань.

Сценарій:

1. Гість заповнює анкету своїми даними на сайті. Для підтвердження особи він прикріплює фотографію або скан будь-якого документу, що засвідчує особу.

2. Гість надсилає анкету організаторам.

3. Автоматично створюється неактивний профіль учасника.

4. Організатор отримує інформацію.
5. Організатор надсилає на адресу електронної пошти гостя запит про підтвердження реєстрації.
6. Гість підтверджує факт реєстрації у змаганнях.
7. Організатор відкриває базу даних системи та вносить відповідні дані про учасника.
8. Організатор створює для учасника особистий електронний кабінет.
9. Учасник отримує персональний ідентифікаційний номер та пароль для подальшого доступу до особистого кабінету.
10. Профіль учасника автоматично активується.
11. По завершенню роботи організатор закриває базу даних.
12. Організатор перевіряє наявність змін у турнірній таблиці на сайті.

Виключні ситуації:

1. Гість невірно заповнив поля анкети або залишив їх порожніми.
2. Гість не підтвердив факт реєстрації у змаганнях.
3. порушено зв'язок між сервером із базою даних та сайтом.

1.5.2. Сценарій доступу до особистого кабінету

Назва: сценарій доступу до особистого кабінету.

Учасники: учасник.

Передумови: учасник зареєстрований у змаганнях.

Результат: перевірено поточні дані учасника.

Сценарій:

1. Учасник вводить ідентифікаційний номер та пароль.
2. Учасник отримує доступ до особистого кабінету.
3. Автоматично створюється сеанс роботи.

Підп. і дата					
Взаєм. інв. №					
Інв. № дубл.					
Підп. і дата					
Інв. № підп					
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа. » 5

4. Учасник переглядає інформацію щодо своїх успіхів, а також місця і часу проведення наступного етапу змагань, у якому він бере участь.

5. Учасник виходить із особистого кабінету.

6. Сеанс роботи автоматично завершується.

Виключні ситуації:

Учасник невірно ввів ідентифікаційний номер та пароль або залишив поля порожніми.

1.5.3. Сценарій відмови від участі у змаганнях.

Назва: сценарій відмови від участі у змаганнях.

Учасники: учасник, організатор.

Передумови: учасник зареєстрований у змаганнях.

Результат: учасник усунений від участі.

Сценарій:

1. Учасник заходить до особистого кабінету.

2. Учасник надсилає організатору повідомлення про бажання відмовитись від участі у змаганнях.

3. Учасник виходить із особистого кабінету.

4. Організатор отримує повідомлення.

5. Організатор виключає запис учасника із бази даних.

6. Організатор видаляє особистий кабінет учасника.

7. Організатор перевіряє наявність змін у турнірній таблиці на сайті.

Виключні ситуації:

1. Порушено зв'язок між сервером із базою даних та сайтом.

2. Учасник не вийшов з особистого кабінету.

1.5.4. Сценарій адміністрування ресурсів організаторами

Назва: сценарій адміністрування ресурсів організаторами.

Учасники: організатор.

3. Суддя працює з оцінками учасників змагань.
4. Суддя надсилає організатору запит на внесення змін до турнірної таблиці.
5. Суддя закінчує роботу.
6. Сеанс роботи автоматично завершується.

Виключні ситуації:

1. Суддя невірно ввів логін та пароль або залишив поля порожніми.
2. Порушено зв'язок між сайтом та сервером із базою даних.

1.5.6. Сценарій переходу з категорії «учасник» до категорії «організатор».

Назва: сценарій переходу з категорії «учасник» до категорії «організатор».

Учасники: учасник, організатор.

Передумови: учасник бажає організувати власні змагання.

Результат: зареєстровано нового організатора.

Сценарій:

1. Учасник заходить до особистого кабінету.
2. Учасник надсилає організаторам повідомлення, що він бажає стати організатором власних змагань.
3. Учасник виходить з особистого кабінету.
4. Організатор проводить співбесіду з іншими організаторами.
5. Організатор надсилає учасникові підтвердження, персональні логін та пароль для доступу до адміністрування на сайті та дані для доступу до бази даних.
6. Кожному учасникові, який перейшов у категорію організаторів, на перший час надається персональний консультант.

Виключні ситуації:

Підп. і дата						Підп. і дата	Арк.
Взаєм. інв. №						Підп. і дата	Арк.
Інв. № дубл.						Підп. і дата	Арк.
Інв. № підп						Підп. і дата	Арк.
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа. »		8

На співбесіді організатори вирішили відхилити запропоновану кандидатуру.

1.5.7. Сценарій переходу з категорії «учасник» до категорії «суддя».

Назва: сценарій переходу з категорії «учасник» до категорії «суддя».

Учасники: учасник, організатор.

Передумови: учасник бажає бути суддею на змаганнях.

Результат: зареєстровано нового суддю.

Сценарій:

1. Учасник заходить до особистого кабінету.
2. Учасник надсилає організаторам повідомлення, що він бажає стати суддею змагань.
3. Учасник виходить з особистого кабінету.
4. Організатор створює загальне голосування на сайті, у якому можуть брати участь усі зацікавлені особи.
5. Організатор надсилає учасникові підтвердження та персональні логін та пароль для внесення змін до турнірної таблиці.
6. Кожному учасникові, який перейшов у категорію судей, на перший час надається персональний консультант.

Виключні ситуації:

В результаті голосування виявилось, що більшість прости запропонованої кандидатури.

1.6. Функціональність системи

Основні вимоги до функціональності, пред'явлені зацікавленими особами, відносяться до чотирьох категорій :

1. Організатор
2. Суддя
3. Учасник
4. Гість

Підп. і дата						Підп. і дата						Інв. № підп
Взаєм. інв. №												
Інв. № дубл.												
Підп. і дата												
Інв. № підп												
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа. »							Арк.
												9

1.6.1. Структура сайту системи «Організація вуличних змаганьWorkOut»

Сайт системи повинен мати наступні розділи:

1. *Реєстрація* – розділ, в якому будь-який бажаючий зможе зареєструватися у змаганнях.
2. *Вхід до особистого кабінету* – дає можливість переглянути персональну сторінку учасника.
3. *Турнірна таблиця* – інформація про оцінки кожного учасника та відомість про поточне місце учасника в турнірі.
4. *Керування та оцінювання* – розділ виключно для організаторів та суддів, в ньому можна зареєструвати учасника, внести зміни до турнірної таблиці, опублікувати оголошення щодо проведення змагань.
5. *Гостям турніру* – розділ, в якому гість може забронювати сидяче місце на змаганні.
6. *Оголошення* – інформація про адресу та час проведення змагань.
7. *Новини* – інформація про поточний хід подій на змаганнях.
8. *Контакти* – телефони для довідок організаторів змагання, адреса за яким можна звертатися.
9. *Архіви змагань* – інформація про турніри, які вже відбулися.

1.6.2. Можливості зацікавлених осіб

А)Можливості учасника

Реєструватися для участі у змаганнях

Мати доступ до свого особистого кабінету

Надіслати організаторам запит щодо організації власних змагань

Надіслати організаторам запит щодо переходу у категорію судей

Переглядати новини

Переглядати оголошення

Коментувати новини

Переглядати турнірну таблицю

Переглядати архів попередніх змагань

Переглядати контактну інформацію.

Б)Можливості гостя (глядача)

Переглядати новини

Переглядати оголошення

Переглядати архів попередніх змагань

Переглядати контактну інформацію

Реєструватися для участі у змаганнях

Переглядати турнірну таблицю.

В)Можливості організаторів

Мати персональний доступ до керування сайтом

Реєструвати учасників

Публікувати новини

Публікувати оголошення

Переглядати новини

Переглядати оголошення

Вносити зміни до турнірної таблиці

Переглядати турнірну таблицю

Завантажувати архів змагань

Переглядати архів змагань

Публікувати контактну інформацію

Переглядати контактну інформацію.

Г)Можливості суддів

Мати персональний доступ до редагування турнірної таблиці

Переглядати новини

Переглядати оголошення

Вносити зміни до турнірної таблиці

Підп. і дата	
Взаєм. інв. №	
Інв. № дубл.	
Підп. і дата	
Інв. № підп	

Зм	Арк.	№ докум.	Підпис	Дат

Ошибка! Неизвестное имя свойства документа. »

Арк.

11

Переглядати турнірну таблицю

Переглядати архів змагань

Переглядати контактну інформацію.

1.7. Практичність

1.7.1. Стандартизація

Система надає доступ до інформації кожному її користувачеві відповідно до його приналежності до однієї з чотирьох категорій.

1.7.2. Інтерфейс користувача

Інтерфейс сайту системи «Організація вуличних змагань WorkOut» відповідає наступним вимогам:

Зрозумілий і не допускає двозначного тлумачення.

Виконаний з урахуванням ергономічних вимог.

Всі кодовані параметри або елементи, наведені скорочення повинні мати розшифрування або вікно-підказку, що буде з'являтися після наведення курсору на елемент або після натискання спеціальної клавіші.

1.8. Надійність.

Протягом терміну зберігання архівів змагань повинна бути забезпечена їх цілісність.

Протягом терміну проведення одного турніру повинна бути забезпечена недоторканність та достовірність особистих сторінок учасників поточних змагань.

Для забезпечення збереження та цілісності організаторами буде використовуватися метод резервного копіювання.

Для забезпечення незмінності та достовірності організаторами буде використовуватися комплекс технологічних і адміністративних процедур, що перешкоджають випадковій або навмисній зміні збережених даних із бази.

Також повинна бути забезпечена конфіденційність персональної інформації. Надання доступу до персональних даних організатори здійснюють у відповідності з правами доступу відвідувача.

РОЗДІЛ 2 РОЗРОБКА ІНФОРМАЦІЙНОГО ЗАБЕЗПЕЧЕННЯ

2.1. Загальна схема прецедентів

Загальна схема прецедентів для ролі гостя показує можливі послідовності дій даної категорії та встановлює зв'язок дій між різноманітними категоріями. Схема прецедентів представлена на рис. 2.1.

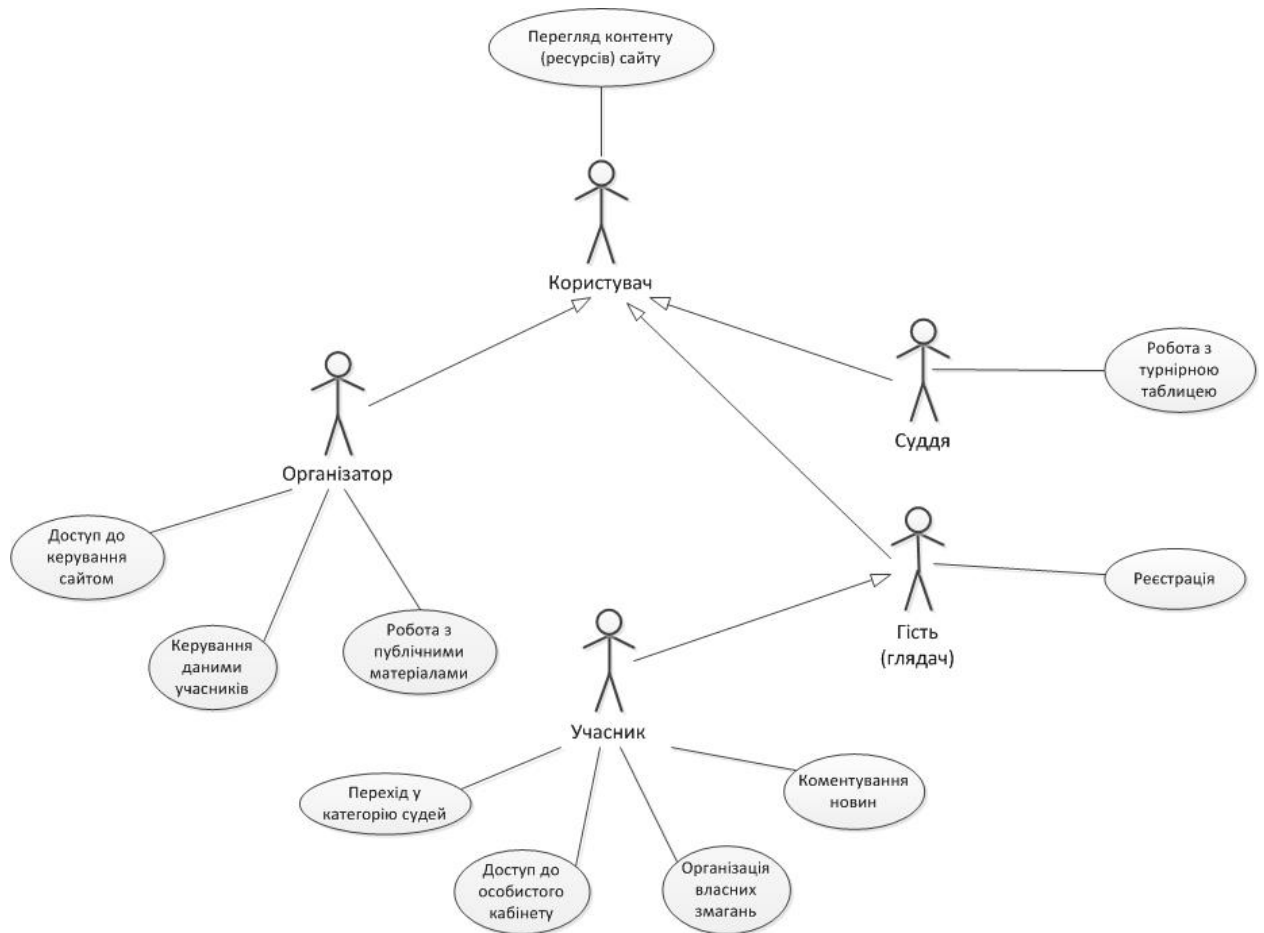


Рис. 2.1 – Загальна схема прецедентів для ролі головного адміністратора

2.2. Прецеденти для ролі гостя

Нижче описані процедури для ролі користувача з вказаними передумовами, результатом, виключними ситуаціями та детальним описом послідовності дій.

Прецедент №1 Реєстрація у змаганнях

Назва: Реєстрація у змаганнях.

Учасники: гість, система.

Передумови: гість не зареєстрований у системі.

Підп. і дата	
Взаєм. інв. №	
Інв. № дубл.	
Підп. і дата	
Інв. № підп	

Зм	Арк.	№ докум.	Підпис	Дат

Результат: зареєстровано нового учасника змагань.

Основний сценарій:

1. Гість заповнює анкету своїми даними на сайті. Для підтвердження особи він прикріплює фотографію або скан будь-якого документу, що засвідчує особу.
2. Гість надсилає анкету організаторам.
3. Автоматично створюється неактивний профіль учасника.
4. Система надсилає на адресу електронної пошти гостя запит про підтвердження реєстрації.
5. Гість підтверджує факт реєстрації у змаганнях.
6. Система створює для учасника особистий електронний кабінет.
7. Учасник отримує персональний ідентифікаційний номер та пароль для подальшого доступу до особистого кабінету.
8. Система активує профіль учасника.

Виключні ситуації:

1. Гість невірно заповнив поля анкети або залишив їх порожніми.
2. Гість не підтвердив факт реєстрації у змаганнях.
3. Порушено зв'язок між сервером із базою даних та сайтом.

Інв. № підп	Підп. і дата	Інв. № дубл.	Взаєм. інв. №	Підп. і дата						Арк.
										14
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа. »					

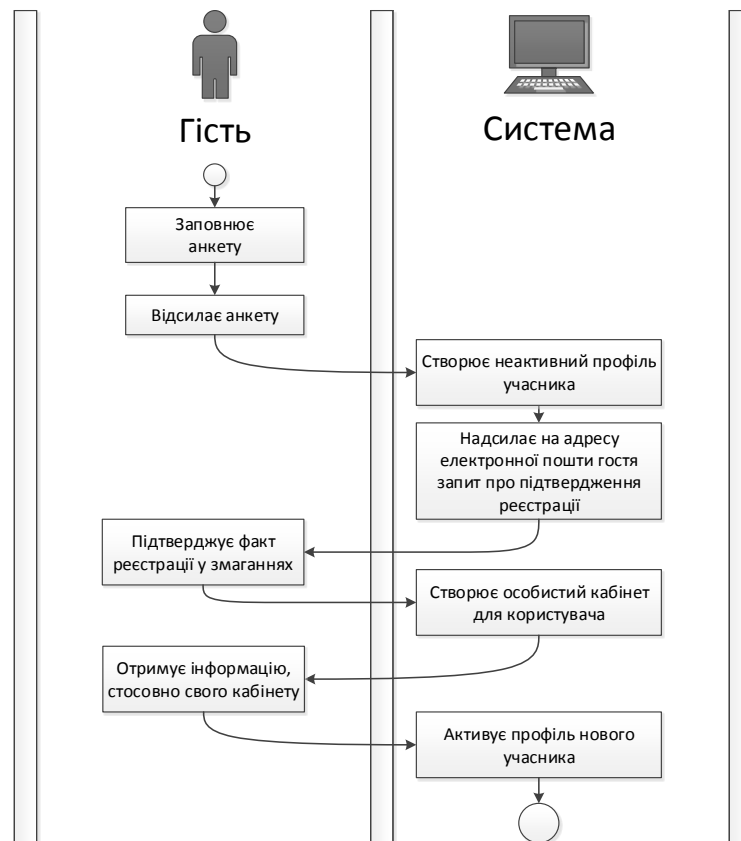


Рис. 2.2 – Схема реєстрації

Прецедент №2. Доступ до особистого кабінету

Назва: Доступ до особистого кабінету.

Учасники: учасник, система.

Передумови: учасник зареєстрований у змаганнях.

Результат: перевірено поточні дані учасника.

Сценарій:

1. Учасник входить в систему.
2. Учасник вводить ідентифікаційний номер та пароль.
3. Учасник отримує доступ до особистого кабінету.
4. Система створює сеанс роботи.
5. Учасник переглядає інформацію щодо своїх успіхів, а також місця і часу проведення наступного етапу змагань, у якому він бере участь.
6. Учасник виходить із особистого кабінету.

7. Система завершує сеанс роботи.

Виключні ситуації:

Учасник невірно ввів ідентифікаційний номер та пароль або залишив поля порожніми.



Рис. 2.3 – Схема доступу до кабінету

Підп. і дата	
Взаєм. інв. №	
Інв. № дубл.	
Підп. і дата	
Інв. № підп	

Зм	Арк.	№ докум.	Підпис	Дат

2.3. Діаграма бізнес-сутностей

Дана діаграма створюється на етапі бізнес моделювання. Вона відображає основні сутності та взаємозв'язки між ними. В даному випадку вона демонструє зв'язок між сутностями: організатор, суддя, гість та учасник. Діаграма бізнес-сутностей проекту зображена на рис. 2.8.

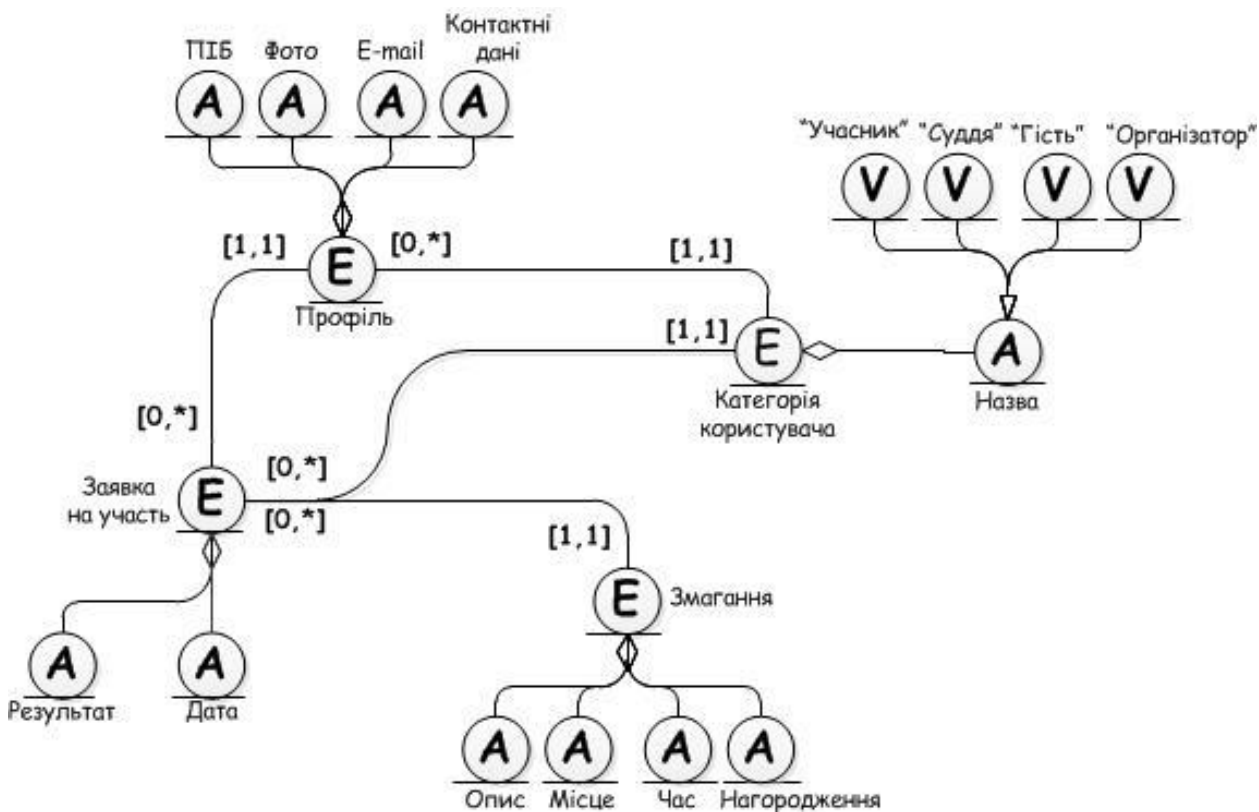


Рис. 2.8 – Діаграма бізнес-сутностей

2.4. Реляційна модель бази даних

Реляційна модель бази даних (рис 2.3) зображує структуру таблиць бази даних, взаємозв'язки між ними та поля кожної з таблиць. Наведена діаграма має багато схожого з діаграмою бізнес-сутностей. Кожній основній бізнес-сутності відповідає таблиця баз даних.

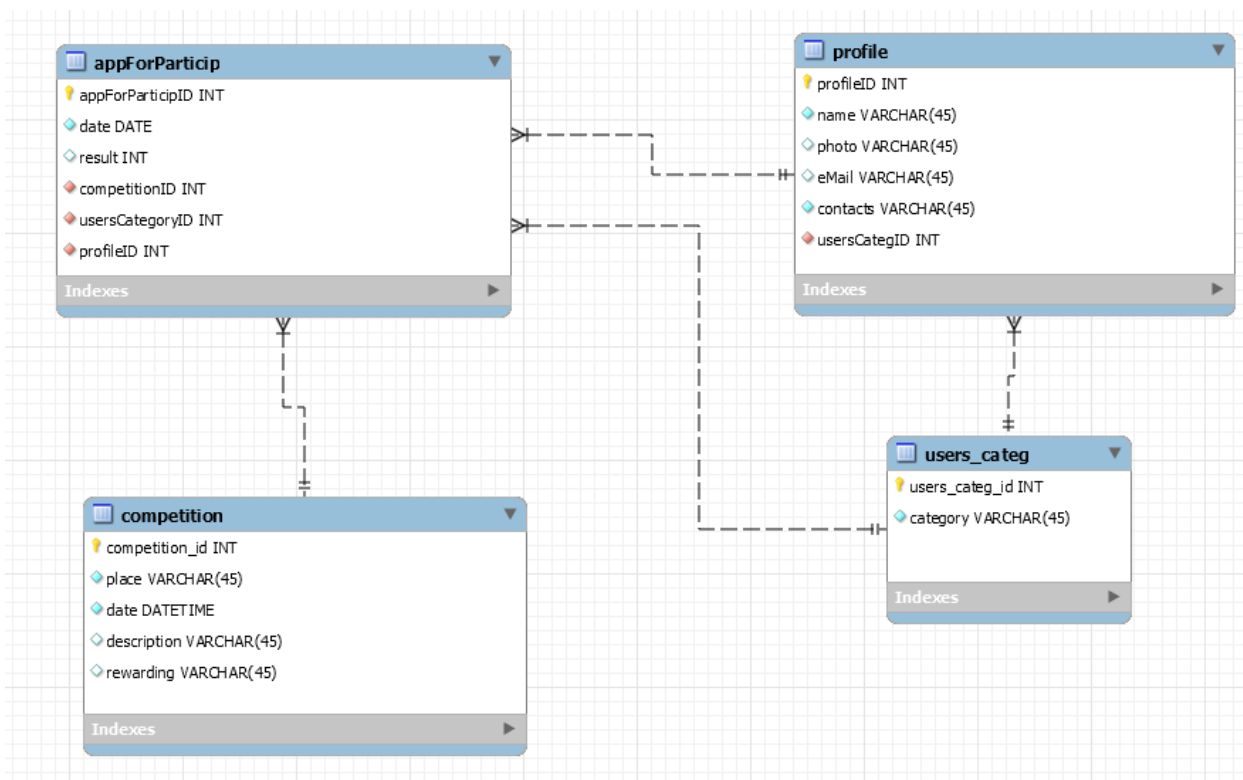


Рис 2.9 – Реляційна модель

2.5. Специфікація таблиць бази даних

Специфікація таблиць бази даних включає в себе інформацію про назви колонок таблиці, їхній тип, інформацію про те, чи є ця колонка первинним ключем, чи поле може бути пустим, чи значення поля автоматично збільшується та коментар щодо призначення колонки. Таблиці зі специфікаціями наведені нижче.

Таблиця 2.1

Таблиця **appForParticip** (application for participation)

appForParticip	date	result	competitionID	usersCategoryID	profileID
1	2014-11-15	NULL	1	1	1
2	2014-11-19	NULL	2	1	1
3	2014-11-15	NULL	1	2	2
4	2014-11-19	NULL	2	2	2
5	2014-11-15	NULL	1	3	3
6	2014-11-19	NULL	2	3	3
7	2014-11-15	NULL	1	4	4
8	2014-11-19	NULL	2	4	4

Таблиця 2.2

Таблиця **Profile**

profileID	name	photo	e-mail	contacts	UsersCategID
1	Pavluchkov Vlad	NULL	NULL	0981223425	1
2	Zmeul Evgeniy	NULL	NULL	0671234569	2
3	Morozov Maks	NULL	NULL	0671234565	3
4	Popenko Ruslan	NULL	NULL	0984582499	4

Таблиця 2.3

Таблиця **competition**

competition_id	Place	Date	Description	Rewarding
1	Stadium Start	2014-11-15 19:00:00	NULL	NULL
2	Metro Gidropark bus station	2014-11-19 19:00:00	NULL	NULL

Таблиця 2.4

Таблиця **users_categ**

users_categ_id	category
1	participant
2	judge
3	guest
4	organizer

<div>Підп. і дата</div> <div>Взаєм. інв. №</div> <div>Інв. № дубл.</div> <div>Підп. і дата</div> <div>Інв. № підп</div>						<div>Ошибка! Неизвестное имя свойства документа. »</div> <div>Арк.</div> <div>19</div>
	Зм	Арк.	№ докум.	Підпис	Дат	

РОЗДІЛ 3

РОЗРОБКА ПРОГРАМНОГО ПРОДУКТУ

3.1. Реляційно-об'єктне відображення

Для реляційно-об'єктного відображення в програмі використовується бібліотека Hibernate. Вона надає можливість легко встановити зв'язок з будь-якою базою даних та створити відображення між об'єктно-орієнтованою моделлю та традиційною реляційною моделлю баз даних. На рис. 3.1 зображено діаграму Entity класів. Детальна специфікація (JavaDoc) наведена нижче.

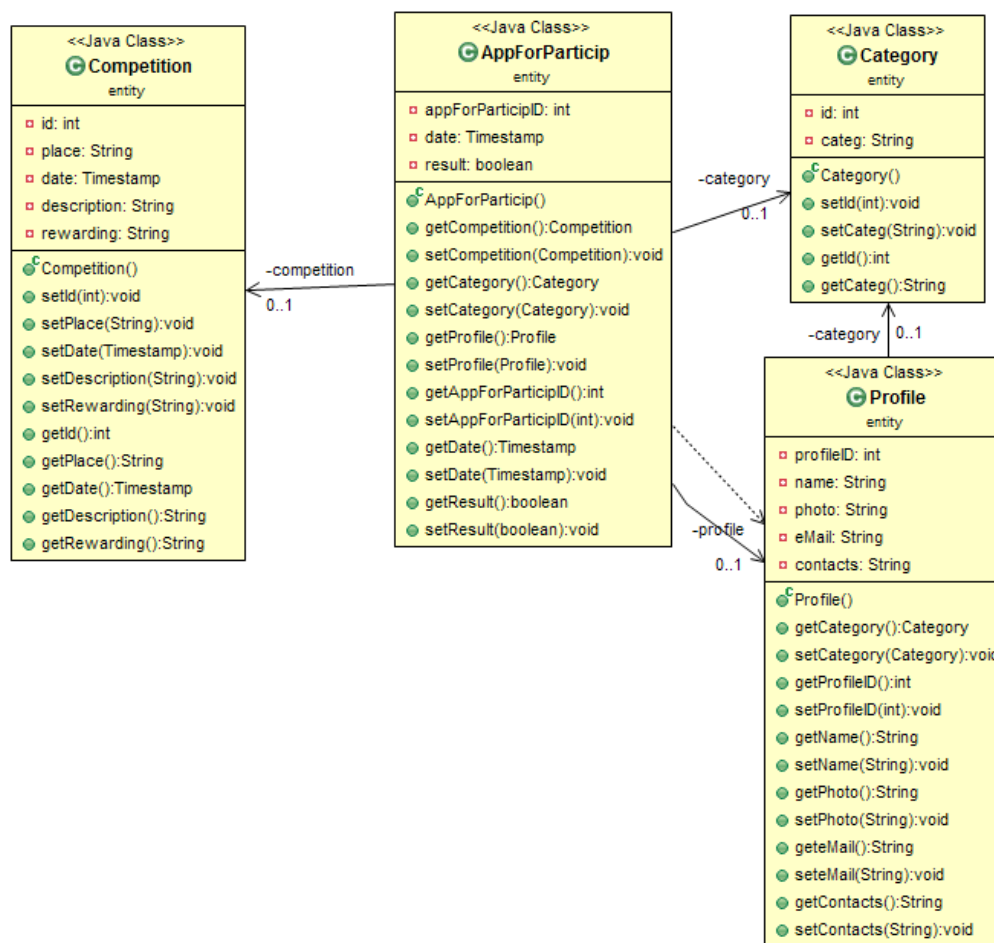


Рис 3.1 – Діаграма entity класів

3.1.1. Клас «AppForParticip»

```
@Entity
public class AppForParticip
extends java.lang.Object

Клас представляє сутність Заявка на участь

Since:
2015-04-14
Version:
1.0
Author:
Руслан Попенко
```

Field Summary

Fields	
Modifier and Type	Field and Description
private int	appForParticipID Ідентифікатор заявки на участь
private Category	category Ідентифікатор категорії користувача
private Competition	competition Ідентифікатор змагання
private java.sql.Timestamp	date Дата створення заявки
private Profile	profile Ідентифікатор профілю
private boolean	result Результат реєстрації заявки

Constructor Summary

Constructors	
Constructor and Description	
AppForParticip()	

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type	Method and Description	
int	getAppForParticipID()	
Category	getCategory()	
Competition	getCompetition()	
java.sql.Timestamp	getDate()	
Profile	getProfile()	
boolean	getResult()	
void	setAppForParticipID(int appForParticipID)	
void	setCategory(Category category)	
void	setCompetition(Competition competition)	
void	setDate(java.sql.Timestamp date)	
void	setProfile(Profile profile)	
void	setResult(boolean result)	

3.1.2. Клас «Category»

```
@Entity
public class Category
extends java.lang.Object
```

Клас представляє сутність Категорія

Since:
2015-04-14

Version:
1.0

Author:
Руслан Попенко

Field Summary

Fields

Modifier and Type	Field and Description
private java.lang.String	categ Категорія користувача
private int	id Ідентифікатор категорії

Constructor Summary

Constructors

Constructor and Description
Category()

Method Summary

All MethodsInstance MethodsConcrete Methods

Modifier and Type	Method and Description
java.lang.String	getCateg()
int	getId()
void	setCateg(java.lang.String categ)
void	setId(int id)

3.1.3. Клас «Competition»

```
@Entity  
public class Competition  
extends java.lang.Object
```

Клас представляє сутність Змагання

Since:
2015-04-14

Version:
1.0

Author:
Руслан Попенко

Fields

Modifier and Type	Field and Description
private java.sql.Timestamp	date Дата змагання
private java.lang.String	description Опис змагання
private int	id Ідентифікатор змагання
private java.lang.String	place Місце змагання
private java.lang.String	rewarding Нагородження учасників змагання

Method Summary

All Methods	Instance Methods	Concrete Methods
Modifier and Type		Method and Description
java.sql.Timestamp		<code>getDate()</code>
java.lang.String		<code>getDescription()</code>
int		<code>getId()</code>
java.lang.String		<code>getPlace()</code>
java.lang.String		<code>getRewarding()</code>
void		<code>setDate(java.sql.Timestamp date)</code>
void		<code>setDescription(java.lang.String description)</code>
void		<code>setId(int id)</code>
void		<code>setPlace(java.lang.String place)</code>
void		<code>setRewarding(java.lang.String rewarding)</code>

3.1.4. Клас «Profile»

```
@Entity
public class Profile
extends java.lang.Object
```

Клас представляє сутність Профіль

Since:

2015-04-14

Version:

1.0

Author:

Руслан Попенко

Fields	
Modifier and Type	Field and Description
private Category	category Ідентифікатор категорії
private java.lang.String	contacts Контактний телефон
private java.lang.String	eMail Пошта
private java.lang.String	name Ім'я
private java.lang.String	photo Фото
private int	profileID Ідентифікатор профіля

Constructor Summary

Constructors
Constructor and Description
Profile()

3.2. Специфікація HibernateUtil класу

Клас HibernateUtil слугує для того, щоб при взаємодії із файлами конфігурації створювати hibernate-сесію. Діаграму цього інтерфейса можна побачити на рис. 3.2. Детальна специфікація (JavaDoc) наведена нижче.

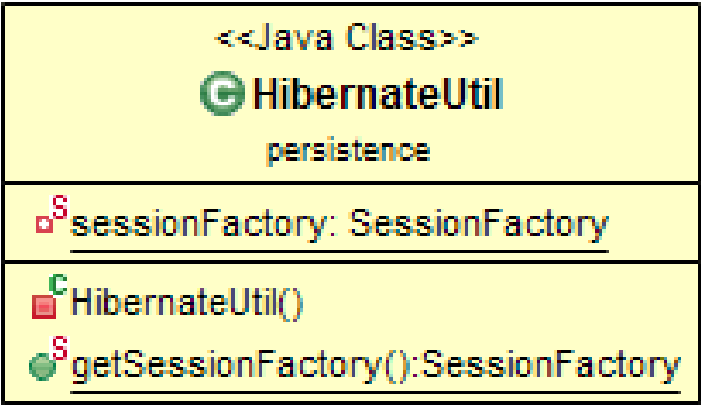


Рис 3.2. – Діаграма HibernateUnit класу

3.2.1. Клас «HibernateUtil»

```
public class HibernateUtil
extends java.lang.Object
```

Клас для взаємодії з конфіг файлами і створення об'єкту SessionFactory,котрий відповідає за створення hibernate-сесії

Since:
2015-04-14

Version:
1.0

Author:
Руслан Попенко

Field Summary

Fields

Modifier and Type	Field and Description
private static org.hibernate.SessionFactory	sessionFactory об'єкт SessionFactory,котрий відповідає за створення hibernate-сесії

Constructor Summary

Constructors

Modifier	Constructor and Description
private	HibernateUtil()

Method Summary

All MethodsStatic MethodsConcrete Methods

Modifier and Type	Method and Description
static org.hibernate.SessionFactory	getSessionFactory() Повертає об'єкт SessionFactory

3.3. Специфікація DAO-класів

Класи, що тут представлені, містять методи для роботи з базою даних. Ці методи використовують бібліотеку Hibernate. Діаграму цих класів можна побачити на рис. 3.3. Детальна специфікація наведена в додатку В.

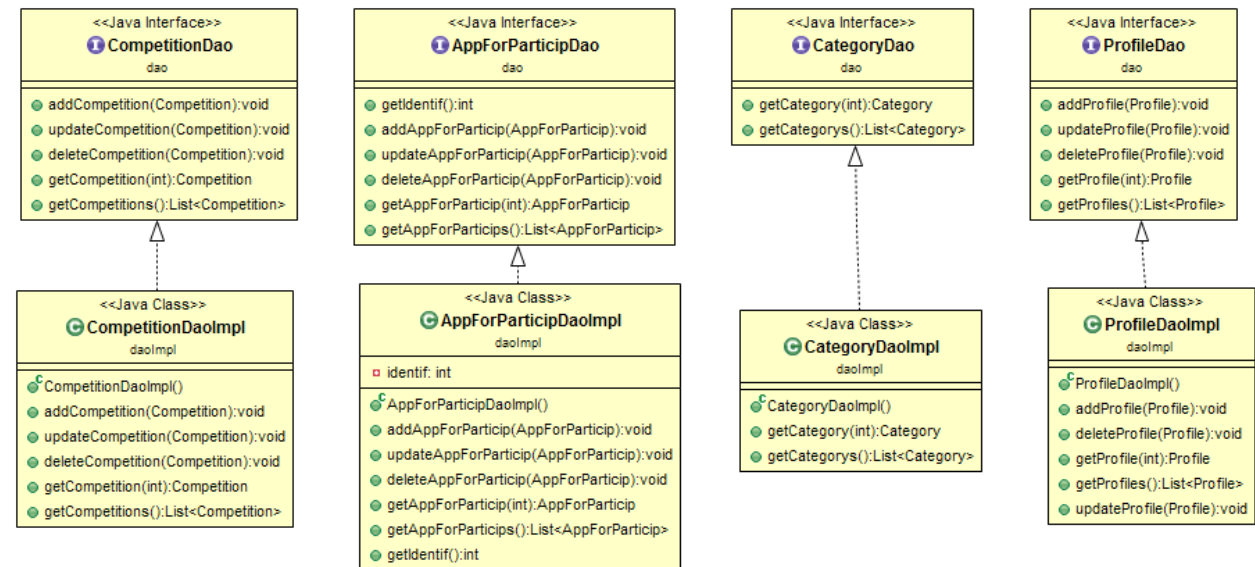


Рис 3.3 – Діаграма DAO класів

3.4. Класи контролерів та їх специфікація

Дані класи призначені для створення зв'язку між сервером та клієнтом. Діаграму цих класів можна побачити на рис. 3.4. Детальна специфікація наведена в додатку В.

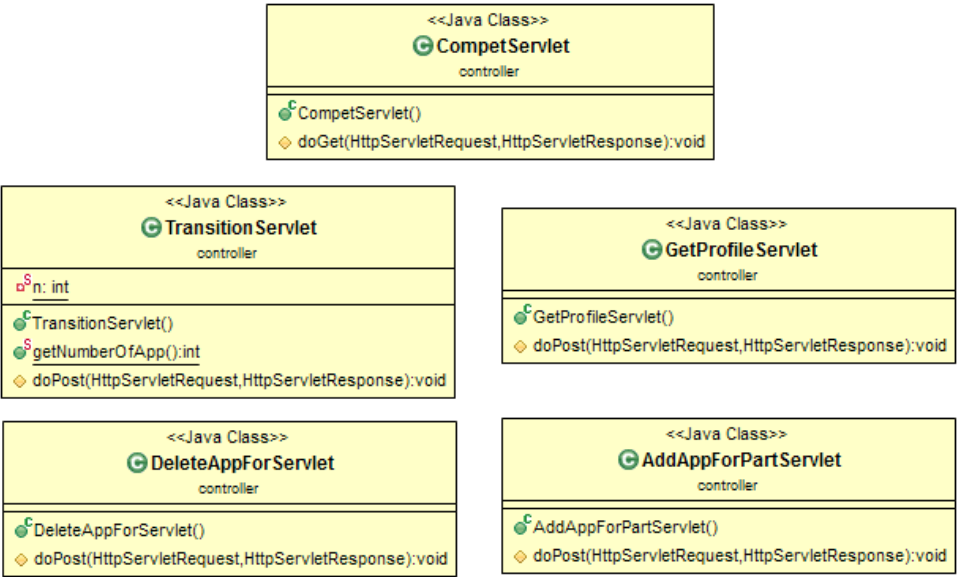


Рис. 3.4 – Діаграма класів контролерів

Підп. і дата
Взаєм. інв. №
Інв. № дубл.
Підп. і дата
Інв. № підп

Зм	Арк.	№ докум.	Підпис	Дат
----	------	----------	--------	-----

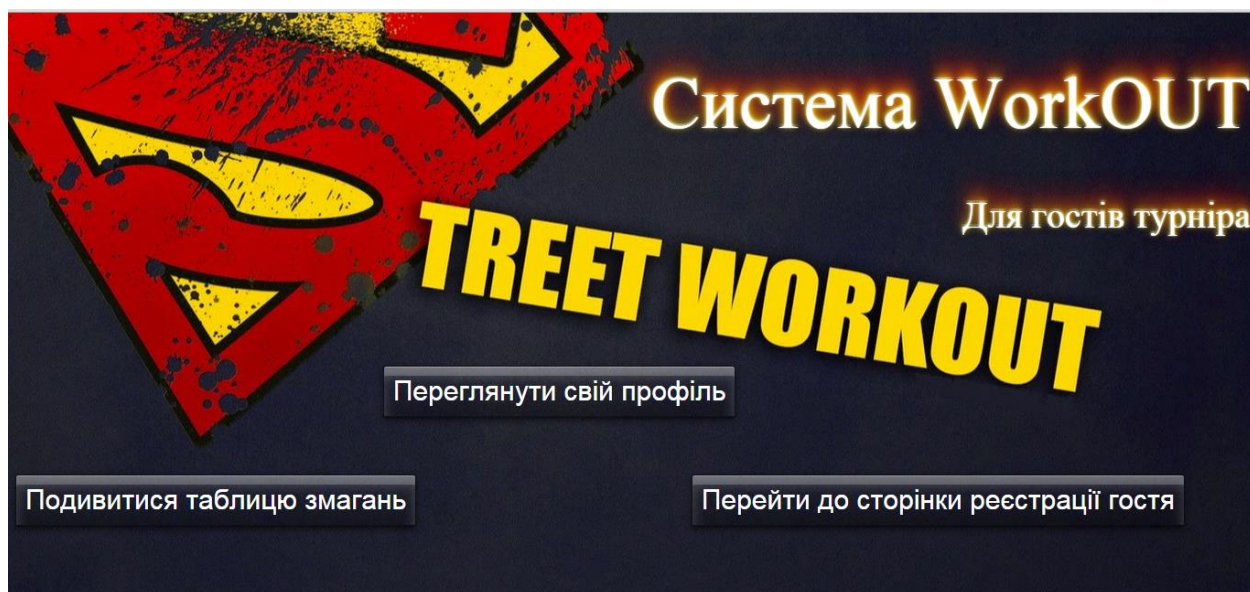
РОЗДІЛ 4

ІЛЮСТРАЦІЯ РОБОТИ ПРОГРАМИ

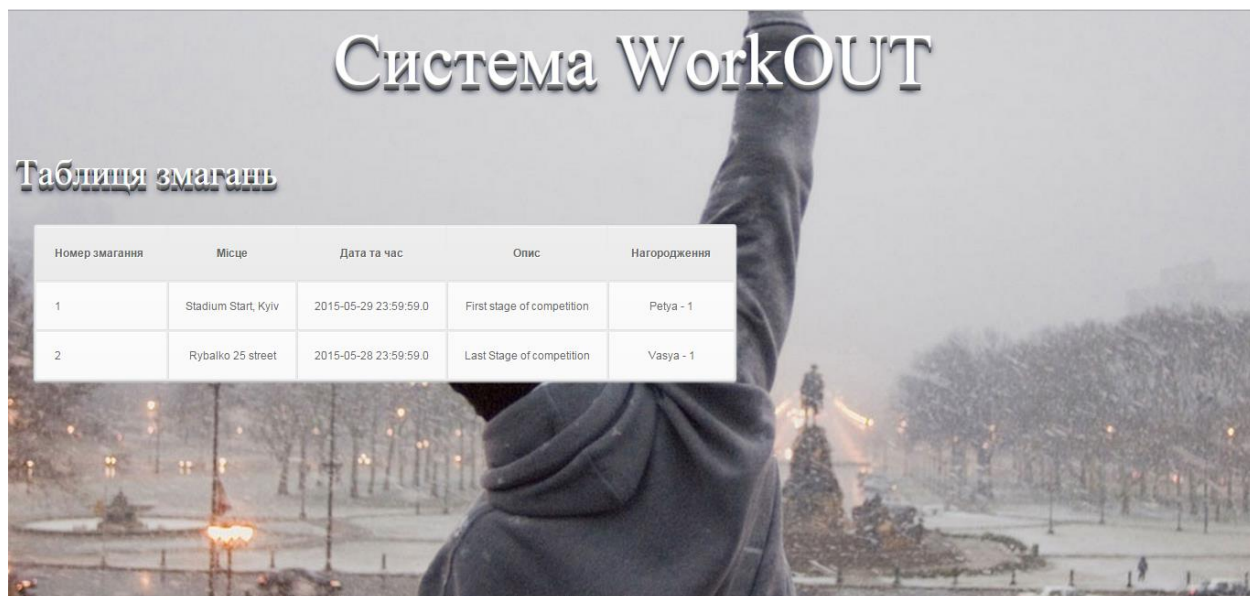
Для ілюстрації роботи програми в цьому розділі наведено графічні сценарії роботи проекту.

4.1. Взаємодія гостя турніра і системи.

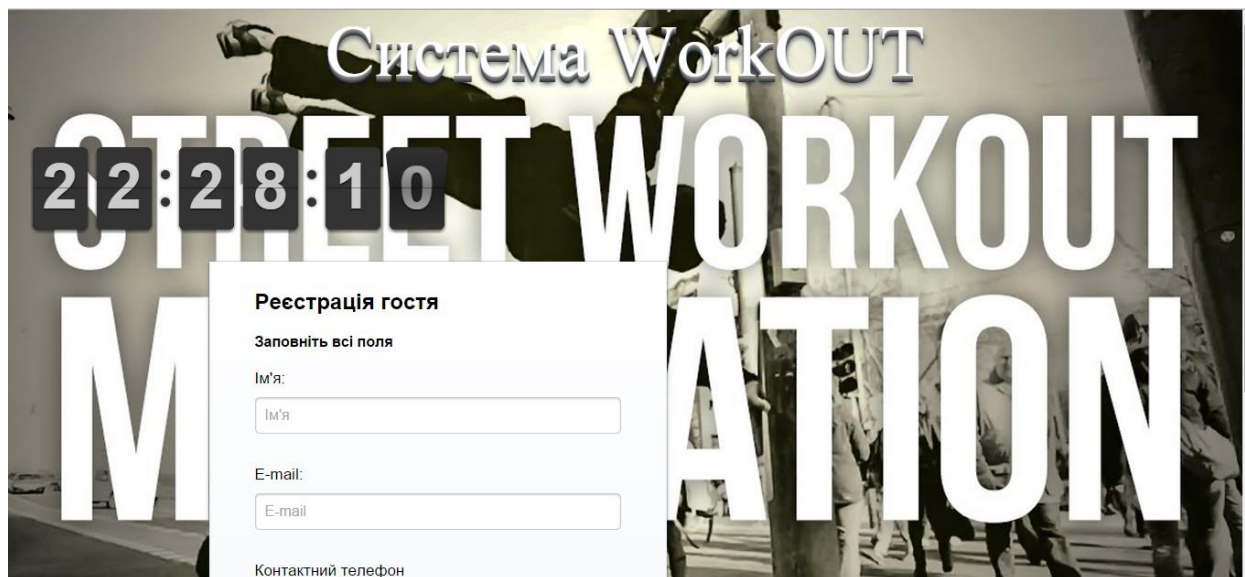
4.1.1. Головна сторінка, на якій можна зробити вибір, яку дію потрібно зробити користувачу.



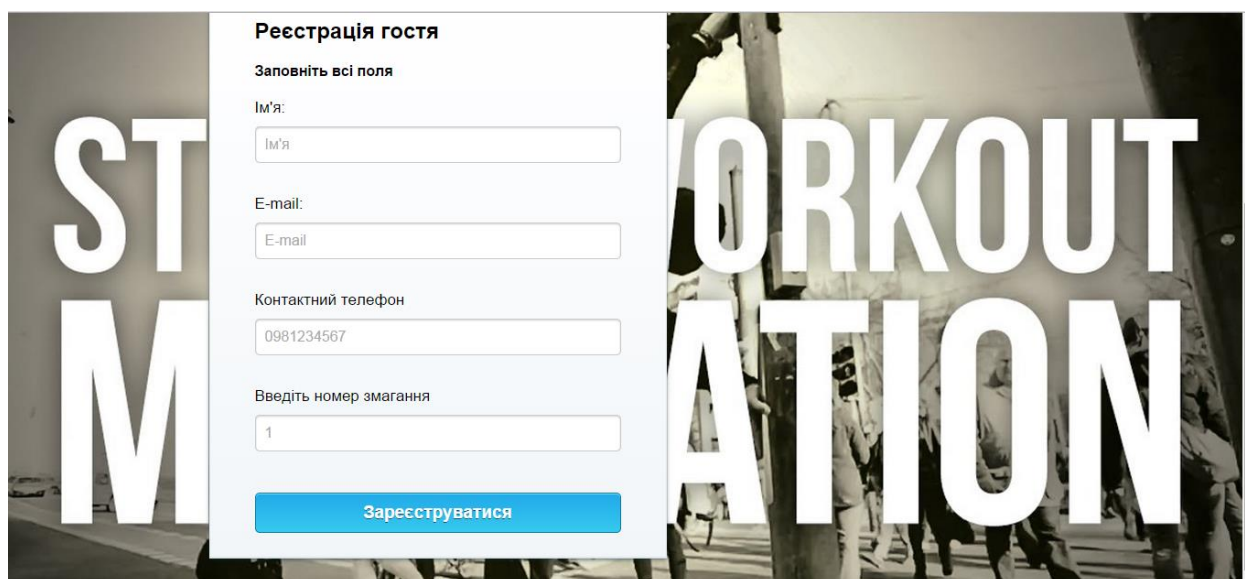
4.1.2. Перегляд турнірної таблиці.



4.1.3. Реєстрація гостя. На сторінці зроблений годинник, для того щоб користувач міг подивитися час реєстрації.



(продовження форми)



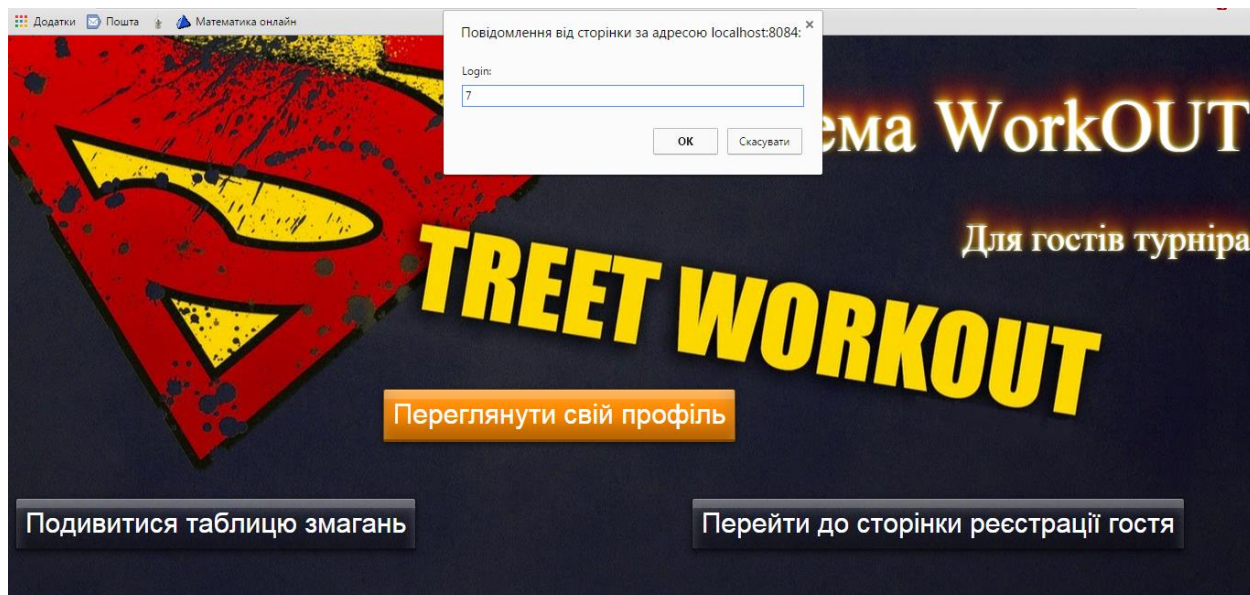
4.1.4. Перегляд заявки і профіля гостя.

4.1.4.1. Спочатку треба пройти авторизацію

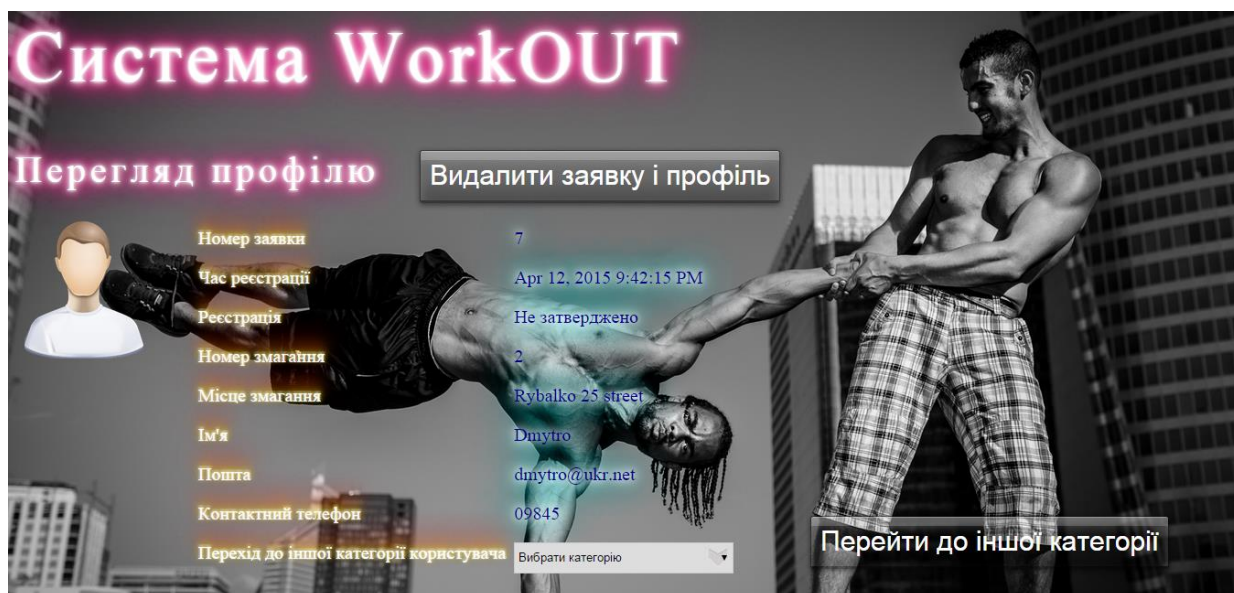
Інв. № підп	Підп. і дата	Інв. № дубл.	Взаєм. інв. №	Підп. і дата

Зм	Арк.	№ докум.	Підпис	Дат

Ошибка! Неизвестное имя свойства документа.»



4.1.4.2. Потім з'явиться інформація про гостя.



Підп. і дата	
Взаєм. інв. №	
Інв. № дубл.	
Підп. і дата	
Інв. № підп	

									Арк.
Зм	Арк.	№ докум.	Підпис	Дат	Ошибка! Неизвестное имя свойства документа.»				28

СПИСОК ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Apache Tomcat. – Посилання: <http://tomcat.apache.org>
2. Hibernate. – Посилання: <http://hibernate.org>
3. Maven. – Посилання: <https://maven.apache.org>
4. Git. – Посилання: <http://git-scm.com>
5. GitHub. – Посилання: <https://github.com>

[illegible]