

НАЦІОНАЛЬНИЙ ТЕХНІЧНИЙ УНІВЕРСИТЕТ УКРАЇНИ  
«КИЇВСЬКИЙ ПОЛІТЕХНІЧНИЙ ІНСТИТУТ  
ІМЕНІ ІГОРЯ СІКОРСЬКОГО»

Факультет прикладної математики

Кафедра прикладної математики

ПОЯСНЮВАЛЬНА ЗАПИСКА  
ДО КУРСОВОЇ РОБОТИ  
з дисципліни “Бази даних та інформаційні системи”  
на тему: Організація футбольних турнірів

Студента IV курсу,  
групи КМ-42  
МАВЛЮТОВА Я.С.

Викладач  
ТЕРЕЩЕНКО І.О.

Оцінка: \_\_\_\_ балів

## ЗМІСТ

Вступ .....	5
1 ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ .....	6
1.1 Мета .....	6
1.2 Граничні умови .....	6
1.3 Ролі користувачів .....	6
1.4 Функції .....	7
2 ПОСТАНОВКА ЗАДАЧІ .....	9
2.1 Визначення підгруп користувачів .....	9
2.1.1 Неавторизовані користувачі .....	10
2.1.2 Авторизовані користувачі .....	10
2.1.3 Адміністратор інформаційної системи .....	11
2.2 Визначення підгрупи даних .....	11
2.2.1 Неавторизовані користувачі .....	11
2.2.2 Авторизований користувач .....	12
2.3 Бізнес-правила .....	12
2.4 Діаграма послідовностей .....	14
3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ .....	16
3.1 Use-case .....	16
3.2 Component diagram .....	19
3.3 Scrum. Sprint Planning .....	20
4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ .....	32
4.1 Організатор .....	33
4.2 Турнір .....	33
4.3 Матч .....	34
4.4 Рахунок .....	34
4.5 Команда .....	34
4.6 Повідомлення .....	35

	3
5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ.....	36
Висновки .....	38
Перелік посилань .....	39

## ЗАВДАННЯ НА ВИКОНАННЯ КУРСОВОЇ РОБОТИ

Основним завданням курсової роботи є побудова архітектури інформаційної системи "Організація футбольних турнірів". Необхідно правильно визначити сутності, їх атрибутів, ключі.

У курсовій роботі необхідно побудати концептуальну, логічну та фізичну моделі. Згенерувати скрипт бази даних, та реалізувати "клієнт-серверну" частину. Необхідно реалізувати перевірку валідності даних на клієнській частині та в базі даних.

Потрібно реалізувати інформаційну систему для керування футбольними турнірами, де користувачі могли би легко неявно додавати, редагувати або видаляти дані з бази даних, звісно під контролем адміністратора БД.

## АНОТАЦІЯ

Курсова робота написана на 39 аркушах. Метою роботи є створення інформаційної системи контролю футбольних матчів. При дослідженні було спроектовано концептуальну, логічну та фізичну моделі бази даних, визначено Use Cases, діаграму компонент та діаграми послідовстей для кожної ролі користувачів.

Основні поняття: модель, турнір, організатор, сутність, інформаційна система.

## ВСТУП

Сьогодні все більше і більше людей, які не стали професійними спортсменами, бажають грати в футбол, але не просто так, а брати участь в повноцінному чемпіонаті. На допомогу їм приходять аматорські змагання з футболу. Проте для організації подібного турніру потрібна система, що здатна повністю обслуговувати команди, організаторів та давати багато можливостей для керування подібним турніром.

Існує декілька інформаційних систем для контролю турнірів, але вони всі переважно обслуговують матчі вищого рангу і є не безкоштовними. Тому такі ресурси не є поширеними у колі аматорських команд.

Метою даного завдання є створення удосконалої інформаційної системи, в порівнянні з іншими, причому буде безкоштовна можливість керування турнірами для будь-кого.

Перш за все, що буде притаманне розроблювані ІС - простота у використанні з потужним функціоналом. Організатори турнірів зможуть з легкістю створювати нові турніри, додавати команди, додавати матчі, редагувати ці дані.

## 1 ПЕРЕДПРОЕКТНЕ ДОСЛІДЖЕННЯ

### 1.1 Мета

Оптимізація процесів організації проведення футбольних турнірів:

- а) створення турнірів, їх контроль, редагування, додавання матчів;
- б) візуалізація результатів кожного з матчів;
- в) сповіщення учасників про зміни у турнірі.

### 1.2 Граничні умови

Створення турніру, додавання команд та матчів, повідомлення про зміни на конкретному футбольному турнірі для команд, що подали заявку до початку змагань.

Кількість команд на одному турнірі – 2-32.

Кількість гравців в одній команді – будь-яка, організатор вирішує.

Термін проведення змагань – будь-який, в залежності від інтенсивності матчів.

### 1.3 Ролі користувачів

Визначено наступні ролі:

- а) учасник турніру (переглядає турнірну сітку)
- б) учасник-капітан (переглядає турнірну сітку; отримує сповіщення на електронну скриньку)
- в) організатор турніру (редагує турнір, додає нові матчі; надсилає сповіщення та інші дії)
- г) адміністратор сайту (контроль функціоналу системи; видалення організаторів, турнірів, матчів)

## 1.4 Функції

Функціонал системи:

а) створення турніру:

- 1) параметри: назва турніру, часовий діапазон проведення, кількість команд;
- 2) час виконання(діапазон: < 20с.);
- 3) джерело інформації (організатор турніру);
- 4) структура даних (заповнення організатором у поле вводу в ІС);
- 5) результат: збереження даних в БД;

б) додавання команд до турніру:

- 1) параметри: код турніру, назва команди, електронна скринька команди;
- 2) час виконання(діапазон: < 20с.);
- 3) джерело інформації (організатор турніру);
- 4) структура даних (заповнення організатором у поле вводу в ІС);
- 5) результат: збереження даних в БД;

в) сповіщення капітана про зміни(повідомлення на електронну скриньку):

1) параметри: електронна скринька організатора(за замовчуванням), контакти, зміни;

- 2) час виконання(діапазон: < 20с.);
- 3) джерело інформації (організатор турніру);
- 4) структура даних (заповнення організатором у поле вводу в ІС);
- 5) результат: надсилення повідомлення на електронну скриньку;

г) візуалізація інформації про турнір:

- 1) параметри: код турніру, логін організатора(за замовчуванням);
- 2) час виконання(діапазон: < 30сек);
- 3) джерело інформації (БД);
- 4) результат: відображення інформації у читабельному вигляді;

д) редагування інформації про турнір:

- 1) параметри: код турніру, нова назва, новий часовий діапазон;
- 2) час виконання(діапазон: < 20сек);



- 3) джерело інформації (організатор турніру);
- 4) результат: збереження даних в БД.

## 2 ПОСТАНОВКА ЗАДАЧІ

За допомогою розроблюваної інформаційної системи організатори футбольних турнірів значно мінімізують часові затрати на проведення змагань. Дана система дозволить повністю контролювати перебіг подій на футбольному турнірі. При розробці даної системи головним завданням є простота у використанні та контролі. Створення турніру, додавання команд, додавання матчів до турніру, вивід необхідної інформації на екран, ведення статистики, фільтрація даних - основні функції системи організації футбольних матчів, тобто організатору залишиться лише одне - заповнити необхідні поля даними. Також при розробці даної системи орієнтація буде напрямлена не лише на організаторів турнірів, а й звичайних користувачів. Роль звичайного користувача матиме гравець кожної з команд. Вони матимуть змогу переглядати статистику турнірів, не зареєструвавшись при цьому.

### 2.1 Визначення підгруп користувачів

Бізнес-логікою ІС було визначено про необхідність реалізації функціоналу для наступних 4-ох типів користувачів:

- а) неавторизовані користувачі;
- б) авторизовані користувачі;
- в) адміністратор інформаційної системи.

Розглянемо детільніше кожен із них.

### 2.1.1 Неавторизовані користувачі

Згідно до принципів визначення неавторизованих користувачів функціонал будемо так, щоб він не потребував збереження особистих даних, тобто він зможе лише переглядати інформацію. А саме, неавторизованим користувачам буде надано можливість:

- а) зареєструватись;
- б) переглядати інформацію про турніри та матчі;
- в) пошук турнірів за деякими параметрами.

### 2.1.2 Авторизовані користувачі

Для авторизованих користувачів відповідно будуть більш розширені можливості, зокрема дії, які пов'язані із авторизацією та ведення облікового запису, з метою користування ІС. Авторизований користувач має наступний додатковий функціонал, у порівнянні з неавторизованим:

- а) створення турнірів;
- б) додавання команд;
- в) додавання матчів;
- г) редагування турнірів;
- д) видалення турнірів, матчів;
- е) надсилання повідомлень на електронну скриньку командам.

### 2.1.3 Адміністратор інформаційної системи

Адміністратор ІС обмежений по функціоналу правами, що йому надає розробник ІС. Основні функції адміністратора:

- а) перегляд усієї відкритої інформації про організаторів та їх турніри;
- б) видалення організаторів, турнірів, матчів.

## 2.2 Визначення підгрупи даних

Розглянемо дані, які потребує функціонал системи для неавторизованих та авторизованих користувачів.

### 2.2.1 Неавторизовані користувачі

Оскільки для неавторизованого користувача не потрібно оброблювати його персональні дані системою, то єдина функція, яка потребує його дій - фільтрація(пошук) турнірів. Тобто, на головній сторінці користувач бачитиме форму для введення тексту і кнопку. Усі дані, введені користувачем, будуть перевірятися на клієнтській частині. На поє вводу буде встановлено допустиму довжину тексту(формат даних - рядок) та допустимі символи. Після коректного вводу, система відбере необхідні турніри та виведе їх на екран користувача.

Проте, якщо користувач хоче зареєструватися, йому необхідно перейти на потрібну вкладку на сторінці та заповнити необхідні поля - логін, пароль, пошта. На всі поля буде встановлено регулярні вирази введення. Якщо всі поля були коректно заповнені, то система зберігає користувача в базі даних, а сайт перенаправляє його

на персональну сторінку.

### 2.2.2 Авторизований користувач

Для авторизації необхідно заповнити дві форми введення - логін та пароль. Причому пароль буде скритий від користувача, йому відображатимуться лише "\*". Довжина кожного з цих полів встановлена регулярними виразами, формат даних типу рядок. Після коректного вводу даних, користувач автоматично переходить на персональну сторінку, де відображається список його турнірів. ІС при авторизації витягує логін користувача та автоматично шукає в базі даних його турніри, що значно полегшує роботу організатору.

Авторизований користувач може створювати турнір. Назва турніру(формат даних типу рядок), дата початку турніру, дата закінчення турніру, назва та пошта команд - необхідні поля для створення турніру. Якщо всі поля були заповнені коректно, то в базу даних додається новий турнір, і на персональній сторінці користувача додається інформація про створений турнір. Подібним чином користувач має змогу редагувати дані про турнір, проте у цьому випадку оновлюються дані в поточному, а не створюється новий турнір.

Організатор турнірів має змогу відправляти повідомлення командам на електронну пошту. Форма для введення пошти матиме регулярний вираз, що перевірятиме правильність вводу інформації. Повідомлення може мати довжину від 2 до 40 символів, формат даних - рядок.

### 2.3 Бізнес-правила

Розглянемо основні бізнес-правила, що функціонують у ІС організація футбольних турнірів.

До основних фактів, що формують бізнес-правила належать наступні твердження:

- а) турнір створюється організатором;
- б) матчі додаються організатором;
- в) рахунок у матчі визначає організатор;
- г) видалити зареєстрованого користувача може лише адмін.

До основних обмежень, що формують бізнес-правила належать наступні твердження:

- а) при видаленні матчу, або турніру організатору потрібно підтвердити його пароль;
- б) не можна створити турнір з 0 кількістю учасників;
- в) дата старту турніру не може бути раніше за поточний день;
- г) дата кінця турніру завжди пізніше за дату старту;
- д) день голу у матчі співпадає з днем матчу.

До основних активаторів операцій, що формують бізнес-правила належать наступні твердження:

- а) при некоректному введенні поля з'являється інформаційне повідомлення;
- б) якщо авторизація пройшла успішно, то система автоматично перенаправляє користувача на персональну сторінку;
- в) при виході з персональної сторінки, система перенаправляє користувача на головну сторінку;
- г) якщо видалено користувача, то автоматично видаляються його турніри, матчі та голи.

Виділимо підпроцеси ІС, причому кожен із них буде реалізовано різними моделями життєвого циклу.

V-моделі життєвого циклу відповідають наступні підпроцеси:

- а) реєстрація;
- б) авторизація;
- в) видалення турніру;
- г) видалення користувача.

Спіральній моделі життєвого циклу відповідають наступні підпроцеси:

- а) створення турніру;
- б) додавання матчів.

ХР-моделі життєвого циклу відповідають наступні підпроцеси:

- а) візуалізація інформації про організаторів, турніри.

## 2.4 Діаграма послідовностей

Розглянемо діаграми послідовностей для незареєстрованих(рисунок 2.1) та зареєстрованих(рисунок 2.2) користувачів.

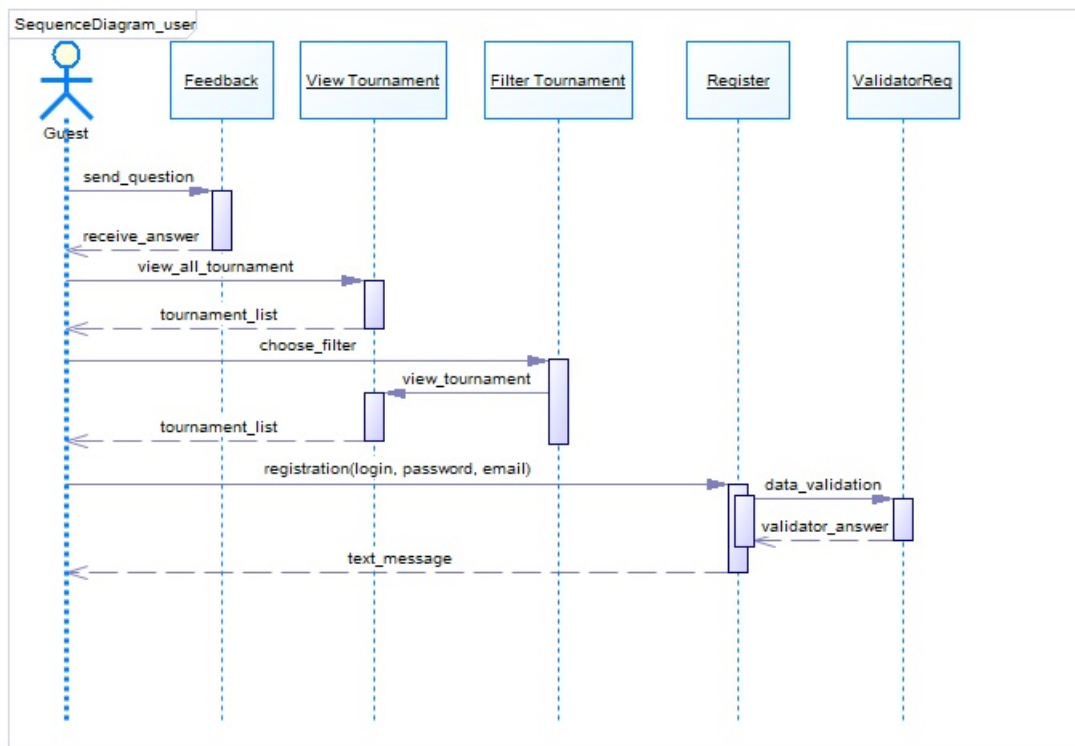


Рисунок 2.1 – Діаграма послідовностей для незареєстрованого користувача

Діаграма послідовності показує життєвий цикл будь-якого певного об'єкта і взаємодію акторів ІС в рамках якого- або певного прецеденту.

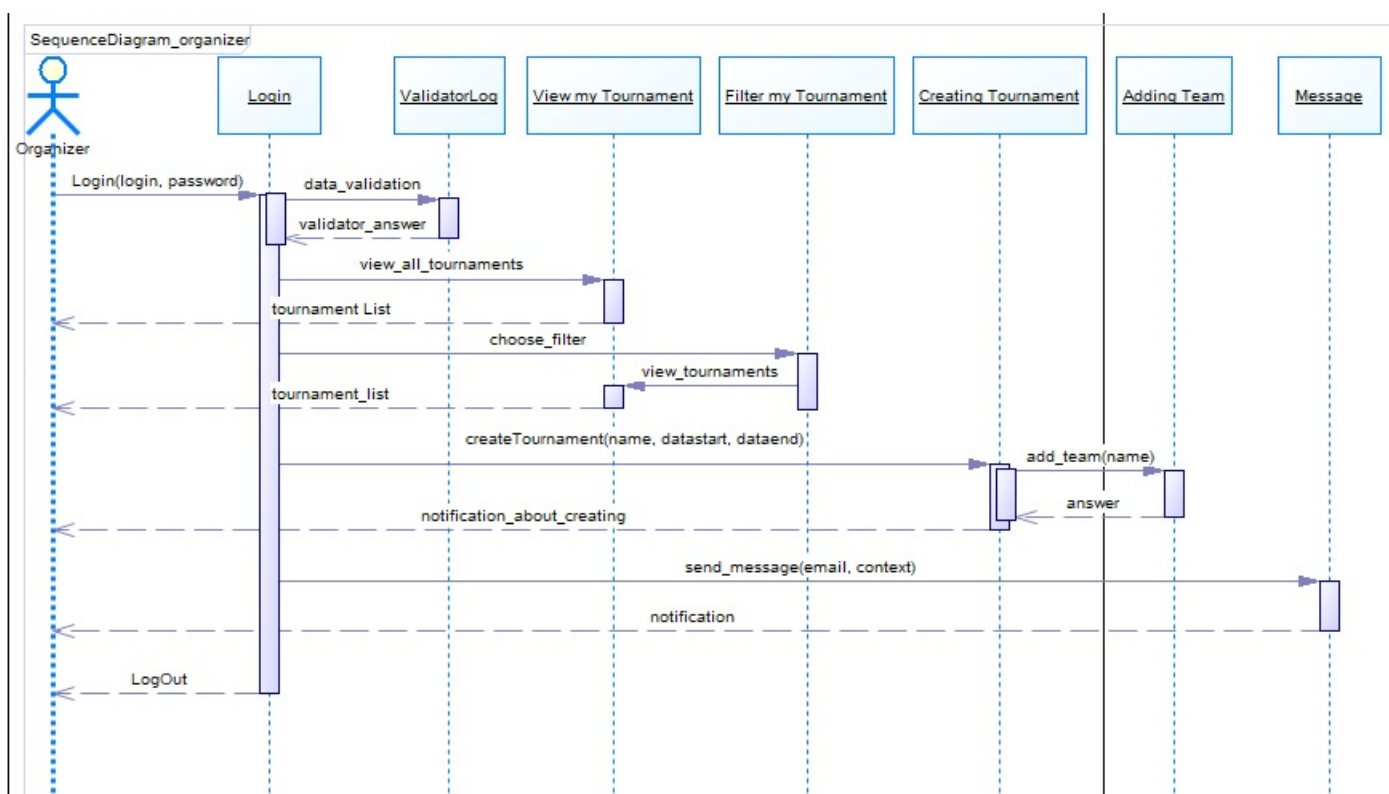


Рисунок 2.2 – Діаграма послідовностей для зареєстрованих користувачів



## 3 МОДЕЛЮВАННЯ БІЗНЕС-ПРОЦЕСІВ

## 3.1 Use-case

В інформаційній системі існує два види звичайних користувачів, які не є адміністраторами. Вони ж в свою чергу розділені на авторизованих і неавторизованих користувачів. Авторизований користувач, на відміну від неавторизованого, має більше можливостей при роботі з інформаційною системою.

Створений Use Case[1] зображено на рисунку 3.1..

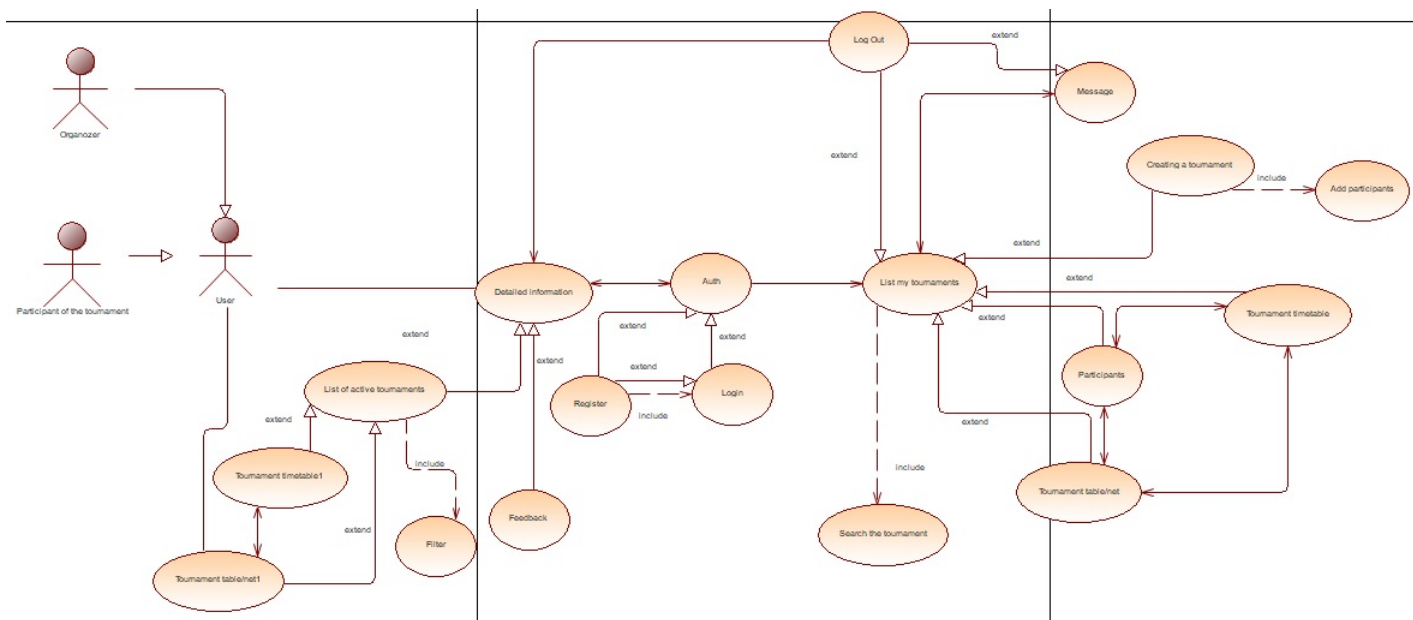


Рисунок 3.1 – Use Case для користувачів

Більш детальна структура даного Use Case на рисунках 3.2 та 3.3.

На діаграмі відображено можливість неавторизованого користувача переглядати список турнірів та результати кожного з них, також показано розширення функціоналу в результаті входу у систему, як авторизований користувач.

Притримуючись логіки, що була встановлена для системи, побудуємо діаграму для адміністратора(рисунок 3.4).

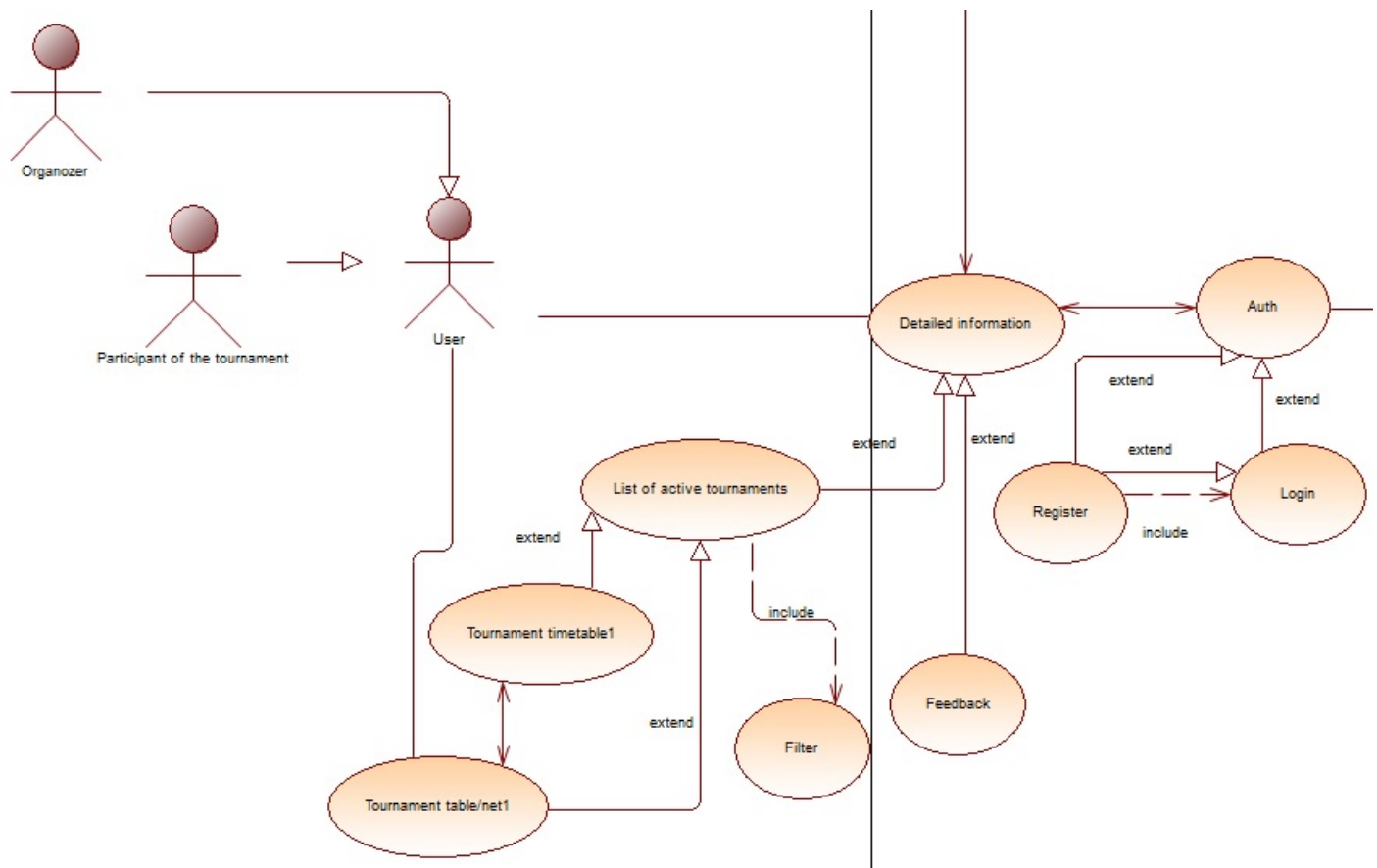


Рисунок 3.2 – Use Case для користувачів(частина 1)

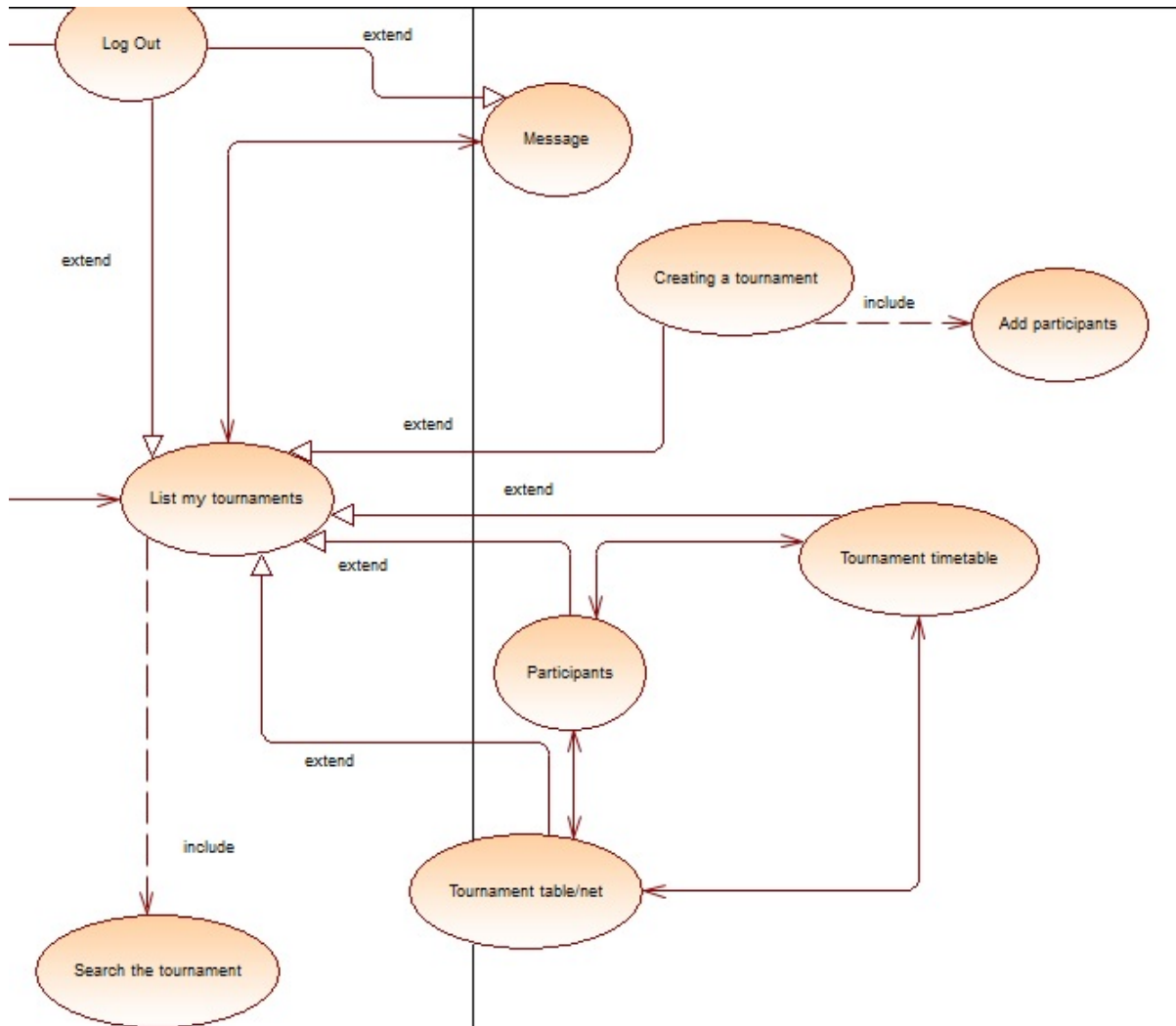


Рисунок 3.3 – Use Case для користувачів(частина 2)

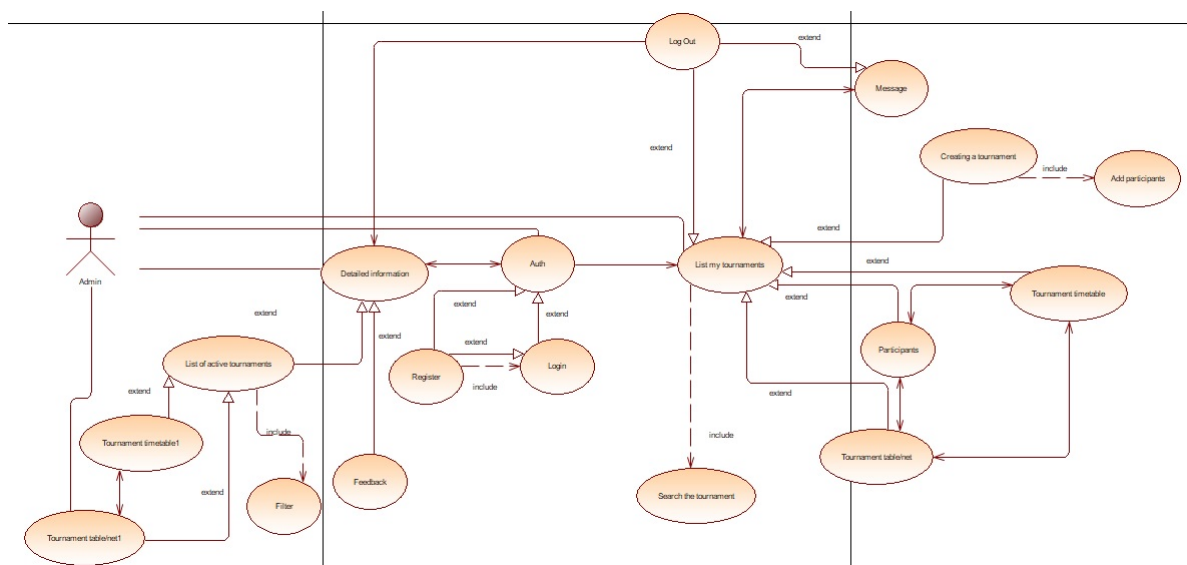


Рисунок 3.4 – Область дії адміністратора

### 3.2 Component diagram

Розглянемо компоненту діаграму[1], реалізовану для інформаційної системи. Загальна схема діаграми зображена на рисунку 3.5.

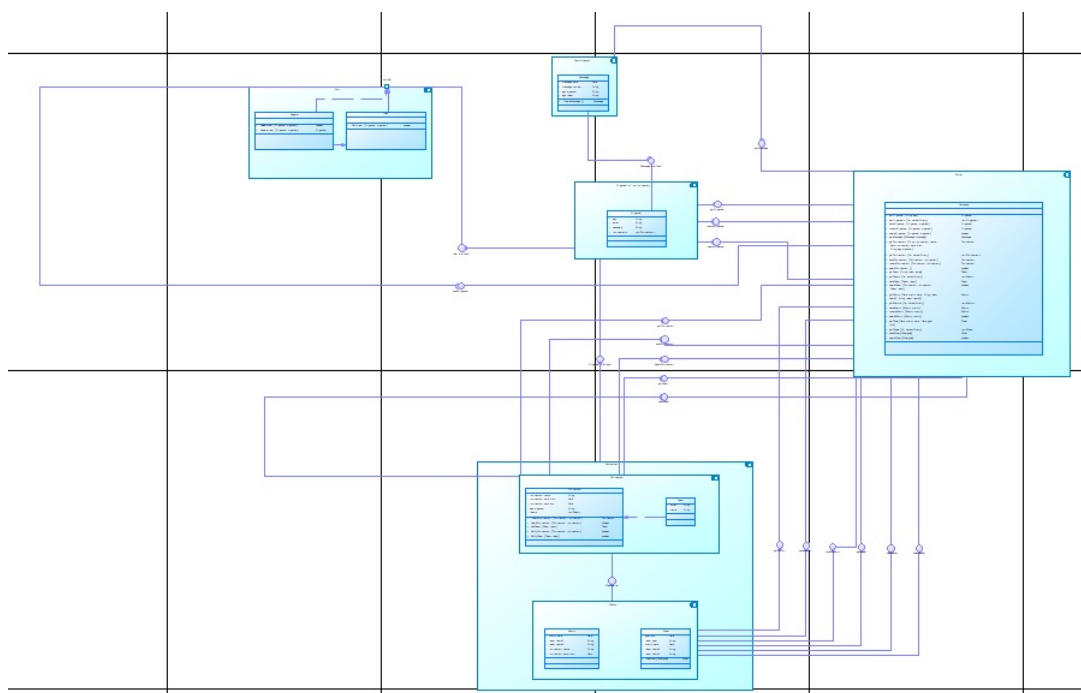


Рисунок 3.5 – Компонентна діаграма

Більш детальна структура даної діаграми на рисунках 3.6 , 3.7, 3.8, 3.9 та 3.10.

Діаграма компонентів дозволяє визначити архітектуру розроблюваної системи, встановивши залежності між програмними компонентами. Основними графічними елементами діаграми компонентів є компоненти, інтерфейси і залежності між ними. Компоненти об'єднуються, разом використовуючи структурні зв'язки. Це ілюструє зв'язок типу «клієнт-сервер». Структурна взаємодія — «зв'язок двох компонент, який передбачає, що один з них надає послуги, потрібні іншому компоненту».

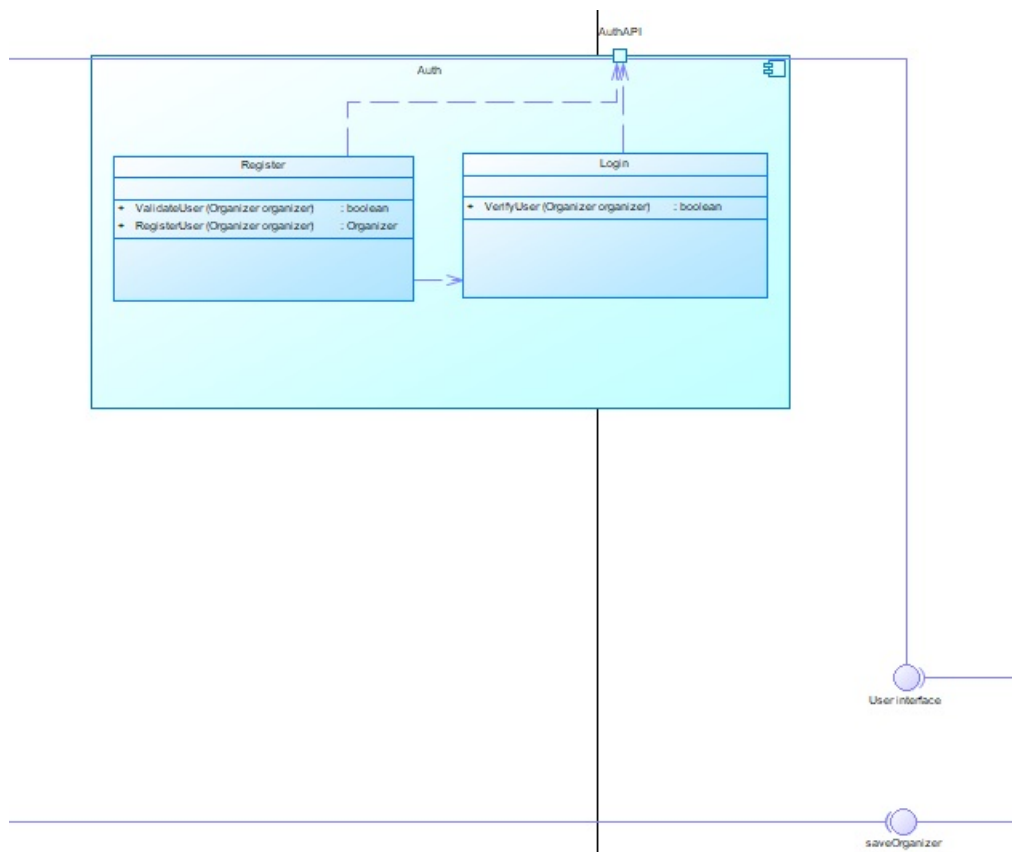


Рисунок 3.6 – Компонентна діаграма(частина 1)

### 3.3 Scrum. Sprint Planning

На рисунку 3.11 зображено перелік усіх спринтів розроблюваної системи. Розглянемо детільніше кожен із них.

Перший спринт містить реалізацію основного функціоналу системи - генерування футбольного турніру, додавання учасників та інше.

На рисунку 3.12 зображено першу картку спринта "Generation and Visualization". Дана картка містить 2 задачі, що являють собою можливість створення футбольного турніру та додавання до нього учасників. На рисунках 3.13 та 3.14 зображено обмеження даних для даної задачі.

На рисунку 3.15 зображено другу картку спринта "Generation and Visualization". Дана картка містить 1 задачу, що являє собою генерацію турнірних матчів. На рисунку 3.16 зображено обмеження даних для даної задачі.

На рисунку 3.17 зображено наступну картку спринта "Generation and

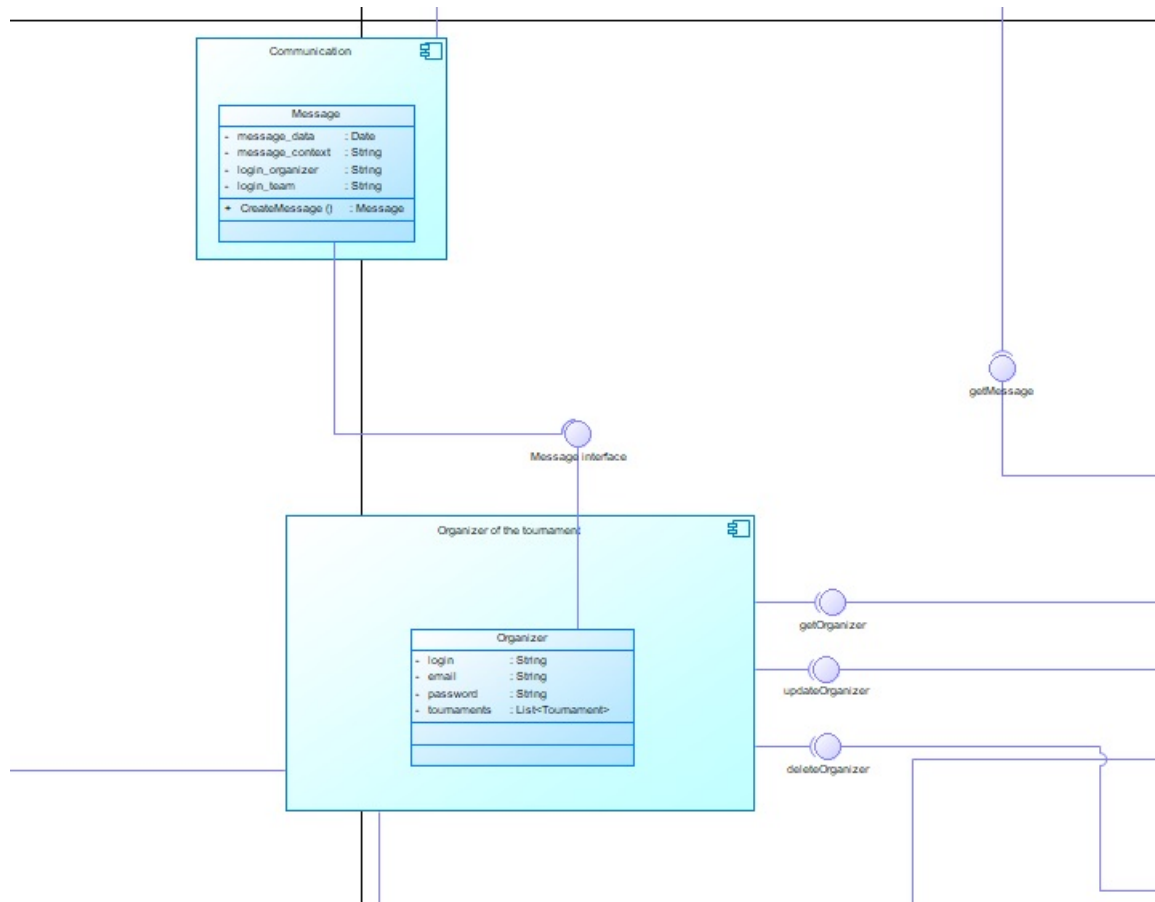


Рисунок 3.7 – Компонентна діаграма(частина 2)

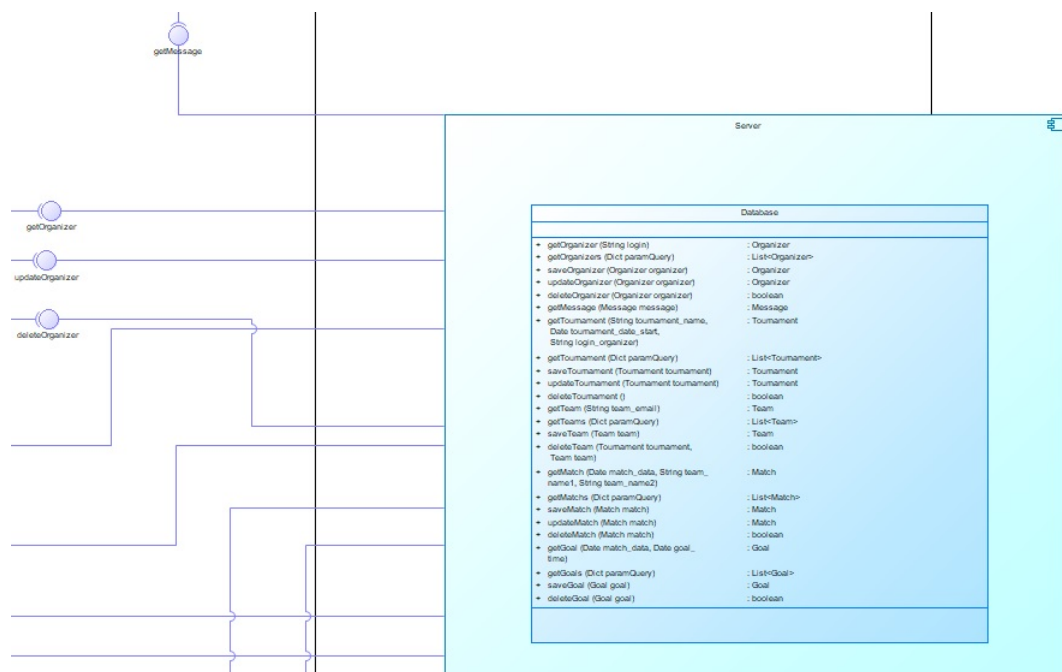


Рисунок 3.8 – Компонентна діаграма(частина 3)

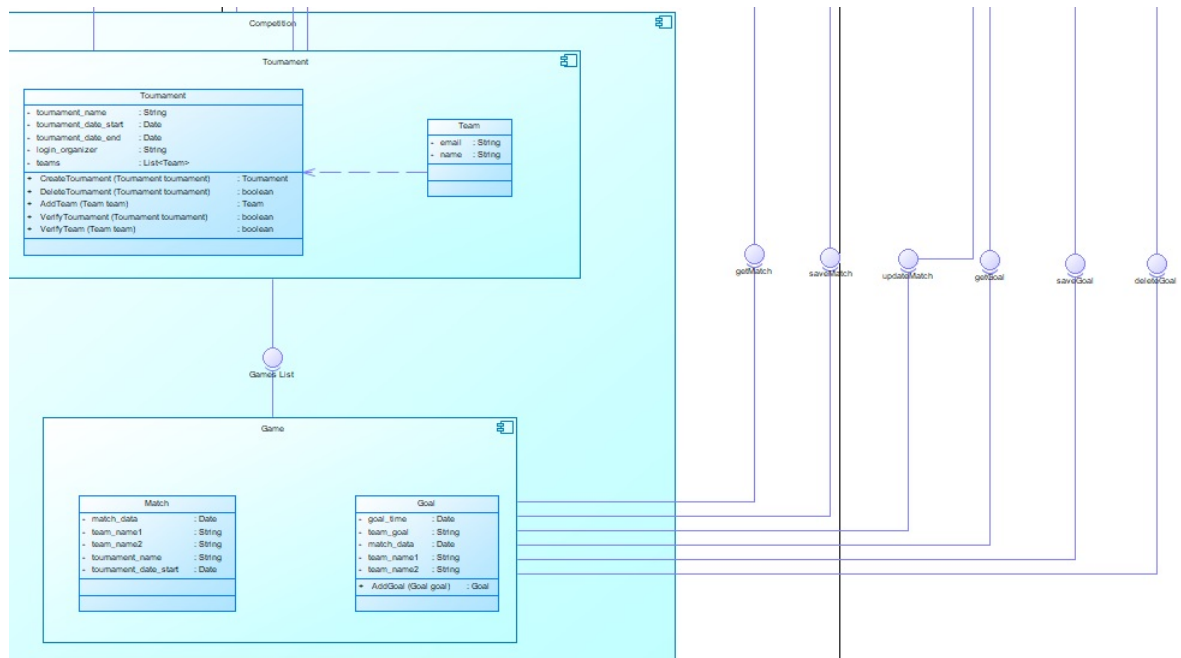


Рисунок 3.9 – Компонентна діаграма(частина 4)

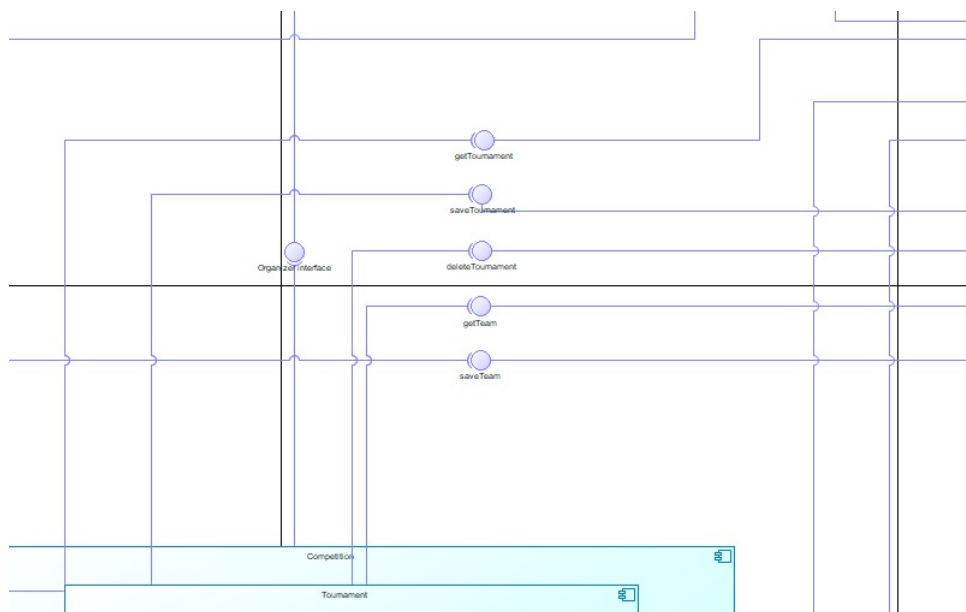


Рисунок 3.10 – Компонентна діаграма(частина 5)



Рисунок 3.11 – Спрінти системи

Generation and Visualization

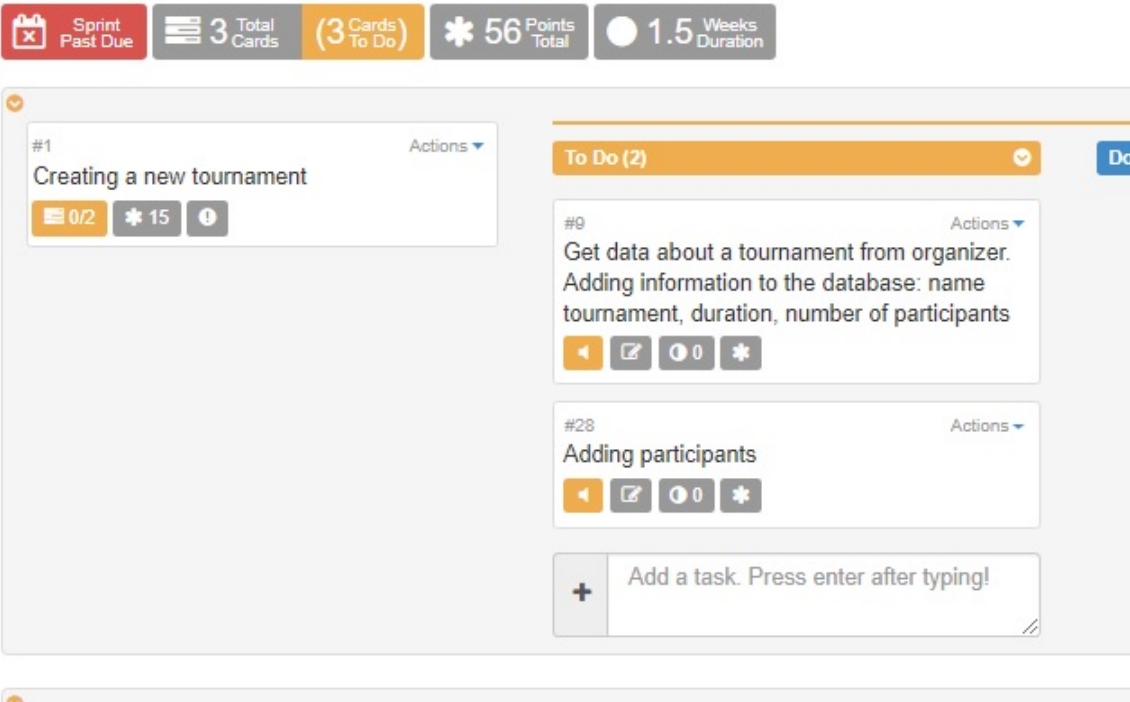


Рисунок 3.12 – Картка "Creating a new tournament"



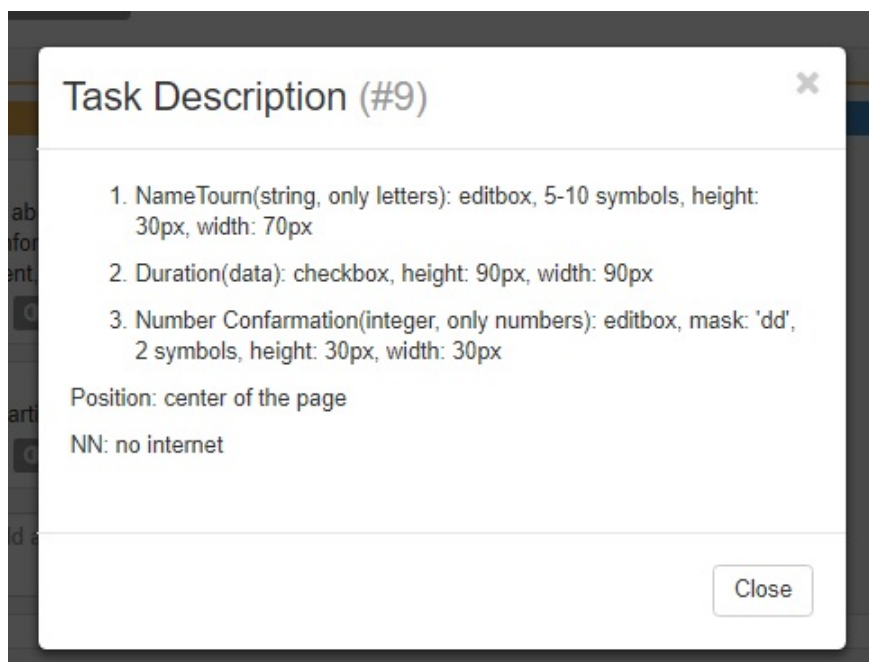


Рисунок 3.13 – Обмеження даних при створенні турніру

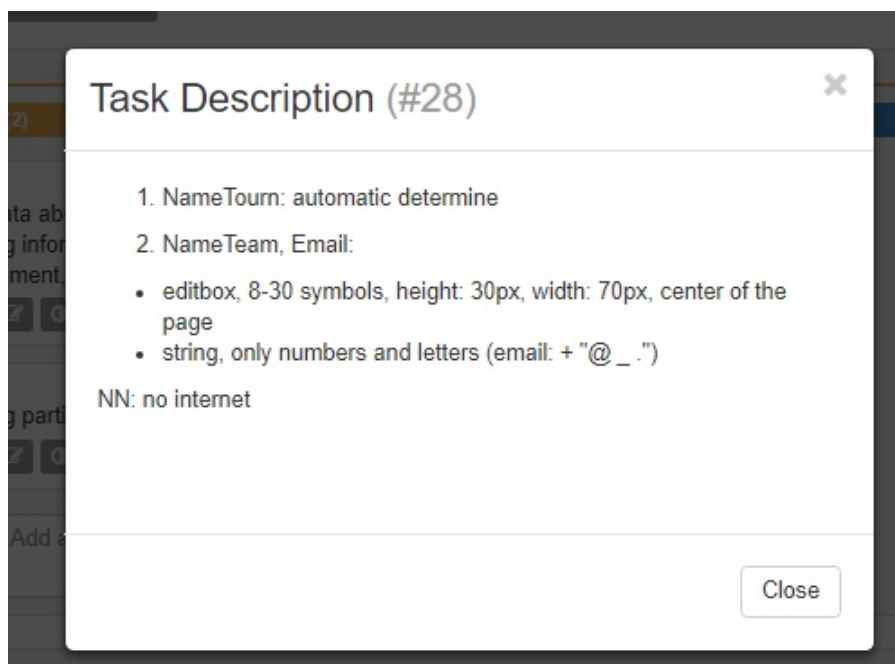


Рисунок 3.14 – Обмеження даних при додавання учасників

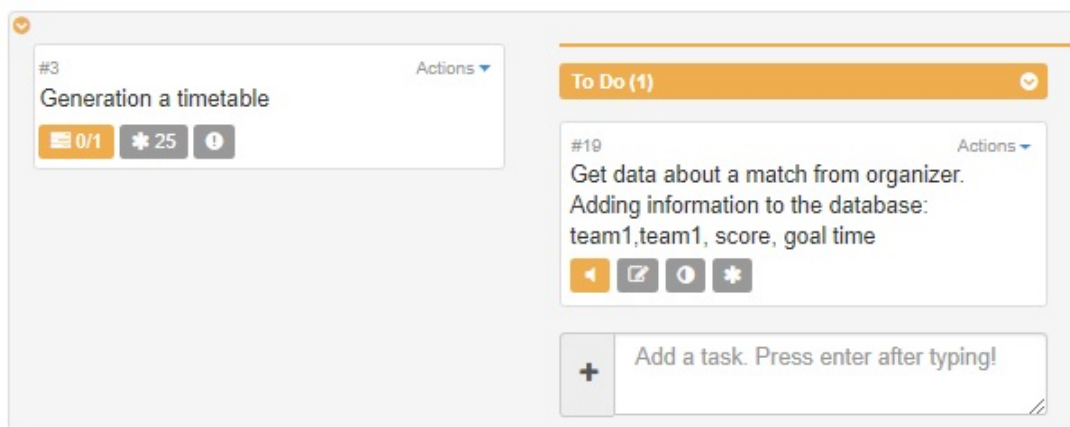


Рисунок 3.15 – Картка "Generation a timetable"

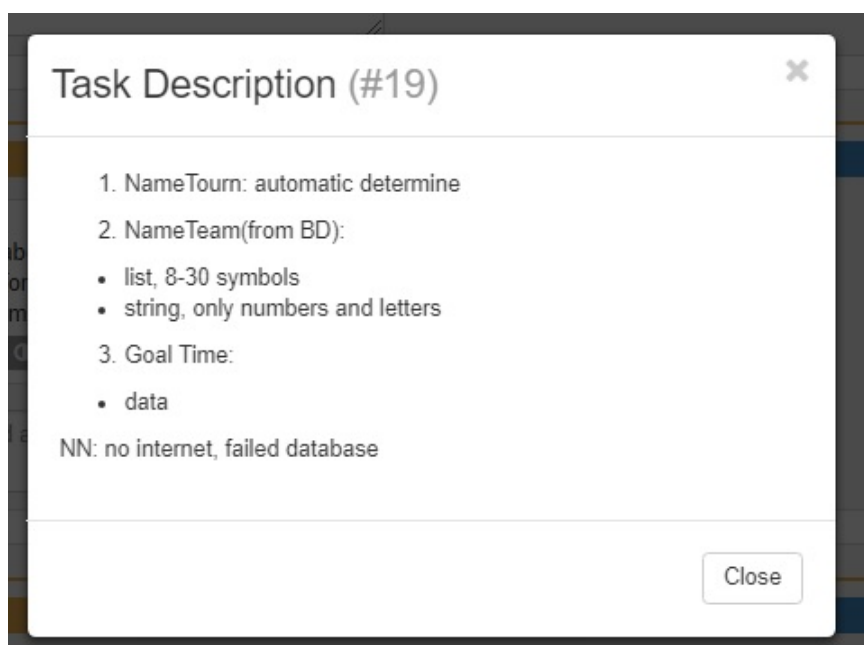


Рисунок 3.16 – Обмеження даних при генерації матчів

Visualization". Дана картка містить 1 задачу, що являє собою візуалізацію турнірної таблиці турніру. На рисунку 3.18 зображено обмеження даних для даної задачі.

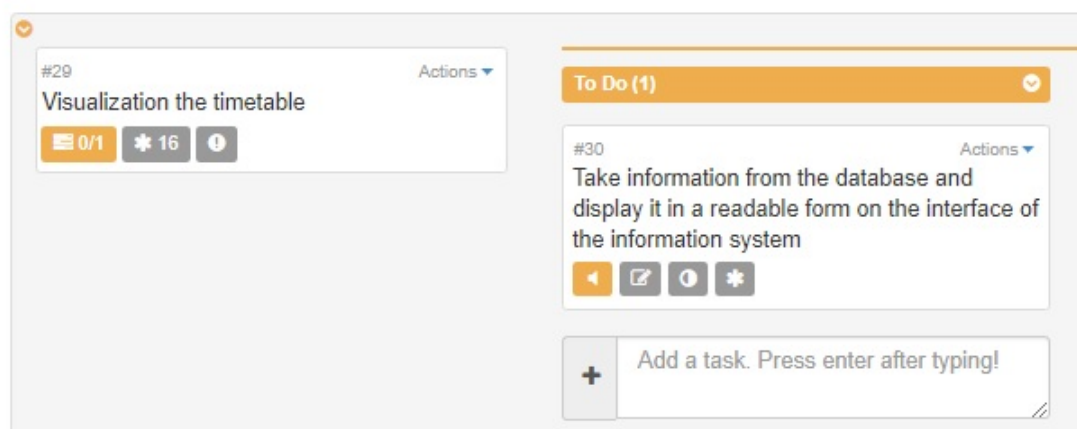


Рисунок 3.17 – Картка "Visualization a table of tournament"

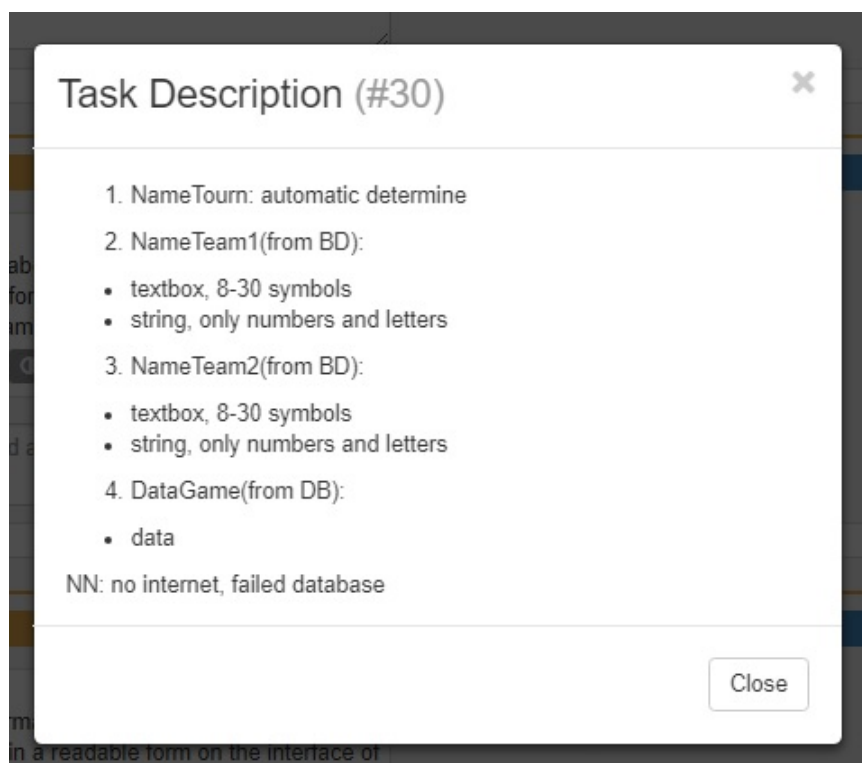


Рисунок 3.18 – Обмеження даних при візуалізації таблиці

Другий спрінт містить реалізацію реєстрації та авторизації користувачів.

На рисунку 3.19 зображено першу картку спрінта "Registration and Authentication". Дана картка містить 1 задачу, що являє собою можливість

реєстрації. Слід зауважити, що можливість організовувати турніри є лише у зареєстрованих користувачів. На рисунку 3.20 зображено обмеження даних для даної задачі.

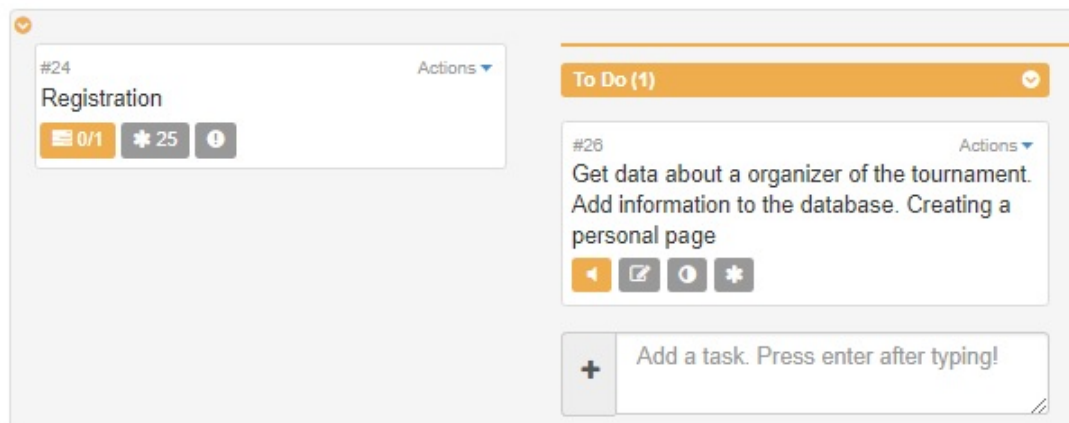


Рисунок 3.19 – Картка "Registration"

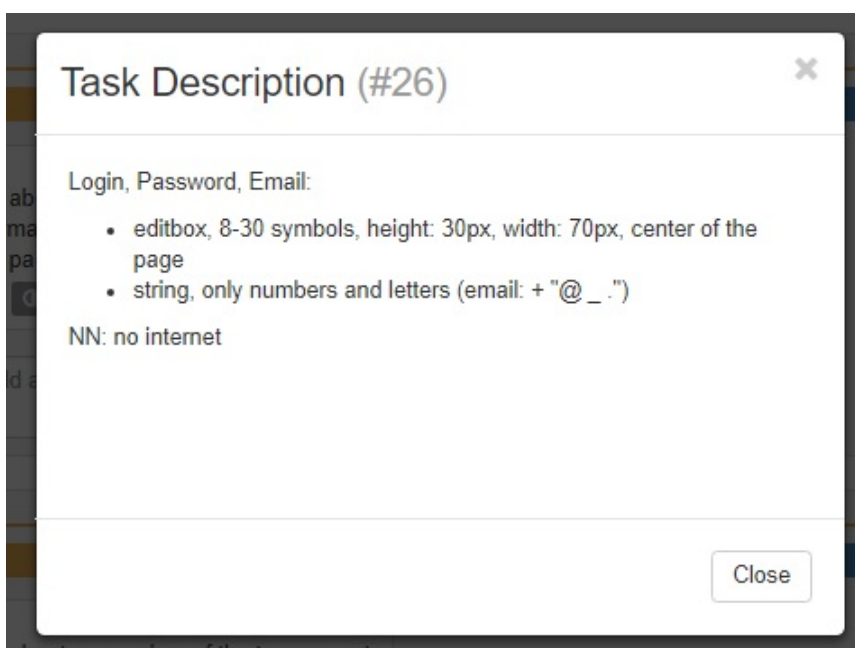


Рисунок 3.20 – Обмеження даних при реєстрації

На рисунку 3.21 зображено другу картку спрінта "Registration and Authentication". Дана картка містить 1 задачу, що являє собою можливість авторизації. На рисунку 3.22 зображено обмеження даних для даної задачі.

Третій спрінт містить додаткові інструменти.

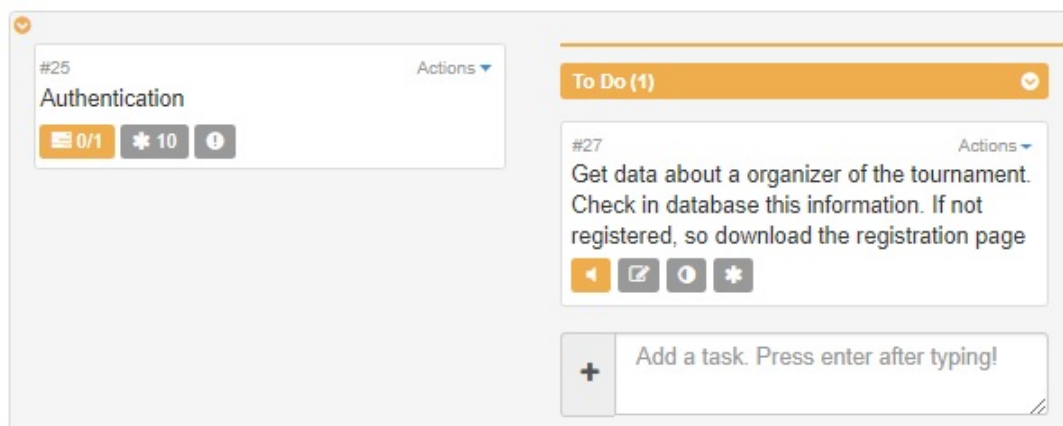


Рисунок 3.21 – Картка "Authentication"

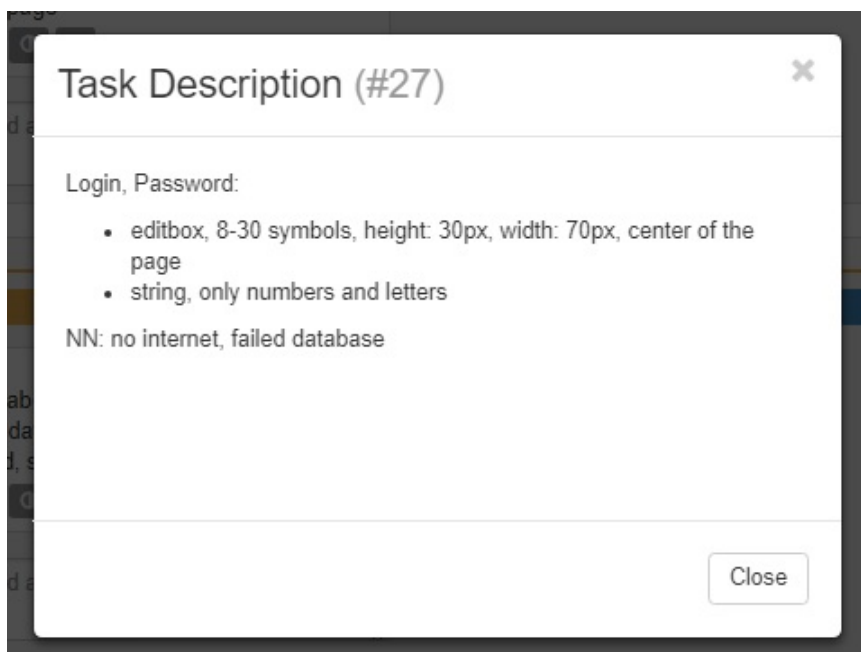


Рисунок 3.22 – Обмеження даних при реєстрації

На рисунку 3.23 зображено першу картку спрінта "Tools". Дана картка містить 1 задачу, що являє собою можливість надсилати повідомлення. Дана можливість є у організатора турніру. Повідомлення надсилається на електронну скриньку. На рисунку 3.24 зображено обмеження даних для даної задачі.

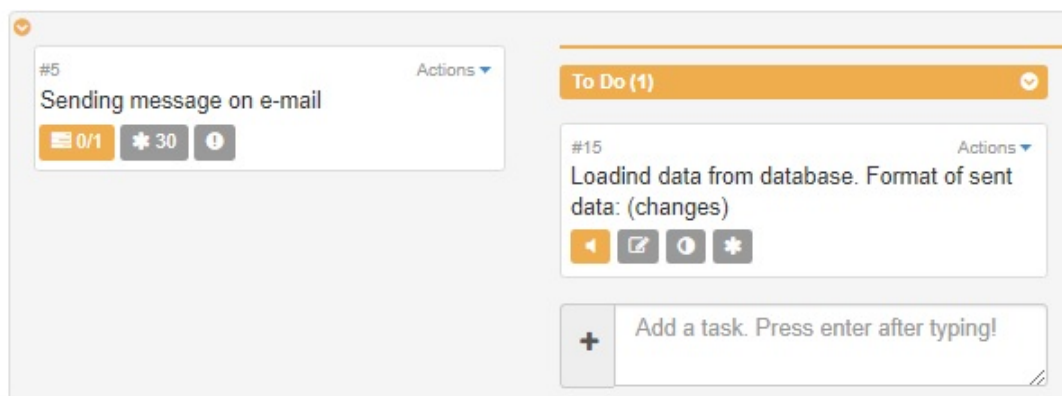


Рисунок 3.23 – Картка "Sending message on e-mail"

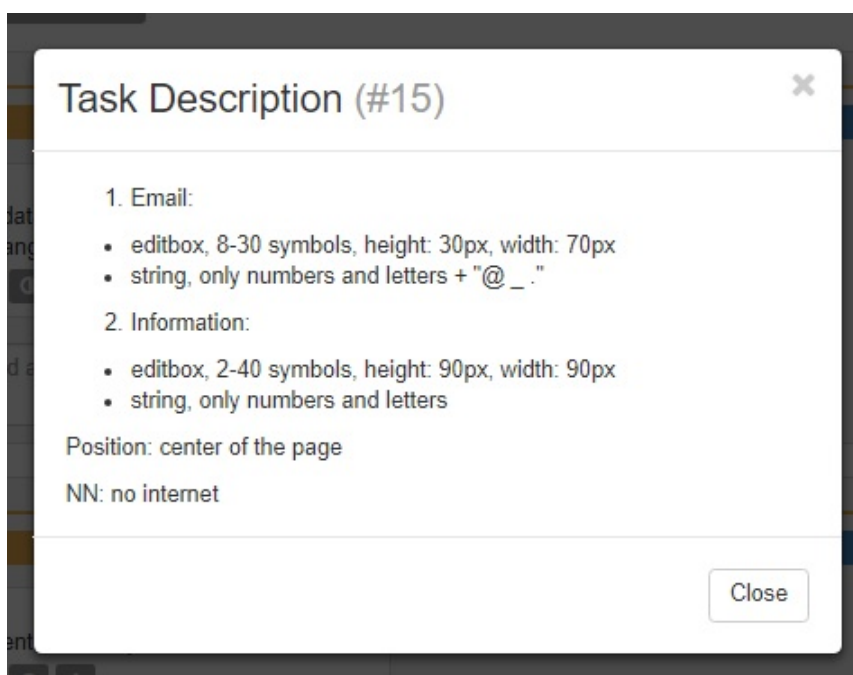


Рисунок 3.24 – Обмеження даних при надсиланні повідомлення

На рисунку 3.25 зображено другу картку спрінта "Tools". Дана картка містить 2 задачі, що являють собою можливість пошуку турнірів. На рисунку 3.26 та 3.27 зображено обмеження даних задач.

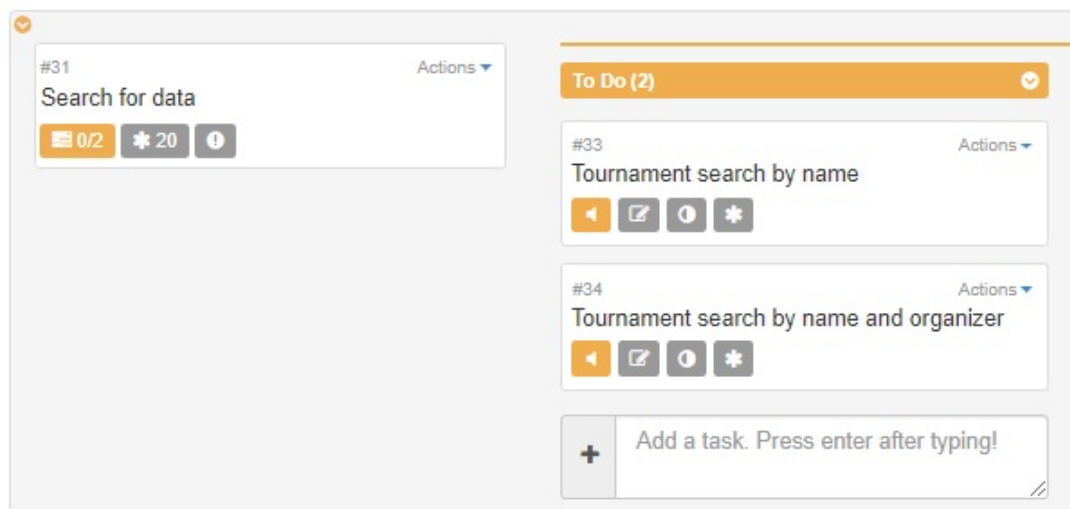


Рисунок 3.25 – Картка "Search for data"

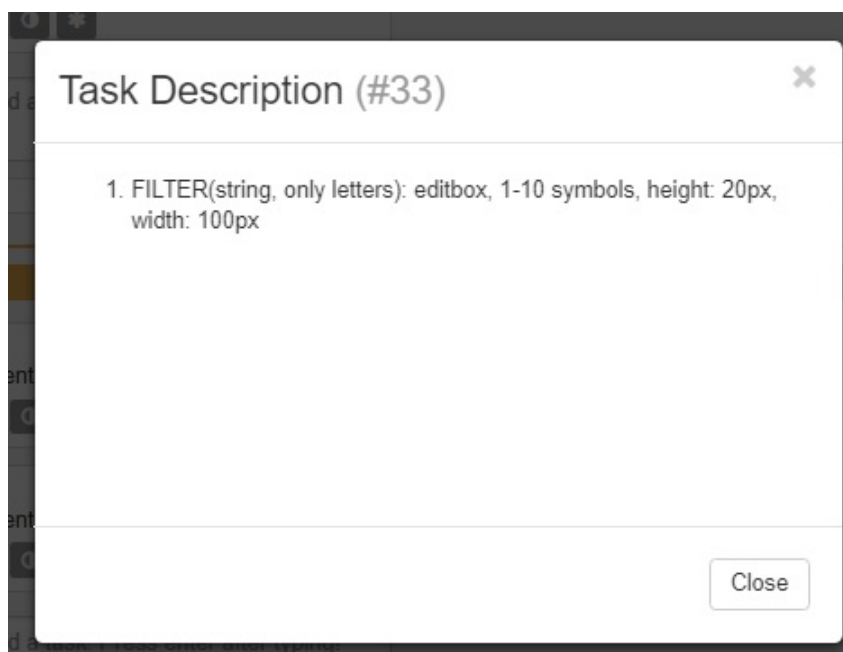


Рисунок 3.26 – Обмеження даних при пошуку турніру(1 версія)

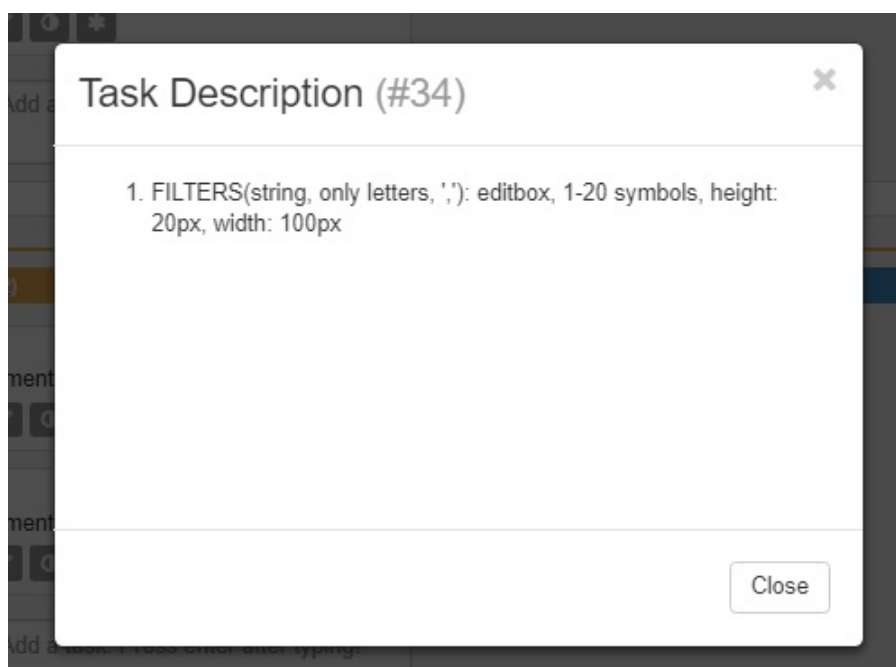


Рисунок 3.27 – Обмеження даних пошуку турніру(2 версія)



## 4 ІНФОЛОГІЧНЕ ПРОЕКТУВАННЯ

Визначимо основні сутності проекту, що реалізується, їх атрибутів, ключів та зв'язків між собою. Отже, до головних сутностей віднесемо:

- а) організатор(користувач);
- б) турнір;
- в) матч;
- г) рахунок;
- д) повідомлення;
- е) команда.

Концептуальна модель[3] інформаційної системи зображена на рисунку 4.1.

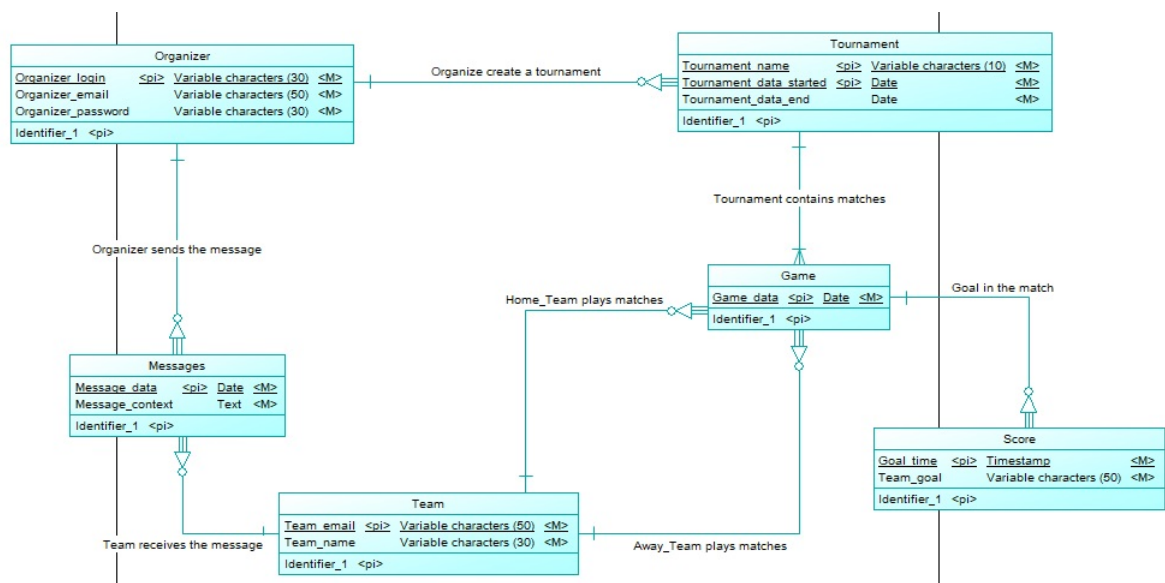


Рисунок 4.1 – Conceptual model

Розглянемо детільніше кожну з сутностей.

#### 4.1 Організатор

Організатор - зареєстрований користувач, який має змогу керувати турнірами. Дана сутність є однією з головних, оскільки без організатора не може існувати турнір та інше. Визначимо наступні атрибути даної сутності

- а) логін, що є ключем;
- б) електронна пошта;
- в) пароль.

Кожен з цих атрибутів обов'язково не нульовий. При побудові логічної моделі ще додадуться ідентифікатори, що визначатимуть чи даний користувач є адмін та чи він не видалений.

Організатор має зв'язок "один до багатьох" до повідомлень, та "один до багатьох" до турнірів.

#### 4.2 Турнір

Турнір - сутність, що фактично являє собою головний об'єкт розробки даної системи. Визначимо наступні атрибути даної сутності

- а) назва турніру, що є ключем;
- б) дата початку, що є ключем;
- в) дата кінця.

Кожен з цих атрибутів обов'язково не нульовий. При побудові логічної моделі ще додадуться ідентифікатори, що визначатимуть чи він не видалений.

Турнір має зв'язок "один до багатьох" до матчів, та "багато до одного" до організатора.

### 4.3 Матч

Матч - сутність, що описує події в турнірі, тобто визначає пари команд, що грали між собою. Визначимо наступні атрибути даної сутності

а) дата матчу, що є ключем.

При побудові логічної моделі ще додадуться ідентифікатори, що визначатимуть чи він не видалений.

Матч має зв'язок "один до багатьох" до голів, та два зв'язки "багато до одного" до команд, оскільки в матчі грають дві команди.

### 4.4 Рахунок

Рахунок - сутність, що визначає голи. За допомогою цієї сутності можна порахувати рахунок у матчі. Визначимо наступні атрибути даної сутності

а) час голу, що є ключем;

б) назва команди, що забила гол.

Кожен з цих атрибутів обов'язково не нульовий. При побудові логічної моделі ще додадуться ідентифікатори, що визначатимуть чи він не видалений.

Гол має зв'язок "багато до одного" до матчів.

### 4.5 Команда

Команда - сутність, що визначає учасників турніру. Визначимо наступні атрибути даної сутності

а) електронна пошта, що є ключем;

б) назва команди.

Кожен з цих атрибутів обов'язково не нульовий.

Команда має зв'язок "один до багатьох" до повідомлень, та два зв'язки "один до багатьох" до матч.

#### 4.6 Повідомлення

Повідомлення - сутність, що визначає повідомлення, надіслані організатором турніру для команд. Визначимо наступні атрибути даної сутності

а) дата повідомлення;

б) вміст повідомлення.

Кожен з цих атрибутів обов'язково не нульовий.

Повідомлення має зв'язки "багато до одного" до організатора та команди.

## 5 ДАТАЛОГІЧНЕ ПРОЕКТУВАННЯ

Зпроектуємо логічну, логічно-оптимізовану та фізичну[1] моделі інформаційної системи організації футбольних турнірів.

Логічна модель інформаційної системи зображена на рисунку 5.1.

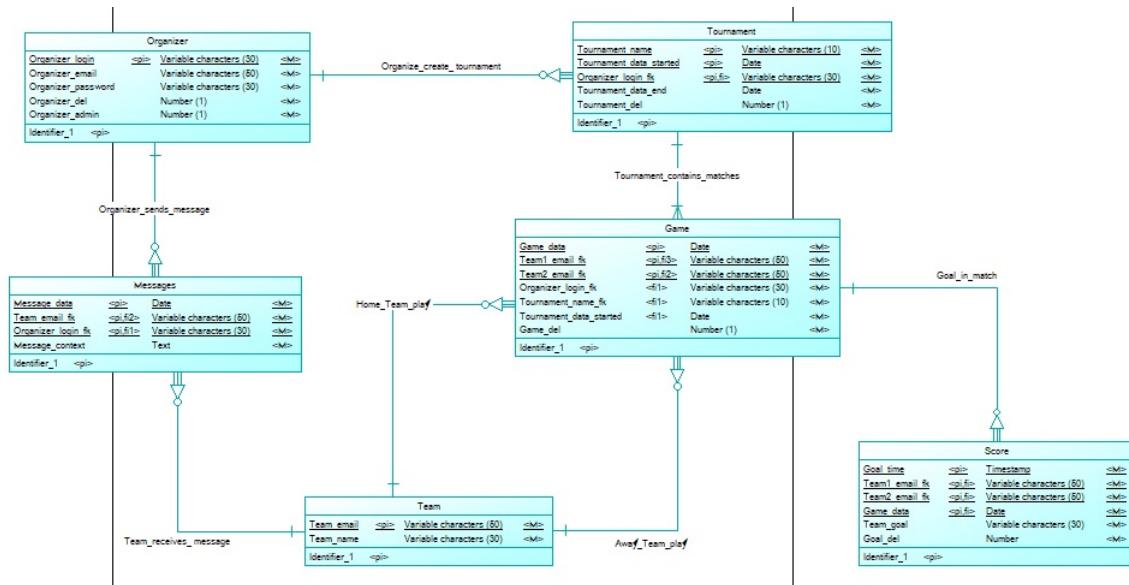


Рисунок 5.1 – Logical model

У порівнянні з концептуальною, відбулися певні зміни:

- з'явилися зовнішні ключі;
- Організатор має атрибути, що визначають чи це адміністратор, та чи користувач видалений. Дані поля можуть набувати значення 0 або 1;
- Турнір, матч та гол мають атрибут, що визначає чи запис видалений, може бути значення 0 або 1.

Логічно-оптимізаційна модель інформаційної системи зображена на рисунку 5.2.

У порівнянні з логічною, відбулися певні зміни:

- турнір має новий атрибут - id.

Ключем даної сутності є id та логін організатора турніру.

Фізична модель інформаційної системи зображена на рисунку 5.3.

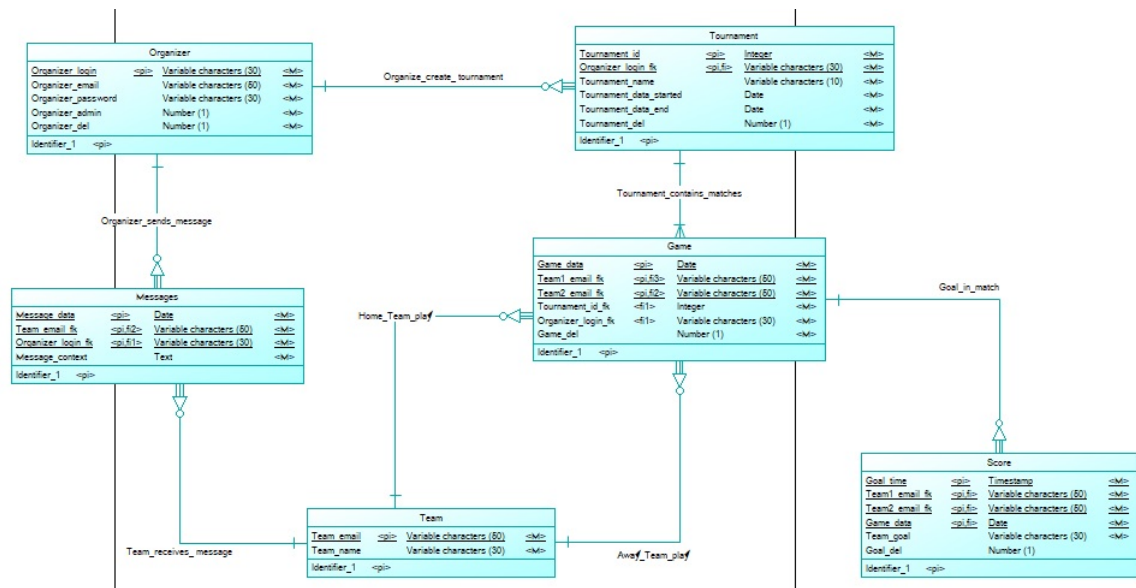


Рисунок 5.2 – Logical optimization model

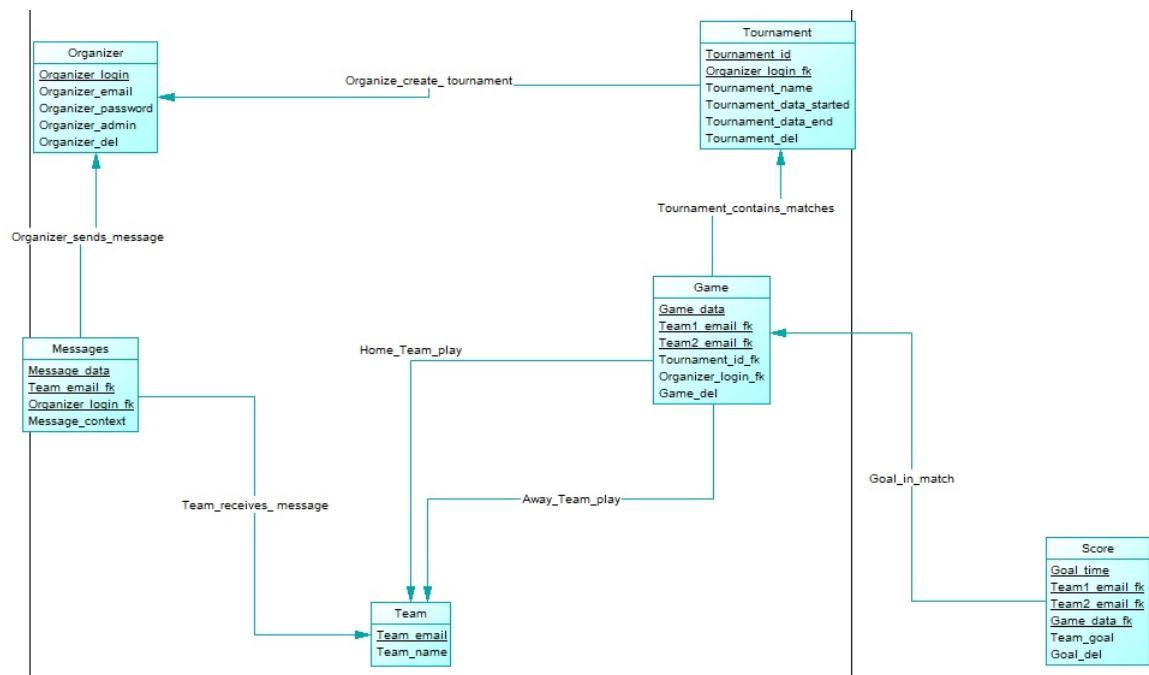


Рисунок 5.3 – Physical model

## ВИСНОВКИ

Було досліджено та реалізовано інформаційну систему контролю футбольних турнірів. При дослідженні було спроектовано концептуальну, логічну та фізичну моделі бази даних, визначено Use Cases, діаграму компонент та діаграми послідовстей для кожної ролі користувачів. Реалізована система має наступний функціонал для користувачів:

- а) реєстрація та авторизація користувачів;
- б) додавання турнірів, матчів, команд;
- в) редагування даних;
- г) відправлення повідомлень на електронну пошту.

Реалізовано одне з головних завдань - простота у використанні та контролі.

В рамках виконання курсової роботи було отримано практичні навички побудови архітектури системи для окремих блоків процесів системи.

## ПЕРЕЛІК ПОСИЛАНЬ

1. UML Diagrams [Електронний ресурс]. <https://www.smartdraw.com/uml-diagram/>
2. SCRUM [Електронний ресурс]. <https://www.flying-donut.com/>
3. Oracle. Help center [Електронний ресурс]. <https://docs.oracle.com/en/>
4. Форт Б. SQL. — [3th ed.] — Москва, 2006. — 169 с.