Міністерство освіти і науки України

Національний технічний університет України

«Київський політехнічний інститут»

Факультет інформатики та обчислювальної техніки

Кафедра обчислювальної техніки

**ПРОЕКТ**

**на тему**

**«*Система організації вуличних змагань WorkOut*»**

Дисципліна: Організація баз даних

Підготував: студент підгрупи № 4 (ІІ курс, група ІО-32)   
Попенко Руслан   
Перевірив: Болдак А. О.

Київ, 2014 р.

**Перелік документів**

1. Запити зацікавлених осіб
2. Прецеденти

а) Діаграми прецедентів

б) Сценарії виконання

1. Моделі:

а) Бізнес-модель системи

б) Діаграма класів

в) Реляційна модель представлення даних

4. Словник предметної сфери

1. База даних:

а) Автоматично згенеровані запити

б) Заповнення таблиць даними

в) Результати заповнення таблиць

1. Управління базою даних за допомогою JDBC

а) Код класу Test

б) Результати виконання програми

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. **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. Короткий огляд продукту

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

1. Сайт, який використовують зацікавлені особи в якості інтерфейсу доступу до усіх даних, що пов’язані із змаганнями, інструменту для роботи з ними;
2. База даних на головному сервері, що відображатиметься на сайті у вигляді турнірної таблиці.
3. Контекст

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

1. Ділові правила і приписи
   1. Призначення системи «Організація вуличних змагань WorkOut»

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.

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. Політика взаємовідносин із зацікавленими особами

При використанні системи «Організація вуличних змагань WorkOut» всі зацікавлені особі 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. Сценарії
     1. 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. **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.
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:

1. Учасник невірно ввів ідентифікаційний номер та пароль або залишив поля порожніми.
   * 1. **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. **Назва:** сценарій адміністрування ресурсів організаторами.

**Учасники:** організатор.

**Передумови:** організатор має доступ до адміністрування.

**Результат:** внесено зміни до даних на сайті.

1. Організатор вводить логін та пароль для доступу до бази даних.
2. Організатор вносить зміни до бази даних відповідно до повідомлень з боку учасників та судей.
3. Організатор завершує роботу та залишає сервер з базою даних.
4. Організатор вводить шестизначний логін та персональний пароль для доступу до адміністрування на сайті.
5. Автоматично створюється сеанс роботи.
6. Організатор працює з ресурсами сайту відповідно до своїх прав доступу.
7. Організатор закінчує роботу.
8. Сеанс роботи автоматично завершується.

Виключні ситуації:

1. Організатор невірно ввів логін та пароль або залишив поля порожніми.
2. Порушено зв’язок між сервером із базою даних та сайтом.
   * 1. **Назва:** сценарій адміністрування ресурсів суддями.

**Учасники:** суддя.

**Передумови:** суддя має частковий доступ до адміністрування.

**Результат:** внесено зміни до турнірної таблиці.

1. Суддя вводить шестизначний логін та персональний пароль для отримання часткового доступу до адміністрування на сайті.
2. Автоматично створюється сеанс роботи.
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. **Назва:** сценарій переходу з категорії «учасник» до категорії «організатор».

**Учасники:** учасник, організатор.

**Передумови:** учасник бажає організувати власні змагання.

**Результат:** зареєстровано нового організатора.

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:**

1. На співбесіді організатори вирішили відхилити запропоновану кандидатуру.
   * 1. **Назва:** сценарій переходу з категорії «учасник» до категорії «суддя».

**Учасники:** учасник, організатор.

**Передумови:** учасник бажає бути суддею на змаганнях.

**Результат:** зареєстровано нового суддю.

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:**

1. В результаті голосування виявилося, що більшість прости запропонованої кандидатури.
2. Функціональність

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

1. Організатор
2. Суддя
3. Учасник
4. Гість
   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. **Можливості зацікавлених осіб**
      1. 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.
  + 1. Можливості гостя (глядача)
* Переглядати новини
* Переглядати оголошення
* Переглядати архів попередніх змагань
* Переглядати контактну інформацію
* Реєструватися для участі у змаганнях
* Переглядати турнірну таблицю.
  + 1. 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.
  + 1. Можливості суддів
* Мати персональний доступ до редагування турнірної таблиці
* Переглядати новини
* Переглядати оголошення
* Вносити зміни до турнірної таблиці
* Переглядати турнірну таблицю
* Переглядати архів змагань
* Переглядати контактну інформацію.

1. Практичність
   1. Стандартизація

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

* 1. Інтерфейс користувача

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

1. Зрозумілий і не допускає двозначного тлумачення.
2. Виконаний з урахуванням ергономічних вимог.
3. 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.
4. 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.

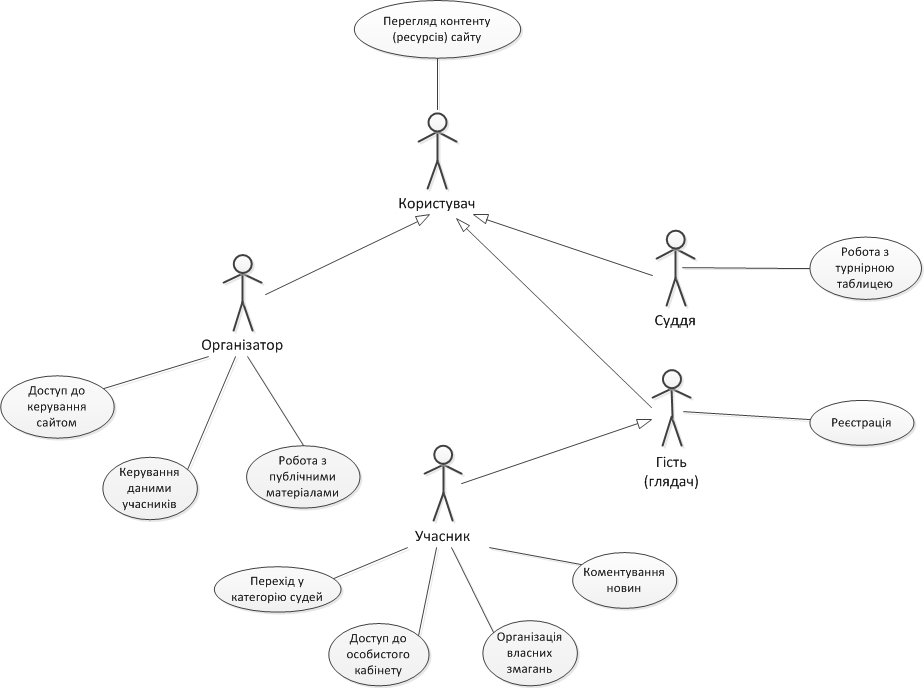
Для забезпечення збереження та цілісності організаторами буде використовуватися метод резервного копіювання.

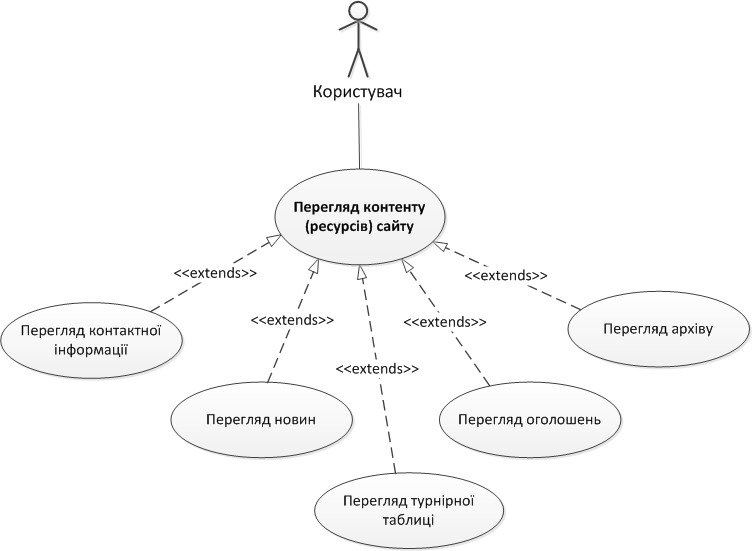
Для забезпечення незмінності та достовірності організаторами буде використовуватися комплекс технологічних і адміністративних процедур, що перешкоджають випадковій або навмисній зміні збережених даних із бази.

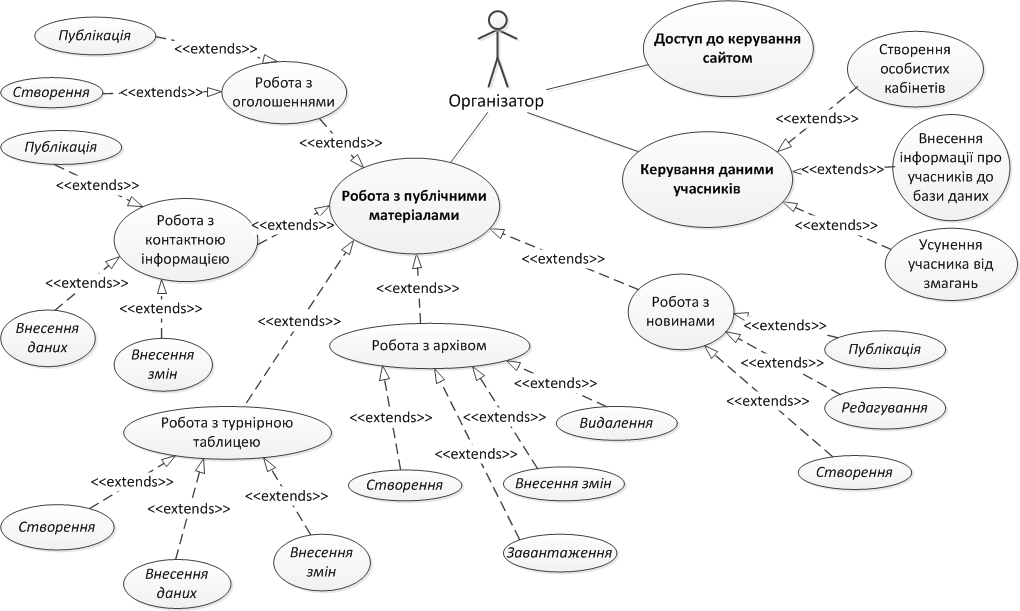
Також повинна бути забезпечена конфіденційність персональної інформації. Надання доступу до персональних даних організатори здійснюють у відповідності з правами доступу відвідувача.

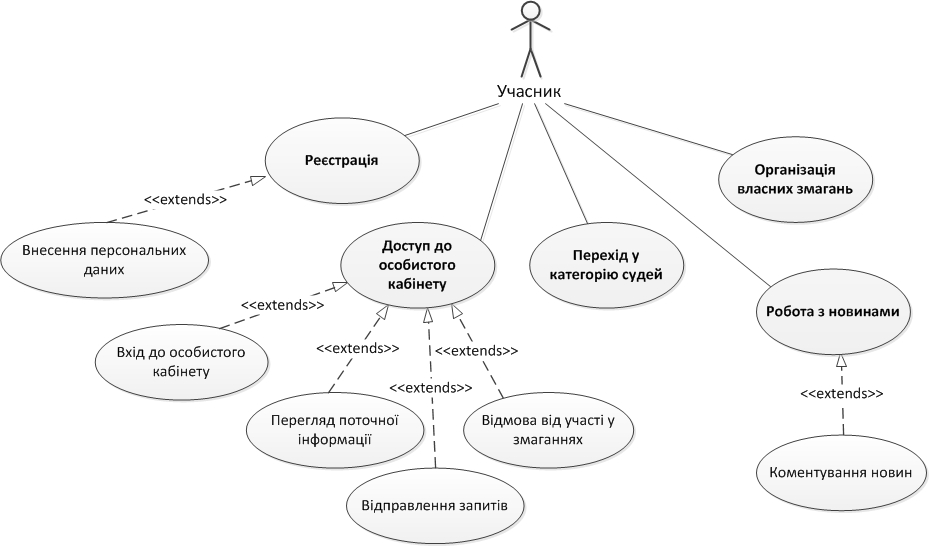
**Прецеденти**

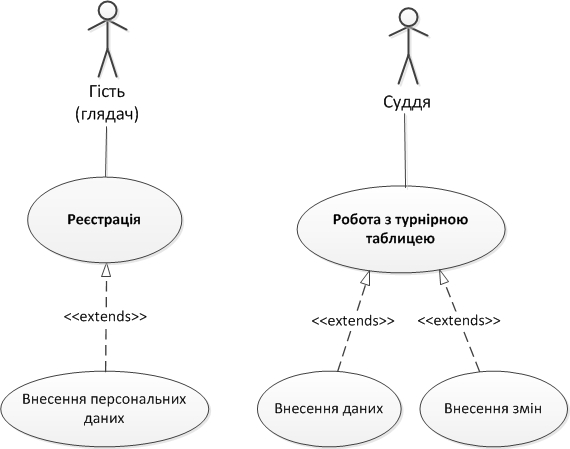
A) Діаграми прецедентів











Б) Сценарії виконання

ID: *UC 001*

Назва: Реєстрація у змаганнях.

Учасники: гість, система.

Передумови: гість не зареєстрований у системі.

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.



ID: *UC 002*

Назва: Реєстрація у змаганнях.

Учасники: організатор, система.

Передумови: гість не зареєстрований у системі.

Результат: зареєстровано нового учасника змагань.

Основний сценарій:

1. Організатор входить в систему.
2. Система надсилає інформацію про гостя, бажаючого стати учасником.
3. Організатор отримує інформацію.
4. Організатор надсилає на адресу електронної пошти гостя запит про підтвердження реєстрації.
5. Організатор надсилає на адресу електронної пошти гостя запит про підтвердження реєстрації.
6. Система надсилає організатору відповідь гостя.
7. Організатор відкриває базу даних системи та вносить відповідні дані про учасника.
8. Організатор створює для учасника особистий електронний кабінет.
9. По завершенню роботи організатор закриває базу даних.
10. 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.



ID: *UC 003*

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:

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



ID: *UC 004*

Назва: Адміністрування ресурсів організаторами.

Учасники: організатор, система.

Передумови: організатор має доступ до адміністрування.

Результат: внесено зміни до даних на сайті.

1. Організатор заходить в систему.
2. Організатор вводить логін та пароль для доступу до бази даних.
3. Організатор вносить зміни до бази даних відповідно до повідомлень з боку учасників та судей.
4. Організатор завершує роботу та залишає сервер з базою даних.
5. Організатор вводить шестизначний логін та персональний пароль для доступу до адміністрування на сайті.
6. Система створює сеанс роботи.
7. Організатор працює з ресурсами сайту відповідно до своїх прав доступу.
8. Організатор закінчує роботу.
9. Система завершує сеанс роботи.

Виключні ситуації:

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



ID: *UC 005*

Назва: сценарій адміністрування ресурсів суддями.

Учасники: суддя, система.

Передумови: суддя має частковий доступ до адміністрування.

Результат: внесено зміни до турнірної таблиці.

1. Суддя вводить шестизначний логін та персональний пароль для отримання часткового доступу до адміністрування на сайті.
2. Автоматично створюється сеанс роботи.
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.



ID: *UC 006*

Назва: Перехід з однієї категорії в іншу

Учасники: Суддя, система.

Передумови: Суддя має данні про нового учасника змагань.

Результат: Внесено нового учасника то турнірної таблиці.

Основний сценарій:

1. Суддя входить у систему .
2. Суддя вводить шестизначний логін та пароль.
3. Суддя заходить до турнірної таблиці.
4. Суддя вносить дані про нового учасника.
5. Суддя відправляє системі новий варіант таблиці.
6. Система вносить нового учасника до турнірної таблиці.
7. Система публікує нову таблицю.

Виключні ситуації:

1. На співбесіді організатори вирішили відхилити запропоновану кандидатуру.



ID: *UC 007*

Назва: Перехід з однієї категорії в іншу

Учасники: Система, організатор.

Передумови: Наявність заяв на суддівство.

Результат: Зареєстровано нових організаторів змагань.

Основний сценарій:

1. Організатор входить у систему.
2. Організатор переглядає наявність запитів на суддівство.
3. Організатор розглядає заявки.
4. Організатори публікує голосування на сайті, у якому можуть брати участь поточні організатори змагань.
5. Впродовж тижня система набирає певну кількість відповідей.
6. Система відправляє результати голосувань організатору.
7. Організатор надсилає прізвища нових суддів.

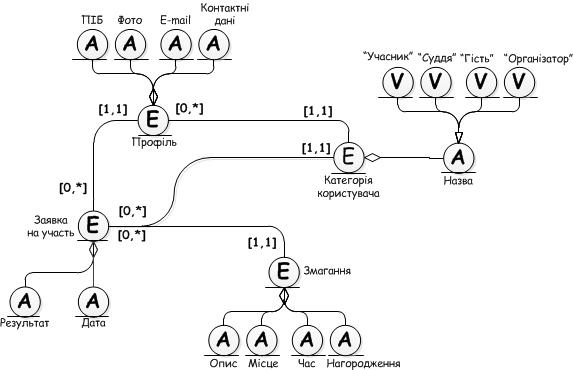
Виключні ситуації:

1. В результаті голосування виявилося, що більшість прости запропонованої кандидатури.

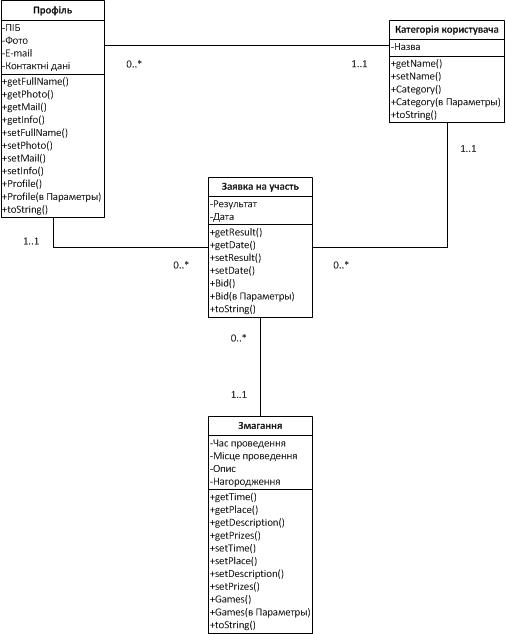


**Моделі**

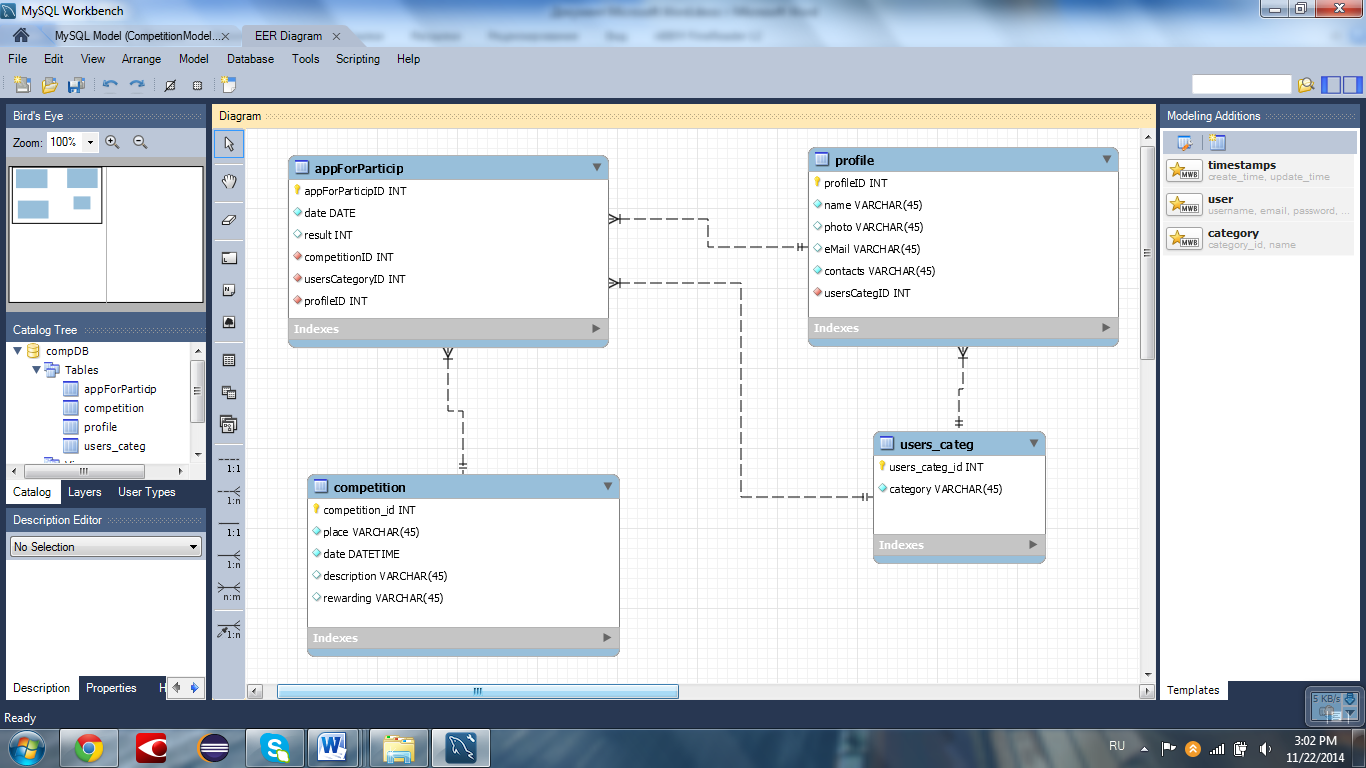
А) Бізнес-модель системи

****

Б) Діаграма класів



В) Реляційна модель представлення даних



**Словник предметної сфери**

Сутності (еntities)

***Профіль*** – внутрішнє представлення користувача, яке задається атрибутами:

* ПІБ;
* Фото;
* E-mail;
* Контактні дані.

***Категорія користувача*** – сукупність людей, які володіють спільними правами, можливостями, доступом. Задається атрибутом:

* Назва.

***Заявка на участь*** – подається зареєстрованим користувачем у фіксованих змаганнях, в яких він бере участь. Визначає категорію користувача. Задається атрибутами:

* Результат;
* Дата.

***Змагання*** – захід, що знаходиться у центрі уваги усіх категорій користувачів. Задається атрибутами:

* Час проведення;
* Місце проведення;
* Опис;
* Нагородження.

Атрибути (atributes)

1. ***ПІБ*** – атрибут профіля;
2. ***Фото*** – атрибут профіля;
3. ***E-mail*** – атрибут профіля;
4. ***Контактні дані*** – атрибут профіля;
5. ***Назва*** – атрибут категорії користувача;
6. ***Результат*** – атрибут заявки на участь;
7. ***Дата*** – атрибут заявки на участь;
8. ***Час проведення*** – атрибут змагання;
9. ***Місце проведення*** – атрибут змагання;
10. ***Опис*** – атрибут змагання;
11. ***Нагородження*** – атрибут змагання.

Значення (values)

1. ***Гість*** - значення назви категорії користувача.
2. ***Учасник*** - значення назви категорії користувача.
3. ***Суддя*** - значення назви категорії користувача.
4. ***Організатор*** - значення назви категорії користувача.

**База даних**

А) автоматично згенеровані запити

Створення таблиць:

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Schema compDB

-- -----------------------------------------------------

DROP SCHEMA IF EXISTS `compDB` ;

-- -----------------------------------------------------

-- Schema compDB

-- -----------------------------------------------------

CREATE SCHEMA IF NOT EXISTS `compDB` DEFAULT CHARACTER SET utf8 ;

SHOW WARNINGS;

USE `compDB` ;

-- -----------------------------------------------------

-- Table `users\_categ`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `users\_categ` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `users\_categ` (

`users\_categ\_id` INT NOT NULL,

`category` VARCHAR(45) NOT NULL,

PRIMARY KEY (`users\_categ\_id`))

ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------

-- Table `profile`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `profile` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `profile` (

`profileID` INT NOT NULL,

`name` VARCHAR(45) NOT NULL,

`photo` VARCHAR(45) NULL,

`eMail` VARCHAR(45) NULL,

`contacts` VARCHAR(45) NOT NULL,

`usersCategID` INT NOT NULL,

PRIMARY KEY (`profileID`),

CONSTRAINT `fk\_Profile\_Users category`

FOREIGN KEY (`usersCategID`)

REFERENCES `users\_categ` (`users\_categ\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SHOW WARNINGS;

CREATE INDEX `fk\_Profile\_Users category\_idx` ON `profile` (`usersCategID` ASC);

SHOW WARNINGS;

-- -----------------------------------------------------

-- Table `competition`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `competition` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `competition` (

`competition\_id` INT NOT NULL,

`place` VARCHAR(45) NOT NULL,

`date` DATETIME NOT NULL,

`description` VARCHAR(45) NULL,

`rewarding` VARCHAR(45) NULL,

PRIMARY KEY (`competition\_id`))

ENGINE = InnoDB;

SHOW WARNINGS;

-- -----------------------------------------------------

-- Table `appForParticip`

-- -----------------------------------------------------

DROP TABLE IF EXISTS `appForParticip` ;

SHOW WARNINGS;

CREATE TABLE IF NOT EXISTS `appForParticip` (

`appForParticipID` INT NOT NULL,

`date` DATE NOT NULL,

`result` INT NULL,

`competitionID` INT NOT NULL,

`usersCategoryID` INT NOT NULL,

`profileID` INT NOT NULL,

PRIMARY KEY (`appForParticipID`),

CONSTRAINT `fk\_Application for participation\_Competition1`

FOREIGN KEY (`competitionID`)

REFERENCES `competition` (`competition\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Application for participation\_Users category1`

FOREIGN KEY (`usersCategoryID`)

REFERENCES `users\_categ` (`users\_categ\_id`)

ON DELETE NO ACTION

ON UPDATE NO ACTION,

CONSTRAINT `fk\_Application for participation\_Profile1`

FOREIGN KEY (`profileID`)

REFERENCES `profile` (`profileID`)

ON DELETE NO ACTION

ON UPDATE NO ACTION)

ENGINE = InnoDB;

SHOW WARNINGS;

CREATE INDEX `fk\_Application for participation\_Competition1\_idx` ON `appForParticip` (`competitionID` ASC);

SHOW WARNINGS;

CREATE INDEX `fk\_Application for participation\_Users category1\_idx` ON `appForParticip` (`usersCategoryID` ASC);

SHOW WARNINGS;

CREATE INDEX `fk\_Application for participation\_Profile1\_idx` ON `appForParticip` (`profileID` ASC);

SHOW WARNINGS;

Б) заповнення таблиць даними

SET @OLD\_UNIQUE\_CHECKS=@@UNIQUE\_CHECKS, UNIQUE\_CHECKS=0;

SET @OLD\_FOREIGN\_KEY\_CHECKS=@@FOREIGN\_KEY\_CHECKS, FOREIGN\_KEY\_CHECKS=0;

SET @OLD\_SQL\_MODE=@@SQL\_MODE, SQL\_MODE='TRADITIONAL,ALLOW\_INVALID\_DATES';

-- -----------------------------------------------------

-- Data for table `users\_categ`

-- -----------------------------------------------------

START TRANSACTION;

USE `compDB`;

INSERT INTO `users\_categ` (`users\_categ\_id`, `category`) VALUES (1, 'participant');

INSERT INTO `users\_categ` (`users\_categ\_id`, `category`) VALUES (2, 'judge');

INSERT INTO `users\_categ` (`users\_categ\_id`, `category`) VALUES (3, 'guest');

INSERT INTO `users\_categ` (`users\_categ\_id`, `category`) VALUES (4, 'organizer');

COMMIT;

-- -----------------------------------------------------

-- Data for table `profile`

-- -----------------------------------------------------

START TRANSACTION;

USE `compDB`;

INSERT INTO `profile` (`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES (1, 'Pavluchkov Vladislav', NULL, NULL, '0631234567', 4);

INSERT INTO `profile` (`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES (2, 'Zmeul Evgeniy', NULL, NULL, '0671234567', 2);

INSERT INTO `profile` (`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES (3, 'Morozov Max', NULL, NULL, '0961234567', 3);

INSERT INTO `profile` (`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES (4, 'Popenko Ruslan', NULL, NULL, '0981234567', 1);

INSERT INTO `profile` (`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES (5, 'Korchak Myhailo', NULL, NULL, '0501234567', 3);

COMMIT;

-- -----------------------------------------------------

-- Data for table `competition`

-- -----------------------------------------------------

START TRANSACTION;

USE `compDB`;

INSERT INTO `competition` (`competition\_id`, `place`, `date`, `description`, `rewarding`) VALUES (1, 'Stadium Start', '2014-11-15 19:00:00', NULL, NULL);

INSERT INTO `competition` (`competition\_id`, `place`, `date`, `description`, `rewarding`) VALUES (2, 'Metro Gidropark bus station', '2014-11-19 19:00:00', NULL, NULL);

COMMIT;

-- -----------------------------------------------------

-- Data for table `appForParticip`

-- -----------------------------------------------------

START TRANSACTION;

USE `compDB`;

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (1, '2014-11-15', NULL, 1, 1, 1);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (2, '2014-11-19', NULL, 2, 1, 1);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (3, '2014-11-15', NULL, 1, 2, 2);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (4, '2014-11-19', NULL, 2, 2, 2);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (5, '2014-11-15', NULL, 1, 3, 3);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (6, '2014-11-19', NULL, 2, 3, 3);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (7, '2014-11-15', NULL, 1, 4, 4);

INSERT INTO `appForParticip` (`appForParticipID`, `date`, `result`, `competitionID`, `usersCategoryID`, `profileID`) VALUES (8, '2014-11-19', NULL, 2, 4, 4);

COMMIT;

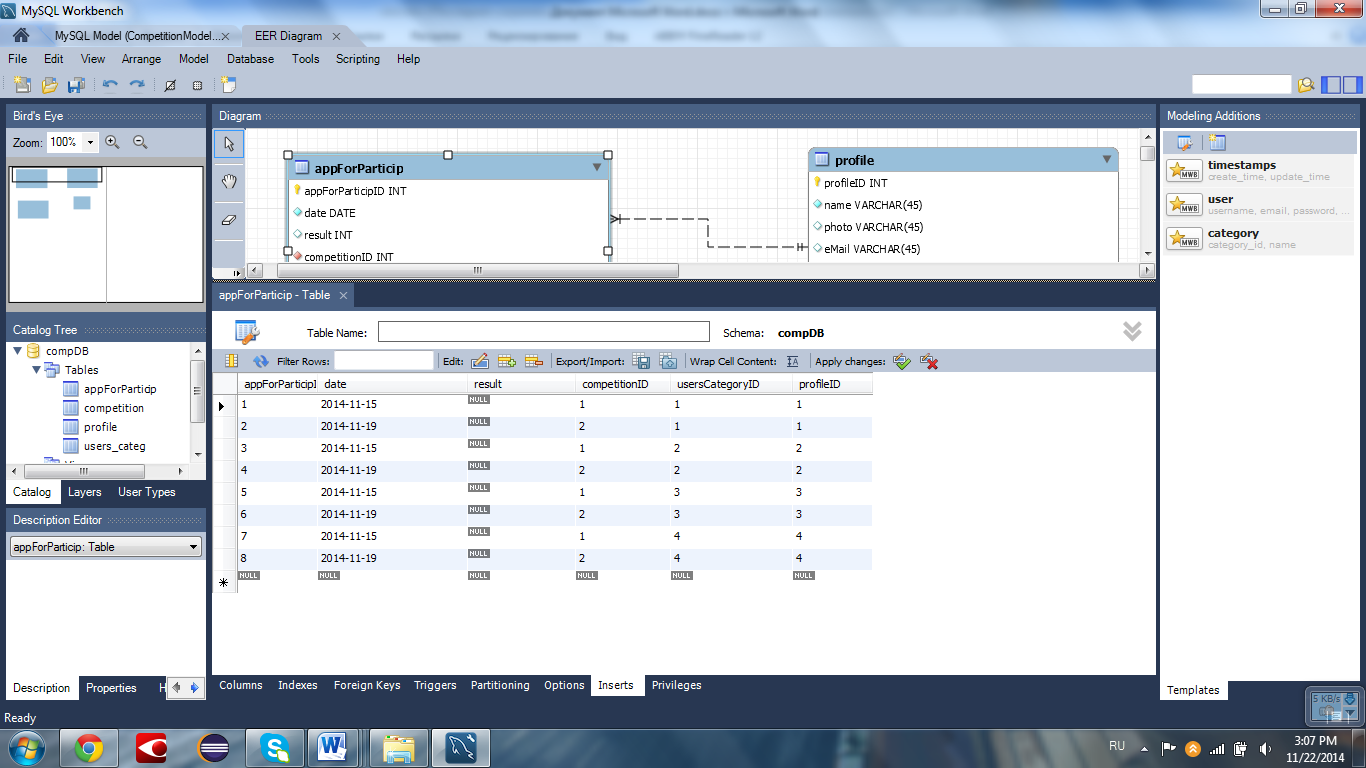
-- -----------------------------------------------------

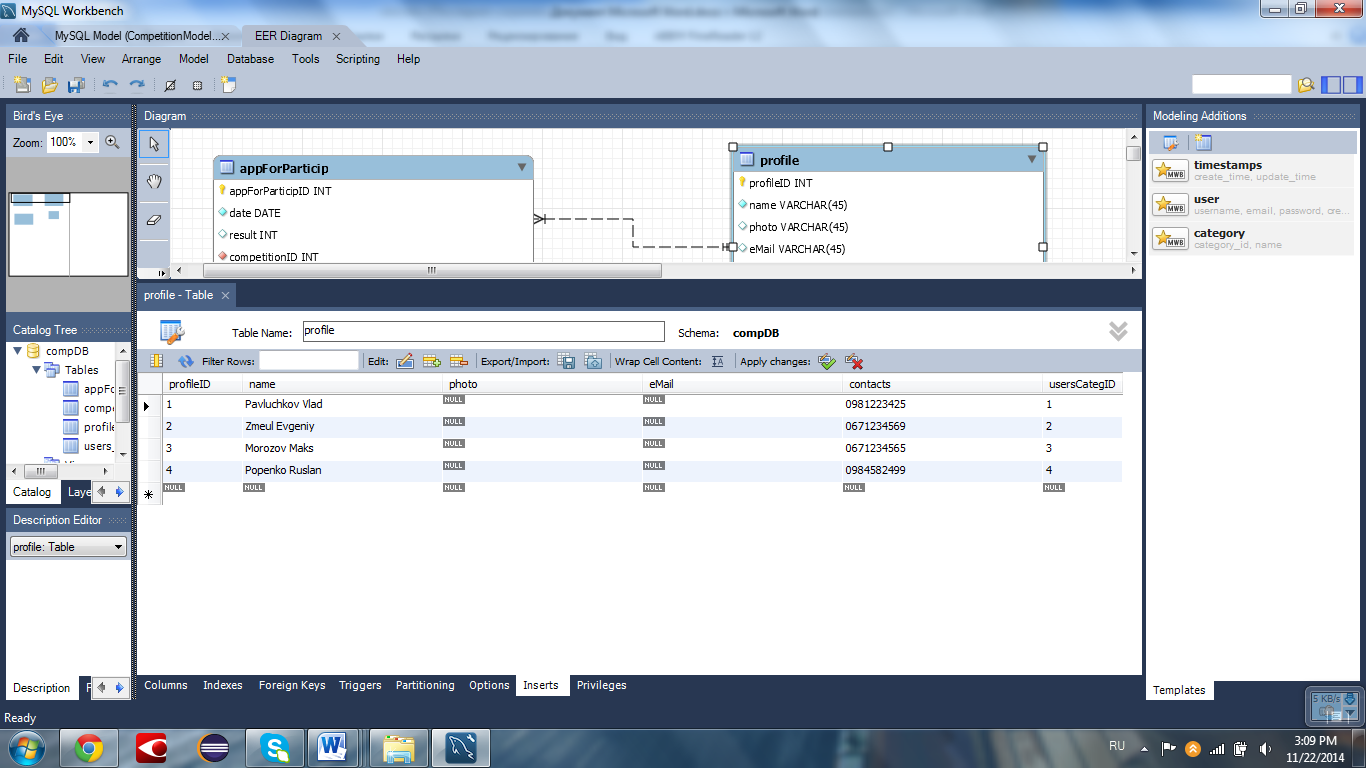
SET SQL\_MODE=@OLD\_SQL\_MODE;

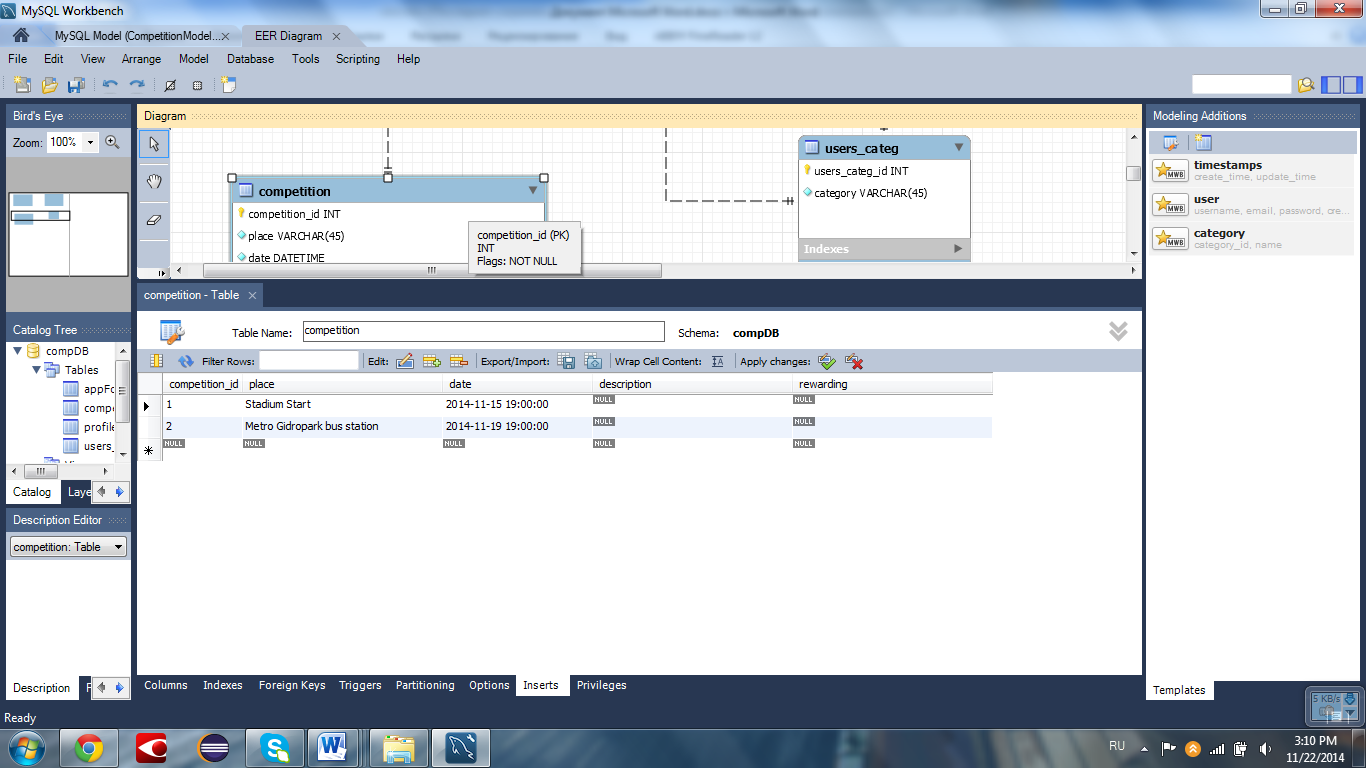
SET FOREIGN\_KEY\_CHECKS=@OLD\_FOREIGN\_KEY\_CHECKS;

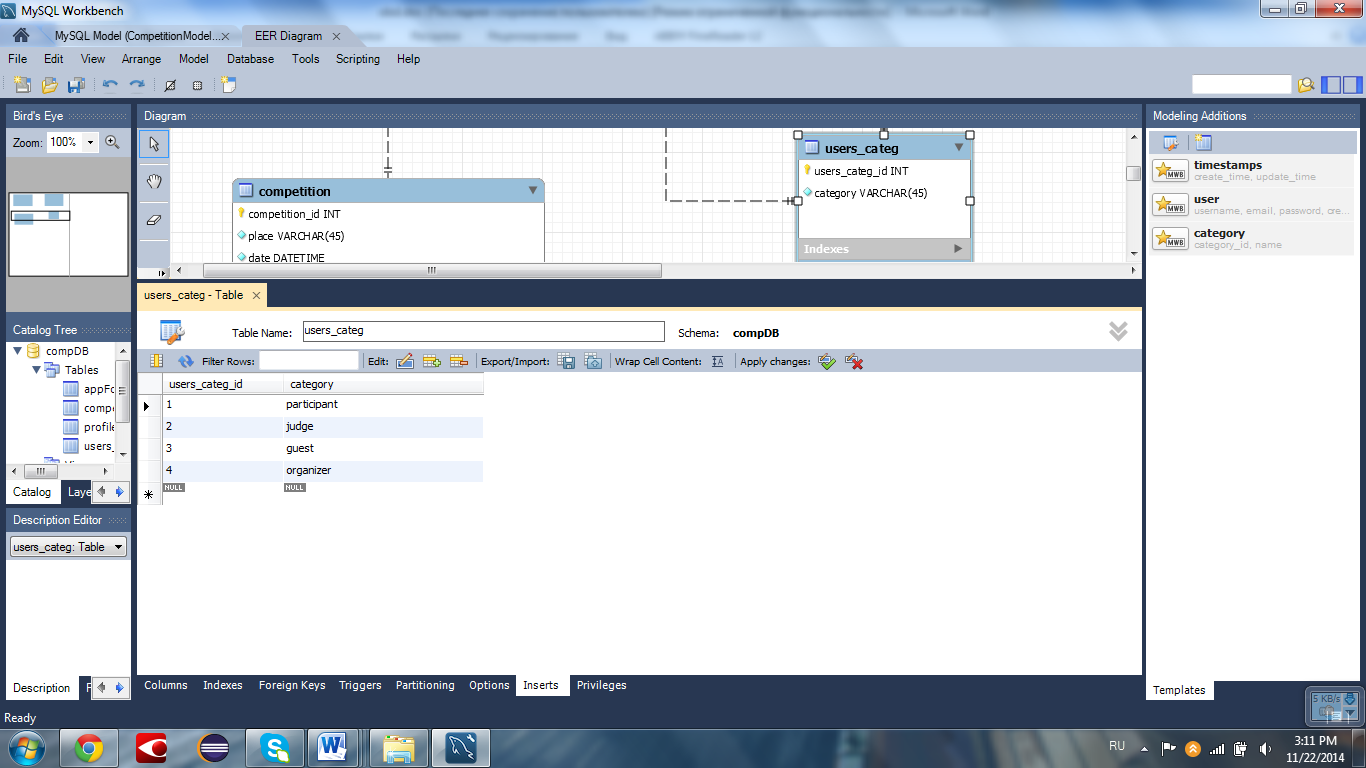
SET UNIQUE\_CHECKS=@OLD\_UNIQUE\_CHECKS;

В) Результати заповнення таблиць









**Управління базою даних за допомогою JDBC**

А) Код класу Test

**package** obd;

**import** java.sql.Connection;

**import** java.util.Date;

**import** java.sql.DriverManager;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**import** java.sql.ResultSet;

**import** java.sql.Time;

/\*\*

\* **@author** Group 4

\*

\* **@version**

\*/

**public** **class** **Test** {

**public** **static** **void** **main**(**String**[] args) {

Connection **connection** = **null**;

Statement **statement** = **null**;

Statement **statement2** = **null**;

Statement **statement3** = **null**;

**String** **query** = "select\*from competition";

**try** {

**Class**.*forName*("com.mysql.jdbc.Driver");

connection = **DriverManager**.*getConnection*(

"jdbc:mysql://localhost/compDB", "root", "1111");

statement = connection.createStatement();

ResultSet **rs** = statement.executeQuery(query);

*printCompetitions*(rs);

query = "insert into `compDB`.`profile`(`profileID`, `name`, `photo`, `eMail`, `contacts`, `usersCategID`) VALUES ('5','Korchak Myhailo','','','05055555555','3')";

statement2 = connection.createStatement();

statement2.executeUpdate(query);

query = "select\*from profile";

rs = statement2.executeQuery(query);

*printProfile*(rs);

statement3 = connection.createStatement();

query = "delete from `compDB`.`profile`" + "where `profileID`=5";

statement3.executeUpdate(query);

connection.close();

} **catch** (**Exception** **e**) {

e.printStackTrace();

}

}

**private** **static** **void** **printProfile**(ResultSet rs) **throws** **SQLException** {

**int** **ProfileID**, **UsersCategID**;

**String** **name**, **photo**, **eMail**, **contacts**;

**while** (rs.next()) {

ProfileID = rs.getInt(1);

name = rs.getString(2);

photo = rs.getString(3);

eMail = rs.getString(4);

contacts = rs.getString(5);

UsersCategID = rs.getInt(6);

**System**.***out***.println("--------------------------");

**System**.***out***.println("ProfileID= " + ProfileID);

**System**.***out***.println("name: " + name);

**System**.***out***.println("photo: " + photo);

**System**.***out***.println("e-mail: " + eMail);

**System**.***out***.println("contacts: " + contacts);

**System**.***out***.println("UsersCategID= " + UsersCategID);

**System**.***out***.println("-----------------------------");

}

}

**private** **static** **void** **printCompetitions**(ResultSet rs) **throws** **SQLException** {

**int** **competitionID**;

**String** **place**, **description**, **rewarding**;

**Date** **date**;

**Time** **time**;

**while** (rs.next()) {

competitionID = rs.getInt(1);

place = rs.getString(2);

date = rs.getDate(3);

time = rs.getTime(3);

description = "";

rewarding = "";

**System**.***out***.println("-----------------------------");

**System**.***out***.println("CompetitionID= " + competitionID);

**System**.***out***.println("Place: " + place);

**System**.***out***.println("Date: " + date + " " + time);

**System**.***out***.println("Description: " + description);

**System**.***out***.println("Rewarding: " + rewarding);

**System**.***out***.println("------------------------------");

}

}

}

Б) Результати виконання програми

CompetitionID = 1

Place: Stadium Start

Date: 2014-11-15 19:00:00

Description:

Rewarding:

------------------------------

-----------------------------

CompetitionID = 2

Place: Metro Gidropark bus station

Date: 2014-11-19 19:00:00

Description:

Rewarding:

------------------------------

--------------------------

ProfileID = 1

name: Pavluchkov Vladislav

photo: null

e-mail: null

contacts: 0631234567

UsersCategID = 4

-----------------------------

--------------------------

ProfileID = 2

name: Zmeul Evgeniy

photo: null

e-mail: null

contacts: 0671234567

UsersCategID = 2

-----------------------------

--------------------------

ProfileID = 3

name: Morozov Max

photo: null

e-mail: null

contacts: 0961234567

UsersCategID= 3

-----------------------------

--------------------------

ProfileID = 4

name: Popenko Ruslan

photo: null

e-mail: null

contacts: 0981234567

UsersCategID = 1

--------------------------

ProfileID = 5

name: Korchak Myhailo

photo:

e-mail:

contacts: 0501234567

UsersCategID = 3