

**А.В. Морозов, к.т.н., доц.
Т.А. Вакалюк, к.пед.н., доц.
Д.С. Антонюк, к.пед.н.
В.В. Дідківський, магістрант**

Державний університет «Житомирська політехніка»

Програмний комплекс створення та проведення числових симуляцій

Актуальність. У сучасному світі набули широкого поширення бізнес-симулятори – інтерактивні системи, які дозволяють організовувати та проводити симуляції економічних та соціально-економічних процесів з достатньою реалістичністю. Водночас опитування, проведене серед студентів, показало недостатні знання та технічні навички студентів вітчизняних закладів вищої освіти у галузі розробки бізнес-симуляцій з використанням сучасних технологій та підходів розробки програмного забезпечення.

Мета статті полягає в описі розробки системи, яка дозволяє викладачам та студентам організовувати й проводити симуляції економічних і соціально-економічних процесів у навчальному процесі.

Методи. В роботі було використано методи об'єктно-орієнтованого проектування, проектування програмних інтерфейсів, веб-програмування та дизайну.

Результати. У цій роботі було описано розробку програмного комплексу створення та проведення числових симуляцій. Для цього було визначено такі вимоги до програмного комплексу: система має бути у вільному доступі в мережі Інтернет; система має бути доступна з будь-якого пристрою, який має доступ до мережі Інтернет, незалежно від типу операційної системи пристрою; система повинна мати сучасний дизайн, який адаптується до розширення екрана учасника; система повинна мати механізм розширення сторонніми симуляціями, які підтримують визначений API; система повинна мати простий інтерфейс, який мінімізує зусилля викладача та студентів при організації та проведенні симуляцій.

У процесі дослідження було обрано хмарну архітектуру програмного комплексу, що дозволяє організувати управління симуляцій учасниками за допомогою централізованого ресурсу. Розроблений програмний комплекс створення та проведення числових симуляцій складається з таких компонентів: *SimulationLab.Admin*, *SimulationLab.Api.Models*, *SimulationLab.Engine*, *SimulationLab.Core*, *SimulationLab.Web*.

У процесі дослідження було встановлено, що використання засобів навчання, що збагачують пізнавальний досвід студентів та забезпечують вироблення умінь і навичок, значною мірою залежить від зручності впровадження таких засобів у освітній процес викладачем, тому велика увага в роботі приділялася саме зручному впровадженню використання розробленої системи під час лекцій чи практичних занять. Система побудована з використанням сучасних технологій та архітектурних рішень.

Висновки. Використання розробленої системи значно спрощує доповнення теоретичного матеріалу практичними та дослідницькими видами навчальної діяльності в межах курсів математичного й соціально-економічних блоків у закладах вищої та загальної середньої освіти, корпоративних тренінгах та в межах заходів неформальної освіти.

Ключові слова: симуляція; числові симуляції; бізнес-симулятори; програмний комплекс; проектування.

Актуальність теми. В сучасному світі набули широкого поширення бізнес-симулятори – інтерактивні системи, які дозволяють організовувати та проводити симуляції економічних і соціально-економічних процесів з достатньою реалістичністю [1]. Такі системи створені для візуалізації процесів та явищ, покращення засвоєння знань, вироблення навичок і сприянню формування та розвитку компетентностей і за умови ефективної інтеграції з навчальними задачами підвищують ефективність освітніх результатів [2].

Застосування бізнес-симуляторів у вивченні курсів фінансово-економічного спрямування стає необхідністю, забезпечуючи елементи активного навчання та практичну складову навчання. Лекції з використанням симуляторів можуть підвищити зацікавленість студента в дисципліні, демонструючи користь набутих теоретичних знань на практиці.

Водночас опитування, проведене серед студентів, показало недостатні знання та технічні навички студентів вітчизняних закладів вищої освіти у галузі розробки бізнес-симуляцій з використанням сучасних технологій та підходів розробки програмного забезпечення.

Аналіз останніх досліджень та публікацій, на які спирається автор. Зокрема, застосування програмно-імітаційних комплексів у процесі навчання розглядали В.Адамс (W.K. Adams) [11], Б.Девейн (B.Devine) [13], Е.Кастронова (E.Castronova) [12], К.Перкінс (K.Perkins) [14], С.Фортман-Рое (S.Fortmann-Roe) [15], в Україні цю тему досліджували О.О. Мацюк [17], В.А. Пермінова [16], О.Б. Шендерук [16].

Метою статті є опис розробки системи, яка дозволяє викладачам та студентам організувати та проводити симуляції економічних і соціально-економічних процесів у навчальному процесі.

Викладення основного матеріалу. Для досягнення поставленої мети було визначено такі вимоги до програмного комплексу:

- система має бути у вільному доступі в мережі Інтернет;
- система має бути доступна з будь-якого пристрою, який має доступ до мережі Інтернет, незалежно від типу операційної системи пристрою;
- система повинна мати сучасний дизайн, який адаптується до розширення екрана учасника;
- система повинна мати механізм розширення сторонніми симуляціями, які підтримують визначений API;
- система повинна мати простий інтерфейс, який мінімізує зусилля викладача й студентів при організації та проведенні симуляції.

Для виконання перерахованих вище вимог було сформовані такі задачі:

1. Розробка системи для додавання нових сторонніх симуляцій до програмного комплексу. Механізм розширення системи сторонніми симуляціями такий: розробник реалізовує визначені інтерфейси відповідно до своєї симуляції та розгортає свій додаток у мережі Інтернет. Для реєстрації нової симуляції в системі створення симуляцій необхідно додати маніфест, в якому обов'язково має бути вказано назву симуляції та шлях (URL) до місця, де система зможе знайти всі інші налаштування симуляції.

2. Розробка каталогу доступних симуляцій. Система має відображати усі симуляції, інформацію про яких вдалося успішно отримати, звернувшись по вказаному у маніфесті URL, інакше такі симуляції мають бути проігноровані. Кожна симуляція повинна бути представлена у вигляді заголовка та опису з правилами симуляції.

3. Розробка системи створення сеансу симуляції. Якщо симуляція підтримує декількох учасників, користувач повинен мати можливість створити багатокористувацький сеанс симуляції. Після створення сеансу симуляції користувач має побачити 2 коди доступу до сеансу: перший код для гравців, який вони зможуть використати, щоб приєднатися до сеансу симуляції та почати гру; другий для інструктора, який він чи вона зможе використати для того, щоб отримати доступ до сторінки презентації. Код для інструктора має бути прихованим у вигляді «зірочок», але повинна бути можливість його переглянути.

4. Розробка функціоналу сторінки презентації. Після створення сеансу симуляції інструктор за допомогою отриманого коду має отримати доступ до спеціальної сторінки презентації. На цій сторінці у верхній частині екрана завжди має бути відображена інформація про те, як приєднатися до сеансу симуляції, а саме: URL до форми входу в сеанс та код доступу. Сторінка інструктора повинна містити 4 вкладки:

a) «QR» – вкладка, на якій зображено код швидкого доступу, який містить зашифровану інформацію для миттєвого приєднання учасників до сеансу симуляції;

b) «Description» – вкладка, на якій зображено опис симуляції;

c) «Monitor» – вкладка, в якій зображено список вільних та зайнятих місць учасників сеансу симуляції разом з інформацією про останню активність учасників. Також, якщо симуляція це підтримує, тут має бути можливість примусово перевести учасника симуляції на наступний крок або замінити учасника ботом. Інформація про активність учасників має оновлюватися в режимі реального часу;

d) «Statistics» – вкладка, на якій представлена статистика сеансу. Може бути відсутньою, якщо симуляція не підтримує генерування статистики. Статистика має оновлюватися в режимі реального часу.

5. Розробка системи проходження симуляції з підтримкою декількох учасників.

6. Розробка системи проходження симуляції з підтримкою лише 1 учасника.

7. Розробка симуляції «The Market for Lemons Game» [4].

8. Розробка симуляції «The Red/Green Simulation» [5].

У процесі дослідження було обрано хмарну архітектуру програмного комплексу, що дозволяє організувати управління симуляцій учасниками за допомогою централізованого ресурсу (Інтернет-сервісу). Також хмарна архітектура системи дозволяє застосовувати концепцію BYOD [7] (принеси свій власний пристрій) при проведенні симуляцій. Концепція BYOD дозволяє організувати проходження симуляцій під час лекцій чи практичних занять.

Для забезпечення модульності програмного комплексу був застосований об'єктно-орієнтований метод програмування (ООП), який заснований на поданні програми у вигляді сукупності взаємодіючих об'єктів [6]. Таким чином, програма являє собою набір об'єктів, що мають стан та поведінку. Стійкість та керованість системи забезпечуються за рахунок чіткого розподілення відповідальності об'єктів (за кожно

дію відповідає певний об'єкт), однозначного означення інтерфейсів взаємодії об'єктів та повної ізольованості внутрішньої структури об'єкта від зовнішнього середовища (інкапсуляції) [10].

Для того щоб система мала механізм розширення сторонніми симуляціями, які підтримують визначений API, було обрано архітектурний стиль мікросервісів, за яким єдиний застосунок будується як сукупність сервісів, кожен з яких працює окремо, комунікація між сервісами відбувається за допомогою HTTP. Сервіси будуються та розгортаються незалежно один від одного і можуть бути написані з використанням різних мов програмування [8].

Особливості архітектури мікросервісів, які використані при розробці програмного комплексу створення та проведення числових симуляцій: незалежна розробка; незалежне розгортання; незалежне масштабування; невелика кодова база зменшує кількість конфліктів; простота заміни однієї реалізації сервісу іншою; простота додавання нового функціоналу в систему; вихід з ладу одного сервісу не призводить до виходу з ладу всієї системи; кожен сервіс незалежно від інших може бути реалізований за допомогою будь-якої мови програмування [8].

Зважаючи на особливості архітектури мікросервісів було розроблено таку структуру програмного комплексу створення та проведення числових симуляцій (рис. 1):

1. «SimulationLab ASP.NET Core Web App»:

- здійснює централізоване управління іншими сервісами, з якими спілкується через визначений API;
- відповідає за збереження даних, в тому числі збереження даних про сеанси симуляцій;
- відповідає за надання інформації клієнту;
- відповідає за неможливість виникнення колізій між запитами декількох учасників однієї симуляції.

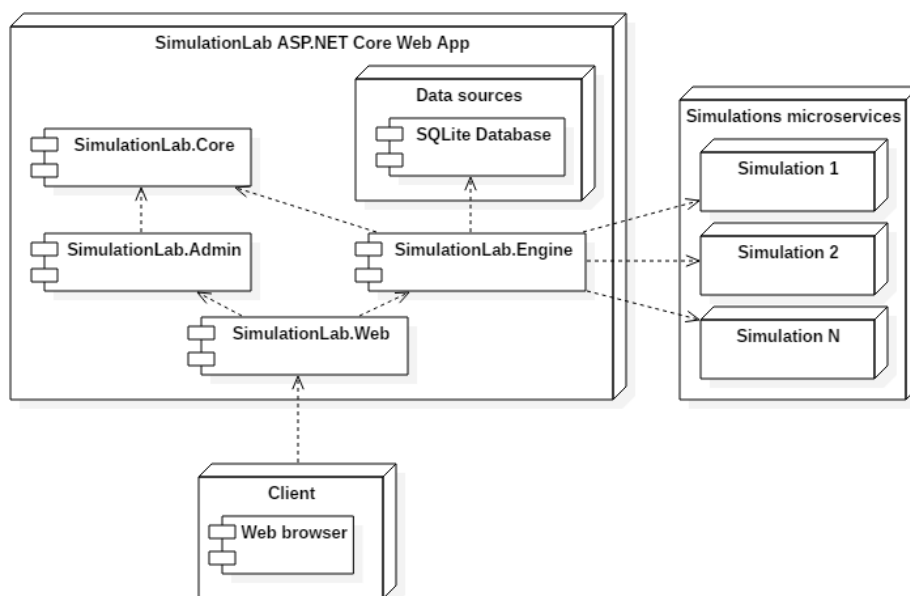


Рис. 1. Діаграма розгортання розроблюваної системи

2. Клієнт:

- здійснює запити до основного сервера програмного комплексу;
- відповідає за представлення даних клієнту.

3. Мікросервіси симуляцій:

○ є незалежними програмами, тобто кожна окрема симуляція немає жодної інформації про іншу симуляцію;

○ декілька симуляцій можуть бути розгорнуті в межах одного домену, якщо вони написані на одній мові програмування;

- реалізують визначений API.

Обмін даними між сервісами відбувається за допомогою HTTP. Спілкування між основною системою та мікросервісами симуляцій здійснюється через визначений API, який має реалізувати кожна окрема симуляція. В розробленому програмному комплексі симуляції мають лише оброблювати дані, які приходять у запиті у форматі JSON, та повертати на основі цих даних відповідь у такому ж форматі.

Основним архітектурним шаблоном сервісу створення та проведення числових симуляцій було обрано MVC – шаблон, який передбачає поділ системи на три взаємопов'язані частини: модель даних, вигляд (інтерфейс користувача) та модуль керування [9]. Використання шаблону MVC забезпечує

розділення відповідальності між компонентами в системі, що полегшує внесення змін і робить програму більш зрозумілою через зменшення її складності.

Для опису найважливіших алгоритмів системи було розроблено діаграми активності.

На рисунку 2 зображено діаграму активності створення сеансу симуляції, яка складається з таких послідовних дій:

1. Користувач обирає симуляцію, сеанс якої він хоче створити.
2. Якщо створення сеансу симуляції потребує додаткових налаштувань, користувачу буде показано вікно з формою для заповнення додаткових даних.
3. Якщо створення сеансу симуляції не потребує додаткових налаштувань або потребує і додаткові дані успішно заповнені – сеанс симуляції буде створено.
4. Якщо симуляція підтримує сторінку презентації, в наступному вікні буде зображено 2 коди і кнопки для входу в сеанс симуляції як інструктор.
5. Інакше у вікні після створення сеансу симуляції буде зображено лише 1 код і кнопка для входу в сеанс як учасник симуляції.

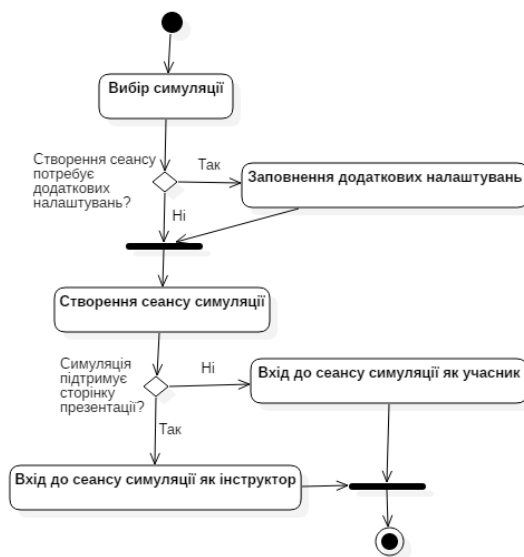


Рис. 2. Діаграма активності створення сеансу симуляції

На рисунку 3 зображена діаграма активності проходження сеансу симуляції, яка складається з таких послідовних дій:



Рис. 3. Діаграма активності проходження сеансу симуляції

1. Учасник підключається до сеансу симуляції за допомогою логіну.
2. Система отримує сторінку симуляції, яка відображається учаснику.
3. Якщо сторінка містить форму – учасник має заповнити її.

4. Якщо форма була заповнена неправильно – система покаже учаснику помилки.
5. Якщо форма була заповнена правильно – учасник переходить на наступний крок.
6. Процес отримання сторінок і заповнення форм продовжується поки учасник не дійде до останньої сторінки, яка не містить форму для заповнення.

Для опису взаємозв'язків, внутрішню структуру і типи відносин між окремими сутностями програмного комплексу було розроблено діаграми класів.

Програмний комплекс створення та проведення числових симуляцій складається з таких компонентів:

1. SimulationLab.Admin – містить специфічні класи системи адміністратора сайту (рис. 4, табл. 1).

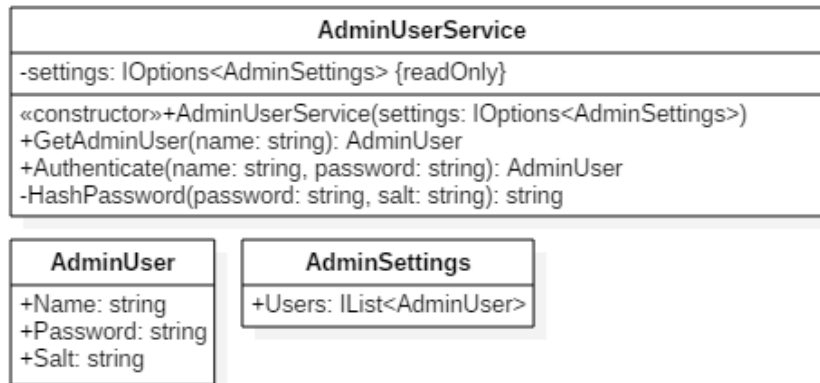


Рис. 4. Діаграма класів «SimulationLab.Admin»

Таблиця 1

Опис класів «SimulationLab.Admin»

Назва класу	Опис
AdminUserService	Клас для роботи з користувачами системи адміністратора сайту
AdminUser	Модель користувача системи адміністратора сайту
AdminSettings	Модель налаштувань системи адміністратора сайту

2. SimulationLab.Api.Models – містить класи моделей передачі даних між підсистемами програмного комплексу (рис. 5, табл. 2).

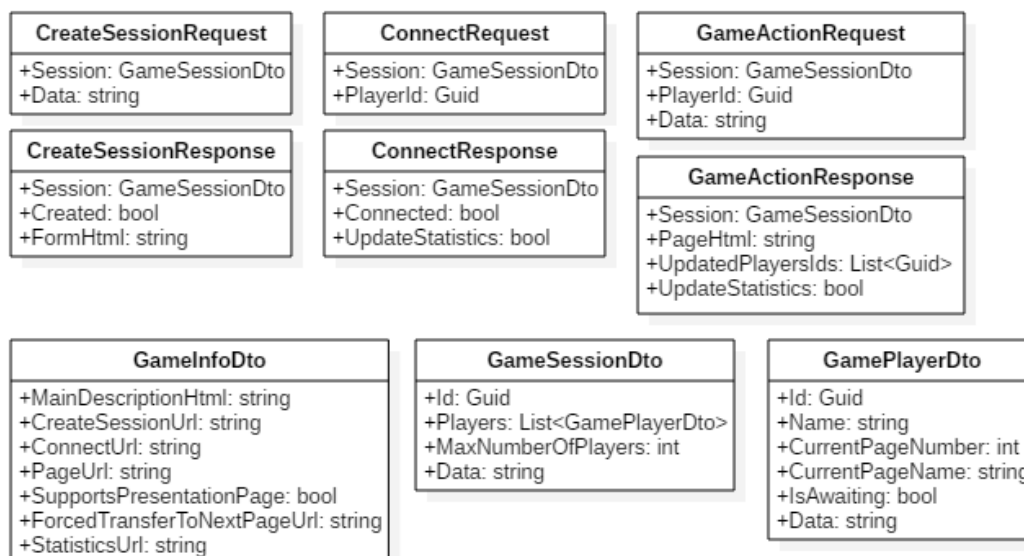


Рис. 5. Діаграма класів «SimulationLab.Api.Models»

Опис класів «SimulationLab.Api.Models»

Назва класу	Опис
CreateSessionRequest	Модель передачі даних створення сесії
CreateSessionResponse	Модель передачі даних результату створення сесії
ConnectRequest	Модель передачі даних підключення до сеансу симуляції
ConnectResponse	Модель передачі даних результату підключення до сесії
GameActionRequest	Модель передачі даних виконання запиту до симуляції
GameActionResponse	Модель передачі даних результату запиту до симуляції
GameInfoDto	Модель передачі даних налаштувань симуляції
GameSessionDto	Модель передачі даних сесії симуляції
GamePlayerDto	Модель передачі даних учасника сесії симуляції

3. SimulationLab.Engine – містить класи, які реалізують основний функціонал програмного комплексу (рис. 6–9, табл. 3–6).

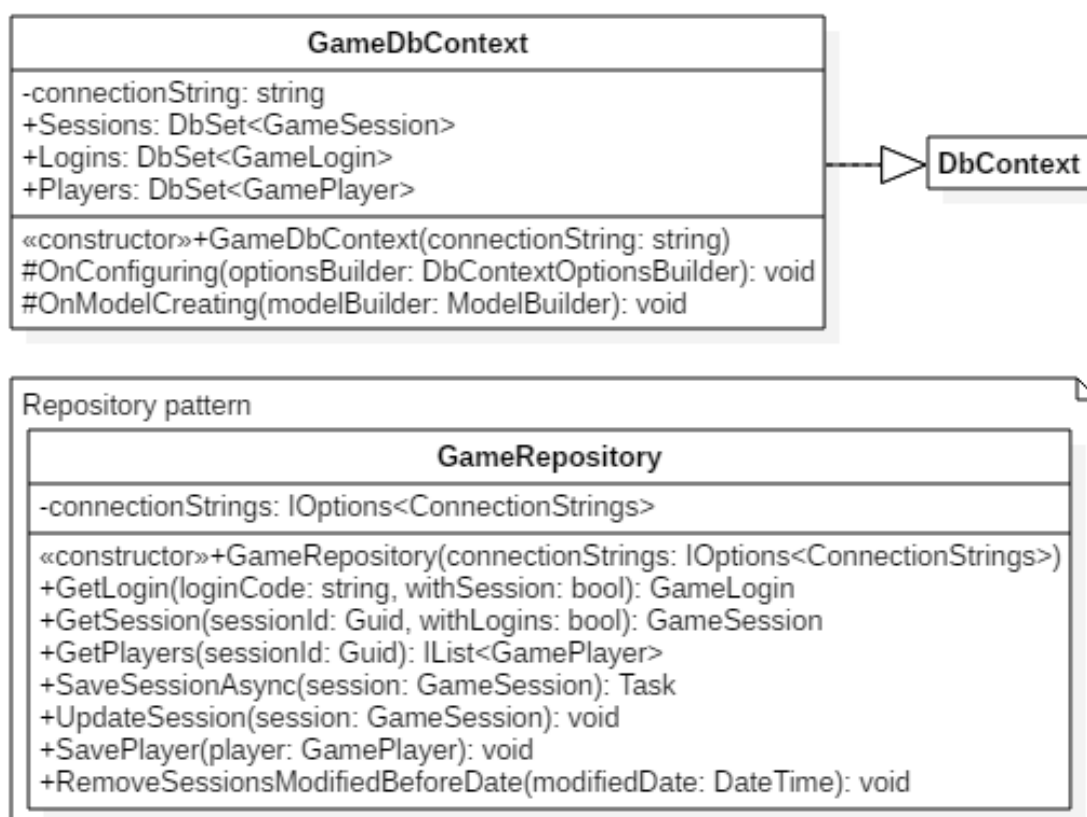


Рис. 6. Діаграма класів «SimulationLab.Engine.Data»

Таблиця 3

Опис класів «SimulationLab.Engine.Data»

Назва класу	Опис
GameDbContext	Клас, що унаслідований від класу «DbContext». Представляє сеанс з базою даних і використовується для запиту та збереження об'єктів
GameRepository	Реалізація патерну «Repository», яка дозволяє абстрагуватися від того як зберігаються дані про проходження симуляції, є проміжною ланкою між класами, які безпосередньо взаємодіють з даними і рештою програми

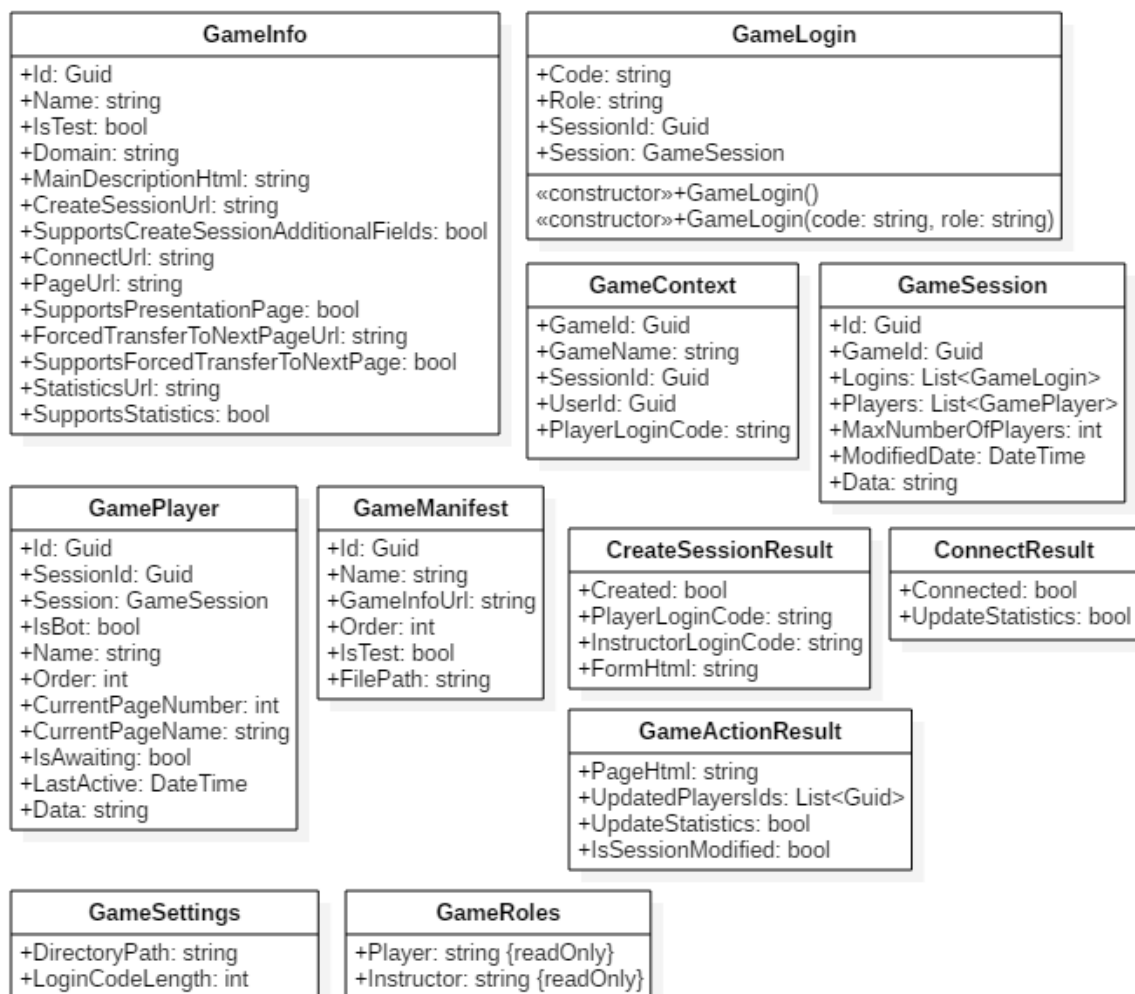


Рис. 7. Діаграма класів «SimulationLab.Engine.Models»

Таблиця 4

Опис класів «SimulationLab.Engine.Models»

Назва класу	Опис
ConnectionStrings	Модель даних строк підключень
GameInfo	Модель даних інформації про симуляцію
GameLogin	Модель даних логіна до сеансу симуляції
GameSession	Модель даних сеансу симуляції
GamePlayer	Модель даних учасника сеансу симуляції
GameManifest	Модель даних реєстрації симуляції в системі
GameSettings	Модель даних загальних налаштувань симуляцій
GameRoles	Модель з константами всіх можливих ролей, які можуть мати користувачі в системі
GameContext	Модель з даними поточного сеансу симуляції
CreateSessionResult	Модель даних результату створення сеансу симуляції
ConnectResult	Модель даних результату підключення до сеансу симуляції
GameActionResult	Модель даних результату виконання запиту до симуляції

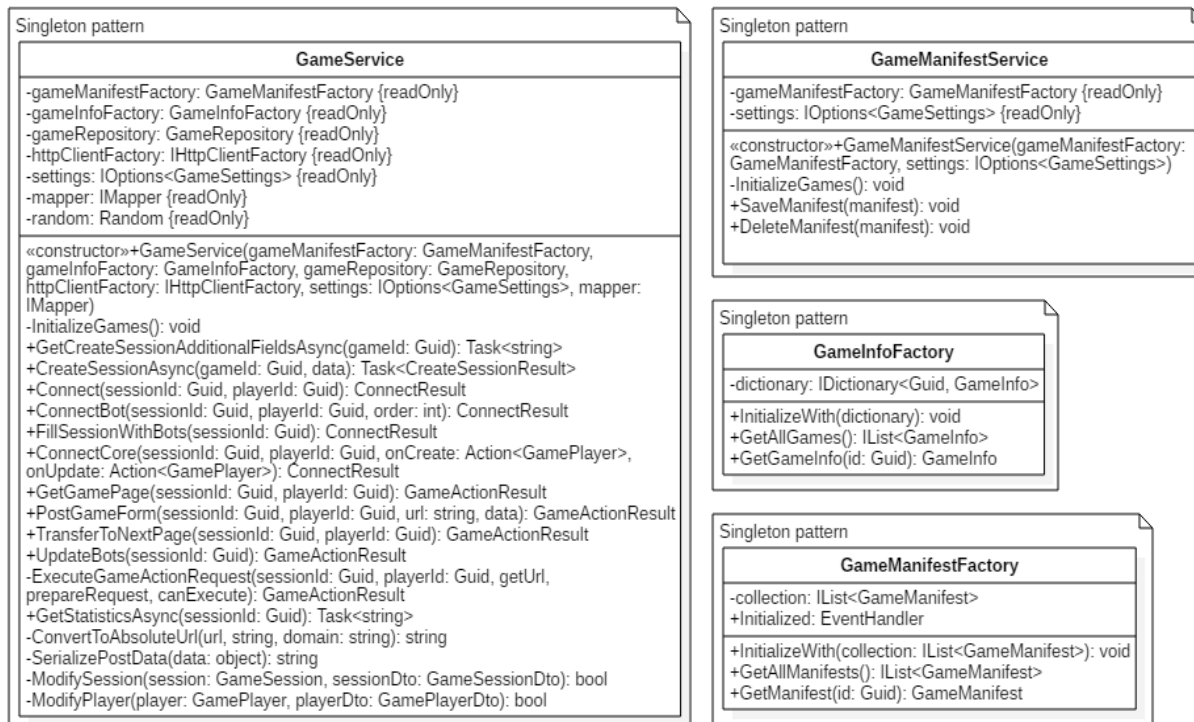


Рис. 8. Діаграма класів «SimulationLab.Engine.Services»

Таблиця 5

Опис класів «SimulationLab.Engine.Services»

Назва класу	Опис
GameService	Клас для створення, виконання та оброблення результатів запитів до мікросервісів симуляцій
GameManifestService	Клас для виконання ініціалізації та виконання операції створення, редагування та видалення маніфестів симуляцій
GameInfoFactory	Реалізація патерну «Flyweight» для збереження об’єктів з інформацією про симуляції
GameManifestFactory	Реалізація патерну «Flyweight» для збереження об’єктів з інформацією про маніфести симуляцій

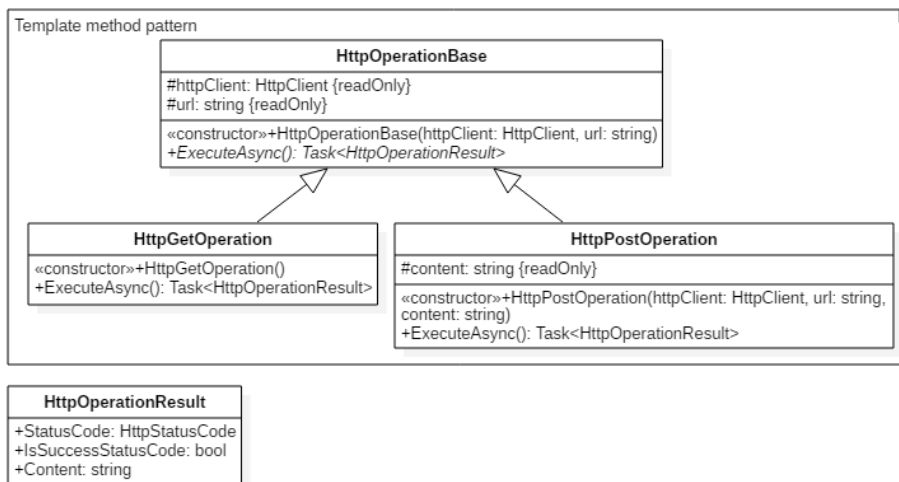


Рис. 9. Діаграма класів «SimulationLab.Engine.Services.Operations»

Таблиця 6

Опис класів «SimulationLab.Engine.Services.Operations»

Назва класу	Опис
HttpOperationResult	Модель даних з результатом запиту до мікросервісу
HttpOperationBase	Реалізація патерну «Template method». Клас є базовим для класів, які призначені для виконання запитів різних типів до мікросервісів
HttpGetOperation	Клас унаслідований від класу «HttpOperationBase». Необхідний для виконання запитів типу «GET» до мікросервісів
HttpPostOperation	Клас унаслідований від класу «HttpOperationBase». Необхідний для виконання запитів типу «POST» до мікросервісів

4. SimulationLab.Core – містить загальні допоміжні класи, які можуть знадобитися в будь-якому іншому компоненті програмного комплексу (рис. 10, табл. 7).

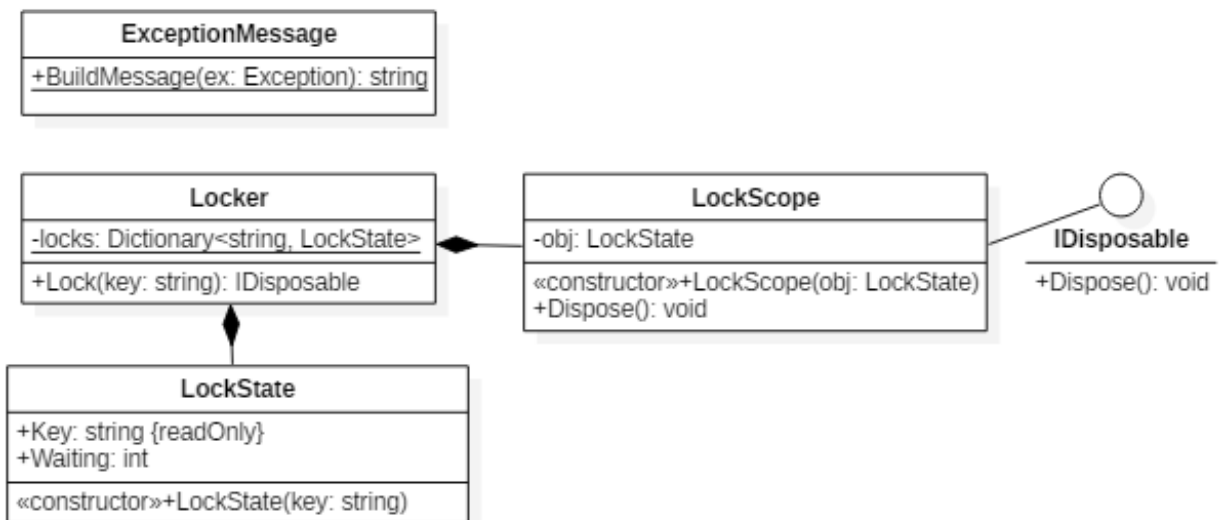


Рис. 10. Діаграма класів «SimulationLab.Core»

Опис класів «SimulationLab.Core»

Таблиця 7

Назва класу	Опис
ExceptionMessage	Допоміжний клас для побудови повідомлення про помилку
Locker	Допоміжний клас для синхронізованого доступу до ділянок коду за ключем
LockState	Внутрішній клас класу «Locker» для збереження ключа блокування та кількості потоків, які чекають доступу для виконання заблокованого коду
LockScope	Внутрішній клас класу «Locker» для створення окремої ділянки коду, доступ до якої має бути синхронізований. Реалізовує інтерфейс «IDisposable»

5. SimulationLab.Web – містить класи контролерів, моделей-представлень, представлення, файли скриптів та файли зі стилями (рис. 11–14, табл. 8–11).

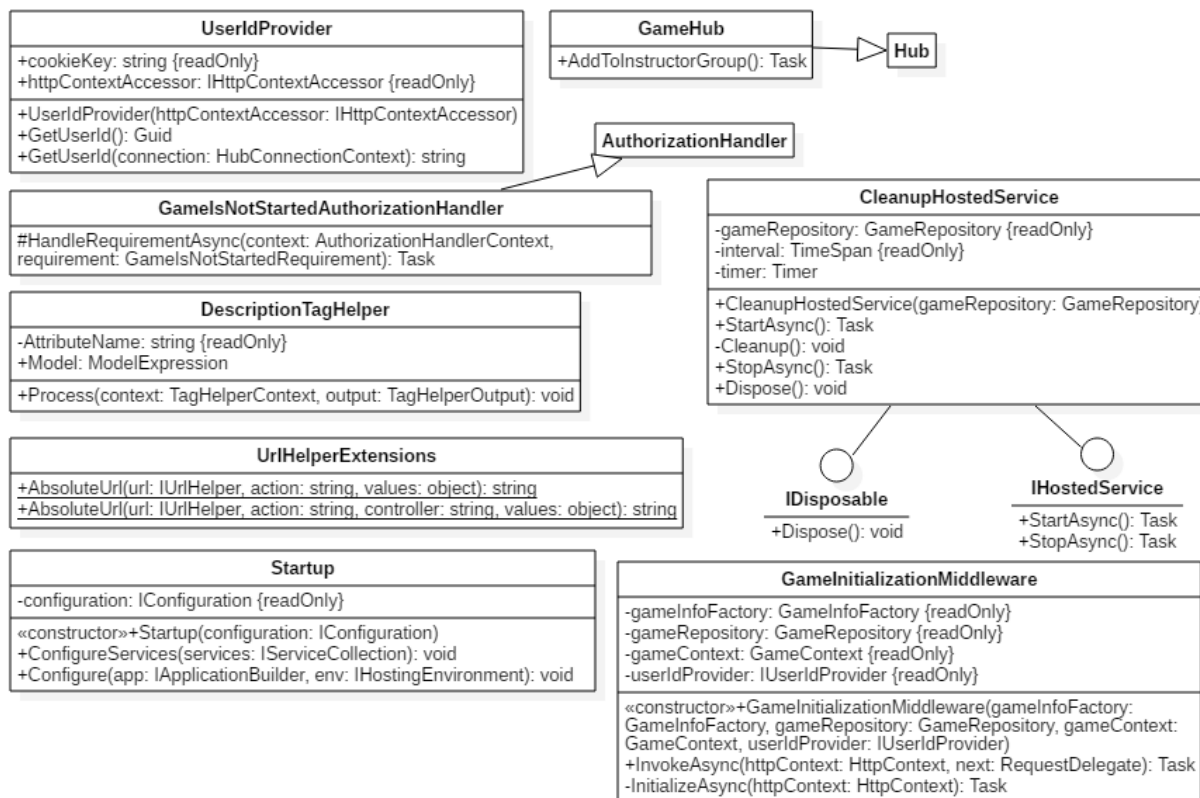


Рис. 11. Діаграма класів «SimulationLab.Web»

Таблиця 8

Опис класів «SimulationLab.Web»

Назва класу	Опис
UserIdProvider	Клас для генерації чи визначення унікального ідентифікатора учасника симуляції
GameHub	Клас унаслідований від «Hub». Необхідний для створення з'єднання між браузером та сервером для обміну повідомленнями в режимі реального часу
CleanupHostedService	Клас реалізує інтерфейси «IHostedService» та «IDisposable». Необхідний для видалення застарілих даних сеансів симуляцій
GameInitializationMiddleware	Клас реалізує інтерфейс «IMiddleware». Необхідний для ініціалізації контексту поточного сеансу симуляції
DescriptionTagHelper	Допоміжний клас унаслідований від «TagHelper» для створення тексту з описом поля
UrlHelperExtensions	Клас з допоміжними методами для побудови абсолютних URL адрес
GameIsNotStartedAuthorizationHandler	Клас унаслідований від класу «AuthorizationHandler» для визначення чи поточний користувач розпочав проходження симуляції
Startup	Клас для конфігурації додатку

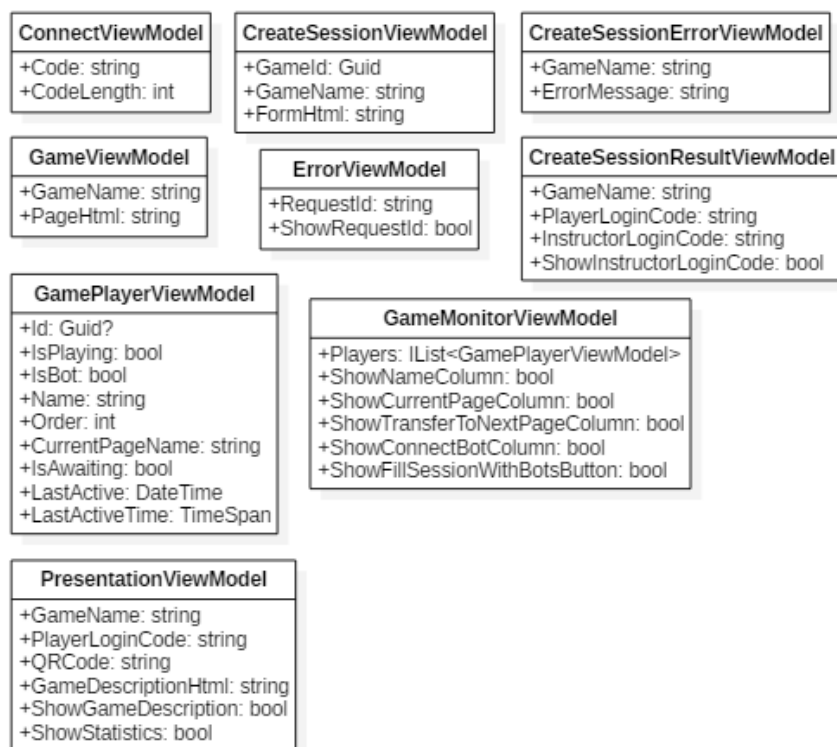


Рис. 12. Діаграма класів «SimulationLab.Web.Models»

Таблиця 9

Опис класів «SimulationLab.Web.Models»

Назва класу	Опис
ConnectViewModel	Модель представлення сторінки з формою підключення до сеансу симуляції
CreateSessionViewModel	Модель представлення форми створення сеансу симуляції
CreateSessionErrorViewModel	Модель представлення повідомлення про те, що сеанс симуляції створити не вдалося
CreateSessionResultViewModel	Модель представлення повідомлення про те, що сеанс симуляції успішно створено
ErrorViewModel	Модель представлення сторінки з інформацією про помилку
GameViewModel	Модель представлення сторінки симуляції
PresentationViewModel	Модель представлення сторінки інструктора
GameMonitorViewModel	Модель представлення вкладки «Monitor» на сторінці інструктора
GamePlayerViewModel	Модель представлення інформації про учасника симуляції

Таблиця 10

Опис класів «SimulationLab.Web.Controllers»

Назва класу	Опис
GamesController	Клас унаслідований від класу «Controller». Є основним класом для генерації моделей та представлень, що пов'язані зі створенням та проведенням числових симуляцій
ErrorsController	Клас унаслідований від класу «Controller». Повертає представлення з інформацією про помилки

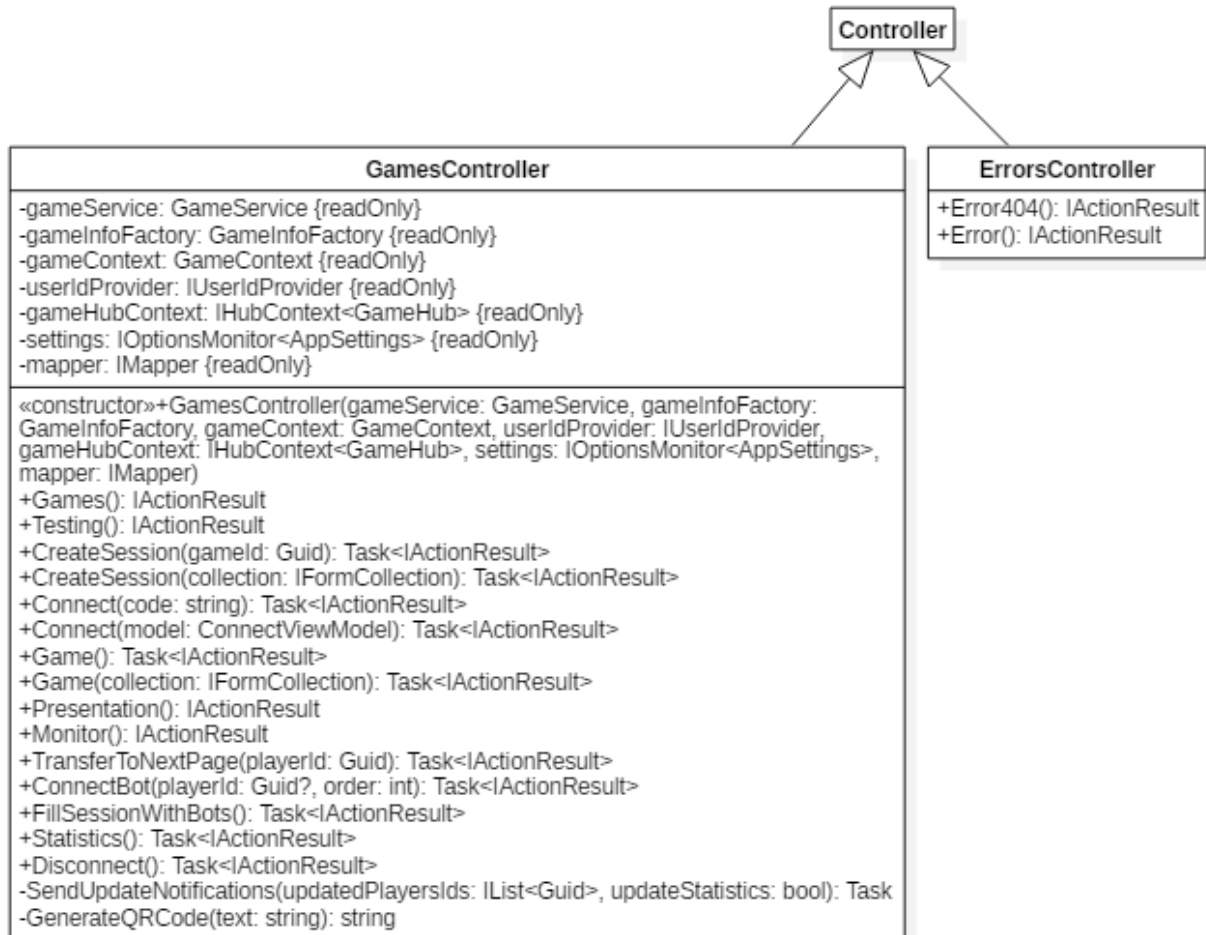


Рис. 13. Діаграма класів «SimulationLab.Web.Controllers»

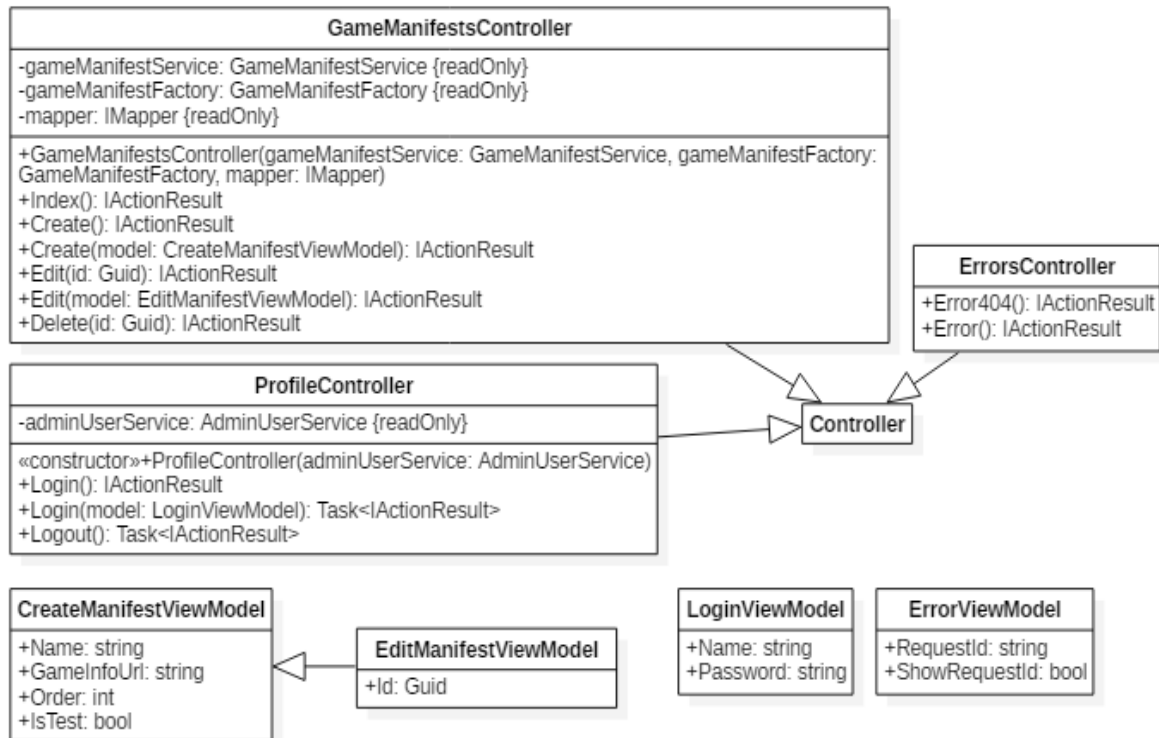


Рис. 14. Діаграма класів «SimulationLab.Web.Admin»

Опис класів «SimulationLab.Web.Admin»

Назва класу	Опис
GameManifestsController	Клас унаслідований від класу «Controller». Генерує моделі та повертає представлення пов'язані з маніфестами симуляцій
ProfileController	Клас унаслідований від класу «Controller». Генерує моделі та повертає представлення пов'язані з аутентифікацією користувача в системі адміністратора
ErrorsController	Клас унаслідований від класу «Controller». Повертає представлення з інформацією про помилки
LoginViewModel	Модель представлення сторінки аутентифікації користувача в системі адміністратора
CreateManifestViewModel	Модель представлення сторінки створення маніфесту симуляції
EditManifestViewModel	Модель представлення сторінки редагування маніфесту симуляції
ErrorViewModel	Модель представлення сторінки з інформацією про помилку

Структура інтерфейсу програмного комплексу

Програмний комплекс створення та проведення числових симуляцій є веб-додатком, який розгорнутий в мережі Інтернет. Режим доступу: <https://simulationlab.azurewebsites.net>.

Розроблений програмний комплекс підтримує такі ролі користувачів:

- «Адміністратор»;
- «Інструктор»;
- «Учасник».

Для того щоб зрозуміти, які можливості мають користувачі з перерахованими вище ролями, було зроблено опис функціональності сайту через його варіанти використання.

На рисунку 15 зображено діаграму варіантів використання сайту користувача з роллю «Адміністратор». На діаграмі видно, що користувач з цією роллю має такі варіанти використання:

1. Доступ до системи контролю над сайтом – спеціальної сторінки, яка є прихованою від користувачів з іншими ролями.
2. Додавання маніфесту симуляції, який має містити такі дані:
 - a) назву симуляції;
 - b) шлях (URL) до місця, де система зможе знайти налаштування симуляції;
 - c) число, яке буде використано при сортуванні симуляцій в каталозі (менше число означає те, що симуляція буде в списку вище);
 - d) вказати, чи дана симуляція знаходиться на стадії тестування, щоб система помістила такі симуляції в окремий каталог.
3. Редагування маніфесту симуляції. При редагуванні маніфесту можна змінювати всі перераховані вище дані, які були заповнені при створенні маніфесту симуляції.
4. Видалення маніфесту симуляції.



Рис. 15. Діаграма варіантів використання сайту користувача з роллю «Адміністратор»

На рисунку 16 зображено діаграму варіантів використання сайту користувача з роллю «Інструктор». На діаграмі видно, що користувач з цією роллю має такі варіанти використання:

1. Доступ до сторінки презентації сеансу симуляції, яка містить дані про те, як учасникам приєднатися до сеансу симуляції.
2. Створення сеансу симуляції.

3. Перегляд даних про активність учасників симуляції. Ця сторінка також дає інформацію про те, скільки всього гравців підключено до сеансу і скільки ще є вільних місць.
4. Заповнення вільних місць сеансу ботами. Для випадків, коли кількість учасників є недостатньою для проходження симуляції.
5. Переведення учасника симуляції на наступний крок. Для випадків, коли учасник почав проходження симуляції, але є довго неактивним.
6. Заміна учасника симуляції ботом. Ця дія є невідворотною і необхідна лише для випадків, коли учасник почав проходження, але більше не зможе продовжити далі.
7. Перегляд загальної статистики сеансу симуляції.



Рис. 16. Діаграма варіантів використання сайту користувача з роллю «Інструктор»

На рисунку 17 зображено діаграму варіантів використання сайту користувача з роллю «Учасник». На діаграмі видно, що користувач з цією роллю має лише можливість приєднатися до сеансу симуляції та проходити його.



Рис. 17. Діаграма варіантів використання сайту користувача з роллю «Учасник»

Висновки та перспективи подальших досліджень. У цій роботі було спроектовано та реалізовано програмний комплекс створення та проведення числових симуляцій. Використання засобів навчання, що збагачують пізнавальний досвід студентів та забезпечують вироблення умінь та навичок, значною мірою залежить від зручності впровадження таких засобів у освітній процес викладачем, тому велика увага в роботі приділялася саме зручному впровадженню використання розробленої системи під час лекцій чи практичних занять. Система побудована з використанням сучасних технологій та архітектурних рішень.

Використання розробленої системи значно спрощує доповнення теоретичного матеріалу практичними та дослідницькими видами навчальної діяльності в межах курсів математичного та соціально-економічних блоків у закладах вищої та загальної середньої освіти, корпоративних тренінгах та в межах заходів неформальної освіти.

Система забезпечує розширення бібліотеки симуляцій, що дає можливість доповнити в подальшому реалізованій програмній комплекс новими симуляціями, які можуть бути розроблені незалежно від основної системи.

Список використаної літератури:

1. Бізнес-симуляція [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/Бізнес-симуляція>.
2. Вакалюк Т.А. Використання хмарних бізнес-симуляторів для вивчення поведінкової економіки / Т.А. Вакалюк Д.С. Антонюк // Наукова молодь-2016 : збірник матеріалів IV Всеукраїнської науково-практичної конференції молодих учених (15 груд. 2016 р., м. Київ) ; за заг. ред. проф. О.М. Спірина. – К. : ІТЗН НАПН України, 2016. – С. 53–56 [Електронний ресурс]. – Режим доступу : <http://lib.iitta.gov.ua/707095>.

3. Антонюк Д.С. Використання програмно-імітаційних комплексів як засобів формування економічних компетентностей студентів технічних спеціальностей : дис. ... канд. пед. наук : 13.00.10 / Д.С. Антонюк. – Київ, 2018.
4. The Market for Lemons [Електронний ресурс]. – Режим доступу : https://en.wikipedia.org/wiki/The_Market_for_Lemons.
5. Bruehler James R. The Red/Green Simulation / James R. Bruehler, Alan P. Grant, Linda S. Ghent // Journal of Economics and Finance Education. – 2017. – Vol. 16, Issue 1 [Електронний ресурс]. – Режим доступу : <https://www.economics-finance.org/jefe/issues/JEFE-Vol-16-Num-1-Winter-2017.pdf>.
6. Об'єктно-орієнтоване програмування [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Об'єктно-орієнтоване_програмування.
7. Bring your own device [Електронний ресурс]. – Режим доступу : https://uk.wikipedia.org/wiki/Bring_your_own_device.
8. Мікросервіси [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/Мікросервіси>.
9. Модель-вид-контролер [Електронний ресурс]. – Режим доступу : <https://uk.wikipedia.org/wiki/Модель-вид-контролер>.
10. Architectural principles [Електронний ресурс]. – Режим доступу : <https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/architectural-principles>.
11. Adams W.K. Student engagement and learning with PhET interactive simulations / W.K. Adams [Електронний ресурс]. – Режим доступу : http://www.jdes.tyc.edu.tw/PhET2013/publications/MPTL_2010_PhET_final.pdf.
12. Castronova Edward Exodus to the Virtual World: How Online Fun is Changing Reality / Edward Castronova // Palgrave Macmillan [Електронний ресурс]. – Режим доступу : <http://us.macmillan.com/exodustothetvirtualworld/edwardcastronova>
13. Liang T. Classical atomistic simulations of surfaces and heterogeneous interfaces with the charge-optimized many body (COMB) potentials / T.Liang, T.R. Shan // Materials Science and Engineering. – R. : Reports, 2013. – Vol. 74, Issue 9, P. 255–279.
14. Podolefsky N.S. Computer simulations to classrooms: tools for change / N.S. Podolefsky, K.K. Perkins, W.K. Adams // Physics Education Research : Conference Proceedings. – AIP Press, 2010 [Електронний ресурс]. – Режим доступу : https://www.colorado.edu/physics/EducationIssues/papers/Podolefsky_etal/Podolefsky_invited_PERC_09_scales_revised.pdf.
15. Fortmann-Roe Scott Insight Maker: A general-purpose tool for web-based modeling & simulation / Scott Fortmann-Roe // Simulation Modelling Practice and Theory. – 2014. – № 47. – P. 28–45 [Електронний ресурс]. – Режим доступу : https://www.researchgate.net/publication/263127834_Insight_Maker_A_general-purpose_tool_for_web-based_modeling_simulation.
16. Шендерук О.Б. Технології ситуативного моделювання. Симуляції або імітаційні ігри // О.Б. Шендерук, В.А. Пермінова [Електронний ресурс]. – Режим доступу : http://www.rusnauka.com/26_SSN_2010/Psichologia/71388.doc.htm.
17. Мацюк О.О. Комп'ютерні симуляції як засіб формування професійної компетентності майбутніх перекладачів в умовах інформаційного суспільства / О.О. Мацюк // Вісник Національної академії Державної прикордонної служби України. – 2013. – Вип. 5 [Електронний ресурс]. – Режим доступу : http://nbuv.gov.ua/UJRN/Vnadps_2013_5_20.

References:

1. *Biznes-symuljacija*, [Online], available at: <https://uk.wikipedia.org/wiki/Biznes-symuljacija>
2. Vakaljuk, T.A. and Antonjuk, D.S. (2016), «Vykorystannja hmarnyh biznes-symuljatoriv dlja vyvchennja povedinkovoi ekonomiky», *Naukova molod'-2016*, zbirnyk materialiv IV Vseukrai'ns'koi naukovo-praktychnoi konferencii molodyh uchenyh (15 grud. 2016 r., m. Kyi'v), za zag. red. prof. O.M. Spirina, IITZN NAPN Ukrainy, K., pp. 53–56, [Online], available at: <http://lib.iitta.gov.ua/707095>
3. Antonjuk, D.S. (2018), «Vykorystannja programno-imitacijnyh kompleksiv jak zasobiv formuvannja ekonomichnyh kompetentnostej studentiv tehnicnyh special'nostej», Ph.D. Thesis of diss., 13.00.10., Kyiv.
4. The Market for Lemons, [Online], available at: https://en.wikipedia.org/wiki/The_Market_for_Lemons
5. Bruehler, James R., Grant, Alan P. and Ghent, Linda S. (2017), «The Red/Green Simulation», *Journal of Economics and Finance Education*, Vol. 16, Issue 1, [Online], available at: <https://www.economics-finance.org/jefe/issues/JEFE-Vol-16-Num-1-Winter-2017.pdf>
6. *Ob'jektno-orijentovane programuvannja*, [Online], available at: https://uk.wikipedia.org/wiki/Ob'jektno-orijentovane_programuvannja
7. *Bring your own device*, [Online], available at: https://uk.wikipedia.org/wiki/Bring_your_own_device
8. *Mikroservisy*, [Online], available at: <https://uk.wikipedia.org/wiki/Mikroservisy>
9. *Model-vyd-kontroler*, [Online], available at: <https://uk.wikipedia.org/wiki/Model-vyd-kontroler>
10. *Architectural principles*, [Online], available at: <https://docs.microsoft.com/en-us/dotnet/standard/modern-web-apps-azure-architecture/architectural-principles>
11. Adams, W.K., *Student engagement and learning with PhET interactive simulations*, [Online], available at: http://www.jdes.tyc.edu.tw/PhET2013/publications/MPTL_2010_PhET_final.pdf
12. Castronova, E., *Exodus to the Virtual World: How Online Fun is Changing Reality*, Palgrave Macmillan, [Online], available at: <http://us.macmillan.com/exodustothetvirtualworld/edwardcastronova>

13. Liang, T. and Shan, T.R. (2013), «Classical atomistic simulations of surfaces and heterogeneous interfaces with the charge-optimized many body (COMB) potentials», *Materials Science and Engineering, Reports, R.*, Vol. 74, Issue 9, pp. 255–279.
14. Podolefsky, N.S., Perkins, K.K. and Adams, W.K. (2010), «Computer simulations to classrooms: tools for change», *Physics Education Research, Conference Proceedings*, AIP Press, [Online], available at: https://www.colorado.edu/physics/EducationIssues/papers/Podolefsky_et al/Podolefsky_invited_PERC_09_scales_revised.pdf
15. Fortmann-Roe, Scott (2014), «Insight Maker: A general-purpose tool for web-based modeling & simulation», *Simulation Modelling Practice and Theory*, No. 47, pp. 28–45, [Online], available at: https://www.researchgate.net/publication/263127834_Insight_Maker_A_general-purpose_tool_for_web-based_modeling_simulation
16. Shenderuk, O.B. and Perminova, V.A. (2010), *Tehnologii' sytuatyvnoho modeljuvannja. Symuljacji' abo imitacijni igry*, [Online], available at: http://www.rusnauk.a.com/26_SSN_2010/Pshilologia/71388.doc.htm
17. Macjuk, O.O. (2013), «Komp'juterni symuljacji' jak zasib formuvannja profesijnoi' kompetentnosti majbutnih perekladachiv v umovah informacijnogo suspilstva», *Visnyk Nacional'noi' akademii' Derzhavnoi' prykordonnoi' sluzhby Ukrainy*, Issue 5, [Online], available at: http://nbuv.gov.ua/UJRN/Vnadps_2013_5_20

Морозов Андрій Васильович – кандидат технічних наук, доцент, доцент Державного університету «Житомирська політехніка».

Наукові інтереси:

– інженерія програмного забезпечення.

E-mail: morozov.andriy@gmail.com.

ORCID: 0000-0003-3167-0683

Вакалюк Тетяна Анатоліївна – кандидат педагогічних наук, доцент, професор Державного університету «Житомирська політехніка».

Наукові інтереси:

– інформаційно-комунікаційні технології;

– хмарні сервіси, теорія і методика навчання (інформатика);

– інженерія програмного забезпечення.

E-mail: tetianavakaliuk@gmail.com.

ORCID: 0000-0001-6825-4697

Антонюк Дмитро Сергійович – кандидат педагогічних наук, доцент Державного університету «Житомирська політехніка».

Наукові інтереси:

– програмно-імітаційні комплекси;

– бізнес-симуляції, розробка бізнес-симуляцій;

– інженерія програмного забезпечення.

E-mail: dmitry_antonyuk@yahoo.com.

ORCID: 0000-0001-7496-3553

Дідківський Владислав Валентинович – студент 1 курсу магістратури факультету інформаційно-комп'ютерних технологій Державного університету «Житомирська політехніка».

Наукові інтереси:

– розробка бізнес-симуляцій;

– інженерія програмного забезпечення.

E-mail: v.didkivskyi@sana-commerce.com.

ORCID: 0000-0002-4615-7578

Стаття надійшла до редакції 20.09.2019.