

аналоги, виявлено їх переваги та недоліки. Була розроблена логіко-структурна схема додатку та необхідний функціонал. Ідея матриці Ейзенхауера полягає у розстановці пріоритетів по ступені важливості та терміновості. Тому потрібно навчитися відрізняти важливі справи від термінових і відкидати справи, які не приносять жодної користі.

#### *Список літератури*

1. [https://chicagorazom.com/matrycy\\_a\\_eizenhauera\\_dlya\\_skladannya\\_spyску\\_sprav](https://chicagorazom.com/matrycy_a_eizenhauera_dlya_skladannya_spyску_sprav)
2. <http://www.management.com.ua/blog/3483>
3. <https://wombat.org.ua/AByteOfPython/AByteofPythonRussian-2.01.pdf>
4. <https://blog.agrokebety.com/matrytsya-eyzenkhauera>

**УДК 004.42**

*Сірко О. О., студент 4 курсу спеціальності 122 «Комп'ютерні науки та інформаційні технології»  
Антонов Ю. С., к.ф.-м.н., доцент, доцент кафедри інформаційних технологій*

## **СТВОРЕННЯ ІНТЕЛЕКТУАЛЬНОЇ МЕРЕЖЕВОЇ ГРИ ШАШКИ**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Зараз, у епоху розвитку інформаційних технологій з'явилась велика кількість різноманітних ігор, як нових, так і багато реалізацій класичних ігор. Зараз ігри у нашому житті займають важливу роль, тому що вони слугують для розвитку корисних навичок а також просто для розваги.

Інтелектуальна гра «Шашки» розвиває уважність, навички стратегічного та логічного мислення та вміння концентруватись. Зараз існує багато реалізацій гри «Шашки», як для гри проти комп'ютера на різних рівнях складності, так і проти інших гравців по мережі. Серед подібних реалізацій можна привести як приклад: Веб-гра на ресурсі logic-games, Додаток для IOS Шашки Онлайн Elite, Веб-гра на ресурсі min2win.

Метою роботи є розробка гри на основі web-серверу та desktop-додатку.

Web-сервер надає Web API для гри, та доступ до бази даних за допомогою Web API, а desktop-додаток виступає клієнтом. Також будуть реалізовані певні заходи безпеки для захисту даних користувачів, що зберігаються на сервері.

Розробка здійснювалась з використанням мови програмування Java, використовуючи такі фреймворки: Spring - для розробки серверу, JavaFX - для розробки клієнту, а також Hibernate (ORM, EntityManager) для об'єктно-реляційного відображення даних, що зберігаються у базі даних. Та управління даними [1].

Насамперед, визначимо, які дані повинні зберігатись у базі даних:

- Дані користувача (дані для авторизації та статистика)

- Дані по іграх (ігри, ходи, запрошення)
- Списки друзів

Насамперед, потрібно реалізувати API, що буде слугувати для авторизації, реєстрації та створення, отримання та зміну даних із бази даних використовуючи протокол http.

Протокол HTTP (HyperText Transfer Protocol) – це протокол прикладного рівня стеку TCP/IP, що дозволяє передавати дані від клієнта до сервера [2]. Схема взаємодії Серверу з Клієнтом наведена у рис. 1

Після розробки Web API відбувається розробка клієнту.

Перше що бачить користувач запустивши програму це форма реєстрації/авторизації. Реєстрація користувача вимагає заповнення наступних полів: аватар користувача, логін, Ім'я користувача, пароль.

Пароль, що ввів користувач хешується алгоритмом SHA-256, та далі використовується його хеш. Після цього, дані відправляються на сервер та створюється новий користувач у базі даних. У базі даних записується хеш-сума конкатенації хешу пароля та секретного ключа, що зберігається на сервері.

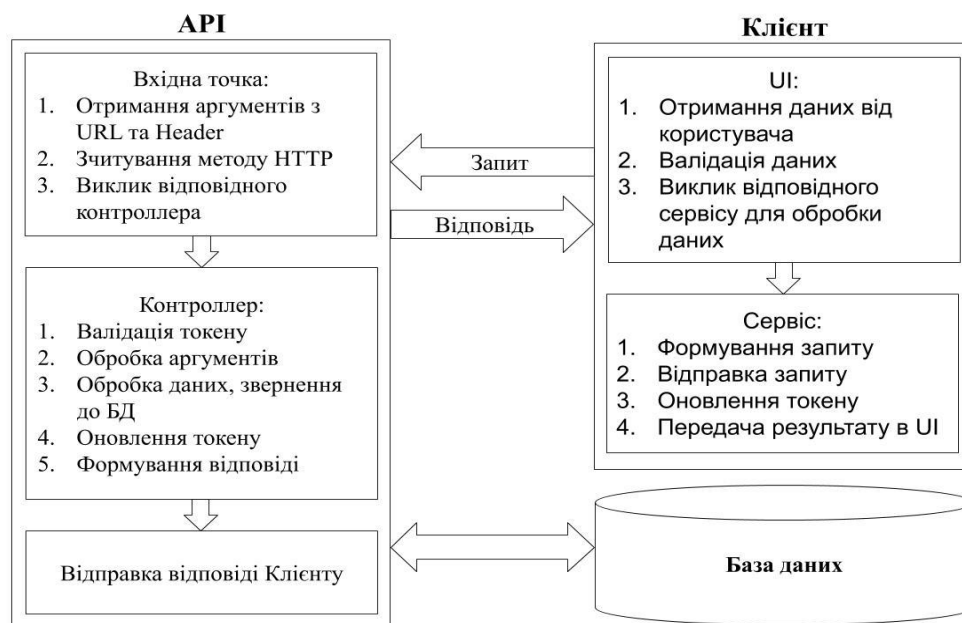


Рисунок 1 – Схема взаємодії Клієнта та сервера

Також для реалізації системи безпеки на API було використано бібліотеку Spring Security та протокол JWT (Json Web Tokens). Протокол JWT використовується для передачі інформації, що стосується ідентичності характеристик клієнта, що несе маркер. JWT є контейнером, що вміщує особисті дані користувача і підписується сервером, щоб уникнути підробки, або підміни токєну.

Після авторизації користувач отримує JWT-токєн, котрий він прикріплює до кожного наступного запиту, що забезпечує безпеку доступу до даних. Токєн має час життя 30хв, містить дані користувача та секретний ключ сервера. Кожен запит на сервер (за винятком авторизації та реєстрації) перевіряє токєн, та оновлює його час життя [3].

Після успішної авторизації користувач потрапляє на головне вікно додатку, де він може побачити інформацію про гравців, що зараз “онлайн”, список своїх друзів, список своїх попередніх ігор та свою статистику, а також має можливість перейти до налаштувань, та до створення нової гри.

Для створення нової гри користувач створює нове запрошення, де він повинен вибрати противника із списку гравців “онлайн” та по бажанню прикріпити повідомлення до запрошення, після створення запрошення воно буде активним на протязі 1 хвилини, за цей час противник може як прийняти запрошення, так і його відхилити.

У процесі гри при виконанні ходу кожен із ходів відправляється на сервер через API та зберігається у базі даних. Також всі ходи будуть записуватись у спільному вигляді нотації гри “Шашки”. При нотації ходу без взяття спочатку вказується поле, з якого шашка пішла, потім через тире пишеться поле, на яке вона встала, а при ході з взяттям поля розділяються не тире, а двокрапкою. [4]

Гравець, котрий чекає ходу противника кожних 2 секунди опитує сервер на предмет нових ходів у грі, та малює їх на формі, якщо такі є.

Таким чином у роботі було розглянуто деякі аспекти створення мережевої гри «Шашки», а саме принципи створення http-серверу та desktop-додатку, що використовує WebAPI. Була розглянута реалізація системи безпеки на сервері, а також процес збереження пароля користувача у базі даних.

#### **Список літератури**

1. Бауэр К., Кинг Г., Грегори Г., *Java Persistence API и Hibernate*, 2017. - 630с
2. Тузовский А. Ф., *Проектирование и разработка web-приложений. Учебное пособие для СПО*, 2018. - 218с
3. Knutson M., Winch R., Mularien P., *Spring Security: Secure your web applications, RESTful services, and microservice architectures*, 2017. - 542с
4. Герцензон Б.М., *Шашки - это интересно*, 1989. - 175 с.

#### **УДК 519.1**

*Степанюк О. С., студентка 2 курсу спеціальності 122 «Комп'ютерні науки»  
Ветров О. С., старший викладач кафедри прикладної математики*

### **РЕАЛІЗАЦІЯ ГРАФІЧНОГО ВАРІАНТУ ІТЕРАЦІЙНОГО АЛГОРИТМУ РІШЕННЯ ГОЛОВЛОМКИ «ХАНОЙСЬКІ ВЕЖІ» ДЛЯ ПЛАТФОРМИ IOS**

*Донецький національний університет імені Василя Стуса, м. Вінниця*

Ханойські вежі[1] є однією з популярних головоломок ХІХ століття. Дано три стержня, на один з яких нанизано  $N$  – кількість кілець, причому кільця відрізняються розміром і лежать у порядку спадання розміру знизу вверху. Ціллю