УДК 004.9

Н. Б. Шаховська, Б. П. Худоба
Національний університет “Львівська політехніка”,
кафедра інформаційних систем та мереж

СИСТЕМА ЗАОХОЧЕННЯ МОЛОДИХ ЛЮДЕЙ ДО НАУКОВОЇ РОБОТИ

Досліджено проблеми, які привели до створення системи заохочення до наукової роботи. Встановлено цільову аудиторію, яка проводить свій час за онлайн-іграми. Середній вік “геймера” – 30 років, а 47 % – це жінки. Вказано на популярність онлайн-ігор у соціальних мережах та прибуток для розробників ігор у різних країнах. Виділено основні завдання розроблення системи. Розглянуто можливість написання ігор самими студентами, що дасть змогу зрозуміти, чого саме вони хочуть, що їх найбільше цікавить. Проаналізовано вже наявні вирішення проблеми, наведено їхні переваги та недоліки, які враховано в проектованій системі. Описано можливості системи. Враховано важливі елементи системи, а саме: надання графічного представлення реалізованих користувачами програм та їх оцінювання, що дає можливість вести статистику і на її основі давати поради. Обрано гравців, які присутні в системі, та вказано можливості кожного. Наведено сучасні архітектурні рішення, які дають можливість використовувати спільний API для декількох клієнтів. Проаналізовано сучасні популярні шаблони проектування, які покладено в основу системи. Змодельовано діаграму класів для проведення змагань, яка показує роботу основного завдання системи. Також наведено технології, які присутні в системі, та архітектуру проекту. Сформовано етапи реалізації та впровадження системи.

Ключові слова: інноваційні методи навчання, архітектура системи, геймифікація.

The issues that led to the establishment of the research incentive scheme have been discovered discovered. The target audience who spend their time playing online games have been identified. The average age of “gamers” is 30 years old, and 47 % of “gamers” are women. The popularity of online games in social networks and revenue for game developers across countries are indicated. The main tasks in the design of the system are selected. The opportunity of writing games by students themselves has been considered. It will help to understand exactly what they want and what they are mostly interested in. Existing solutions have been analyzed. Their advantages and disadvantages have been included in projected system. Activities of the system have been described. Important elements of the system, namely: providing graphical representation for implemented programs from solutions and evaluation, which makes it possible to keep statics and to give advice have been included. Players present in the system and their possibilities have been given. Advanced architectural solutions, which enable the use of a common API for multiple clients have been provided. The current popular design patterns that underlie the system have been analyzed. The chart of grade for competition which shows the work of the main objectives of the system have been modeled. Also technologies that are present in the system architecture have been given. The stages of implementation have been formed.

Key words: innovative teaching methods, system architecture.

Вступ

Інноваційні способи зробити технічну освіту і наукову діяльність привабливою для молодих людей – проблема не лише в Україні, а й загалом у світі. Освіта – це процес, у результаті якого людина отримує знання та практичні навички. В результаті цього процесу людина розвиває розумово-пізнавальний та морально-естетичний рівні. Питання заохочення студентів та аспірантів до навчання і наукової діяльності в Україні та світі стає все актуальнішим для всіх верств населення, оскільки падає кваліфікаційний рівень майбутніх працівників, що призводить, своєю чергою, до зниження якості продукції в матеріальній та нематеріальній сферах. Відсутність молодих людей, які хочуть займатися науковою діяльністю, призведе до того, що буде недостатньо людей на робочих місцях. Ще один фактор, який випливає з відсутності мотивації – це гальмування розвитку, а в деяких випадках і деградація населення. Саме тому виникає потреба в розробленні системи, яка б поєднувала цікаву гру і одночасно заохочувала студентів до навчання, а також стимулювала їх до самовдосконалення та спрямовувала до наукової діяльності.

Постановка задачі

Сьогодні онлайн-ігри стали популярними серед молодих людей. Встановлено, що середній вік "геймера" – 30 років, причому 47 % – це жінки. Відомо, що ігрова індустрія – одна з найрозвиненіших галузей розроблення програмного забезпечення. В Польщі обсяги коштів, отриманих від їх продажу, та кількість програмістів, що працюють над створенням ігор, є найбільшою серед усіх ділянок ІТ. Маючи змогу розробляти ігри, змагатися з іншими, студенти отримають реальні практичні знання та навички, які знадобляться їм для успішної кар'єри. Також залучення студентів до написання ігор дасть змогу збільшити кількість та якість пошукувачів наукового ступеня, адже кожна гра вимагатиме розроблення власних алгоритмів, архітектур, методів тощо, що, за умови стимулювання та заохочення, дасть змогу студентові розпочати наукову кар'єру.

Ринок ігор зараз дуже популярний (див. http://bizua.org/413/mobilni-i-kompyuterni-igri-xto-yak-i-za-skilki-rozroblyaye-ix-v-ukraini). В http://watcher.com.ua/2010/11/03/yak-rozvyvayetsya-rynok-onlayn- ihor-chastyna-1/ наведено такі факти:

- "Ігри сьогодні у США є головною активністю користувачів в соціальних мережах. Американці грають 39 % часу, проведеного у Facebook, а близько половини користувачів заходять на сайт спеціально з метою пограти в ігри.
- В Європі та США чітко прослідковується тенденція до збільшення часу, який гравці витрачають на гру.
- Соціальні ігри з кожним роком ускладнюються, але водночас зростає й рівень самих гравців.
- Середньостатистичний гравець у соціальних мережах одночасно грає не в одну, а в 2–3 гри.
- За інформацією компанії "6 waves", середньостатистичний користувач соціальної гри приносить протягом року власнику цієї гри в США $0,55, в Європі – $0,41, в Азії – $0,22, Австралії – $0,32, Латинській Америці – $0,16.
- ARPU в країнах Європи росте, і очікується, що в 2012 році наздожене американський. Усе це свідчить про наявність попиту на фахівців з розроблення ігор як в Україні, так і в світі. Розроблення "Віртуальної лабораторії ігор" зможе заохотити цільову аудиторію до наукової роботи.

Тому метою статті є детальний опис архітектури системи заохочення молодих людей до навчання.

До поставлених задач системи належать:

- розроблення задач для предметної області;
- реалізація системи оцінювання та перевірки рішень;

- надання графічного представлення користувачеві;
- збереження даних.

1. Аналіз джерел

Сьогодні одним з найпопулярніших методів навчання є проведення відеоуроків з поясненнями лекторів. Також відеоуроки доповнюються тестами та домашнім завданням, щоб можна було слідкувати за прогресом учнів. Такий метод має достатню кількість переваг, які зробили його популярним, а саме [11]:

- можливість переглядати в будь-який момент і будь-де;
- матеріал, який викладають, є якісним завдяки підтримці великих компаній;
- контент оновлюється.

Також існують системи для імітування реальних умов для досліджень. Ці системи допомагають підібрати цікаву для кожного індивіду галузь досліджень.

The Virtual Lab дає змогу студентам увійти у роль вченого, лікаря-імунолога. Користувачі виконують лабораторні роботи за такими темами, як кардіологія, імунологія та ідентифікація бактерій [6].

Існуючі онлайн-ресурси

Ресурс	Опис
prometheus prometheus.org.ua	відеоуроки різної тематики
codeschool codeschool.com	тести і практичні завдання з поясненнями
codeforces codeforces.com	онлайн-змагання, орієнтовані на програмування і математику
Udacity udacity.com	відеоуроки, які орієнтовані на розроблення ПЗ
Coursera coursera.org	відеоуроки різної тематики
EDX edx.org	відеоуроки різної тематики
codeacademy codecademy.com	тести і практичні завдання з поясненнями
Topcoder topcoder.com	онлайн-змагання, орієнтовані на програмування і математику

Розроблені в тісній співпраці з педагогами та вченими, ці віртуальні лабораторії повністю інтерактивні та можуть надати студентам навички та методи, використовувані в наукових дослідженнях. **BioInteractive –** серія віртуальних лабораторій, що дає змогу студентам вивчати особливості діагностики пацієнтів із захворюваннями серця, аналізувати послідовності ДНК, перевірити нервову систему п'явки, використовувати людські антитіла для діагностики захворювання і вимірювання фізичних рис. Кожна лабораторія забезпечує інтерактивне середовище, в якому студенти проводять експерименти, збирають свої дані і відповідають на питання для оцінювання їх розуміння. Лабораторії об'єднують анімації, ілюстрації та відео, щоб передати ключову інформацію та залучити студентів до процесу організації науки [7].

Такі лабораторії дають відчуття відповідальності за свої дії і розуміння того, що неврахування чогось може загрожувати життю інших людей.

Stellarium – це безкоштовний відкритий планетарій, що показує реалістичне небо в 3D [8]. Звісно, Stellarium не замінить реальності, але він легко може зацікавити людей до того, щоб спробувати себе у цій галузі науки.

Особливості Stellarium:

- За замовчуванням понад 600 000 зірок;
- Додаткові каталоги з понад 210 000 000 зірок;
- Астеризм та ілюстрації сузір'їв;
- Реалістичний Чумацький Шлях;
- Зображення туманностей (повний каталог Мессьє);
- Планети та їхні супутники.

LinguaLeo – навчальний онлайн-сервіс, побудований для вивчення та підвищення рівня іноземної мови. Платформа побудована на грі [9]. Користувач переходить на нові рівні після того, як виконає завдання і вивчить слова, граматичні правила тощо. Завдяки геймифікації LinguaLeo налічує понад 13 млн. користувачів. Сервіс має підтримку таких клієнтів:

- iOS;
- Android;
- Windows Phone;
- Web Application.

Проаналізувавши існуючі варіанти вирішення проблеми, потрібно зазначити, що ідея геймифікації приводить до очікуваних результатів. Основні переваги геймифікації [10]:

- висока популярність ігор;
- гра не завжди має єдине рішення, що дозволяє користувачеві розвивати мислення;
- результат гри завжди можна покращити, мінімізувати витрати часу, ресурсів, а це не дасть змоги зупинитися на досягнутому;
- можливість проведення в режимі змагань додає азарту молодим людям.

Головну роль цієї системи має зіграти наповнення інформаційної системи. Треба врахувати історію LinguaLeo, який також використовував геймифікацію, але сьогодні цим сервісом користується лише 1% людей, які зареєструвались. Оскільки в іграх немає чітко описаного плану перемоги, то користувачі зможуть самі придумати свою стратегію, яку зможуть доповнити концепціями штучного інтелекту (нейронні мережі або/і генетичні алгоритми).

Також важливим пунктом в розробленні такої системи є врахувати побажання дівчат для забезпечення гендерної рівності, бо іграми цікавляться переважно хлопці.

2. Основний матеріал

Першим пунктом в проектуванні інформаційної системи має бути бути побудована діаграма прецедентів (рис. 1). Діаграма прецедентів – це діаграма, яка описує відносини між акторами та прецедентами, які обмежені границею [1]. На цій діаграмі чітко описано всі можливості системи.

У проектованій системі присутні 3 актори:

- Студент;
- Викладач;
- Адміністратор.

Актори утворюють ієрархію, в якій на верху є студент. Між Студентом і Викладачем та між Викладачем і Адміністратором є відношення генералізації.

Прецеденти доступні Студенту, Викладачеві та Адміністратору:

1. Зареєструватися – дозволяє користувачам реєструватися;
2. Автентифікуватися – процес введення логіна і пароля для ідентифікація користувача;
3. Переглянути або редагувати профіль – дозволяє переглянути реєстраційні та додаткові дані або змінити їх;
4. Переглянути матеріали курсу та інструкції – дозволяє ознайомитися з інформацією про курс, теоретичні відомості, ігри, які присутні, опис ігор;

5. Додати рішення до гри – процес додавання рішення є вкладенням текстового файлу, який система відкомпілює та виконає разом з іншими рішеннями і порівняє результативність та оптимальність запропонованого рішення;

6. Переглянути результат – дозволяє переглянути в короткому форматі результати або ж візуалізований варіант.

7. Вийти – дозволяє вийти з облікового запису.

Прецеденти, доступні Викладачеві та Адміністратору:

1. Додати курс або гру – користувач має можливість створити новий курс і/або гру;

2. Додати матеріали курсу – дозволяє наповнювати курс додатковими матеріалами, які допоможуть написати краще рішення до гри або застосувати швидший і/або ресурсоекономніший алгоритм;

3. Переглянути прогрес – дозволяє користувачеві переглянути прогрес відносно часу, змін в курсі або наповненні курсу.

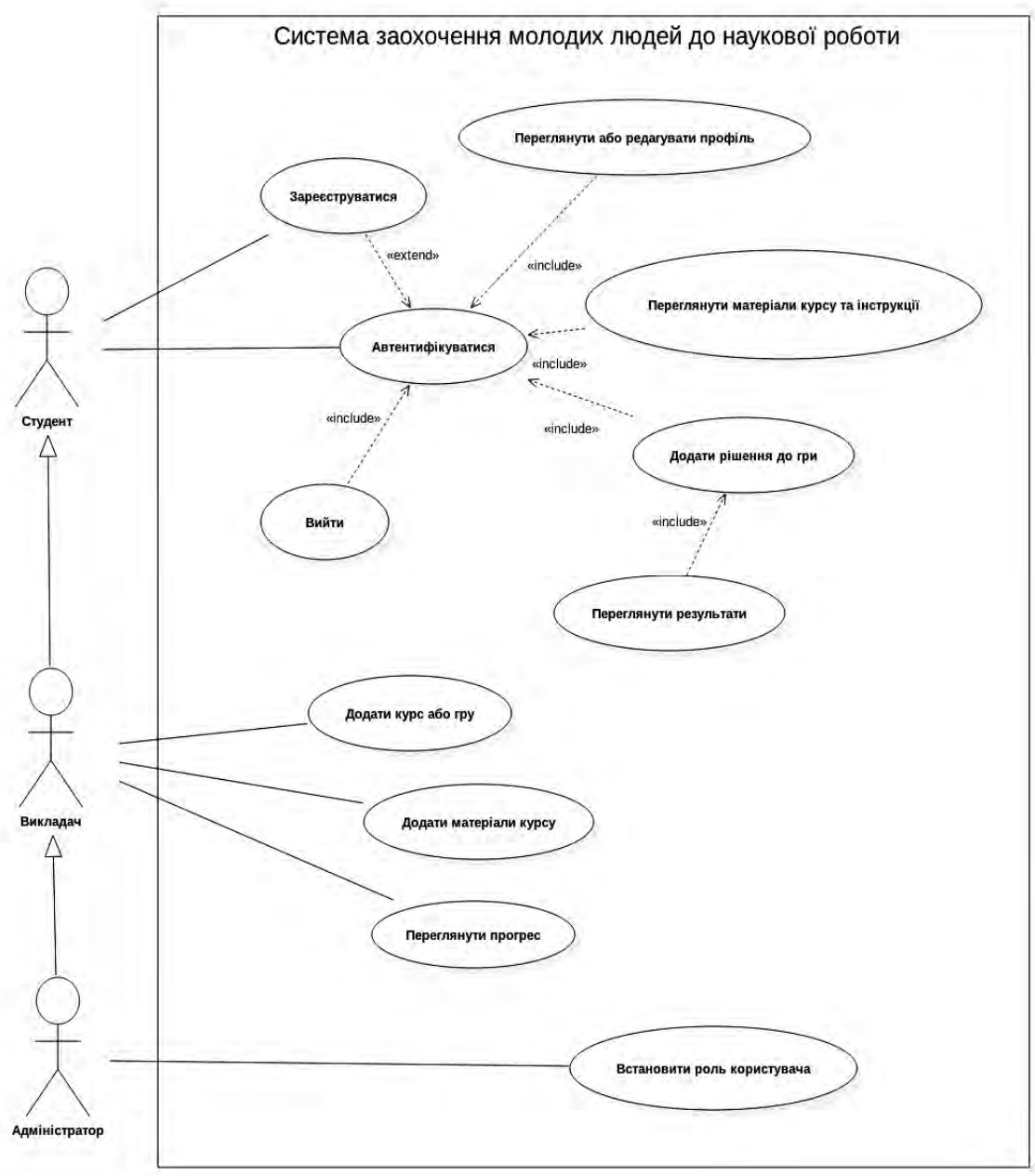

Рис. 1. Діаграма прецедентів

Прецеденти, доступні Адміністратору: встановити роль користувача – користувач може надати/забрати права викладача в інших користувачів.

Наступним важливим кроком є проектування архітектури інформаційної системи [1]. Розглянемо сучасне застосування, яке може включати кілька мобільних додатків на різних платформах і веб-додатки [3]. Без API базова архітектура може виглядати так, як на рис. 2, де кожен клієнт має власну вбудовану бізнес-логіку. Варто звернути увагу, що кожен клієнт має власну вбудовану бізнес-логіку (ймовірно, написану різними мовами), яка підключається безпосередньо до бази даних, щоб отримувати, оновлювати і маніпулювати даними. Це означає, що клієнт- додатки можуть легко стати складними, і всі повинні бути синхронізовані один з одним. Коли потрібна нова функція, то доведеться оновлювати кожен додаток відповідно. Це може бути дуже дорогим процесом, який часто призводить до помилок.

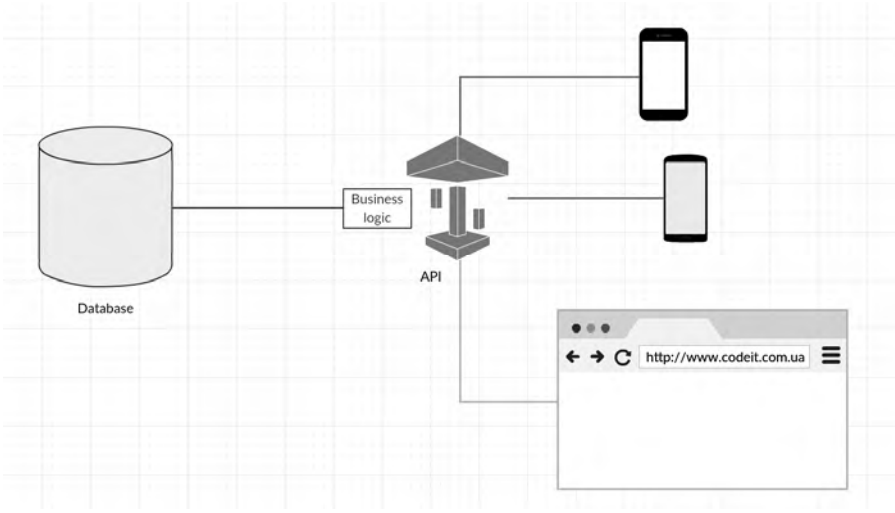

Рис. 2. Схема без спільного API

Тепер розглянемо ту ж архітектуру з центральним API, що має всю бізнес-логіку.

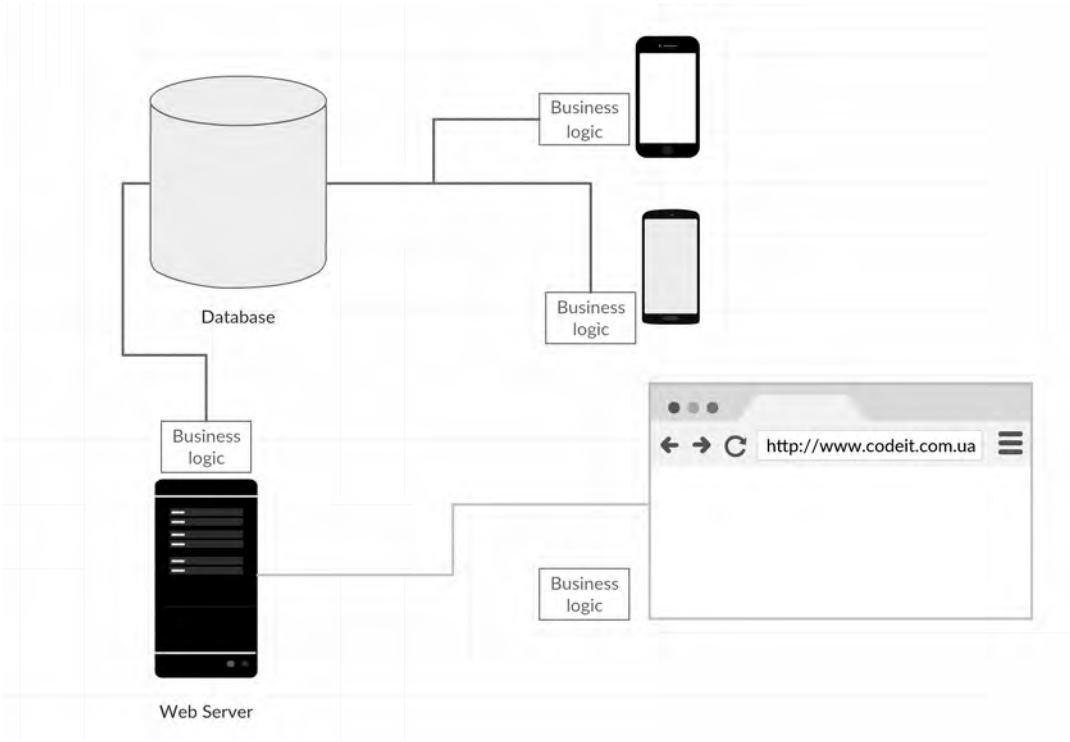

Рис. 3. Схема з спільним API

Кожна програма використовує той самий API, щоб отримувати, оновлювати і маніпулювати даними. Оскільки додатки мають той самий набір функцій, потрібно змінювати лише в одному місці певний функціонал, і він одразу зміниться на всіх клієнтах. Це означає необхідність зміни UI частини для кожного клієнта.

Схема зі спільним API є пріоритетнішою, бо дасть змогу легко створити клієнти для різних мобільних платформ, які зараз є дуже популярними. Більшість сучасних гаджетів підтримують можливість створення текстового файла, що дасть їм змогу реалізовувати свої ідеї в будь-який момент часу.

Також потрібно вибрати MV* шаблон проектування для веб-аплікації [4]:

- MVVM – використовується в ситуаціях, коли немає необхідності реалізовувати View частину. Тобто не потрібно надавати інтерфейси для введення інформації;
- MVP – використовується в ситуаціях, коли неможливо зв’язувати дані;
- MVC – використовується в ситуаціях, коли зв’язок між представленням та іншими частинами аплікації неможливий.

Для проектованої системи шаблон проектування MVC підходить найбільше, бо зв’язок між даними і представленням є неможливим.

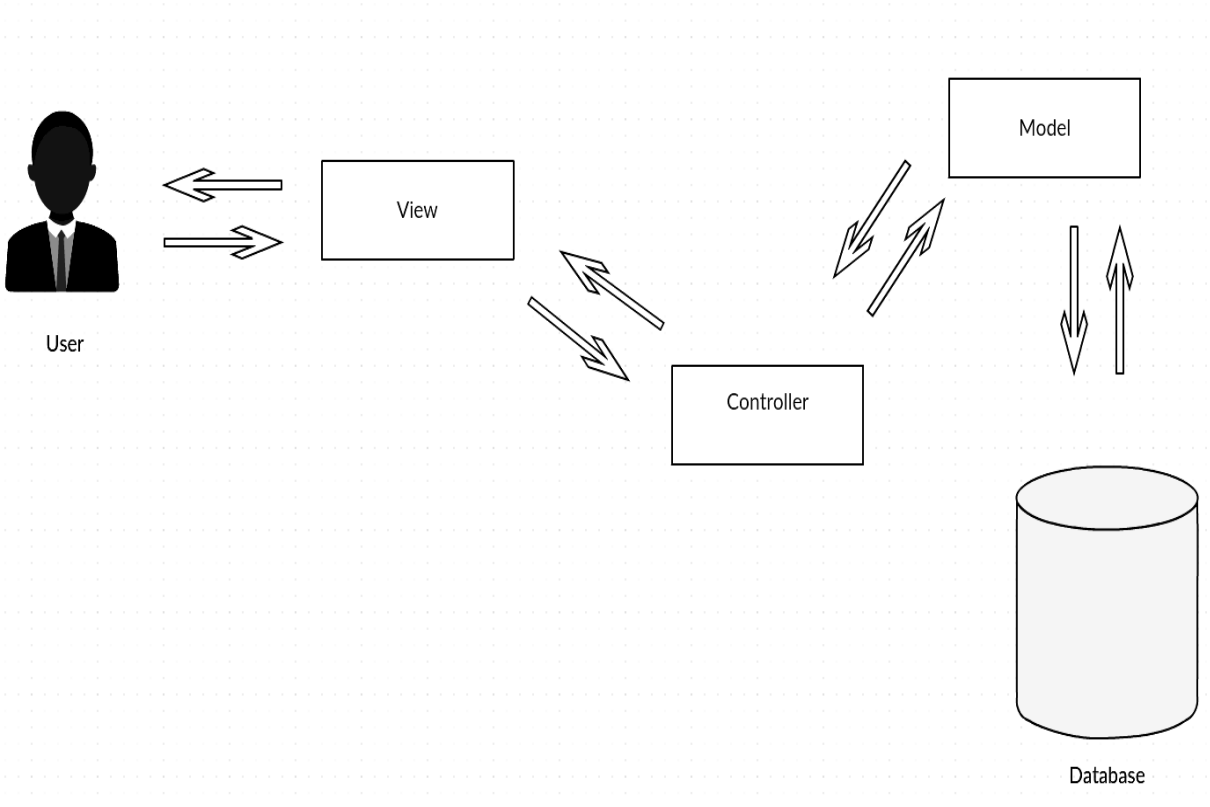

Рис. 4. Шаблон проектування MVC

Користувач матиме прямий зв’язок лише з представленням (View), а контролер (Controller) маніпулюватиме даними між представленням і моделлю (Model). Зв’язок з базою даних відбувається через модель. Значні переваги має Spring MVC Framework [2]:

- Spring забезпечує дуже чистий поділ між контролерами, моделями JavaBean і представленнями;
- незалежна від Servlet API валіадція;
- MVC Spring є дуже гнучким. Spring MVC повністю ґрунтується на інтерфейсах. Крім того, майже кожну частину Spring MVC налаштовують за допомогою підключення у вашому власному інтерфейсі;

− Spring забезпечує перехоплювачі, а також контролери, що дає змогу легко винести загальну поведінку для обробки багатьох запитів;

− не обов'язково використовувати JSP, а, наприклад, Velocity, XSLT або можете самостійно реалізувати інтерфейс Spring View;

− Spring контролери використовують шаблон проектування Inversion of Control, що дає змогу легко відокремити об'єкти, а завдяки цьому їх можна легко перевірити і вони вдало інтегровані з іншими об'єктами;

− Присутній строгий інтерфейс для бізнес-логіки;

− Відсутня залежність на контролерах;

− Прив'язка йде безпосередньо до доменних об'єктів.

Spring Framework містить не лише Spring MVC Module[3, 5], а й інші модулі:

− Spring AOP – містить метадані;

− Spring ORM – містить реалізацію проектування інформаційної системи;

− Spring DAO – механізм, який виконує CRUD-операції.

На наведеній діаграмі класів (рис. 5) можна побачити, як відбуватиметься процес гри. Користувач вибере гру і передаватиме контролеру (GameController) рішення до гри. Наступними діями системи буде створення в базі даних нового рішення від користувача (успішного чи ні). У бізнес-логіці системи закладений клас GameManager, який виконуватиме процес гри, тестування та перевірятиме, чи проходить рішення за обмеженням часу виконання і використаною пам'яттю. Після цього згенерується JSON-Object, який повернеться до користувача і зможе візуально відобразити дії бота, якого він написав. Це дасть змогу легко побачити свої помилки і що можна покращити в наступних версіях свого бота. Така можливість сподобається молодим людям, які витрачають багато часу на відлагодження помилок під час виконання звичайних лабораторних робіт, оскільки вони бачитимуть результат моментально і знатимуть, що можна змінити.

Перевагами такої архітектури є те, що GameManager – невелика компонента, яку можна легко підтримувати і вдосконалювати (рис. 6). Також можна використовувати різні реалізації GameManager. Його можна поєднати з eJudge системою, що спростить розроблення системи. Звісно, можна буде зробити свою реалізацію. Також існує цікавий варіант: написати свій компілятор з уже реалізованими методами та функціями, які будуть потрібні для реалізації бота до гри. Це спростить влиття в навчання для початківців, які тільки почали займатися.

Також потрібно врахувати апаратну частину, яка відповідатиме за розгортання системи. Роль веб-сервера виконуватиме контейнер сервлетів Apache Tomcat 8, який має підтримку Java 8 та Servlet API 3.1, що дасть змогу довго підтримувати систему.

Роль сервера виконує хмарний сервер, який має свої переваги:

− свобода модифікування серверного та програмного забезпечення для певних потреб. Містить ядро операційної системи, яке не завжди є таким самим ефективним у випадку з іншими рішеннями віртуалізації, такими як віртуальні приватні сервери.

− стабільність і безпека, оскільки проблема програмного забезпечення – ізолюватися від навколишнього середовища. Хмарний сервер не може зашкодити системі користувача та іншому хмарному серверу. Крім того, якщо інші користувачі перевантажують свої хмарні сервери, це не матиме жодного впливу на поточний сервер, тому що ресурси збережені і стабільність гарантується. На додаток до цього хмарні сервери не страждають від проблем з апаратним забезпеченням.

− хмарні сервери є економічно ефективнішими, ніж стандартні виділені сервери. За аналогічною ціною з хмарних серверів можна отримати більше ресурсів, сервер працюватиме швидше.

− хмарні сервери дуже добре масштабуються. Це дає змогу легко і швидко додати або поновити CPU, пам'ять, дисковий простір до сервера.

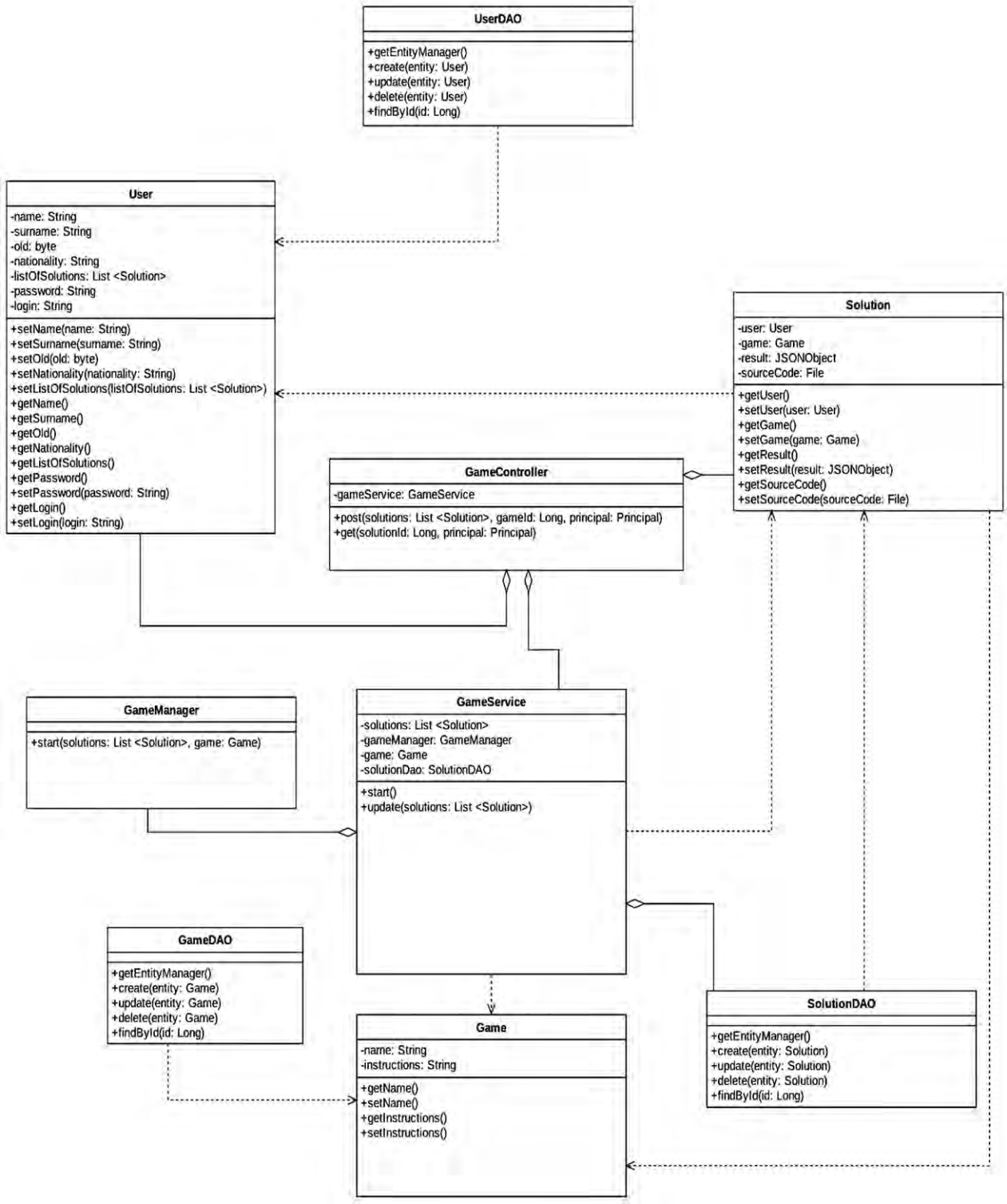

Рис. 5. Діаграма класів

Врахувавши всі деталі, можна навести схему розроблення системи (рис. 8). Поділимо процес розроблення на такі 4 етапи:

1. Розроблення – написання програмного коду;
2. Керування – процес контролю програми за допомогою систем контролю версій;
3. Побудова та тестування – тестування програмного продукту після збирання кожної нової версії;
4. Хостинг – етап, на якому системою користується цільова аудиторія.

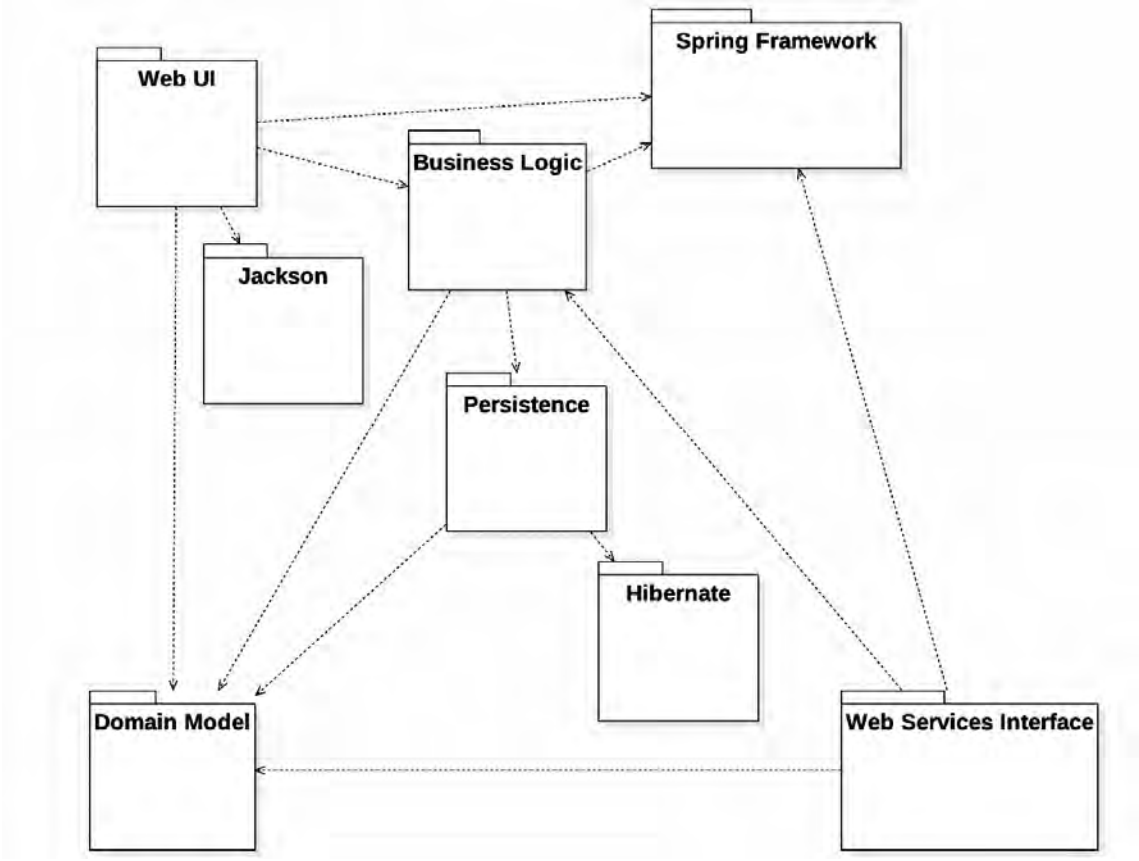

Рис. 6. Діаграма пакетів

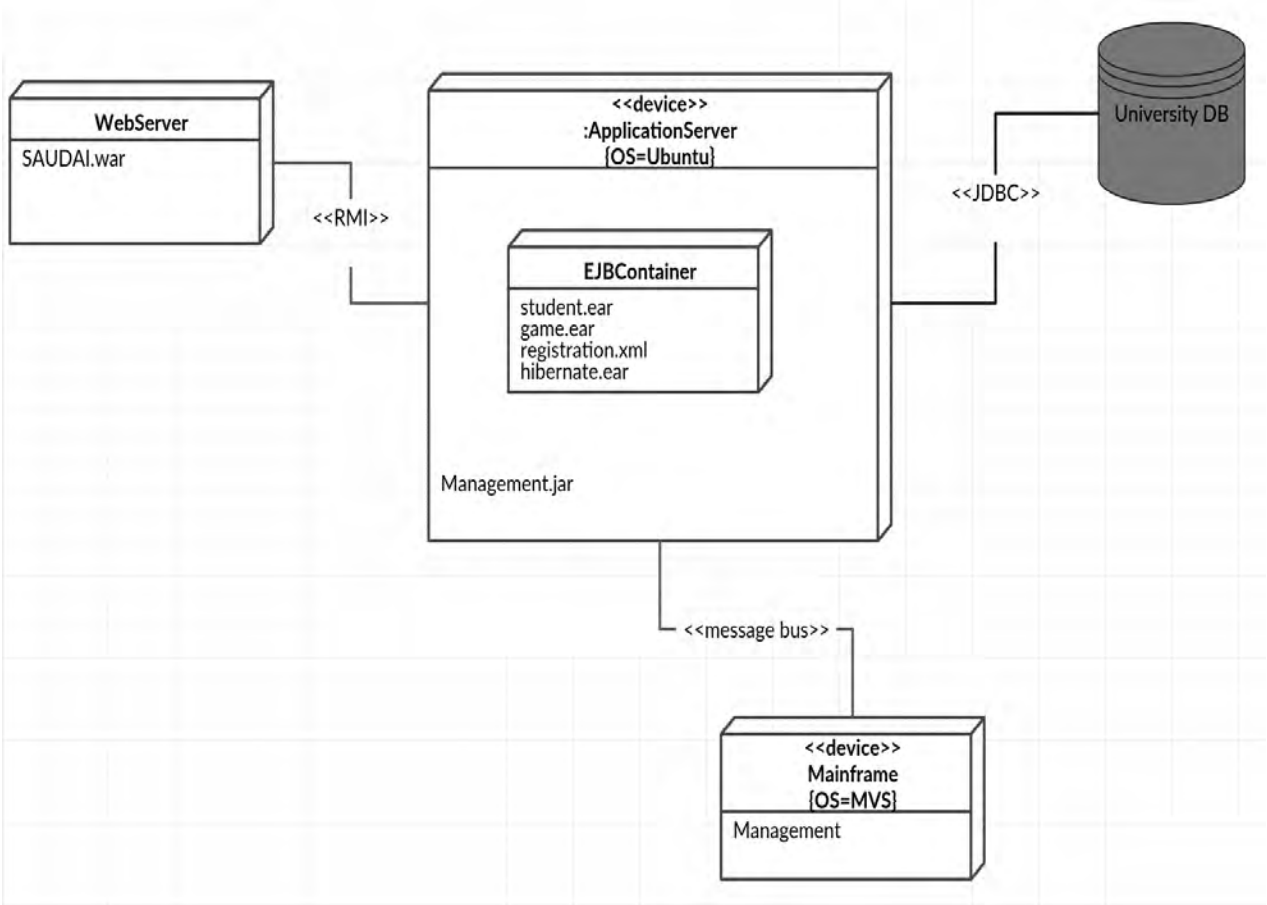

Рис. 7. Діаграма розгортання

Засобом розроблення програмного продукту вибрано середовище Intellij idea. Системою контролю версій є Git, що дасть змогу зацікавленим людям переглянути процес розробки в будь-який момент і запропонувати свої ідеї. Важливою частиною є система безперервної інтеграції. Для цього можна використати написаний на Java Jenkins Server[4]. Система безперервної інтеграції дає проекту певні переваги: інтеграційні помилки виявляються швидко, легко виправляються і спричиняють менші витрати; якщо в новій версії системи не пройшли успішно тести, то система повертається до останньої робочої версії без жодних втручань, потокова побудова програми перед тестуванням, модульна розробка. Процес тестування відбуватиметься до впровадження системи у використання. На рис. 9 наведено приклад гри “Шлях до вузла”. Завдання гри: маючи інформацію про сусідні вузли, дійти до заданого вузла за найменшу кількість кроків. Рухи можна виконувати праворуч, ліворуч, догори, донизу.

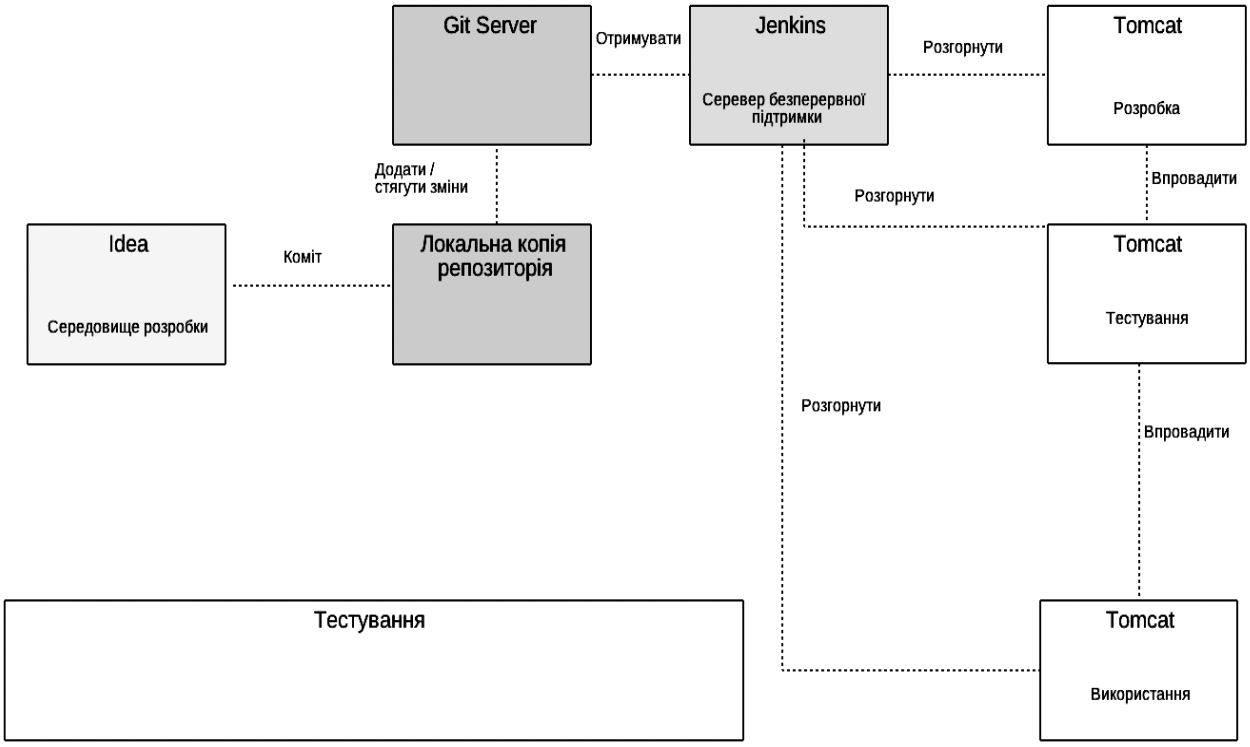

Рис. 8. Схема розроблення

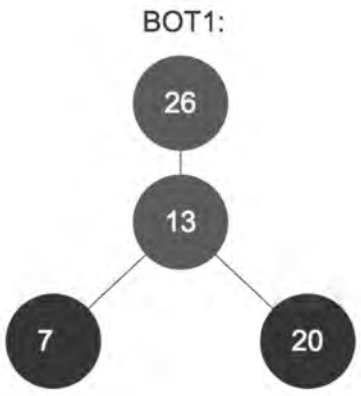

Рис. 9. Приклад гри

Висновки та перспективи подальших наукових ровзздок

Підтверджено важливість створення системи заохочення молодих людей до навчання. Показано, що з цією метою доцільно використовувати не лише традиційні форми навчання та самонавчання, а й геймифікацію. Проаналізовано існуючі системи для самонавчання та визначено їхні переваги та недоліки. Наведено сучасну архітектуру, яка допоможе популяризувати систему і дати очікуваний результат.

1. Rosenfeld L. Information Architecture For the Web and Beyond / Rosenfeld Louis, Morville Peter, Arango Jorge. 4th Edition, .- Sebastopol: O'Reilly Media, 2015. – 478 с. *2. Gerald Gierer Enterprise Application Development with Ext JS and Spring / Gerald Gierer. – Birmingham: Packt Publishing, 2013. – 304 p. 3. Ludovic Dewailly Building a RESTful Web Service with Spring / Ludovic Dewailly. – Birmingham: Packt Publishing, 2015. 4. John Ferguson Smart: Jenkins: The Definitive Guide Continuous integration for the masses / John Ferguson. – Sebastopol: O'Reilly Media, 2011. 5.. Spring Dynamic Modules Reference Guide / Adrian M Colyer (SpringSource), Hal Hildebrand (Oracle), Costin Leau (SpringSource), Andy Piper (BEA). – [Електронний ресурс]. – Режим доступу: http://docs.spring.io/osgi/docs/current/reference/html/. 6. LMS has moved. – [Електронний ресурс]. – Режим доступу:http://www.plm.automation. siemens.com/en_us/products/lms/lms-redirect.shtml. 7. BioInteractive. – [Електронний ресурс]. – Режим доступу: http://www.hhmi.org/biointeractive/explore-virtual-labs. 8. Stellarium. – [Електронний ресурс]. – Режим доступу: http://www.stellarium.org/uk/. 9. Lingualeo. Покори язык. – [Електронний ресурс]. – Режим доступу: https://lingualeo.com/ru. 10. Huang, Wendy Hsin-Yuan, and Dilip Soman. Gamification Of Education. Research Report Series: Behavioural Economics in Action, 2013. 11. Gamification in Education: A Systematic Mapping Study / Darina Dicheva, Christo Dichev, Gennady Agre and Galia Angelova // Journal of Educational Technology & Society. – 2015. – Vol. 18, No. 3. – P. 75–88.*