2 АРХІТЕКТУРА ПРОГРАМНОГО ЗАБЕЗПЕЧЕННЯ

* 1. Перелік використаних технологій

Для реалізації даного інтернет-додатку були використані декілька технологій.

ASP.NET — технологія створення веб-застосунків і веб-сервісів від компанії Майкрософт. Вона є складовою частиною платформи Microsoft. NET і розвитком старішої технології Microsoft ASP. У цей час останньою версією цієї технології є ASP.NET 4.5 [1].

Язик програмування JavaScript. JavaScript - назва реалізації стандарту мови програмування ECMAScript компанії Netscape, базується на принципах прототипного програмування. Найпоширеніше і найвідоміше застосування мови — написання сценаріїв для веб-сторінок, але, також, використовується для впровадження сценаріїв керування об’єктами вбудованими в інші програми [2].

Бібліотека JQuery. jQuery — популярна JavaScript-бібліотека з відкритим сирцевим кодом. Вона була представлена у січні 2006 року у BarCamp NYC Джоном Ресіґом (John Resig). Згідно з дослідженнями організації W3Techs, JQuery використовується понад половиною від мільйона найвідвідуваніших сайтів. jQuery є найпопулярнішою бібліотекою JavaScript, яка посилено використовується на сьогоднішній день. Синтаксис jQuery розроблений, щоб зробити орієнтування у навігації зручнішим завдяки вибору елементів DOM, створенню анімації, обробки подій, і розробки AJAX-застосунків. jQuery також надає можливості для розробників, для створення плагінів у верхній частині бібліотеки JavaScript. Використовуючи ці об'єкти, розробники можуть створювати абстракції для низькорівневої взаємодії та створювати анімацію для ефектів високого рівня. Це сприяє створенню потужних і динамічних веб-сторінок [3].

AJAX (Asynchronous JavaScript And XML) — підхід до побудови користувацьких інтерфейсів веб-застосунків, за яких веб-сторінка, не перезавантажуючись, у фоновому режимі надсилає запити на сервер і сама звідти довантажує потрібні користувачу дані. AJAX — один з компонентів концепції DHTML [4].

Web-service - програмна система, що ідентифікується рядком URI, та публічні інтерфейси та прив'язки якої визначені та описані мовою XML. Опис цієї програмної системи може бути знайдено іншими програмними системами, які можуть взаємодіяти з нею відповідно до цього опису з використанням повідомлень, що базуються на XML та передаються за допомогою інтернет-протоколів [5].

Extensible Stylesheet Language Transformations, або XSLT — функціональна повна за Тьюрінгом мова програмування, яка використовується для програмування переробки XML документів. При цьому вихідний документ не змінюється, натомість, на основі результатів переробки створюється новий. Новий документ може бути серіалізовано (виведено) обробником в стандартний синтаксис XML, або інший формат, такий як HTML або простий текст. Найчастіше, XSLT використовується для перетворення структурованих XML документів із однієї XML схеми в іншу, або для перетворення у веб сторінки або PDF документи [6].

Microsoft SQL Server — комерційна система керування базами даних, що розповсюджується корпорацією Microsoft. Мова, що використовується для запитів — Transact-SQL, створена спільно Microsoft та Sybase. Transact-SQL є реалізацією стандарту ANSI/ISO щодо структурованої мови запитів (SQL) із розширеннями. Використовується як для невеликих і середніх за розміром баз даних, так і для великих баз даних масштабу підприємства [7].

ORM-засіб LINQ to SQL. LINQ to SQL є компонентом .NET Framework версії 3.5, який представляє можливість для управління реляційними даними як об’єктами. В Linq to SQL модель даних реляційної бази даних використовується через об’єктну модель, вираженої в мові програмування розробника. При запуску програмного забезпечення Linq to SQL перетворює запити LINQ із об’єктної моделі в SQL і відправляє їх до бази даних для виконання. Коли база даних повертає результати, LINQ to SQL перетворює їх назад в об’єкт, з яким можна працювати на особистій мові програмування [8].

Windows Forms – назва інтерфейсу програмного додатку (API), що відповідає за графічний інтерфейс користувача і є частиною Microsoft .NET Framework [9].

HTML (англ. HyperText Markup Language — Мова розмітки гіпертексту) — 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 [10].

Каскадні таблиці стилів (англ. Cascading Style Sheets або скорочено CSS) — спеціальна мова, що використовується для відображення 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. Специфікації CSS були створені та розвиваються Консорціумом Всесвітньої мережі [11].

2.2 Архітектура ПЗ

Програмне забезпечення відповідаючи за додавання інформації із football.ua до бази даних має наступні збірки:

* parseHtml – збірка для запуску програми;
* createDataInDB – збірка для аналізу xml, занесення отриманої інформації до колекції та збереження даних в базу даних;
* footballDB – збірка, яка містить в собі ORM-засіб LINQ to SQL для представлення реляційних даних у вигляді об’єктів;
* managerDBFootballFan – збірка для роботи з FootballDB збіркою.

Збірка Parsehtml.exe отримує URL адресу чемпіонату, отримує html сторінки, робить із html валідний xml. Після цього віддає цей xml для перетворення, яке відбувається за допомогою xslt. Отриманий xml аналізується і потім з нього вже будується остаточний xml, який буде зберігатися у файловій системі. Після того як збережений xml буде обраний для аналізу, ParseHtml.exe збірка звертається до CreateDataInDB.dll, передаючи їй фізичну адресу файлу xml у файловій системі. CreateDataInDB.dll аналізує цей xml, завантажує необхідні ресурси з інтернету, та добавляє отриману інформацію до колекції. Коли всі етапи пройдені, всі колекції побудовані, CreateDataInDB.dll проходить по всім колекціям, звертаючись до збірки ManagerDBFootballFan.dll і передаючи їй дані для збереження в БД. В збірці FootballDB.dll виконується побудова об’єктів для створення або оновлення даних. Після цього FootballDB.dll відбувається процес перетворення запитів LINQ із об`єктної моделі в SQL і відправляє їх до бази даних FootballFan для виконання. Потім база даних повертає результати, LINQ to SQL перетворює їх назад в об’єкт. Після цього управління повертається знову до збірки ManagerDBFootballFan.dll, потім до CreateDataInDB.dll, потім до збірки ParseHtml.exe.

Схематично зв'язок цих збірок показано на рисунку 2.1.



Рисунок 2. 1 – Зв'язок збірок системи

Інтернет-додаток для збору та обробки інформації про спортивні заходи має наступні збірки:

* footballFan.dll – збірка, яка представляє собою інтернет-додаток;
* footballDB – збірка, яка містить в собі ORM-засіб LINQ to SQL для представлення реляційних даних у вигляді об’єктів;
* managerDBFootballFan – збірка для роботи з FootballDB збіркою.

Схематично зв'язок цих збірок показано на рисунку 2.2.



Рисунок 2.2 – Зв'язок збірок системи інтернет-додатку

Схема роботи додатка така ж сама як і програми для додавання інформації в базу даних. Збірка FootballFan.dll це і є сайт, який бачить користувач. Ця збірка звертається до збірки з метою отримати інформацію для відображення на сайті. Далі робота проходить за тією ж схему, що описана для програми додавання інформації в базу даних.

В збірці FootballFan.dll знаходяться всі необхідні ресурси, які потрібні для нормальної роботи додатку, веб-сервіси, javaScript-файли, JQuery-файли, сторінки .aspx, а також Web User Controls.

2.3 Структура бази даних

Для написання даного програмного продукту було обране СУБД MS SQL Server. Дана СУБД є досить легкою в розумінні створення таблиць та схеми бази даних. Створювати таблиці в даній СУБД досить легко за рахунок зрозумілості інтерфейсу та відсутності зайвих елементів, які можуть дезорієнтувати користувача. Також в СУБД MS SQL Server може зберігатися досить велика кількість даних.

База даних містить наступні таблиці:

* + Aspnet\_Membership;
  + aspnet\_Users;
  + aspnet\_Roles;
  + aspnet\_UsersInRoles
  + aspnet\_Profile
  + league;
  + club;
  + staff;
  + playInClubs;
  + table;
  + calendar;
  + news;
  + onlineTextTranslation;
  + comments\_Match;
  + comments\_News

Схема бази даних наведена на рисунку 2.3. Вона складається з 15 таблиць, п`ять з яких використовуються програмним інтерфейсом Membership API. Membership API – це каркас, побудований на базі існуючої інфраструктури аутентифікація форм [12]. Останні десять були створені для роботи інтернет-додатку.

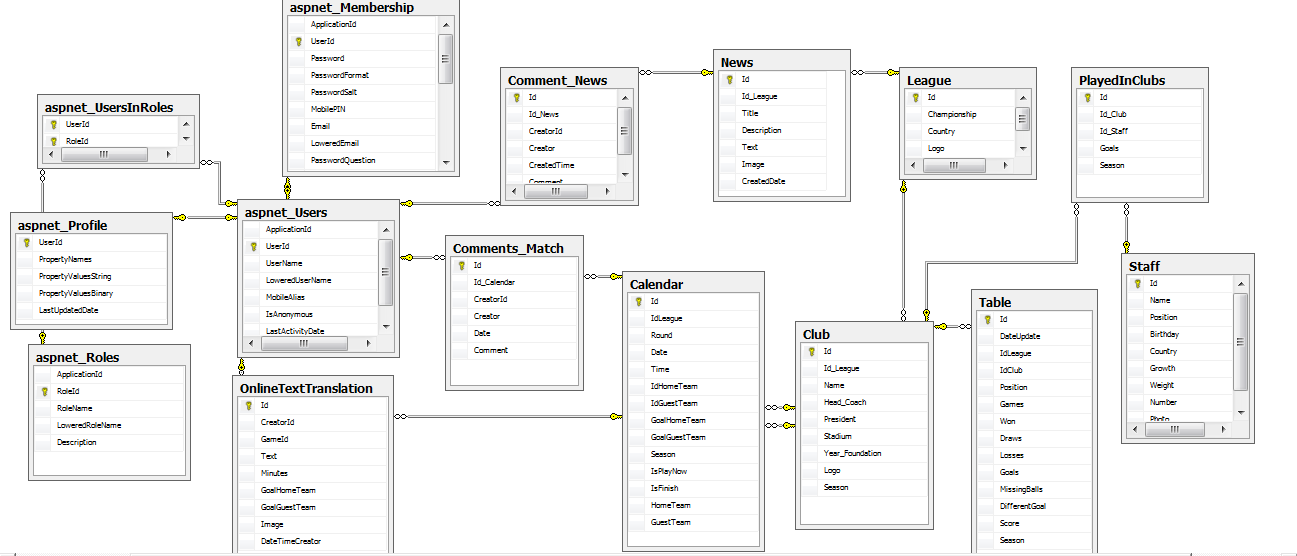


Рисунок 2.3 – Схема бази даних

В таблиці “League” зберігається інформація про чемпіонати. Вона має наступні поля: Id\*, Championship, Country, Logo. Дана таблиця зв’язана з таблицями “Club” зв’язком «один до багатьох» та “News” також зв’язком «один до багатьох». Структура таблиці “League” наведена на рисунку 2.4.

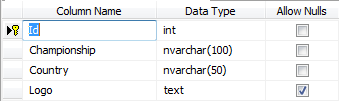


Рисунок 2.4 – Структура таблиці “League”

В таблиці “Club” зберігається інформація про клуби. Таблиця має наступні поля: Id\*, Id\_League, Name, Head\_Coach, President, Stadium, Year\_Foundation, Logo, Season. Дана таблиця зв’язана з таблицями: “Table”, “Calendar” , ”League”. Зв’язки з цими таблицями однакові «один до одного». Між таблицями «Club» та «Calendar» існує два зв’язка, один для команди, яка грає вдома, а другий зв'язок для команди гостей. Дана таблиця має наступну структуру:

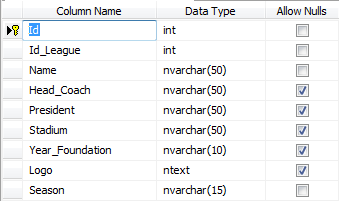


Рисунок 2.5 – Структура таблиці “Club”

В таблиці “Table” зберігається інформація про результати клубів у чемпіонаті різних сезонів. Дана таблиця має наступні поля: Id\*, DateUpdate, IdLeague, IdClub, Position, Games, Won, Draws, Losses, Goals, MissingBalls, DifferentGoal, Score, Season. Хоча дана таблиця не зв’язана з таблицею “League” в ній присутнє поле IdLeague. Це поле є службовим для спрощення пошуку та отримання даних. Структура таблиці “Table” наведена на рисунку 2.6.

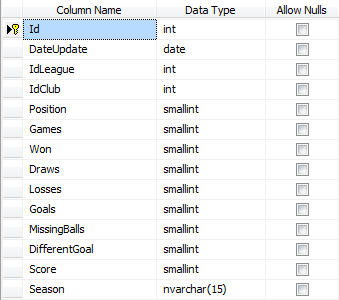


Рисунок 2.6 – Структура таблиці “Table”

В таблиці “Calendar” зберігається інформація про календар ігор. Вона має наступні поля: Id\*, IdLeague, Round, Date, Time, IdHomeTeam, IdGuestTeam, GoalHomeTeam, GoalGuestTeam, Season, IsPlayNow, IsFinish, HomeTeam, GuestTeam. Поле IsPlayNow потрібне для відображення того що в даний момент йде гра, а поле IsFinish відображує, що гра вже закінчилась. Ці поля потрібні для збереження службових даних. Таблиця “Calendar” зв’язана з таблицею “OnlineTextTranslation” зв’язком «один до багатьох». Таблиця має наступну структуру:

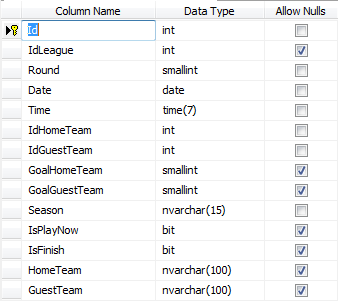


Рисунок 2.7 – Структура таблиці “Calendar”

Таблиця “Staff” зберігає інформацію про гравців. Вона має наступні поля: Id\*, Name, Position, Birthday, Country, Growth, Weight, Number, Photo. Ця таблиця зв’язана з таблицею “PlayInClub” зв’язком «один до багатьох» та має наступну структуру:

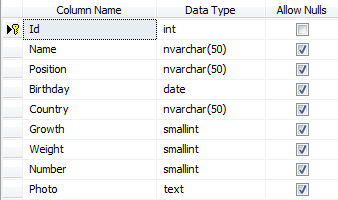


Рисунок 2.8 – Структура таблиці “Staff”

Таблиця PlayInClubs зв’язує таблиці “Club” та “Staff” та має з ними зв'язок «один до багатьох». Це потрібно для того, щоб можна було фільтрувати інформацію по сезонам, бо в одному сезоні гравець грає за один клуб, а після трансферного ринку вже за інший клуб. Ця таблиця має наступні поля: Id\*, IdClub, IdStaff, Season, Goals. Таблиця має наступну структуру:

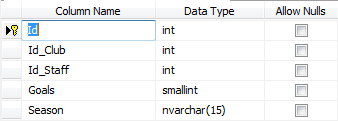


Рисунок 2.9 – Структура таблиці “PlayInClubs”

Таблиця “News” зберігає інформацію про новини чемпіонатів. Вона має наступні поля: Id\*, Id\_League, Title, Description, Text, Image, CreateDate. Дана таблиця зв’язана з таблицею “Comment\_News” зв’язком «один до багатьох», та має наступну структуру:

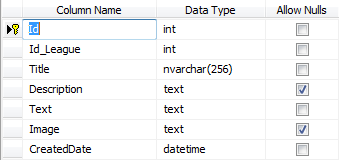


Рисунок 2.10 – Структура таблиці “News”

Таблиця “Comment\_News” зберігає коментарі користувачів про новини. Таблиця має наступні поля: Id\*, Id\_News, CreatorId, Creator, CreatedTime, Comment. Таблиця має наступну структуру:

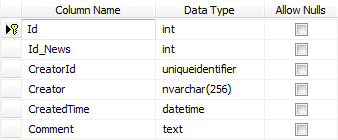


Рисунок 2.11 – Структура таблиці “ Comment\_News ”

Таблиця “Comment\_Match” зберігає коментарі користувачів про ігри. Таблиця має наступні поля: Id\*, Id\_Calendar, CreatorId, Creator, CreatedTime, Comment. Таблиця має наступну структуру:

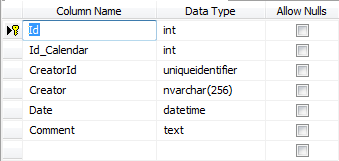


Рисунок 2.12 – Структура таблиці “ Comment\_Match”

Таблиця “OnlineTextTranslation” зберігає інформацію про матч, яку створює користувач з відповідною роллю. Таблиця має наступні поля: Id\*, CreatorId, GameId, Text, Minutes, GoalHomeTeam, GoalGuestTeam, Image, DateTimeCreator. Дана таблиця зв’язана з таблицями “aspnet\_Users” та “Calendar” зв’язком «один до багатьох» та має наступну структуру:

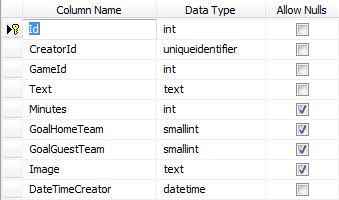


Рисунок 2.13 – Структура таблиці “OnlineTextTranslation”

3 ОПИС ПРОГРАМИ

3.1 Системні вимоги

Зі сторони сервера розроблена програмна система потребує, щоб були наступні інструментарії:

* Microsoft Windows Server 2008 R2 - серверна операційна система для підтримки роботи веб-додатку та сервисів, що він використовує;
* Internet Information Services (IIS) 7.0 - це набір серверів для декількох служб Інтернету від компанії Майкрософт. Основний компонент IIS — веб-сервер, який дозволяє розміщувати в Інтернеті сайти [10].
* Microsoft SQL Server 2008 R2 – реляційна СУБД для зберігання даних додатка та надання доступу до них;
* .net Framework 4.0.

Для користування веб-системою з боку користувача, йому потрібно мати установленим один із перекислених браузерів:

* Google Chrome будь-яка версія починаючи з 20;
* Mozilla Firefox будь яка версія, починаючи з версії 12;
* Internet Explorer починаючи з версії 8;
* Opera, починаючи з версії 10.

Використовуючи дану веб-систему, у користувача обов’язково повинен бути включеним підтримка javaScript у браузері, в іншому випадку додаток буде працювати не коректно.

* 1. Опис інтерфейсу користувача

Веб-системою може користуватися як зареєстрований користувач, так і анонімний. У зареєстрованого користувача є можливість залишати коментарі для матчів та новин. Цей користувач має роль ‘Fan’. Також в системі є ще одна роль ‘Admin’. Користувач з даною роллю має можливість керувати користувачами: створювати, редагувати, видаляти, назначати ролі. А також керує ролями: створює, редагує, видаляє. Також адміністратор сайту має можливість здійснити пошук користувачів по наступним критеріям:

* логін;
* адреса електронної пошти;
* ім’я користувача.

Сторінка керування користувачами має наступний вигляд:



Рисунок 3.1 – Сторінка керування користувачами

Сторінка додавання користувачів показана на рисунку 3.2, а сторінка керування ролями на рисунку 3.3.

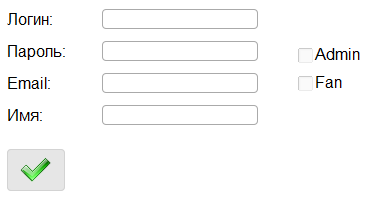


Рисунок 3.2 – Сторінка додавання користувачів



Рисунок 3.3 – Сторінка додавання та видалення ролей

Також у зареєстрованого користувача є можливість переглянуту та редагувати свої персональні данні (див. рис. 3.4).

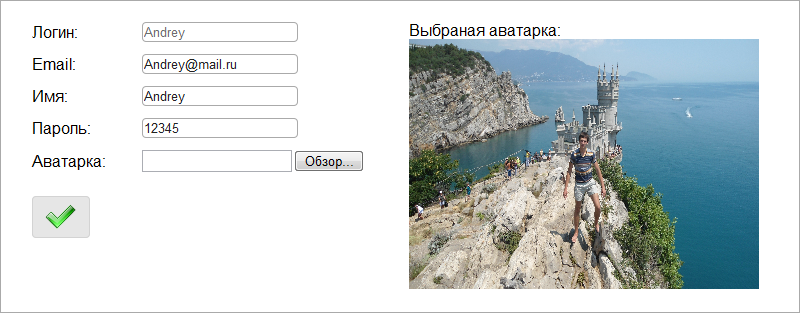


Рисунок 3.4 – Сторінка перегляду особистих даних

Анонімний користувач може лише проглядати інформацію, читати коментарі для новин або матчів, які були написані зареєстрованими користувачами, а також читати онлайн-текстову трансляцію.

Головна сторінка даної веб-системи відображає останні п’ять новин, які виконані за допомогою JQuery ui. Ці новини показані у вигляді слайдера, який через кожні десять секунд гортає новини і відображає їх у більшому вікні (див. рис. 3.5). Також на даній сторінці знаходяться «Центр новостей» та «Матч-цент». Центр новин дозволяє переглянути останні двадцять новин, які відносяться до будь-якого з чемпіонатів. Дані новини отримуються із бази даних по часу добавлення.. Центр новин наведений на рисунку 3.6. “Матч-центр” дозволяє користувачу переглянути результати ігор поточного дня, які вже закінчились, або ідуть в даний момент. Якщо гра ще йде, то рахунок матчу відображається на зеленому фоні. Якщо гра ще не розпочалась або вже закінчилась, то рахунок відображається на чорному фоні. Матч-цент показаний на рисунку 3.7.

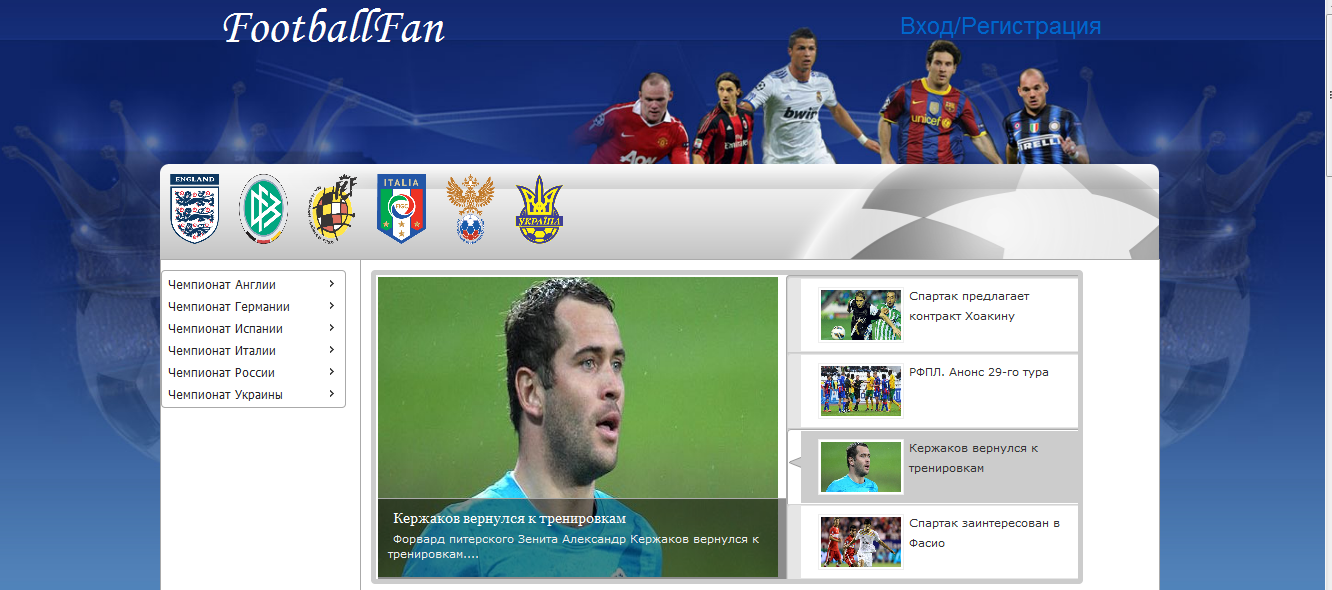


Рисунок 3.5 – Головна сторінка та слайдер новин

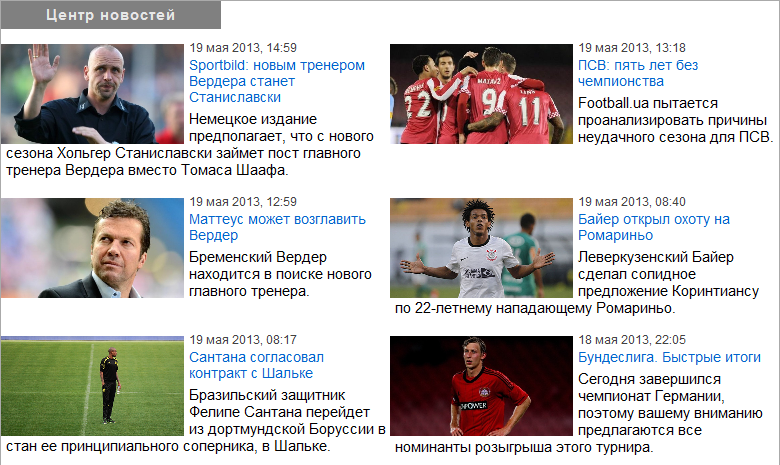


Рисунок 3.6 – Цент новин

На всіх сторінка присутнє шапка сайту, головне меню та меню вибору чемпіонаті. Меню вибору чемпіонату це посилання на сторінку відображення всіх клубів чемпіонату (див. рис. 3.8). На цій сторінці користувач може вибрати якийсь клуб для перегляду інформації про клуби.

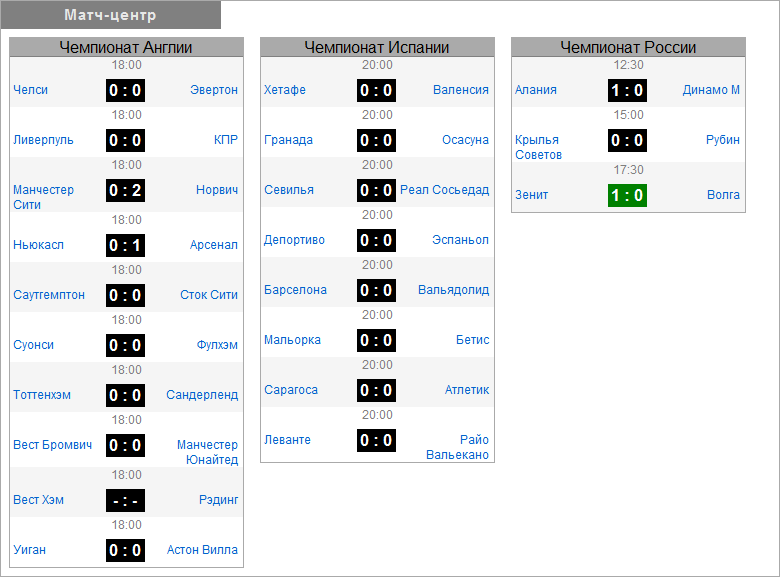


Рисунок 3.7 – Сторінка перегляду особистих даних



Рисунок 3.8 – Сторінка перегляду клубів

Також на сторінці перегляду інформації про клубів можна дізнатись про те які гравці належать до даного клубу та їхню інформацію.

У користувача є можливість переглянути таблицю результатів чемпіонату (див. рис. 3.9) та календарний графік ігор (див. рис. 3.10).



Рисунок 3.9 – Сторінка перегляду таблиці результатів чемпіонату



Рисунок 3.10 – Сторінка перегляду календарного графіку чемпіонату

У адміністратора сайту є можливість коментувати матч в онлайн режимі. Блок для відображення коментувань поновлюється кожну хвилину, при цьому користувач навіть не помітить, що якісь дії відбувалися на сервері, бо відправка запиту на сервер виконується у фоновому режимі завдяки AJAX. Сторінка для створення та відображення коментарів:

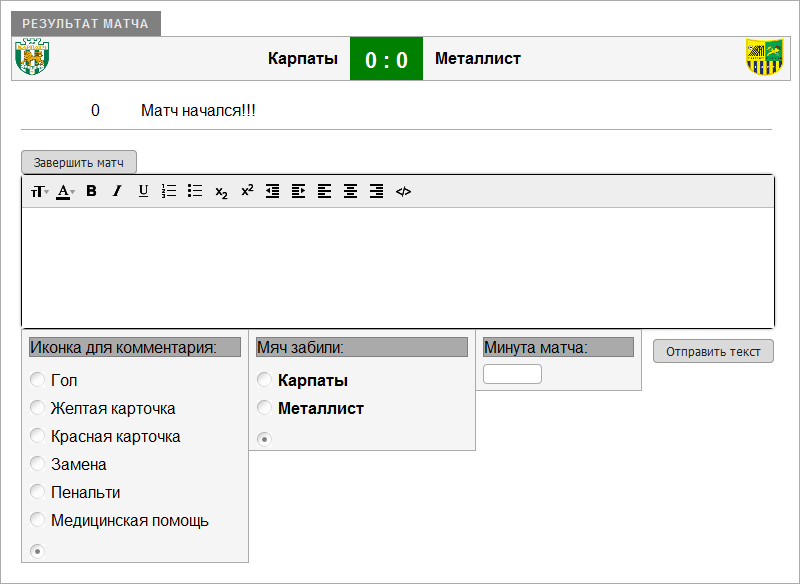


Рисунок 3.11 – Сторінка для перегляду та створення коментарів матчу

Якщо користувач є анонімним, або має роль “Fan” то в нього не буде можливості коментувати матч, бо редактор створення коментарів для них не створюється.

Також зареєстрований користувач має можливість створювати коментарі до матчу та новин. Редактор написання коментарі є таким же самим як і редактор для створення коментарів онлайн-текстової трансляції. Тобто користувач має можливість робити текст жирним, підкресленим, курсивом, задавати колір, збільшувати, зменшувати та форматувати по центру, лівому та правому краям. Після написання коментаря йому потрібно натиснути кнопку «Отправить текст» і коментар збережеться в базі даних. Редактор для створення коментарів показаний на рисунку 3.12. Також слід відзначити, що для коментарів розроблений спеціальний “paging” для того щоб в разі великої кількості записів, сторінка не відображала всі коментарі, а по частинам.

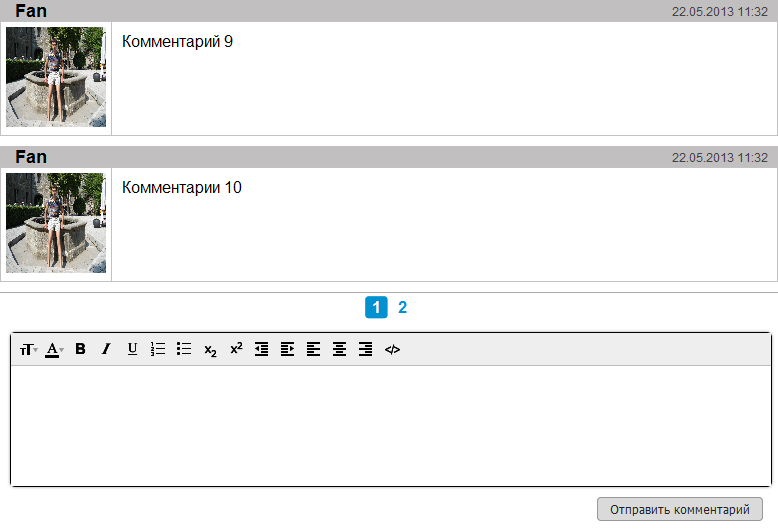


Рисунок 3.12 – Відображення коментарів та редактора

Одним із поставлених завдань, було написання програми для заповнення бази даних інформацією отриманою з football.ua. Дана програма показана на рисунку 3.13. Вона складається із трьох кнопок та двох текстових полів. Коли користувачеві потрібно отримати інформацію з football.ua, йому треба, у поле під назвою «Введите URL-адрес сайта», ввести адресу будь-якого чемпіонату (на малюнку показана адреса чемпіонату України). Після цього треба натиснути кнопку «Распарсить» і програма почне аналізувати html-сторінку і будувати xml з потрібними даними. При цьому програма відображає процес роботи у новому вікні (див. рис. 3.14). Якщо програма десь виконається не правильно або не зможе виконати якусь дію, це буде написано в даному вікні.

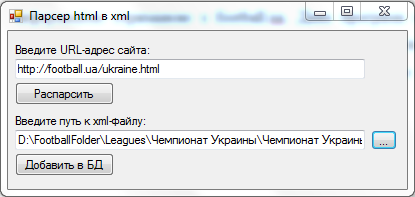


Рисунок 3.13 – Головне вікно програми

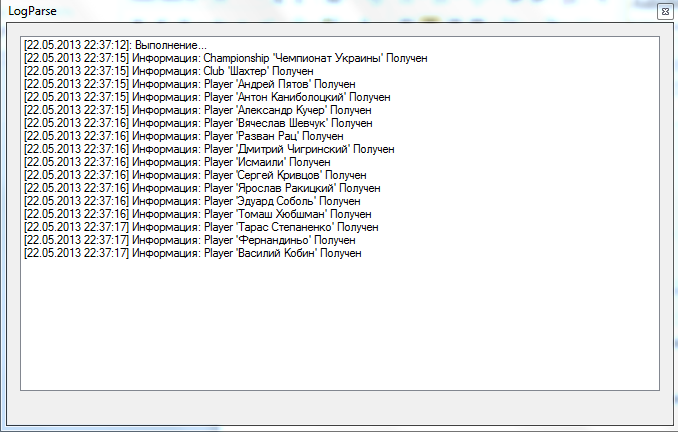


Рисунок 3.14 – Вікно для відображення ходу роботи програми

Після виконання дій отримання інформації з сайту, буде створено чотири файли:

* [назва чемпіонату].xml – в даному файлі знаходиться інформація про чемпіонат, клуби та гравців;
* [назва чемпіонату].Calendar.xml – в цьому файлі знаходиться інформація про календар ігор чемпіонат;
* [назва чемпіонату].News.xml – даний файл містить в собі інформацію про новини чемпіонату;
* [назва чемпіонату].Table.xml – файл для зберігання інформації про таблицю результатів чемпіонату.

Після створення всіх цих фалів, їх треба по черзі указувати в поле «Введите путь к xml-файлу». Важливо щоб перший файл був «[назва чемпіонату].xml» в такому випадку програма відпрацює нормально. Якщо користувач спробує спочатку указати якийсь інший файл, то програма не збереже інформацію в базу даних, бо не зможе знайти інформацію, яка потрібна для створення в БД нових значень.

Коли користувач вписав фізичний путь до xml-файлу і натиснув на кнопку «Добавить в БД», відкриється знову вікно відображення ходу роботи програми. Якщо все виконано правильно, то в базу даних добавиться новий чемпіонат, новини, календар ігор, таблиця.

3.3 Моделювання програмного забезпечення

Діаграма прецедентів (use case) [13] - це діаграма, на якій зображено відношення між акторами та прецедентами. В даному проекті акторами є користувачі системи, тобто гість, зареєстрований користувач та адміністратор. Прецедентами є ті дії, які користувачі можуть виконувати в системі.

Діаграма прецедентів наведена на рисунку 3.15

В даному випадку анонімний користувач має можливість:

* пройти реєстрацію;
* увійти до системи;
* переглядати результати матчів;
* переглядати інформацію про клуби та гравців;
* читати новини та текстову трансляцію;
* читати коментарі;
* переглядати таблицю результатів та календар ігор.

Зареєстрований користувач (з роллю “Fan”) наслідує всі дії анонімного користувача, та має додаткові можливості:

* створювати коментарі до матчів;
* створювати коментарі до новин;
* переглядати та редагувати особисті данні;
* вийти із системи.

Користувач з роллю “Admin” може робити всі дії зареєстрованого та анонімного користувача та має ще наступні можливості:

* створювати користувачів;
* редагувати користувачів;
* видаляти користувачів;
* назначати ролі користувачам;
* створювати, редагувати та видаляти ролі;
* створювати коментарі для онлайн трансляції.



Рисунок 3.15 – Діаграма прецедентів (use case)

Діаграма діяльності — в UML, візуальне представлення графу діяльностей. Граф діяльностей є різновидом графу станів скінченного автомату, вершинами якого є певні дії, а переходи відбуваються по завершенню дій [13]. Діаграма діяльності показана на рисунку 3.16.

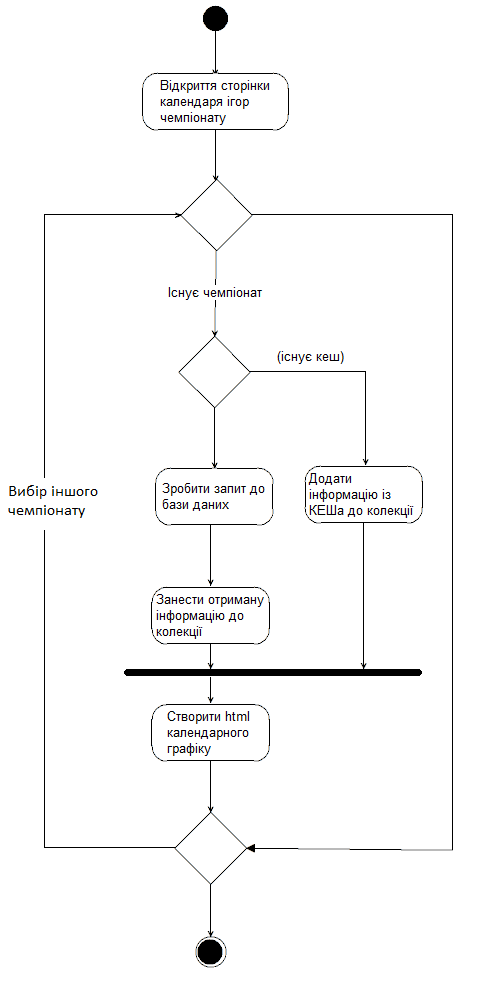


Рисунок 3.16 – Діаграма діяльності

Дана діаграма показує етапи побудови інформації про календарний графік ігор. Коли користувач відкриває сторінку календаря якогось чемпіонату система провіряє існує такий чемпіонат чи ні. Якщо не існує то більше ніяких дій виконуватись не буде. Система буде чекати доки користувач не обере чемпіонат зі списку чемпіонатів або не відкриє календар для іншої ліги. Якщо чемпіонат існує, система робить перевірку, чи є календар в КЕШі з таким ідентифікаційним номером ліги. Якщо є то система бере календар із КЕШа, в іншому випадку здійснюється запит до бази даних. Після отримання інформації із КЕШу чи бази даних, вона записується до колекції і вже після цього будується динамічно html сторінки.