

МІНІСТЕРСТВО ОСВІТИ ТА НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121«Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08

Дипломний проєкт

здобувач освіти денної форми навчання

РП.08.20.000.ДП

ТРОХИМЧУКА

ЯРОСЛАВА СЕРГІЙОВИЧА

м. Одеса

2025 р.

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Спеціальність: 121 «Інженерія програмного забезпечення»

Освітньо-професійна програма: «Розробка програмного забезпечення»

Група: 4РП-08


ПОЯСНЮВАЛЬНА ЗАПИСКА

до дипломного проекту на тему:

Розробка комп'ютерної гри "Техаський холдем" у web-середовищі

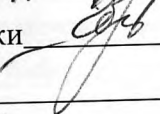
Проектний матеріал складається з пояснювальної записки на 76 сторінках та графічного (презентаційного) матеріалу на 10 аркушах (слайдах)


Дипломник  (Трохимчук Я.С.)

Керівник  (Залапін О.І.)

Консультанти:

з економічного розділу  (Канський М.Ю.)

з розділу охорони праці та техніки безпеки  (Чорновол Н.І.)

з нормоконтролю  (Петрашова В.І.)

старший консультант  (Кривченко Ю.В.)


До захисту допущений

Голова циклової комісії  (Кривченко Ю.В.)

Завідувач відділення  (Краснокутська К.Г.)

Захист « 21 » 06 2025 р. Протокол ЕК № 1

Оцінка ЕК 4/85

Секретар ЕК 

МІНІСТЕРСТВО ОСВІТИ І НАУКИ УКРАЇНИ
ВСП «ОДЕСЬКИЙ ТЕХНІЧНИЙ ФАХОВИЙ КОЛЕДЖ ОНТУ»

Відділення комп'ютерних систем Комісія КТ та ПІ
Спеціальність 121 «Інженерія програмного забезпечення»
Освітньо-професійна програма «Розробка програмного забезпечення»

ЗАТВЕРДЖУЮ:
Заст. дир. з НВР Беркань І.В.
« 12 » 05 2025 р.

ЗАВДАННЯ

на дипломний проект

Здобувачеві освіти Трохимчук Ярослав Сергійович
(прізвище, ім'я, по батькові)

1. Тема проекту Розробка комп'ютерної гри "Техаський холдем" у веб-середовищі

затверджена наказом по коледжу від « 14 » листопада 2024 р. № 246

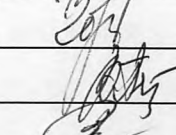
2. Термін здачі закінченого проекту 16 грудня 2025 р.

3. Вихідні данні до проекту 1. Використовувати архітектуру клієнт-сервер з REST API для взаємодії між фронтендом та бекендом, забезпечити роботу в реальному часі для синхронізації ігрових процесів; 2. Реалізувати повну логіку техаського покеру, включаючи автоматичну роздачу карт, підрахунок покерних комбінацій, визначення переможця у кожному раунді та обробку всіх ігрових фаз; 3. Програмне забезпечення реалізувати засобами Node.js/Express для бекенду та React для фронтенду у середовищі розробки Visual Studio Code.

4. Зміст розрахунково-пояснювальної записки (перелік питань, які необхідно розробити)
Аналіз існуючих рішень у сфері онлайн покерних платформ та веб-ігор реального часу; Розробка архітектури системи та структури бази даних для зберігання користувачів, ігрових сесій; Алгоритми реалізації логіки техаського покеру: роздача карт, підрахунок покерних комбінацій та визначення переможця; Програмна реалізація серверної частини на Node.js/Express з інтеграцією MySQL

5. Перелік графічного (презентаційного) матеріалу (з точним зазначенням обов'язкових креслень, кількості слайдів)
Блок-схема архітектури веб-системи техаського покеру (фронтенд React, бекенд Node.js, база даних MySQL); Блок-схема алгоритму роздачі карт та створення колоди з валідацією унікальності; Блок-схема алгоритму підрахунку покерних комбінацій та визначення переможця; Блок-схема системи аутентифікації користувачів;

6. Консультанти по проекту (роботі), із зазначенням розділів проекту, що їх стосується

Розділ	Консультант	Підпис, дата	
		Завдання видав	Завдання прийняв
Основний розділ	Залапін О.І.		
Економічний розділ	Канський М.Ю.		
Розділ охорони праці	Чорновол Н.І.		
Нормоконтроль	Петрашова В.І.		
Старший консультант	Кривченко Ю.В.		

7. Дата видачі завдання 15.05.25р.

Керівник

Залапін О.І.


(підпис)

Завдання прийняв до виконання

Трохимчук Я.С.


(підпис)


КАЛЕНДАРНИЙ ПЛАН

№ з/р	Назва етапів дипломного проекту (роботи)	Термін виконання етапів дипломного проекту (роботи)	Відмітка про виконання
1	Постановка задачі проектування	15.05.2025	Виконано
2	Аналіз онлайн-покерних платформ та ігрових технологій	16.05.2025	Виконано
3	Проектування архітектури системи та визначення структури бази даних	17.05.2025	Виконано
4	Аналіз технологій real-time взаємодії	19.05.2025	Виконано
5	Алгоритм роздачі карт та управління ігровими раундами	22.05.2025	Виконано
6	Реалізація визначення покерних комбінацій й переможця	26.05.2025	Виконано
7	Алгоритм роботи ботів та автоматичних дій	01.06.2025	Виконано
8	Реалізація системи ставок: blinds, call, raise, fold	06.06.2025	Виконано
9	Реалізація системи користувачів: реєстрація, автентифікація	10.06.2025	Виконано
10	Інтеграція універсального API для гри Blackjack	11.06.2025	Виконано
11	Тестування ігрової логіки, користувацького інтерфейсу	12.06.2025	Виконано
12	Виконання економічних розрахунків	13.06.2025	Виконано
13	Розробка заходів з охорони праці	14.06.2025	Виконано
14	Підготовка доповіді та презентації для захисту	16.06.2025	Виконано

Дипломник


(підпис)

Керівник


(підпис)

ЗМІСТ

Вступ.....	7
1 Основний розділ	8
1.1 Аналіз сучасних онлайн-покерних платформ та ігрових технологій	8
1.1.1 Огляд основних покерних платформ та їх порівняльний аналіз	8
1.1.2 Вивчення архітектури веб-додатків для карткових ігор.....	11
1.1.3 Аналіз технологій real-time взаємодії	14
1.1.4 Дослідження технологій збереження стану гри та синхронізації.....	18
1.2 Розробка алгоритмів ігрової логіки техаського покеру	21
1.2.1 Проектування архітектури системи та структури бази даних	22
1.2.2 Алгоритм роздачі карт та управління ігровими раундами	25
1.2.3 Розробка алгоритму визначення покерних комбінацій й переможця .	31
1.2.4 Реалізація системи ставок: blinds, call, raise, fold	35
1.2.5 Алгоритм роботи ботів та автоматичних дій з таймерами	38
1.3 Програмна реалізація веб-додатку для гри в техаський покер	43
1.3.1 Вибір і обґрунтування технологічного стеку розробки	43
1.3.2 Реалізація системи користувачів: реєстрація та автентифікація	46
1.3.3 Інтеграція універсального API для гри Blackjack	50
1.3.4 Тестування розробленого програмного забезпечення	52
2 Економічний розділ.....	53
2.1 Резюме	53
2.2 Визначення трудомісткості розробки програмного забезпечення ..	53
2.3 Розрахунок ціни програмного продукту	55
3 Розділ хорони праці і техніки безпеки	57
3.1 Аналіз небезпечних і шкідливих факторів, що впливають на користувача ПК	57
3.2 Гігієнічні вимоги до виробничого середовища	60
3.2.1 Освітлення	61
3.2.2 Шум.....	62
3.2.3 Вимоги до організації робочого місця	63

3.2.4 Мікроклімат.....	64
3.2.5 Електробезпека.....	65
3.3 Пожежна безпека.....	66
Висновки.....	69
Перелік використаних інформаційних джерел.....	70
Додаток А. Лістинги основних класів та модулів.....	71
Додаток Б. Слайди мультимедійної презентації.....	73

ВСТУП

Сучасний розвиток інформаційних технологій та зростання популярності онлайн-розваг створюють нові можливості для реалізації інтерактивних веб-додатків, зокрема карткових ігор. Техаський покер, будучи однією з найпопулярніших різновидів покеру у світі, представляє особливий інтерес для розробки веб-платформ завдяки своїй складній ігровій логіці та необхідності забезпечення взаємодії між гравцями в реальному часі.

Розробка якісних онлайн-покерних платформ вимагає комплексного підходу до вирішення таких ключових завдань, як реалізація повної логіки техаського покеру з автоматичною роздачею карт, точним підрахунком покерних комбінацій та коректним визначенням переможців у кожному раунді. Особливо критичними є питання забезпечення чесності гри через використання криптографічно стійких алгоритмів перемішування карт, а також створення надійної архітектури клієнт-сервер з REST API для ефективної взаємодії між фронтендом та бекендом. Додатковою складністю є необхідність підтримки багатокористувацького режиму з синхронізацією ігрових процесів між учасниками та збереженням консистентності даних у реальному часі.

Сучасні веб-технології, такі як Node.js/Express для серверної частини та React для клієнтського інтерфейсу, надають потужні інструменти для створення масштабованих ігрових платформ.

У рамках даного дипломного проекту передбачається розробка повнофункціональної веб-платформи для гри в техаський покер, яка включатиме реалізацію всіх аспектів ігрової логіки, створення інтуїтивного користувацького інтерфейсу та забезпечення стабільної роботи. Система аутентифікації користувачів з JWT-токенами гарантуватиме безпеку персональних даних, а розширена функціональність API дозволить у майбутньому інтегрувати додаткові карткові ігри, зокрема Blackjack.

Основним результатом проекту стане створення масштабованого веб-додатку, що продемонструє здатність до самостійної роботи над усіма етапами створення програмного продукту – від ідеї до реалізації.

					<i>РП 08. 20 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		7

1 ОСНОВНИЙ РОЗДІЛ

1.1 Аналіз існуючих онлайн-покерних платформ та ігрових технологій

У сучасних умовах стрімкого розвитку цифрових технологій онлайн-покер займає провідне місце серед карткових ігор в інтернеті, демонструючи стабільне зростання популярності та технологічну еволюцію. Ринок онлайн-покеру характеризується високою конкуренцією між платформами, що стимулює постійне вдосконалення ігрових механік, інтерфейсів користувача та технологічних рішень для забезпечення чесної та захищеної гри.

Сучасні покерні платформи повинні відповідати підвищеним вимогам до продуктивності, надійності та безпеки, оскільки обробляють значні обсяги транзакцій та забезпечують одночасну взаємодію тисяч користувачів у режимі реального часу. Особливо актуальним є створення інноваційних рішень, які поєднуюватимуть класичні правила техаського покеру з сучасними веб-технологіями, забезпечуючи при цьому інтуїтивний користувацький досвід та високий рівень безпеки.

1.1.1 Огляд основних покерних платформ та їх порівняльний аналіз

У рамках дослідження було розглянуто чотири найвпливовіші онлайн-покерні платформи сучасності: PokerStars як беззаперечний лідер ринку, 888poker у якості інноваційного рішення з фокусом на соціальну взаємодію, partypoker як представник класичного підходу до організації ігрового процесу, та GG Poker як новатор у сфері мобільних технологій. Вибір цих платформ обґрунтований їх сукупною часткою понад 75% світового ринку онлайн-покеру, а також різноманітністю технологічних підходів до вирішення ключових завдань галузі.

Методологія дослідження базується на комплексному аналізі технічної документації, архітектурних рішень та практичних тестів продуктивності кожної платформи (Рисунок 1.1).

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		8

Технологічну основу становить гібридна мікросервісна архітектура на базі Java Enterprise Edition з використанням Apache Kafka для асинхронної обробки подій та Redis Cluster для кешування ігрових станів у пам'яті.

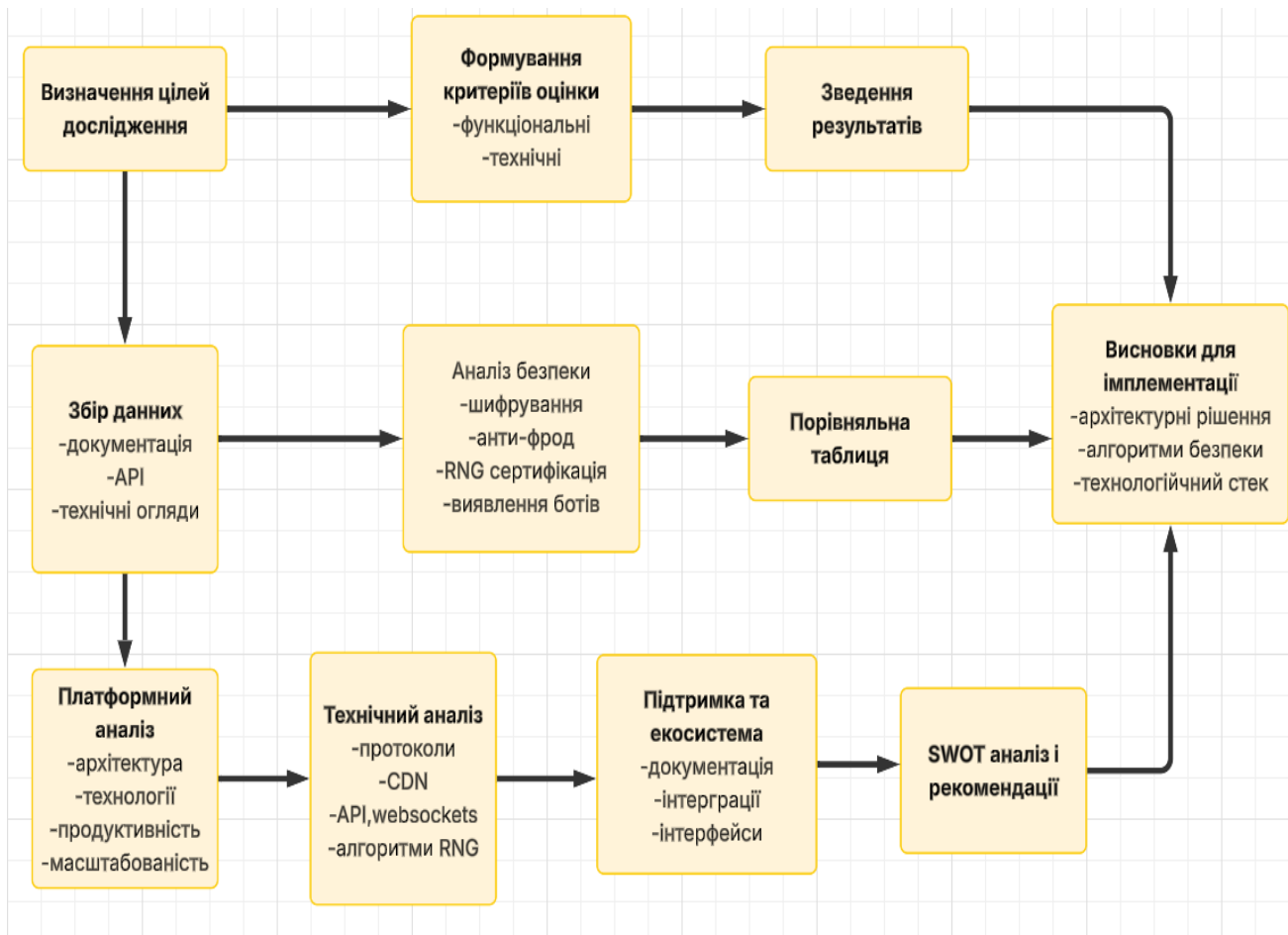


Рисунок 1.1 Методологічна схема комплексного технічного аналізу онлайн-платформ

Інфраструктурні особливості:

- Географічно розподілена система дата-центрів з реплікацією даних через Master-Slave PostgreSQL кластери;
- Сертифікований Random Number Generator (RNG) на базі алгоритму Mersenne Twister з додатковою ентропією від системних подій;
- WebSocket API з підтримкою протоколу compression (deflate), що знижує обсяг трафіку на 60-70%;
- HTTP/2 з Server Push для мобільних додатків;

Система безпеки: Антишахрайська система базується на машинному навчанні з аналізом понад 200 метрик поведінки гравців: від timing tells

(швидкості прийняття рішень) до статистичних аномалій у результатах. Алгоритм використовує ensemble методи (Random Forest + SVM) для виявлення ботів та колюзії між гравцями з точністю понад 97%.

888poker позиціонується як соціально-орієнтована платформа з акцентом на візуальну привабливість та інтерактивність. Технологічний стек побудований на Node.js з використанням Socket.io для real-time комунікації та MongoDB для зберігання профілів користувачів і соціальних даних. Платформа інтегрована з провідними соціальними мережами та підтримує персоналізовані аватари з використанням технологій машинного навчання для створення унікальних ігрових персонажів.

Ключові інновації:

- 3D-движок на базі Three.js для photorealistic рендерингу ігрового столу;
- Інтеграція відеочату через WebRTC із серверами TURN/STUN;
- Amazon Rekognition для автоматичної модерації контенту;
- "Snap Cam" функціональність для створення відео-реакцій;

partypoker фокусується на забезпеченні максимальної надійності та низької латентності. Архітектура побудована на C++ серверах з custom TCP протоколом для мінімізації overhead.

Технічні особливості:

- Lock-free алгоритми для обробки ігрових подій;
- Memory-mapped files для швидкого доступу до турнірних даних;
- Підтримка понад 50 варіантів покеру;
- Proprietary compression algorithm для стиснення трафіку до 15-20%;

GGPoker представляє найсучаснішу mobile-first архітектуру з використанням Flutter для cross-platform development. Бекенд реалізований на Go з gRPC для high-performance API calls та NATS Streaming для event sourcing.

Інноваційні рішення:

- "Smart HUD" система з Apache Spark для big data processing;
- Аналіз понад 500 параметрів гри кожного користувача;
- Інтеграція з poker trackers через REST API з OAuth 2.0;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		10

Таблиця 1.1 Порівняльні характеристики технічної реалізації кожної платформи

Платформа	Архітектура backend	Протокол зв'язку	Database	Механізм захисту
PokerStars	Java EE та Kafka microservices	WebSocket та compression	PostgreSQL Master-Slave	ML anti-fraud (RF та SVM)
888poker	Node.js та Socket.io	WebRTC та WebSocket	MongoDB та Redis	Amazon Rekognition та manual moderation
partypoker	C++ native servers	Custom TCP protocol	Memory-mapped files та PostgreSQL	Proprietary pattern detection
GGPoker	Go та gRPC та NATS	HTTP/2 та Server Push	Apache Spark та Cassandra	Behavioral analytics та OAuth 2.0

Архітектурний аналіз демонструє три основних підходи до масштабування онлайн-покерних систем:

- Монолітно-мікросервісний гібрид (PokerStars): core game engine залишається монолітним для мінімізації латентності, тоді як допоміжні сервіси винесені в окремі мікросервіси;
- Event-driven архітектура (GGPoker): вся система побудована навколо event sourcing pattern, де кожна дія гравця записується як immutable event;
- Hybrid rendering (888poker): поєднання server-side рендерингу для критичних ігрових елементів та client-side для UI/UX покращень;

1.1.2 Вивчення архітектури веб-додатків для карткових ігор

Сучасні веб-додатки для карткових ігор базуються на складній багаторівневій архітектурі, що повинна забезпечувати одночасну обробку тисяч ігрових сесій при мінімальній латентності та гарантованій консистентності даних. Архітектурні рішення в цій галузі еволюціонували від простих client-server моделей до розподілених систем з мікросервісною архітектурою та event-driven підходами.

Клієнт-серверна архітектура залишається фундаментальною основою для більшості карткових ігор, проте її реалізація суттєво ускладнилася

завдяки вимогам до real-time синхронізації. Для техаського покеру особливо критичним є забезпечення справедливості гри та синхронізація стану між усіма гравцями за столом.

Управління станом гри (State Management) є критичним аспектом архітектури. Найефективнішим виявився підхід з authoritative server architecture, де сервер є єдиним джерелом істини (single source of truth), а клієнти отримують лише дозволені представлення (views) поточного стану. Це забезпечує захист від client-side модифікацій та синхронізацію між усіма учасниками.

Мікросервісна декомпозиція:

Мікросервісна архітектура дозволяє розділити систему на логічні компоненти з незалежним життєвим циклом та масштабуванням:

- Authentication Service: управління користувачами, JWT токени, управління сесіями;
- Game Engine Service: основна ігрова логіка, правила, валідація ходів;
- Matchmaking Service: формування ігрових столів, балансування рівня гравців;
- Payment Service: обробка транзакцій, віртуальна валюта, операції виведення коштів;
- Analytics Service: збір метрик, виявлення шахрайства, аналіз поведінки;

Міжсервісна комунікація реалізується через асинхронні повідомлення (Apache Kafka, RabbitMQ) для event-driven взаємодії та синхронні API виклики (gRPC, REST) для request-response операцій. Такий підхід забезпечує горизонтальне масштабування та відмовостійкість.

Патерн Event Sourcing набуває особливої актуальності для карткових ігор, оскільки дозволяє:

- Повне відтворення ігрової історії для аудиту;
- Операції відкату у випадку системних збоїв;
- Real-time streaming подій для режиму спостереження;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		12

— Відповідність регуляторним вимогам щодо чесної гри;

Кожна ігрова дія записується як незмінна подія (immutable event) з timestamp, player_id, action_type та payload:

```
{  
  "event_id": "uuid",  
  "game_id": "game_123",  
  "player_id": "player_456",  
  "action": "BET",  
  "amount": 100,  
  "timestamp": "2024-01-15T10:30:00Z",  
  "sequence": 15  
}
```

Архітектура бази даних:

Карткові ігри потребують поєднання різних типів сховищ даних:

- RDBMS (PostgreSQL): структуровані дані користувачів, баланси, транзакції;
- In-Memory Cache (Redis): активні ігрові сесії, real-time стани столів;
- Document Store (MongoDB): налаштування користувачів, ігрові переваги;
- Time-Series DB (InfluxDB): метрики продуктивності, аналітика, моніторинг;

Консистентність даних забезпечується через протокол Two-Phase Commit для критичних операцій (ставки, виграші).

Архітектура безпеки:

Багаторівневий захист включає:

- Transport Layer: TLS 1.3 з certificate pinning;
- Application Layer: JWT токени з refresh механізмом, обмеження частоти запитів;
- Business Logic: server-side валідація всіх ігрових дій;
- Data Layer: шифрування в спокої (AES-256), шифрування на рівні полів для чутливих даних;

					РП 08. 20 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		13

Протоколи оптимізовані для мінімізації затримки:

- WebSocket для real-time ігрової комунікації з binary форматом повідомлень;
- HTTP/2 з Server Push для доставки асинхронних подій;
- UDP для некритичних повідомлень (чат, анімації);

Cloud-Native архітектура:

Сучасні покерні платформи масово мігрують до cloud-native архітектури з використанням Kubernetes для оркестрації мікросервісів. Docker контейнери забезпечують консистентне середовище розгортання та спрощені операції масштабування.

AWS/Azure managed сервіси активно використовуються:

- Managed databases: Amazon RDS для PostgreSQL з автоматичним backup та failover;
- Message queues: AWS SQS/Amazon MQ для асинхронної обробки подій;
- CDN: CloudFlare для швидкої доставки статичних ресурсів;
- Auto-scaling: горизонтальне автомасштабування на основі CPU/memory метрик;

Архітектура веб-додатків для карткових ігор є складною екосистемою, що поєднує різноманітні технології та підходи. Успішна реалізація вимагає врахування специфічних вимог до real-time синхронізації, безпеки, масштабованості та відмовостійкості.

1.1.3 Аналіз технологій real-time взаємодії

Технології real-time взаємодії є критичною компонентою сучасних онлайн-покерних платформ, оскільки забезпечують синхронну взаємодію між гравцями та миттєву реакцію системи на ігрові дії. Ефективність цих технологій безпосередньо впливає на користувацький досвід та конкурентоспроможність платформи на ринку онлайн-гемблінгу.

Дослідження провідних покерних платформ, таких як PokerStars, 888poker, partypoker та GG Poker, показує, що всі вони використовують комбінацію різних

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		14

протоколів для забезпечення безперебійної роботи та мінімізації затримок у передачі даних. Детальний порівняльний аналіз цих платформ представлено в (Таблиці 1.3.)

WebSocket протокол є де-факто стандартом для двостороннього real-time зв'язку між клієнтом та сервером у веб-додатках карткових ігор. На відміну від традиційного HTTP request-response циклу, WebSocket створює persistent connection, що дозволяє серверу ініціювати передачу даних до клієнта без попереднього запиту.

Ключові переваги WebSocket для покерних додатків включають:

- Мінімальну латентність (5-15ms для локальних з'єднань);
- Низький overhead - лише 2-4 байти per frame замість HTTP headers;
- Підтримку binary та text форматів повідомлень;
- Automatic reconnection mechanisms з exponential backoff;

Сучасні покерні платформи оптимізують WebSocket з'єднання через message batching - групування декількох ігрових подій в один frame для зменшення network roundtrips. Наприклад, одночасна передача інформації про ставку гравця, оновлення pot size та timer countdown об'єднується в єдине повідомлення.

Приклад структури WebSocket повідомлення в покері:

```
{  
  "type": "game_update",  
  "data": {  
    "action": "player_bet",  
    "player_id": "user123",  
    "amount": 50,  
    "pot_total": 275,  
    "timer": 28,  
    "timestamp": 1672531200000  
  }  
}
```

Альтернативні протоколи та fallback механізми:

					РП 08. 20 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		15

Server-Sent Events (SSE) використовуються як fallback механізм для клієнтів, що не підтримують WebSocket, або у випадках обмежень корпоративних firewall. SSE забезпечує односторонню комунікацію від сервера до клієнта через стандартні HTTP з'єднання, що дозволяє підтримувати basic real-time функціональність.

Polling mechanisms залишаються актуальними для некритичних компонентів:

- Long polling для lobby updates та tournament notifications;
- Short polling для статистичних даних та leaderboard;
- Adaptive polling з dynamic intervals залежно від активності користувача;

Бінарні протоколи набувають популярності для high-frequency ігрових подій завдяки компактності та швидкості обробки. Як показано в таблиці 1.2, MessagePack та Protocol Buffers дозволяють зменшити розмір повідомлень на 40-60% порівняно з JSON, що критично важливо для mobile connections. Зокрема, з таблиці 1.2 видно, що MessagePack забезпечує на 45% швидшу передачу даних при підтримці 95% браузерів з поліфілом, тоді як Protocol Buffers досягають на 60% швидшої передачі при 90% підтримці браузерів.

Таблиця 1.2 Порівняльний аналіз протоколів передачі даних для real-time

Формат	Розмір (байт)	Швидкість парсингу	Підтримка браузерів
JSON	156	Базова	100%
MessagePack	98	45% швидше	95%
Protocol Buffers	87	60% швидше	90%

React та Vue.js домінують у розробці інтерфейсів покерних додатків завдяки component-based архітектурі та ефективному state management. React Hook система дозволяє елегантно обробляти real-time оновлення через useEffect та custom hooks для WebSocket з'єднань.

Angular зберігає позиції у enterprise-рішеннях завдяки TypeScript інтеграції та robust dependency injection system. RxJS потужно підтримує reactive programming patterns, що ідеально підходить для обробки потоків ігрових подій.

Progressive Web App (PWA) технології:

- Service Workers для offline caching ігрових assets;
- Web App Manifest для home screen installation;
- Push Notifications для турнірних alerts;
- Background Sync для збереження ігрових дій при поганому з'єднанні;

Графічні оптимізації:

Canvas та WebGL застосовуються для анімації карт та фішок, забезпечуючи smooth 60fps experience. Hardware acceleration через CSS transforms (translate3d, scale3d) мінімізує browser reflow та repaint операції.

Таблиця 1.3. Порівняльний аналіз популярних покерних платформ

Платформа	Fallback	Mobile Support	Offline Mode
PokerStars	SSE та Polling	PWA	Обмежений
888poker	Long Polling	Native App	Відсутній
GGPoker	SSE	Hybrid	Частковий
partypoker	Short Polling	PWA та Native	Відсутній

WCAG 2.1 compliance забезпечується через:

- Semantic HTML структуру для screen readers;
- Keyboard navigation support для всіх ігрових дій;
- High contrast mode для користувачів з порушеннями зору;
- Voice commands integration через Web Speech API;

Порівняльний аналіз платформ:

Аналіз популярних покерних платформ, представлений в таблиці, демонструє різноманітність підходів до реалізації real-time функціональності. З таблиці бачимо, що PokerStars використовує комбінацію SSE + Polling з PWA підтримкою, проте має обмежений offline режим. 888poker застосовує Long Polling з нативним додатком та повністю відсутнім offline режимом. GGPoker обирає SSE з гібридним підходом до мобільної підтримки та частковим offline режимом, тоді як partypoker використовує Short Polling з комбінацією PWA + Native підходу та відсутнім offline режимом.

Управління станом у real-time додатках:

Optimistic updates покращують perceived performance - клієнт миттєво відображає результат дії користувача, а згодом синхронізується з authoritative server state. У випадку конфлікту відбувається rollback до серверного стану.

Безпека та цілісність даних:

Content Security Policy (CSP) запобігає XSS атакам через обмеження джерел JavaScript та inline scripts. Subresource Integrity забезпечує цілісність third-party бібліотек, що критично важливо для фінансових операцій у покерних додатках.

1.1.4 Дослідження технологій збереження стану гри та синхронізації

Збереження стану гри та забезпечення синхронізації між численними клієнтами — одне з ключових технічних завдань при розробці онлайн-платформ для покеру. Від ефективності реалізації цих механізмів залежить справедливість гри, стабільність під навантаженням та якість користувацького досвіду.

Помилки в синхронізації можуть призвести до спотворення результатів гри або навіть втрати даних про ставки, що неприпустимо в системах з фінансовою відповідальністю. Саме тому побудова надійної архітектури фундаментальний етап у створенні покерного движка.

У реальних умовах навантаження ключовим показником стає здатність системи обробляти тисячі подій на секунду без втрат і затримок.

Особлива складність у техаському покері зумовлена необхідністю координації багатьох динамічних процесів:

- роздача карт у суворій послідовності;
- миттєва обробка ставок та відображення для всіх гравців;
- синхронізація таймерів дій з точністю до секунди.

Це вимагає як ефективних мережевих протоколів, так і оптимізованої внутрішньої логіки оновлення стану.

Управління станом (state management) у таких іграх охоплює:

- поточні карти гравців та стан колоди;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		18

- розмір банку та історія ставок;
- позиції гравців за столом;
- фази гри (pre-flop, flop, turn, river);
- таймери дій та статуси гравців;

Архітектурні патерни для багатокористувацьких ігор:

Одним з найбільш ефективних підходів є Authoritative Server Pattern, який передбачає централізацію всієї бізнес-логіки на сервері, що виступає єдиним джерелом істини (Single Source of Truth). Як показано на рисунку, всі ігрові дії валідуються та застосовуються виключно на сервері, який визначає їх допустимість відповідно до поточного стану гри (Рисунок 1.2). Клієнти отримують лише контрольоване представлення стану через API.

У такій архітектурі:

- логіка гри реалізується виключно на сервері;
- база даних зберігає актуальний стан;
- API-ендпоінти обмежують доступ клієнтів до критичних дій;
- автентифікація й авторизація гарантують безпечне виконання операцій.

На схемі видно, що бізнес-логіка централізована на сервері та взаємодіє з базою даних для збереження стану, в той час як клієнти мають доступ лише через контрольовані API endpoints. Система автентифікації та авторизації забезпечує додатковий рівень безпеки для всіх операцій.

Цей підхід особливо важливий для ігор з жорсткими вимогами до послідовності дій. Наприклад, дія "підвищити ставку" повинна перевірятися з урахуванням мінімального розміру ставки, балансу гравця та поточної фази гри.

Для сповіщення зацікавлених компонентів (UI, аналітика, логування) про зміни стану гри ефективно застосовується Observer Pattern. Цей патерн забезпечує підписку на події без прямого зв'язку з ігровою логікою, сприяючи слабкому зв'язуванню (loose coupling) та спрощенню розширення функціональності.

Крім того, він дозволяє динамічно підключати нові модулі, не порушуючи основної логіки гри.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		19

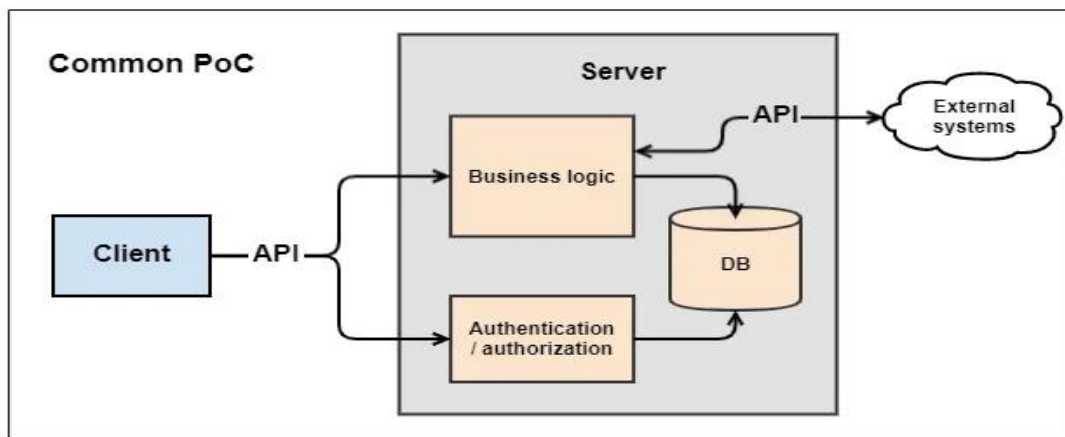


Рисунок 1.2 Діаграма клієнт-серверної архітектури

Таблиця 1.4 Порівняння основних підходів до синхронізації стану в додатках

Технологія	Переваги	Недоліки	Використання
WebSockets	Низька латентність, двосторонній зв'язок	Складність масштабування	Real-time оновлення стану
Server-Sent Events	Простота реалізації, автоматичне перепідключення	Односторонній зв'язок	Оновлення UI
Polling	Простота, сумісність з усіма браузерами	Високе навантаження	Резервний механізм
WebRTC	Найнижча латентність, P2P можливості	Складність NAT traversal	Voice chat, відеопотоки

Додаткові підходи:

— Event Sourcing — зберігає кожну дію як незмінну подію (immutable event) з timestamp та унікальним ідентифікатором. Це дозволяє відтворювати будь-який стан гри, забезпечує повну аудитність та спрощує вирішення спірних ситуацій.

— Distributed Consensus Algorithms (Raft, PBFT) — гарантують узгодженість стану між декількома серверними інстансами, що критично для систем із високою доступністю.

1.2 Розробка алгоритмів ігрової логіки техаського покеру

Розробка алгоритмів ігрової логіки техаського покеру є найбільш критичним і технічно складним компонентом веб-додатку, що визначає якість, справедливість та функціональну повноту ігрового процесу. Цей розділ охоплює створення комплексної системи алгоритмів, здатної забезпечити повноцінну реалізацію всіх аспектів покерної гри: від базових операцій з картами до складних механізмів визначення переможців у багатогравцевих сценаріях.

Основною метою розробки ігрової логіки є створення надійної, масштабованої та захищеної від маніпуляцій системи, яка здатна обробляти всі можливі ігрові сценарії техаського покеру з урахуванням специфічних правил, винятків та крайових випадків. Система повинна забезпечувати математично точне обчислення ймовірностей, справедливу роздачу карт з використанням криптографічно стійких генераторів випадкових чисел, а також коректну обробку всіх типів ставок та ігрових дій.

Архітектурно ігрова логіка побудована за принципом модульності, де кожен компонент виконує специфічну функцію та може бути незалежно тестований і модифікований. Це включає:

Модуль управління колодою та роздачі карт - забезпечує створення стандартної колоди з 52 карт, її перемішування з використанням алгоритму Fisher-Yates та гарантовану унікальність роздаваних карт. Модуль включає механізми валідації цілісності колоди та відновлення після помилок.

Модуль оцінки покерних комбінацій - реалізує повний набір алгоритмів для розпізнавання та ранжування всіх можливих покерних комбінацій від старшої карти до роял-флеша. Система використовує оптимізовані алгоритми порівняння з часовою складністю $O(1)$ для більшості операцій.

Модуль системи ставок - обробляє всі типи ігрових дій включаючи блайнди, call, raise, fold, з урахуванням поточного стану банку, мінімальних ставок та обмежень гравців.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		21

1.2.1 Проектування архітектури системи та структури бази даних

Архітектура веб-додатку для техаського покеру побудована на основі тришарової моделі client-server з використанням REST API та WebSocket для real-time комунікації. Система розроблена з урахуванням принципів мікросервісної архітектури, що забезпечує гнучкість, масштабованість та можливість незалежного розгортання компонентів (Рисунок 1.3).

Архітектурна схема системи:

Загальна архітектура системи представлена трирівневою структурою, де кожен рівень має чітко визначені обов'язки та інтерфейси взаємодії:



Рисунок 1.3 Трьохрівнева архітектура веб-додатка

Клієнтський рівень (React Client) забезпечує інтерактивний користувацький інтерфейс з наступними компонентами:

- Ігровий інтерфейс з відображенням покерного столу, карт та фішок;
- WebSocket хуки для real-time оновлень стану гри;
- State management через Redux для централізованого управління станом;
- Optimistic updates для покращення відгуку інтерфейсу;

Серверний рівень (Node.js Server) реалізує бізнес-логіку та забезпечує:

- WebSocket сервер для синхронізації ігрових подій в реальному часі;
- Ігрову логіку з валідацією всіх дій на сервері;
- Систему аутентифікації з JWT токенами;
- Message routing для правильного розподілення повідомлень між гравцями;

— API endpoints для HTTP запитів (реєстрація, логін, статистика);

Рівень даних (MySQL Database) забезпечує надійне зберігання профілів користувачів з повною статистикою.

Структура бази даних розроблена з урахуванням принципів нормалізації до третьої нормальної форми з селективною денормалізацією для оптимізації продуктивності критичних запитів (Рисунок 1.4). База даних включає три основні сутності з чітко визначеними зв'язками та обмеженнями цілісності.

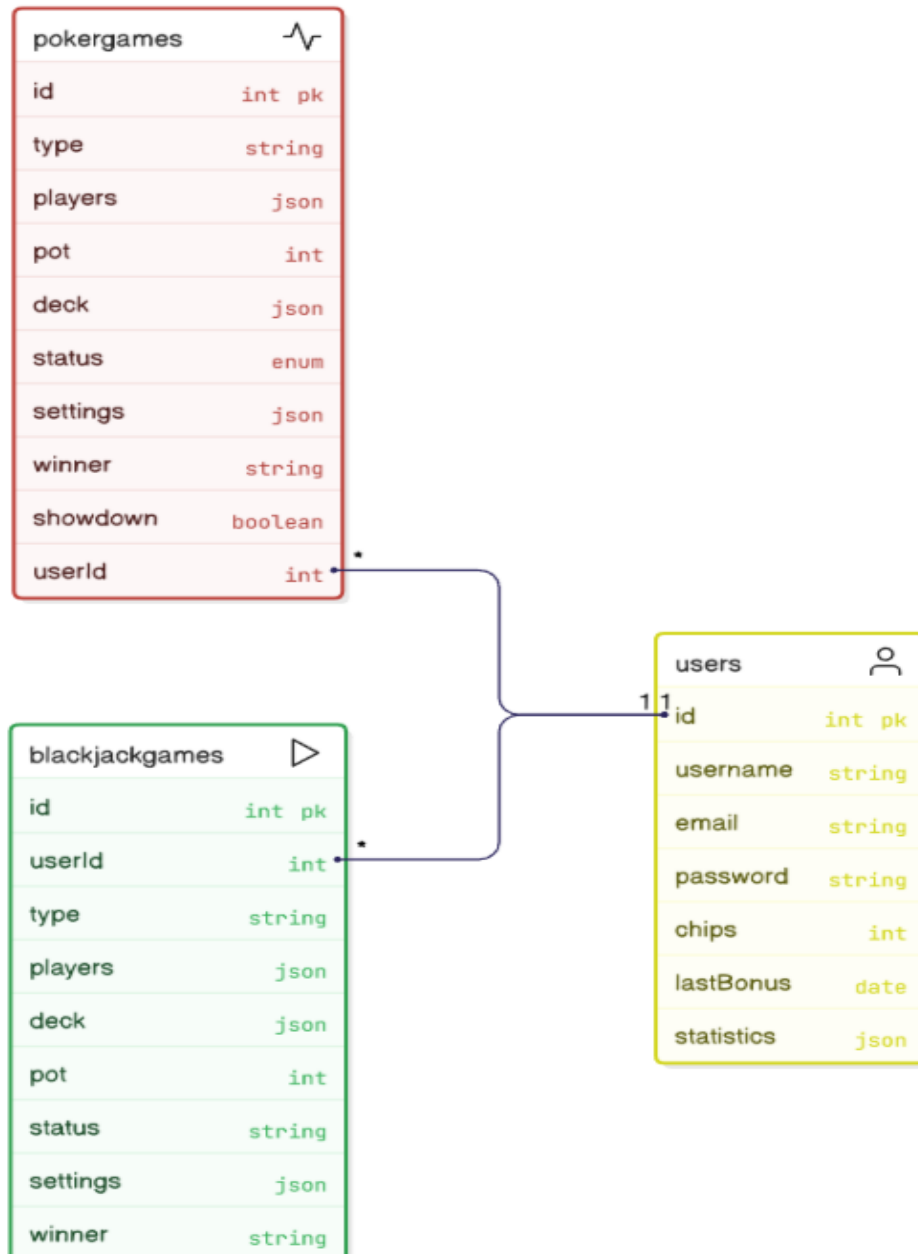


Рисунок 1.4 Схема бази даних (структура таблиць pokergames, blackjackgames, users)"

Сутність "User" є центральною в системі та містить наступні атрибути:

- id (INTEGER, PRIMARY KEY) - унікальний ідентифікатор користувача;
- username (STRING, UNIQUE) - унікальне ім'я користувача з валідацією довжини 3-20 символів;
- email (STRING, UNIQUE) - електронна пошта з валідацією формату та автоматичним приведенням до нижнього регістру;
- password (STRING) - хешований пароль з використанням bcrypt та salt factor 10;
- chips (INTEGER) - кількість ігрових фішок з мінімальним значенням 0 та початковим балансом 1000;
- lastBonus (DATE) - час останнього отримання бонусу для реалізації daily bonus системи;
- statistics (JSON) - статистика гравця, що включає кількість зіграних та виграних ігор, загальний прибуток;

Сутність "PokerGame" зберігає повний стан покерної гри та включає:

- id (INTEGER, PRIMARY KEY) - унікальний ідентифікатор гри;
- type (STRING) - тип гри з значенням за замовчуванням 'poker';
- players (JSON) - масив гравців з повною інформацією про їх стан у грі;
- pot (INTEGER) - розмір поточного банку;
- deck (JSON) - стан колоди карт з перевіркою унікальності;
- status (ENUM) - статус гри: 'waiting', 'playing', 'finished', 'eliminated', 'replaced';
- settings (JSON) - налаштування гри включаючи позиції гравців, поточний раунд, спільні карти;
- winner (STRING) - ім'я переможця або переможців;
- showdown (BOOLEAN) - індикатор того, чи дійшла гра до розкриття карт;
- userId (INTEGER, FOREIGN KEY) - посилання на користувача-власника гри;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		24

База даних реалізована в MySQL з використанням ORM Sequelize для Node.js, що забезпечує type-safe операції та автоматичну міграцію схеми. Використання JSON полів дозволяє зберігати складні структури даних (масиви гравців, налаштування гри) без необхідності створення додаткових таблиць, що спрощує архітектуру та покращує продуктивність запитів.

Особливу увагу приділено механізмам валідації даних на рівні моделі. Для забезпечення цілісності ігрових даних реалізовано hooks на рівні Sequelize, що автоматично перевіряють унікальність карт перед збереженням гри. Це критично важливо для запобігання дублікатам карт, які можуть виникнути через race conditions у багатокористувацькому середовищі.

Система індексації оптимізована для частих запитів: створено індекси на поля username, email у таблиці користувачів та status, userId у таблиці ігор. Це забезпечує швидкий пошук активних ігор та профілів користувачів навіть при значному навантаженні на систему.

Архітектурно система підтримує горизонтальне масштабування через connection pooling з налаштуваннями max: 5, min: 0 connections, що забезпечує оптимальне використання ресурсів бази даних. Підключення до хмарного сервісу Aiven.io реалізовано з SSL шифруванням та автоматичним reconnect механізмом для забезпечення надійності в production середовищі.

Розроблена структура бази даних забезпечує ефективне зберігання ігрових станів, швидкий доступ до даних користувачів та можливість розширення функціональності додатковими типами карточкових ігор без кардинальних змін у схемі даних.

1.2.2 Алгоритм роздачі карт та управління ігровими раундами

Ефективна реалізація алгоритму роздачі карт та управління ігровими раундами є фундаментальним елементом системи техаського покеру, що забезпечує справедливість гри та правильну послідовність ігрових подій. Розроблений алгоритм гарантує унікальність карт, криптографічну стійкість перемішування та точне дотримання правил техаського покеру щодо послідовності раундів.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		25

Процес створення колоди реалізується через генерацію повного набору з 52 карт, що складається з чотирьох мастей (піки, черви, бубни, трефи) та тринадцяти значень (2-10, J, Q, K, A).

Для забезпечення криптографічної стійкості кожна карта отримує унікальний ідентифікатор, що включає значення, масть, часову мітку та криптографічний хеш. Перемішування колоди здійснюється за допомогою модифікованого алгоритму Фішера-Йейтса з використанням криптографічно безпечного генератора випадкових чисел.

Алгоритм Фішера-Йейтса з криптографічним посиленням:

Модифікований алгоритм Фішера-Йейтса забезпечує рівномірний розподіл ймовірностей для всіх можливих перестановок колоди. Формула алгоритму:

$$P(i, j) = 1/(i + 1) \quad (1)$$

де:

$P(i, j)$ - ймовірність обміну елемента i з елементом j

n - розмір колоди

i - поточна позиція (від $n-1$ до 0)

Криптографічне посилення досягається через використання `crypto.randomBytes()` замість стандартного `Math.random()`, що забезпечує непередбачуваність послідовності:

```
function shuffleDeck(deck) {
  const crypto = require('crypto');
  for (let i = deck.length - 1; i > 0; i--) {
    const randomBytes = crypto.randomBytes(4);
    const j = randomBytes.readUInt32BE(0) % (i + 1);
    [deck[i], deck[j]] = [deck[j], deck[i]];
  }
  return deck;
}
```

Алгоритм роздачі карт гравцям:

Роздача карт виконується послідовно для кожного активного гравця з верхньої частини перемішаної колоди. Система виконує багаторівневу

					РП 08. 20 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		26

валідацію унікальності карт через структуру даних Set, що забезпечує $O(1)$ складність для операцій перевірки дублікатів.



Рисунок 1.5 Блок-схема для валідації унікальності карт та перемішування алгоритмом Фішера-Йейтса

Зм.	Арк.	№ докум.	Підпис	Дата

Кожному гравцю видається рівно дві карти відповідно до правил техаського покеру. Критично важливим є механізм перевірки достатності карт у колоді перед кожною роздачею та валідація унікальності через спеціалізовану функцію `validateGameCards()`.

Процес валідації унікальності карт:

Система валідації реалізована через комплексну перевірку всіх карт у грі за допомогою множини (Set), що гарантує відсутність дублікатів. Алгоритм включає наступні етапи:

- Збір всіх карт: Формування єдиного масиву з карт гравців та общих карт;
- Створення унікальних ключів: Генерація ідентифікаторів у форматі "значення-масть";
- Перевірка дублікатів: Використання Set для виявлення повторюваних карт;
- Звітування: Повернення детального звіту про стан валідації;

Система техаського покеру включає п'ять основних фаз з чітко визначеною послідовністю переходів:

Таблиця 1.5. Фази гри та їх характеристики

Фаза	Опис	Кількість спільних карт	Дії гравців
Preflop	Початкова фаза після роздачі карт гравцям	0	Торгівля на основі власних карт
Flop	Розкриття перших трьох общих карт	3	Торгівля з урахуванням флопу
Turn	Додавання четвертої общей карти	4	Торгівля після терну
River	Розкриття п'ятої общей карти	5	Фінальна торгівля
Showdown	Визначення переможця	5	Порівняння комбінацій

Перехід між раундами контролюється функцією nextRound(), що реалізує строго послідовність дій з валідацією умов переходу:

- Preflop → Flop: Роздача трьох общих карт за умови відсутності попереднього флопу;
- Flop → Turn: Додавання четвертої карти при наявності рівно трьох общих карт;
- Turn → River: Роздача п'ятої карти за умови наявності чотирьох общих карт;
- River → Showdown: Ініціація визначення переможця без додаткових карт;

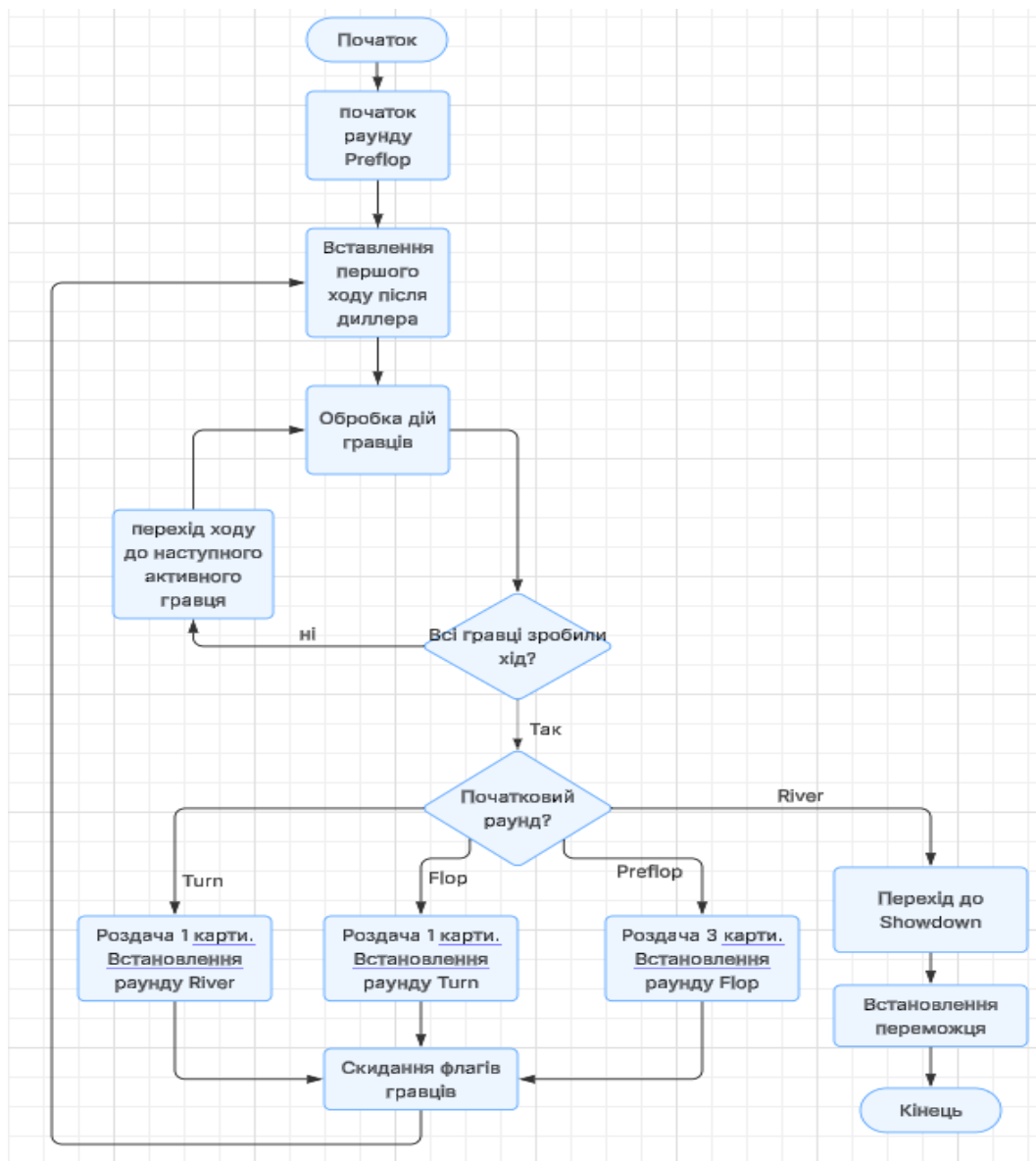


Рисунок 1.6 Блок-схема алгоритму роздачі карт та управління раундами

Зм.	Арк.	№ докум.	Підпис	Дата

Кожен перехід супроводжується скиданням прапорців `hasActed` для всіх активних гравців та встановленням чергового ходу першому гравцю після позиції дилера.

Логіка переходу ходів між гравцями:

Функція `nextTurn()` реалізує складну логіку визначення завершення раунду торгівлі на основі аналізу стану всіх активних гравців. Раунд вважається завершеним за умови виконання наступних критеріїв:

- Універсальність дій: Всі активні гравці виконали принаймні одну дію (`hasActed = true`);
- Рівність ставок: Поточні ставки всіх гравців (`currentBet`) дорівнюють максимальній ставці раунду;
- Активність гравців: Принаймні два гравці залишаються в грі (не склали карти);

Алгоритм виконує циклічний пошук наступного активного гравця, автоматично пропускаючи тих, хто склав карти (`folded = true`). При залишенні лише одного активного гравця система автоматично переходить до визначення переможця з присудженням банку без розкриття карт.

Таблиця 1.6 Характеристики ефективності алгоритмів

Операція	Алгоритм	Часова складність	Просторова складність
Створення колоди	Генерація 52 карт	$O(52) = O(1)$	$O(52) = O(1)$
Перемішування	Фішера-Йейтса	$O(n)$	$O(1)$
Валідація карт	Set lookup	$O(k)$, де $k \leq 23$ карти	$O(k)$
Роздача карт	Послідовний <code>pop()</code>	$O(1)$ за карту	$O(1)$
Перехід раунду	Аналіз стану гравців	$O(m)$, де m - кількість гравців	$O(1)$

1.2.3 Розробка алгоритму визначення покерних комбінацій та переможення

Алгоритм визначення покерних комбінацій та переможення є критично важливим компонентом системи тexasького покеру, що забезпечує справедливість та коректність ігрового процесу. Згідно з офіційними правилами Texas Hold'em, існує десять базових типів покерних комбінацій, що мають чітку ієрархічну структуру від найслабшої (старша карта) до найсильнішої (роял-флеш). Система повинна обробляти до 100,000 оцінок рук на секунду в режимі реального часу, забезпечуючи мінімальну затримку між діями гравців.

Математичні основи алгоритму базуються на комбінаториці та теорії ймовірностей. При аналізі семи доступних карт (дві кишенькові плюс п'ять спільних) система повинна знайти найкращу комбінацію з п'яти карт. Загальна кількість можливих комбінацій становить $C(7,5) = 21$, що робить повний перебір практично здійсненним навіть при високому навантаженні системи.

Система оцінки покерних рук реалізована через багаторівневий ієрархічний підхід з використанням паттерну "Chain of Responsibility". Кожна комбінація має свій числовий ранг від 1 до 10, що дозволяє швидке порівняння між різними типами рук без складних логічних конструкцій.

Етапи роботи алгоритму:

- Нормалізація вхідних даних: Приведення всіх карт до єдиного числового формату, де J=11, Q=12, K=13, A=14. Мاستі кодуються як: піки=1, черви=2, бубни=3, трефи=4.
- Попередня обробка: Сортування карт за убутанням номіналів для оптимізації подальшого аналізу. Створення допоміжних структур даних для швидкого доступу до інформації про частоту карт та масті;
- Послідовна перевірка комбінацій: Використання алгоритму "waterfall checking" - послідовної перевірки всіх можливих комбінацій від найсильнішої до найслабшої з негайним поверненням результату при знайденні відповідності;
- Формування результуючої структури: Повернення об'єкту з рангом комбінації, назвою, основними картами та кікерами для подальшого порівняння;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		31

Таблиця 1.7 Ієрархія покерних комбінацій у Texas Hold'em

Ранг	Назва комбінації	Опис	Імовірність
10	Роял-флеш	10-J-Q-K-A однієї масті	0.000154%
9	Стрит-флеш	5 карт поспіль однієї масті	0.00139%
8	Карє	4 карти одного рангу	0.0240%
7	Фул-хаус	3+2 карти одного рангу	0.144%
6	Флеш	5 карт однієї масті	0.197%
5	Стрит	5 карт поспіль	0.392%
4	Сет	3 карти одного рангу	2.11%
3	Дві пари	2+2 карти одного рангу	4.75%
2	Пара	2 карти одного рангу	42.3%
1	Старша карта	Жодної комбінації	50.1%

Роял-флеш (ранг 10) перевіряється як частковий випадок стрит-флешу, де найстарша карта є тузом. Стрит-флеш (ранг 9) визначається через комбінацію перевірок на флеш та стрит. Особливий випадок стриту А-5-4-3-2 обробляється окремо, оскільки туз може виступати як найнижча карта.

```
function evaluateHand(cards) {
    const normalizedCards = normalizeCards(cards);
    const sortedCards = sortCards(normalizedCards);
    // Перевірка від найсильнішої комбінації
    if (isRoyalFlush(sortedCards)) return {rank: 10, name: 'Royal Flush'};
    if (isStraightFlush(sortedCards)) return {rank: 9, name: 'Straight Flush'};
    if (isFourOfAKind(sortedCards)) return {rank: 8, name: 'Four of a Kind'};
    // ... інші перевірки
    return {rank: 1, name: 'High Card'};
}
```

Для забезпечення максимальної продуктивності алгоритм використовує техніку мемоізації результатів для часто зустрічних комбінацій карт та битові операції для швидкого визначення мастей. Система також включає механізм валідації вхідних даних з перевіркою на дублікати карт та коректність форматування.

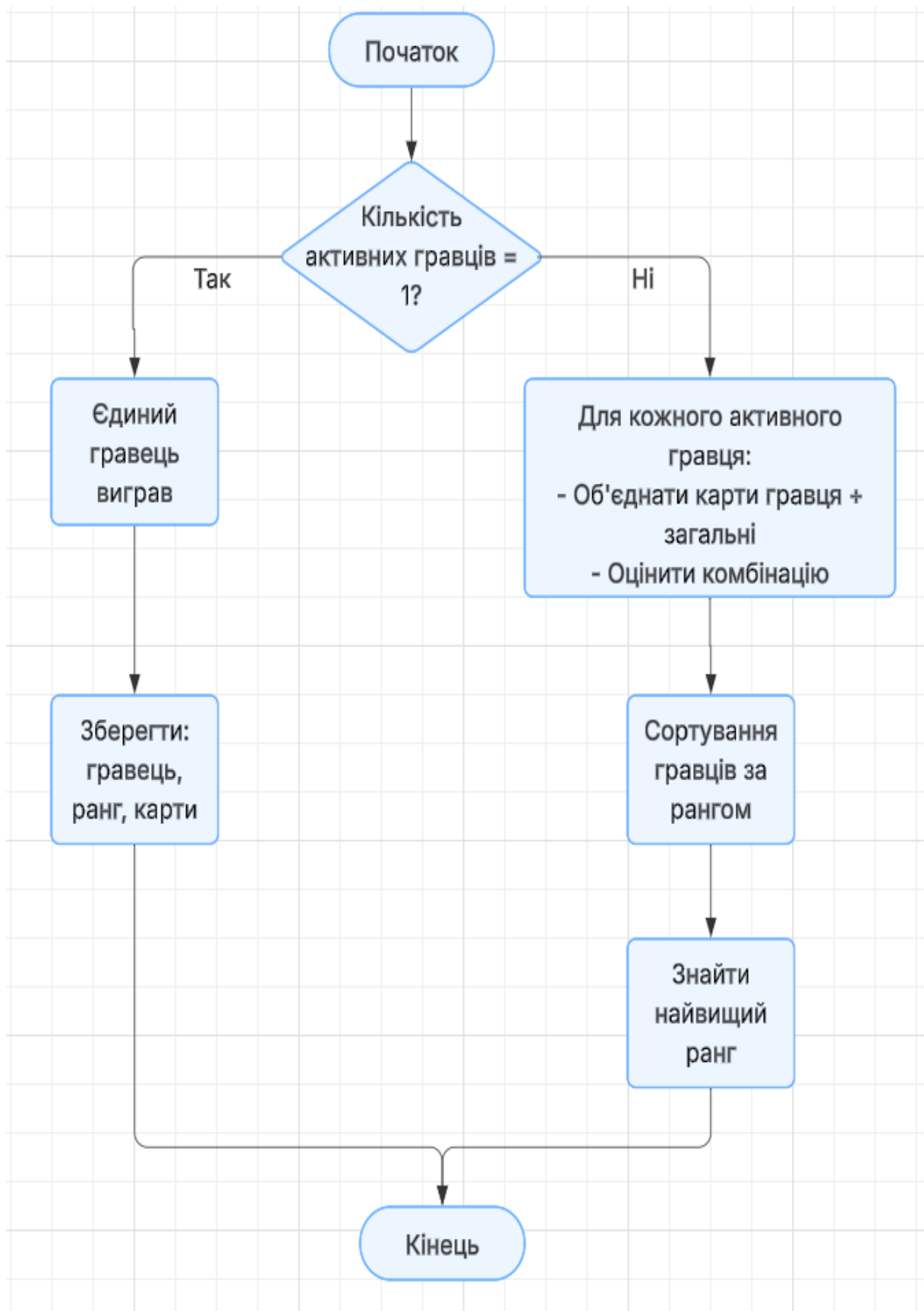


Рисунок 1.7 Блок-схема визначення переможця

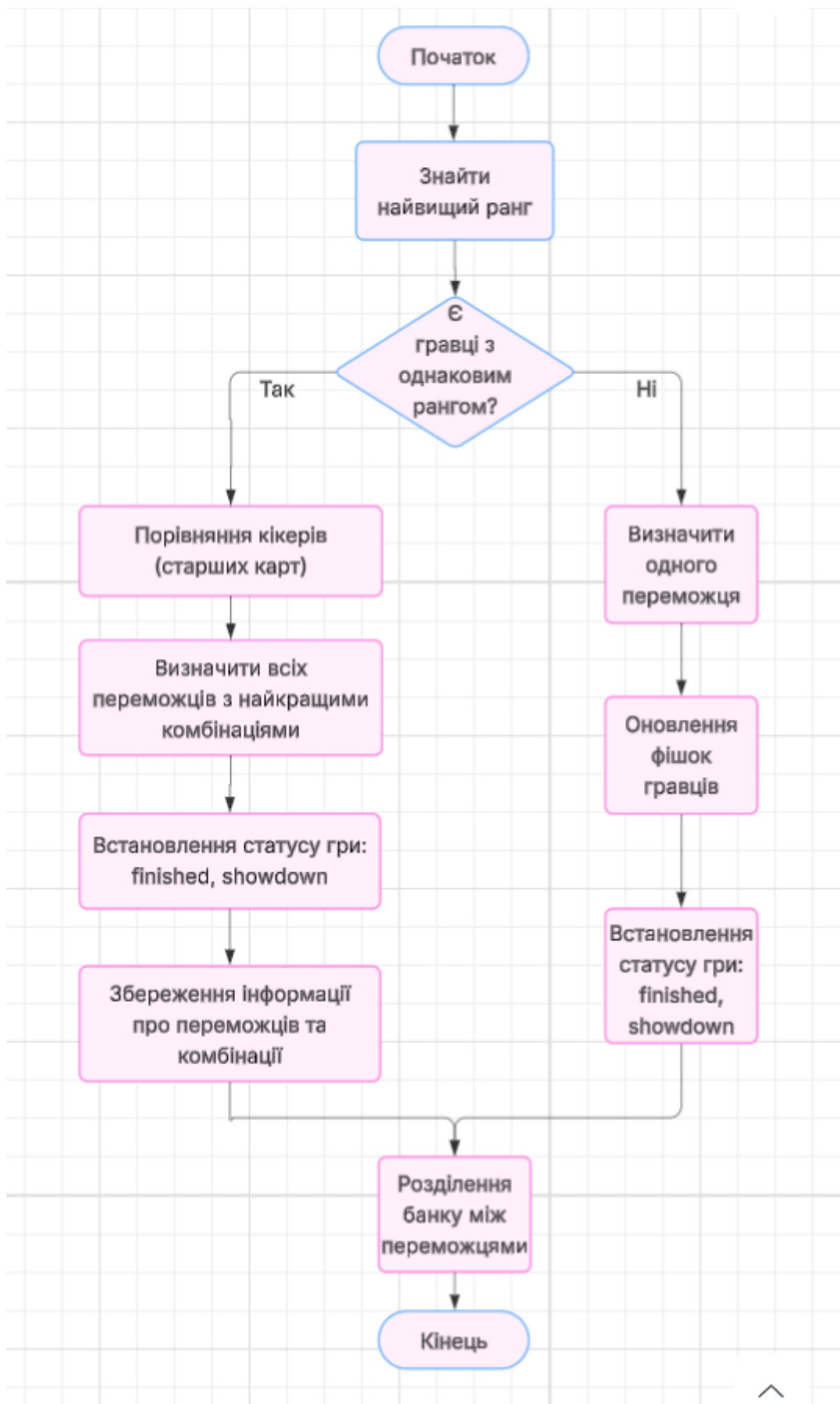


Рисунок 1.8 Схеми алгоритму карткової гри

1.2.4 Реалізація системи ставок: blinds, call, raise, fold

Система ставок є фундаментальним елементом техаського покеру, що забезпечує механіку торгів між гравцями та формування банку для кожної роздачі. Правильна реалізація системи ставок потребує точного дотримання правил позиційної гри, валідації дій гравців та синхронізації стану гри між всіма учасниками партії.

Архітектура системи ставок базується на державному автоматі (state machine), де кожен гравець може знаходитися в одному з наступних станів: активний (active), скинув карти (folded), або очікує ходу (waiting). Перехід між станами регулюється строгими правилами валідації, що забезпечують справедливість гри.

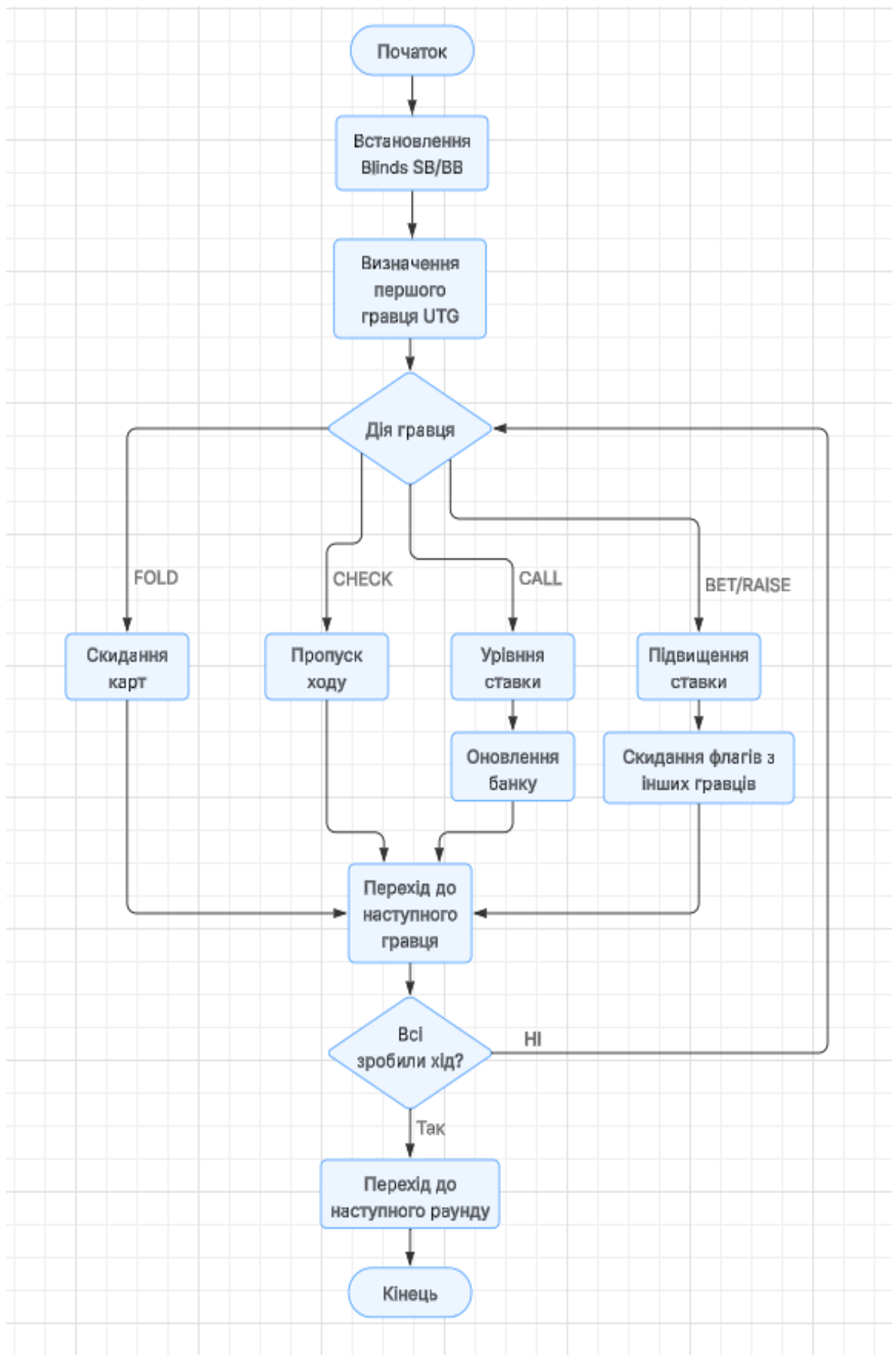
Реалізація системи блайндів:

Основу системи ставок складають обов'язкові ставки - блайнди, які забезпечують початкове наповнення банку та стимулюють активну гру. У техаському покері використовуються два типи блайндів: малий блайнд (Small Blind, SB) та великий блайнд (Big Blind, BB). Розмір великого блайнду зазвичай вдвічі перевищує малий блайнд, що створює базову структуру ставок для партії.

Алгоритм обробки дій гравців представлено на рисунку. (Рисунок 1.9)

FOLD - (Скидання карт) гравець відмовляється від участі в поточній роздачі, втрачаючи всі попередньо внесені ставки. Ця дія не потребує додаткових фішок і завжди доступна незалежно від поточної ситуації за столом. При скиданні карт гравець виключається з активних учасників раунду, але зберігає можливість участі в наступних роздачах.

CALL (Урівнювання ставки) - гравець доплачує різницю між своєю поточною ставкою та найвищою ставкою за столом. Ця дія дозволяє гравцю залишитися в грі без підвищення ставок, просто зрівнявшись з поточним рівнем торгів. Система автоматично розраховує точну суму, необхідну для урівнювання, та перевіряє достатність фішок у гравця. Дія CALL завжди доступна, якщо є активні ставки для урівнювання та у гравця є принаймні один фішок.



Рисунку 1.9 Блок-схема алгоритму реалізації системи ставок

Зм.	Арк.	№ докум.	Підпис	Дата

Особливості реалізації:

- Миттєве виконання без валідації фішок;
- Автоматичне оновлення лічильника активних гравців;

ЧЕКС (Пропуск ходу) - дозволяється лише коли відсутні активні ставки для урівнювання. Гравець передає хід наступному учаснику без внесення додаткових коштів. Дія доступна на пре-флопі тільки для гравця на позиції великого блайнду, якщо не було підвищень.

Умови використання:

- Відсутність активних ставок для урівнювання ($callAmount = 0$);
- Гравець не знаходиться в позиції обов'язкового блайнду з активними рейзами;
- Черга ходу належить поточному гравцю;

Алгоритм обчислення:

```
const callAmount = highestBet - player.currentBet;
```

```
const availableChips = player.chips;
```

```
if (availableChips >= callAmount) {
```

```
    player.chips -= callAmount;
```

```
    player.currentBet = highestBet;
```

```
} else {
```

```
    // Partial call / All-in scenario
```

```
    player.currentBet += availableChips;
```

```
    player.chips = 0;
```

```
    player.isAllIn = true;
```

```
}
```

BET/RAISE (Ставка/Підвищення) - встановлення нової ставки або підвищення існуючої. Мінімальний розмір рейзу дорівнює розміру великого блайнду або розміру попереднього рейзу, що запобігає уповільненню гри мікроставками.

Система валідації рейзів включає перевірку декількох критеріїв: достатність фішок для здійснення мінімальної ставки, відповідність розміру ставки встановленим лімітам та правильність позиційного ходу.

					РП 08. 20 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		37

Правила мінімальних рейзів:

- Перший рейз: мінімум = розмір великого блайнду
- Наступні рейзи: мінімум = розмір попереднього рейзу
- Максимальний рейз обмежений кількістю фішок гравця

1.2.5 Алгоритм роботи ботів та автоматичних дій з таймерами

Система ботів реалізована за принципом асинхронної обробки дій, що дозволяє уникнути блокування основного ігрового циклу. Боти інтегровані в загальну структуру гравців та обробляються тими ж функціями, що й реальні користувачі, забезпечуючи уніфікованість коду. Основні компоненти архітектури ботів представлені в таблиці.

Таблиця 1.7 Компоненти системи ботів

Компонент	Функціональність	Технічна реалізація
Модуль прийняття рішень	Аналіз ігрової ситуації та вибір дії	<code>getBotAction()</code> з стохастичним алгоритмом
Система таймерів	Обмеження часу на хід	<code>setTimeout()</code> з автоматичним <code>fold</code>
Менеджер ботів	Керування життєвим циклом ботів	<code>processBotAction()</code> з асинхронною обробкою
Синхронізатор стану	Узгодження дій між клієнтом та сервером	WebSocket з валідацією на сервері

Алгоритм прийняття рішень ботами: Боти приймають рішення про свої дії на основі ймовірнісного підходу та поточного стану гри.

Алгоритм роботи бота складається з наступних етапів:

1. Визначення черги: Коли настає хід бота, система перевіряє, чи дійсно зараз його черга, чи він ще не зробив хід і не скинув карти.
2. Оцінка ситуації: Бот аналізує поточну ставку (`currentBet`), кількість своїх фішок (`chips`) та необхідну суму для колу (`callAmount`).

3. Генерація випадкового числа: Для імітації різних стилів гри бот генерує випадкове число від 0 до 1, що визначає його дію згідно з ймовірностями.

4. Вибір дії:

- Якщо ставка для колу дорівнює нулю (тобто можна зробити check), бот з ймовірністю 40% обирає check, з ймовірністю 40%
- bet (якщо вистачає фішок), з ймовірністю 15% великий bet, і з ймовірністю 5% — знову check.
- Якщо є ставка для колу, бот з ймовірністю 15% обирає fold, з ймовірністю 50%
- call, з ймовірністю 20%
- raise (якщо вистачає фішок), з ймовірністю 10%
- великий raise, і якщо не вистачає фішок то fold.

5. Виконання дії: Залежно від обраної дії бот:

- Скидає карти (fold);
- Уравнює ставку (call);
- Піднімає ставку (bet/raise);
- Пропускає хід (check);

6. Оновлення стану гри: Після дії бота оновлюється стан гри, зберігаються зміни у базі даних, визначається наступний гравець.

7. Перевірка завершення гри: Якщо залишився лише один активний гравець або завершено всі раунди, гра завершується.

Основні ймовірності дій бота:

- Fold 15%;
- Call 50%;
- Bet/Raise 20% (або 10% для великого raise);
- Check 15% (якщо можливо);

Схема алгоритму ботів з різними стратегіями демонструє повний процес прийняття рішення ботом від аналізу ситуації до виконання конкретної дії (Рисунок 1.10).

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		39

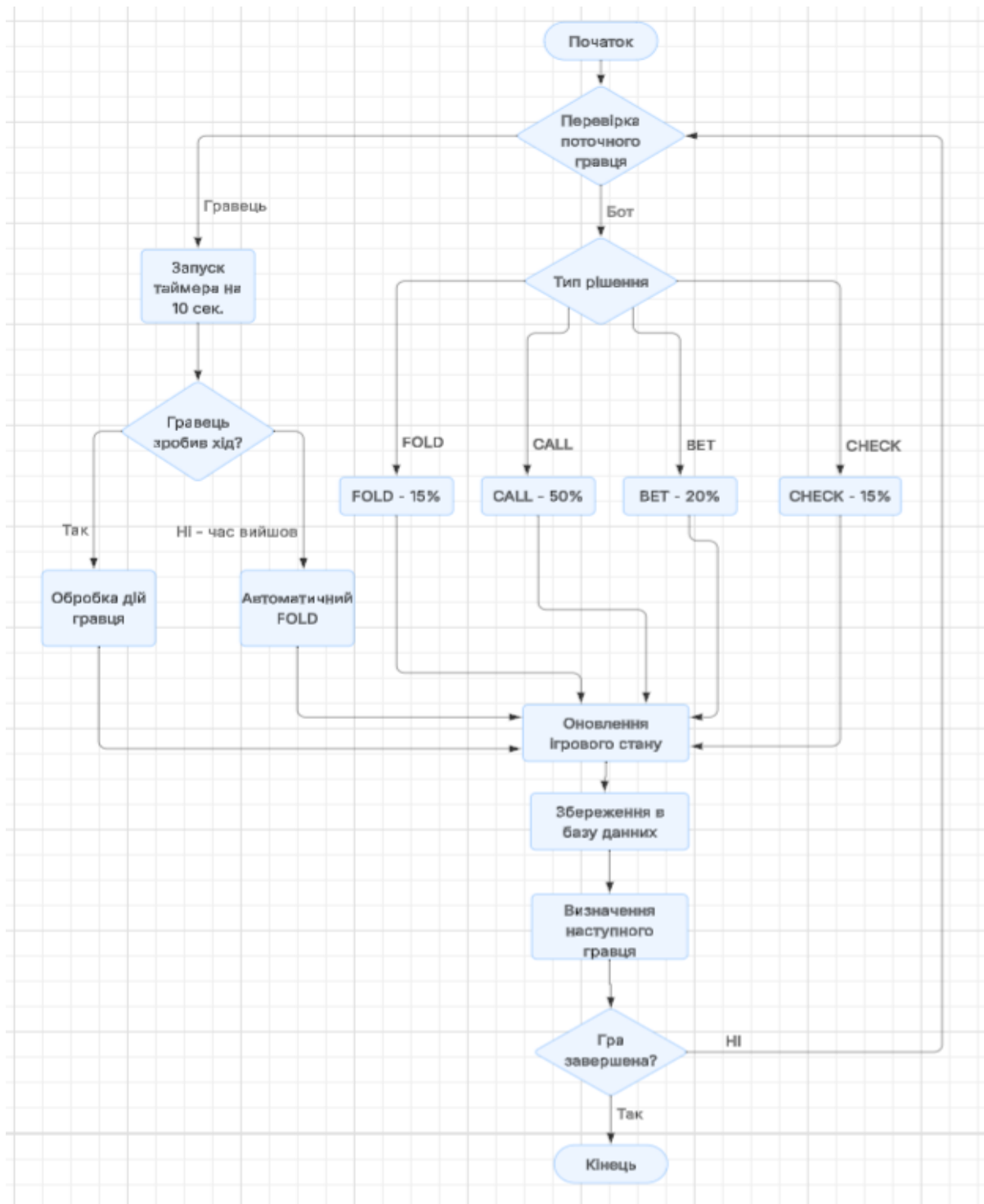


Рисунок 1.10 Блок-схема покерного бота зі змішаною стратегією

Для забезпечення динамічності гри реалізована система таймерів, що обмежує час на прийняття рішення. При перевищенні ліміту часу автоматично виконується дія "fold" для поточного гравця.

Основні компоненти системи таймерів:

- Клієнтський таймер - відображає зворотний відлік для гравця
- Серверна валідація - перевіряє дотримання часових обмежень

Зм.	Арк.	№ докум.	Підпис	Дата

— Автоматичні дії - виконуються при завершенні таймера

Система таймерів та автоматичних дій показує повний цикл роботи таймера від запуску до виконання автоматичної дії (Рисунок 1.11).

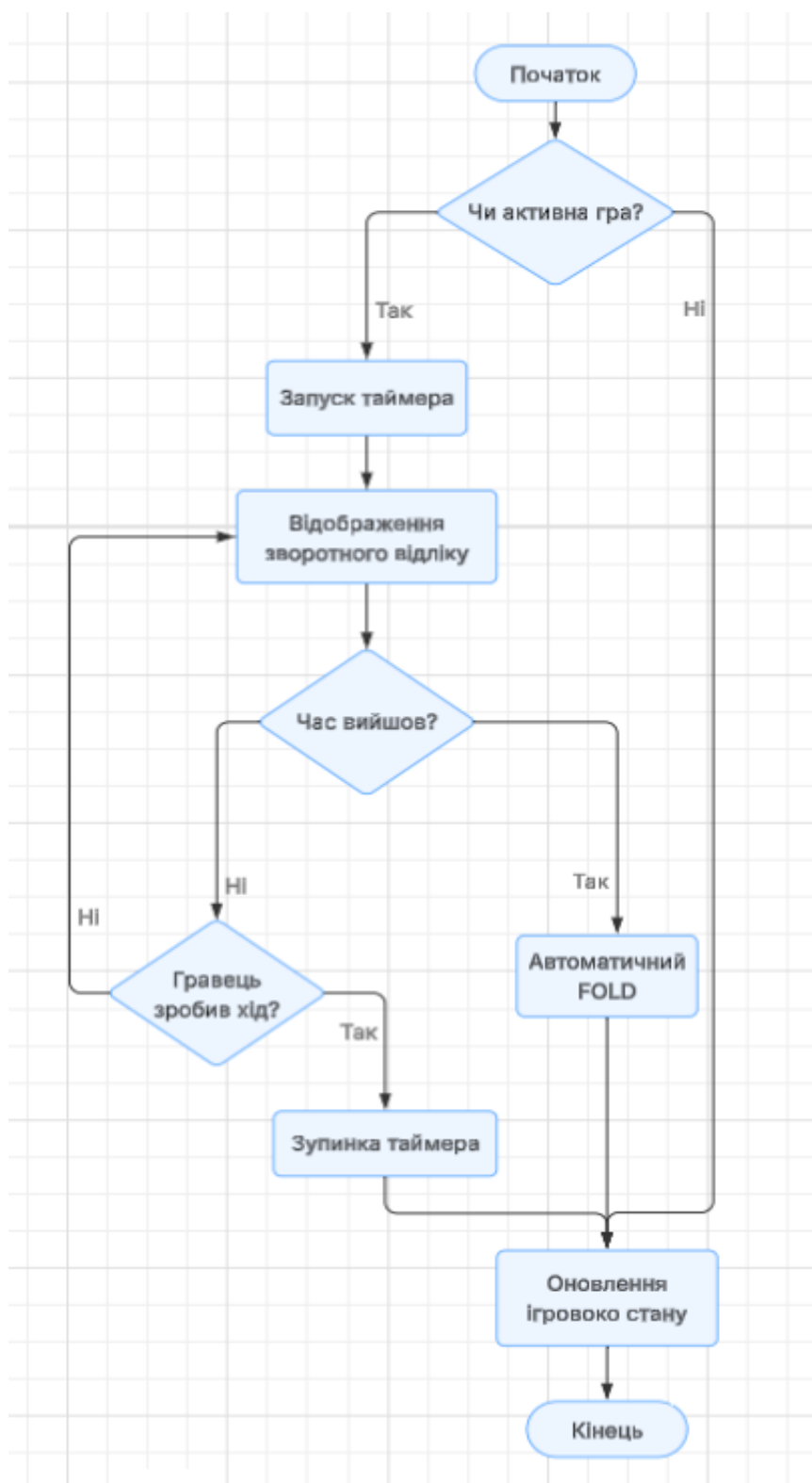


Рисунок 1.11 Блок-схема системи таймерів та автоматичних дій

Зм.	Арк.	№ докум.	Підпис	Дата

1.3 Програмна реалізація веб-додатку для гри в техаський покер

Розроблений веб-додаток для гри в техаський покер побудований за класичною клієнт-серверною архітектурою з чітким розмежуванням шарів представлення, бізнес-логіки та зберігання даних. Такий підхід забезпечує масштабованість, гнучкість та безпеку системи.

Клієнтська частина реалізована на основі бібліотеки React, що дозволяє створювати динамічний, адаптивний та інтуїтивно зрозумілий інтерфейс користувача. Всі ігрові компоненти (стіл, карти, фішки, панель ставок) побудовані як незалежні React-компоненти, що спрощує підтримку та розширення функціоналу.

Серверна частина написана на Node.js з використанням фреймворку Express. Сервер відповідає за обробку ігрової логіки, управління сесіями,

аутентифікацію користувачів, зберігання та обробку даних у базі MySQL (через ORM Sequelize). Для забезпечення роботи в реальному часі використовується WebSocket (Socket.io), що дозволяє синхронізувати ігровий стан між усіма учасниками столу.

База даних розміщена на хмарному сервісі Aiven.io, що гарантує високу доступність, резервне копіювання та захист даних.

1.3.1 Вибір і обґрунтування технологічного стеку розробки

Структурна схема архітектури веб-системи:

- Клієнт надсилає REST-запити для реєстрації, логіну, отримання історії ігор.
- Для ігрових дій (роздача карт, ставки, зміна стану гри) використовується WebSocket-з'єднання.
- Сервер взаємодіє з базою даних для зберігання інформації про користувачів, сесії, результати партій.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		42

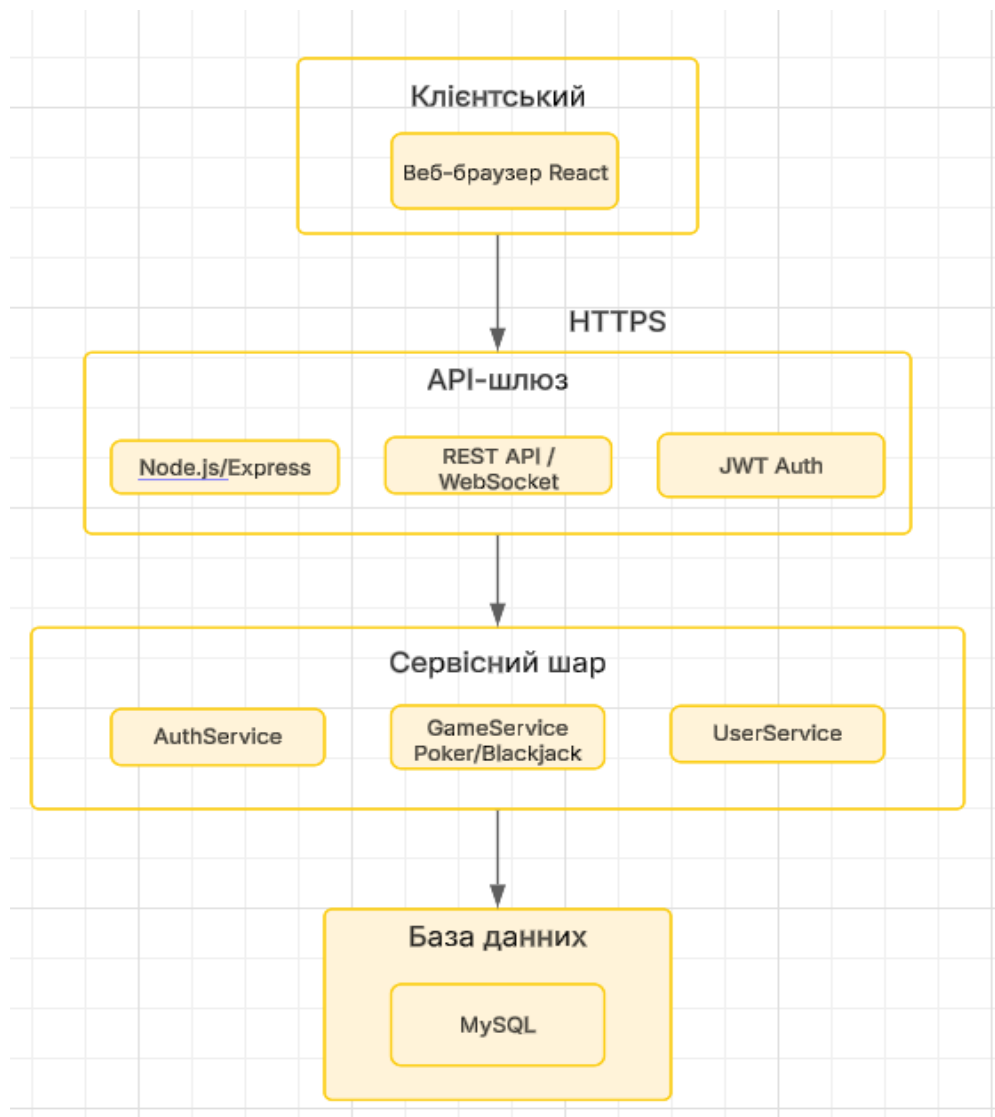


Рисунок 1.12 Схема архітектури веб-платформи

На верхньому рівні знаходиться клієнтський застосунок — веб-браузер, у якому виконується логіка інтерфейсу, реалізована на React (JavaScript). Кожен елемент ігрового столу, карти, панель ставок, чат — це окремі React-компоненти, що взаємодіють між собою через глобальний стан (Redux Toolkit або Context API). Вся бізнес-логіка, пов'язана з роздачею карт, підрахунком комбінацій, визначенням переможця, реалізується виключно на сервері (Рисунок 1.12).

Клієнт надсилає авторизовані HTTP/HTTPS-запити до API-шлюзу для реєстрації, логіну, отримання історії ігор та профілю. Для ігрових дій (роздача карт, ставки, зміна стану гри) використовується WebSocket-з'єднання, що забезпечує миттєву синхронізацію стану між усіма учасниками столу.

API-шлюз реалізовано на Node.js/Express, який обробляє як REST-запити, так

і WebSocket-повідомлення. Основні завдання API-шлюзу:

уніфікація точок входу (/api/..., /ws/...);

автентифікація користувачів через JWT-токени;

застосування CORS-політик, rate-limiting, логування та обробка помилок; маршрутизація запитів до відповідних сервісів (AuthService, GameService, UserService).

Важливо, що клієнтська частина не має прямого доступу до бізнес-логіки чи бази даних — усі дії виконуються через API-шлюз, що підвищує безпеку та дозволяє централізовано контролювати доступ.

Сервісний шар складається з окремих модулів:

AuthService — відповідає за реєстрацію, логін, генерацію та валідацію JWT-токенів;

GameService — реалізує ігрову логіку для покеру та (опціонально) блекджека, обробляє дії гравців, роздачу карт, підрахунок комбінацій, визначення переможця, управління ставками;

UserService — керує профілями користувачів, статистикою, історією ігор. Всі сервіси взаємодіють із базою даних через ORM Sequelize, що забезпечує абстракцію над SQL-запитами, підтримку транзакцій, валідацію даних та міграції.

MySQL виступає реляційним ядром системи, структура якого приведена до третьої нормальної форми (3NF) для уникнення дублювання даних та забезпечення цілісності. Основні таблиці: users, pokergames, blackjackgames, bets, sessions, game_results. Для оптимізації SELECT-запитів використовуються складені індекси (наприклад, по game_id + user_id для швидкого пошуку ставок конкретного гравця у грі).

Резервне копіювання та масштабування бази даних забезпечується можливостями хмарного сервісу Aiven.io, що дозволяє швидко відновлювати дані у разі збоїв та розподіляти навантаження при зростанні кількості користувачів.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		44

1.3.2 Реалізація системи користувачів: реєстрація та автентифікація

Система користувачів у веб-додатку для гри в техаський покер побудована на принципах об'єктно-орієнтованого проєктування та чіткої декомпозиції відповідальностей. Архітектура системи базується на моделі клієнт-сервер з використанням REST API для взаємодії між фронтендом та бекендом.

Система користувачів у веб-додатку для гри в техаський покер побудована на принципах об'єктно-орієнтованого проєктування та чіткої декомпозиції відповідальностей. Архітектура системи базується на моделі клієнт-сервер з використанням REST API для взаємодії між фронтендом та бекендом.

На рисунку представлена ER-діаграма структури бази даних, яка відображає взаємозв'язки між основними сутностями системи. (Рисунок 1.13) Центральною сутністю є таблиця Users, яка містить особисті дані користувачів, інформацію про баланс, роль (user/admin), а також ідентифікатори для інтеграції з різними системами. Таблиця Sessions забезпечує збереження активних сесій користувачів із JWT-токенами, що дозволяє контролювати автентифікацію. Game_sessions зберігає інформацію про участь користувачів у ігрових сесіях, а таблиця Transactions веде облік фінансових операцій користувачів.

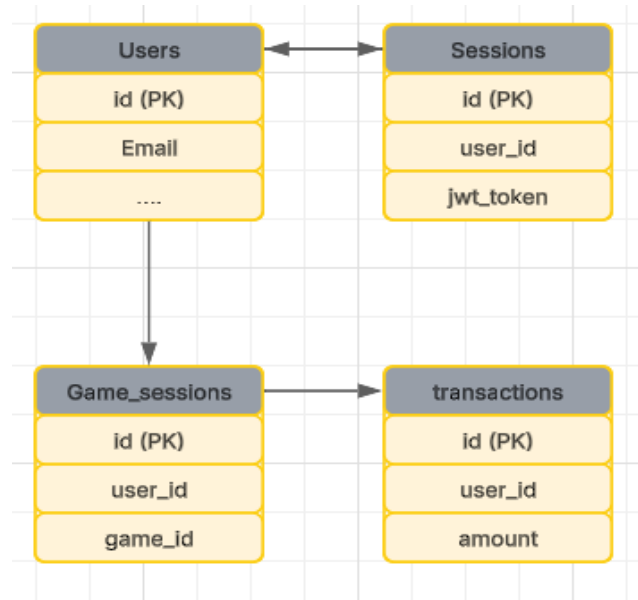


Рисунок 1.13 Схеми бази даних користувачів

Загальна архітектура системи, зображена на рисунку, демонструє

взаємодію між компонентами. Frontend (React) відповідає за форми реєстрації та збереження токенів, Backend (Express) обробляє запити реєстрації, логіну та профілю користувача через JWT-middleware, а Database (MySQL/Aiven) зберігає всі дані користувачів, сесій, транзакцій та ігрових сесій.

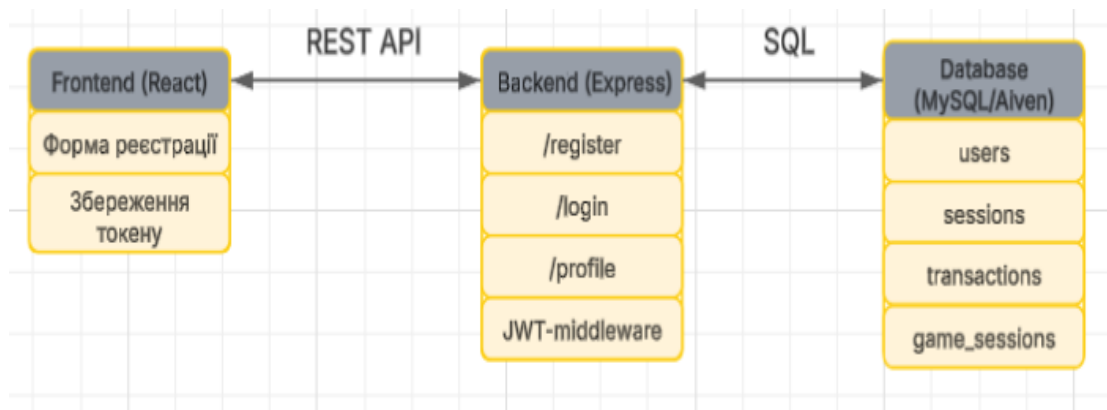


Рисунок 1.14 Архітектура системи користувачів

Процес реєстрації та автентифікації користувачів реалізований згідно з блок-схемою на рисунку (Рисунок 1.14).

Алгоритм складається з наступних етапів:

- Початкова фаза: Користувач заповнює форму реєстрації з необхідними даними
- Відправка запиту: Виконується POST запит до /api/register з даними користувача
- Валідація: Перевіряється унікальність email у базі даних
- Хешування пароля: Пароль шифрується за допомогою bcrypt з 10 раундами хешування
- Створення користувача: Новий запис додається до бази даних з захешованим паролем
- Генерація JWT: Створюється токен автентифікації з ідентифікатором користувача
- Відповідь клієнту: Токен відправляється на фронтенд
- Збереження токену: Токен зберігається у localStorage для подальшого використання

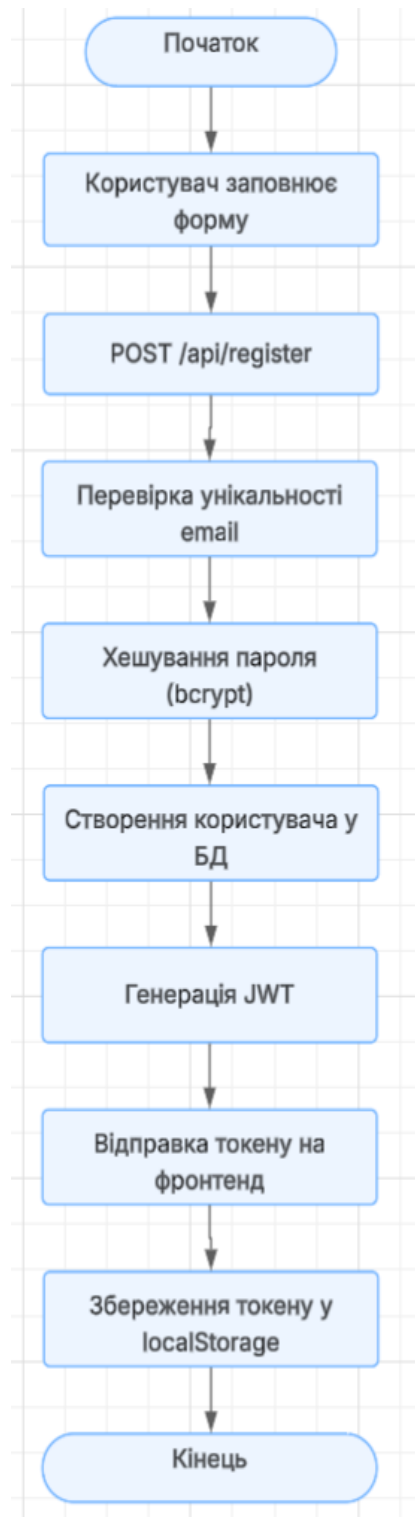


Рисунок 1.15 Блок-схема процесу реєстрації користувача

Інтерфейс системи автентифікації складається з двох основних форм. На рисунку показана форма входу, яка містить поля для введення email та пароля, кнопку "Увійти" та посилання на відновлення пароля і реєстрацію нового користувача (Рисунок 1.15).

Зм.	Арк.	№ докум.	Підпис	Дата

Рисунок 1.16 Вхід в систему та форма реєстрації користувача

Форма реєстрації, представлена на рисунку, включає поля для email, імені користувача (мінімум 3 символи) та пароля (мінімум 6 символів). Інтерфейс забезпечує валідацію даних на клієнтській стороні та відображає відповідні повідомлення про помилки (Рисунок 1.16).

Система користувачів реалізована як окремий модуль бекенду на Node.js/Express, який забезпечує високий рівень безпеки завдяки наступним механізмам:

- Захист паролів: Використання bcrypt для хешування паролів з коефіцієнтом складності 10 раундів;
- JWT автентифікація: Токени з обмеженим терміном дії та підписом секретним ключем;
- Middleware захист: Перевірка валідності токенів для всіх захищених маршрутів;
- Додаткова безпека: Захист від brute-force атак через rate limiting, налаштування CORS та Content Security Policy;

Усі паролі зберігаються виключно у хешованому вигляді, що унеможливорює їх компрометацію навіть у випадку несанкціонованого доступу до бази даних. JWT токени забезпечують безпечну передачу інформації про автентифікацію між клієнтом та сервером без необхідності зберігання сесій на сервері.

					РП 08. 20 001. 00 ДП ПЗ	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		48

1.3.3 Інтеграція універсального API для гри Blackjack

Для підвищення привабливості та функціональності веб-платформи покеру було реалізовано інтеграцію додаткової гри - Blackjack. Це дозволяє користувачам перемикатися між різними ігровими режимами в межах єдиного інтерфейсу, а також забезпечує універсальність API для подальшого розширення функціоналу.

Для підвищення привабливості та функціональності веб-платформи покеру було реалізовано інтеграцію додаткової гри - Blackjack. Це дозволяє користувачам перемикатися між різними ігровими режимами в межах єдиного інтерфейсу, а також забезпечує універсальність API для подальшого розширення функціоналу.

Універсальний API:

Для обох ігор використовується єдиний API, що дозволяє обробляти запити на створення сесій, реєстрацію гравців, роздачу карт, ставки та визначення переможця. Це спрощує підтримку та масштабування системи.

Структура основних API-запитів:

- POST /api/game/create створення нової ігрової сесії (тип гри передається у body)
- POST /api/game/:id/action виконання дії (ставка, fold, hit, stand тощо)
- GET /api/game/:id/state отримання поточного стану гри

Особливості реалізації Blackjack:

- Ігрова логіка реалізована у вигляді окремого модуля, що підключається до основного бекенду.
- Валідація дій гравця (наприклад, перевірка можливості взяти карту або завершити раунд) здійснюється на сервері.
- Зберігання стану гри та історії дій реалізовано через ORM Sequelize у таблицях бази даних.

На рисунку наведено діаграму компонентів React-додатку, що ілюструє взаємодію між Poker, Blackjack та спільними модулями (Auth, UserProfile, GameTable). (Рисунок 1.17)

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		49

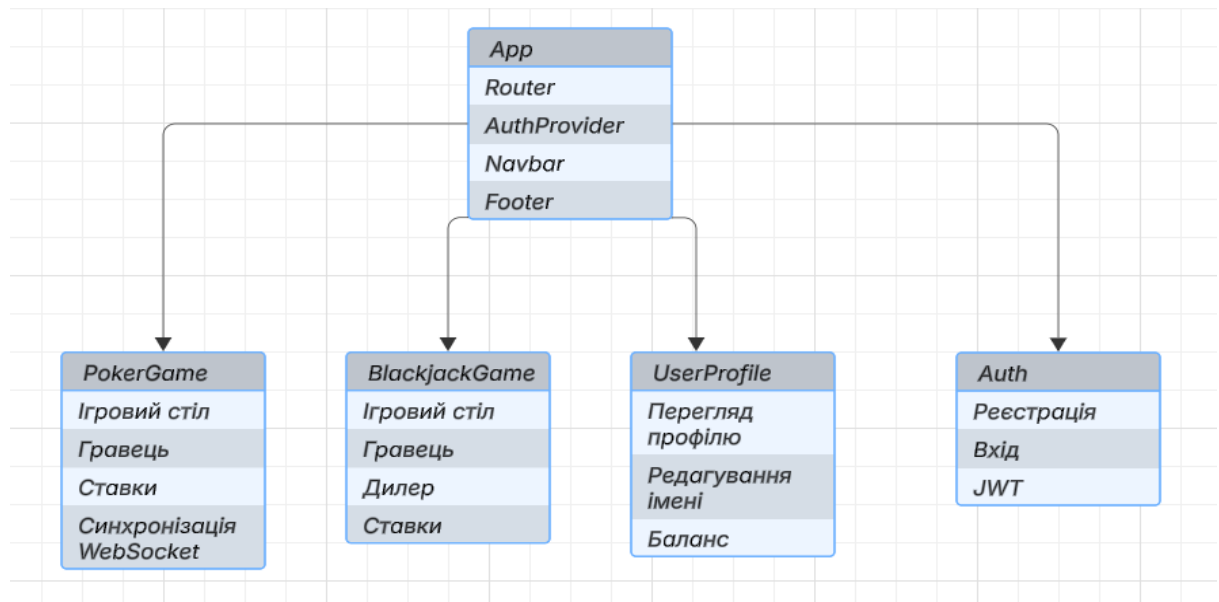


Рисунок 1.17 Діаграма компонентів клієнтської частини

Для обох ігор використовується єдиний механізм синхронізації стану через WebSocket. Сервер надсилає оновлення всім підключеним клієнтам при зміні стану гри (наприклад, після роздачі карт або завершення раунду).

На рисунку наведено приклад ігрового інтерфейсу для гри в Blackjack, що реалізований у рамках універсальної архітектури (Рисунок 1.18).

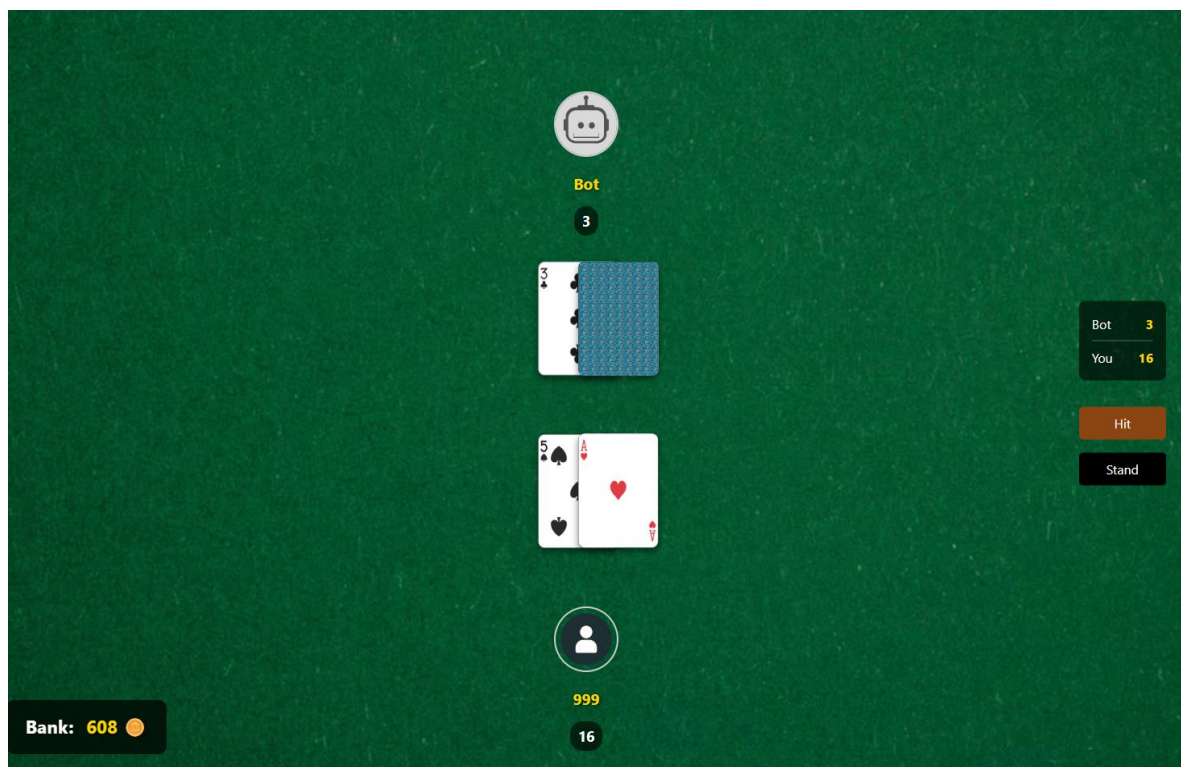


Рисунок 1.18 Інтерфейс гри Blackjack

1.3.4 Тестування розробленого програмного забезпечення

Тестування розробленої покерної системи проводилось з метою перевірки її стабільності, відповідності функціональним вимогам та коректності обробки ігрових механік. Основний акцент було зроблено на перевірку логіки гри, взаємодії між гравцями та ботами, обробки ставок та синхронізації ігрового стану.

У процесі тестування було перевірено такі ключові функціональні модулі:

- створення нової гри та підключення гравців;
- коректність роздачі карт та визначення черги ходу;
- роботу таймерів для гравців та автоматичне виконання дій у разі завершення часу;
- логіку прийняття рішень ботами (fold, call, bet, check) відповідно до заданих ймовірностей;
- обробку ставок, підняття та скидання карт;
- збереження ігрового стану після кожної дії у базі даних MySQL;
- визначення переможця та завершення гри.

Тестування проводилося переважно на рівні API за допомогою інструменту Postman. Для цього були створені різні сценарії, що імітують дії реальних гравців та ботів:

- створення ігрової сесії;
- виконання дій гравцем (ставка, скидання, чек);
- автоматичне виконання дій ботом згідно з алгоритмом прийняття рішень;
- перевірка переходу ходу до наступного гравця;
- завершення гри при визначенні переможця;

Особливу увагу було приділено тестуванню сценаріїв з помилками, зокрема:

- спроба виконати дію не у свою чергу;
- виконання дії гравцем, який вже скинув карти;
- обробка ситуацій, коли у гравця недостатньо фішок для ставки;
- перевірка коректності відновлення гри після збоїв або втрати з'єднання з базою даних;

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		51

Окрему увагу було приділено перевірці алгоритму прийняття рішень ботами. Боти у системі діють згідно з ймовірнісною моделлю, що враховує поточний стан гри, розмір ставки та кількість фішок. Для кожного ходу бота перевірялась коректність вибору дії відповідно до заданих ймовірностей (fold, call, bet, raise, check), а також правильність оновлення стану гри після дії бота.

					<i>РП 08. 20 001. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		52

2. ЕКОНОМІЧНИЙ РОЗДІЛ

2.1 Резюме

У даному розділі виконується економічне обґрунтування доцільності розробки та впровадження веб-додатку для гри в техаський покер у браузері, розробленого в межах дипломного проєкту. Розрахунки проводяться з метою оцінки трудомісткості розробки, фінансових витрат на створення програмного продукту, а також формування кінцевої вартості продукту для потенційного споживача або замовника.

Проведення такого аналізу дає змогу визначити доцільність інвестицій у розробку ігрового веб-додатку, рівень витрат на оплату праці фахівців, супровідні витрати, а також оцінити рівень прибутковості та конкурентоспроможності розробленого програмного забезпечення. У процесі аналізу використано типові показники трудомісткості для веб-розробки, витрат на розробку ігрових додатків, нормативи оплати праці та методику формування вартості відповідно до чинних нормативних документів.

Основною метою даного аналізу є демонстрація економічної доцільності проєкту не лише як навчального кейсу, а й як потенційного комерційного продукту в сфері онлайн-ігор. Визначення загальної трудомісткості, собівартості, рівня рентабельності та кінцевої ціни програмного продукту дає змогу зробити висновки щодо можливості реалізації такого рішення в реальних ринкових умовах.

2.2 Визначення трудомісткості розробки веб-додатку

Тривалість розробки веб-застосунку для гри в техаський покер залежить від обсягу коду, складності реалізації ігрової логіки та кваліфікації виконавців.

Враховуючи специфіку проєкту, було обрано аналог з категорії «Веб-додатки з інтерактивною логікою» з обсягом $V_0 = 5,5$ тис. умовних машинних команд. Згідно з табл. 2.2 базова норма часу становить 410 людино-годин.

З урахуванням коефіцієнта застосування сучасних технологій веб-розробки

					<i>РП 08. 20 002. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		53

($K_k = 0,8$) одержано укрупнену трудомісткість $T_{ан} = 328$ люд-год.

Код новизни проекту — В (аналог з модифікаціями), оскільки використовуються існуючі технології та фреймворки з адаптацією під специфічні вимоги покерної гри, тому $K_n = 0,75$ Частка повторного використання модулів та бібліотек складає приблизно 25%, що дає коефіцієнт $K_T = 0,85$.

Питомі коефіцієнти стадій при коді новизни В для веб-додатків:

- $L_1 = 0,15$ (технічне завдання та аналіз);
- $L_2 = 0,12$ (технічний проект та архітектура);
- $L_3 = 0,58$ (програмна реалізація);
- $L_4 = 0,15$ (тестування та документація);

Таблиця 2.1 Нормативні параметри для оцінки трудомісткості розробки

Параметр	Значення
$T_{ан}$, люд-год	328
K_n	0,75
K_T	0,85
L_1	0,15
L_2	0,12
L_3	0,58
L_4	0,15

Загальна трудомісткість проекту розраховується за формулою: $T_{заг} = T_{ан} \times K_n \times K_T = 328 \times 0,75 \times 0,85 \approx 209$ люд-год

Це еквівалентно 0,12 розробника-року при 8-годинному робочому дні та коефіцієнті календарного планування 0,73.

Таблиця 2.2 Розрахована трудомісткість етапів життєвого циклу ПЗ

Стадія розробки	T, год
Технічне завдання та аналіз	$110 \times 0,15 = 16,5$
Технічний проект	$110 \times 0,12 = 13,2$
Робочий проект	$110 \times 0,58 = 63,8$
Тестування та документація	$110 \times 0,15 = 16,5$
Разом	110,0

Такі фінансові результати підтверджують економічну доцільність проекту: закладена маржа покриває усі прямі та непрямі витрати, водночас забезпечуючи конкурентну вартість для замовника й прийнятний рівень прибутковості для розробника.

2.3 Розрахунок ціни програмного продукту

Даний підрозділ визначає планову собівартість та кінцеву відпускну ціну веб-додатку для гри в техаський покер, розробленого на рівні Junior. Розрахунки виконано нормативним методом із урахуванням чинних тарифів оплати праці, матеріальних витрат, накладних та податкових відрахувань.

У проекті задіяно одного розробника рівня Junior і керівник-консультант; нормоконтроль виконується окремим спеціалістом. Загальна трудомісткість 110 люд./год.

Таблиця 2.3 Розрахунок основної заробітної плати виконавців

Найменування робіт	Трудомісткість, год	Тарифна ставка, грн/год	Сума, грн.
Розробка ПП	63,8	48	3062
Тестування та налагодження	16,5	48	792
Контроль керівника	16,5	48	792
Нормоконтроль	13,2	48	634
Усього Z_0	110	—	5 280

Загальний фонд основної заробітної плати:

$$Z_0 = \sum T_{mj} \cdot Z_{год} = 110 \text{ год} \times 48 \text{ грн.} = 5 280 \text{ грн.}$$

Таблиця 2.4 Розрахунок матеріальних витрат на розробку веб-додатку

Найменування	Тип / модель	Кількість, шт	Ціна, грн.	Вартість, грн
Папір А4	пачка 500 арк.	1	400	400
Канцелярські товари	комплект	1	120	120
Разом B_{M1}	—	—	—	520
Транспортно-заготівельні (10 %)	—	—	—	52
Усього B_M	—	—	—	572

Таблиця 2.6 узагальнює матеріальні витрати, необхідні для документального супроводу проекту. Основними статтями є витрати на папір різних форматів та канцелярські товари. Додатково враховано 10% транспортно-заготівельних витрат, що формує загальну суму 572 грн.

Таблиця 2.5 Калькуляція планової собівартості веб-додатку

Стаття витрат	Значення, грн	Формула
Матеріали	572	Вм
Основна заробітна плата	5 280	Зо
Додаткова заробітна плата (10 %)	528	$0,10 \times Zo$
Відрахування ЄСВ (22 %)	1 278	$0,22 \times (Zo + Zd)$
Накладні витрати (50 % Зо)	2 640	$0,50 \times Zo$
Повна собівартість Спов	10 298	Σ поз. 1–5

У таблиці 2.7 наведено покрокову калькуляцію повної собівартості веб-додатку. Сумарні витрати на матеріали, оплату праці та накладні формують планову собівартість у розмірі 10 298 грн, що є базою для подальшого формування ціни.

Закладена рентабельність 15% забезпечує покриття всіх витрат на розробку та гарантує прийнятний рівень прибутковості, що підтверджує комерційну привабливість проекту.

3 РОЗДІЛ ОХОРОНИ ПРАЦІ ТА ТЕХНІКИ БЕЗПЕКИ

Охорона праці є невід'ємною складовою будь-якої виробничої діяльності та має на меті створення безпечних і здорових умов праці для працівників. Забезпечення безпеки праці, дотримання вимог гігієни праці та виробничої санітарії, а також пожежної безпеки є основними завданнями системи охорони праці на підприємстві.

Відповідно до Закону України "Про охорону праці" та інших нормативно-правових актів, роботодавець зобов'язаний забезпечити безпечні та здорові умови праці на кожному робочому місці, проводити інструктажі з охорони праці, забезпечувати працівників засобами індивідуального захисту та створювати умови для їх ефективного використання.

У процесі виробничої діяльності на працівників можуть впливати різноманітні небезпечні та шкідливі виробничі фактори, які потребують детального аналізу та розробки відповідних заходів щодо їх усунення або мінімізації негативного впливу.

3.1 Аналіз небезпечних і шкідливих факторів, що впливають на користувача ПК

Робота користувача персонального комп'ютера пов'язана з наявністю різноманітних небезпечних і шкідливих виробничих факторів, які можуть негативно впливати на здоров'я та працездатність. Системний аналіз цих факторів виконується відповідно до вимог ДСанПіН 3.3.6.042-99 "Гігієнічні вимоги до організації роботи з відеодисплейними терміналами", ДБН В.2.5-28-2006, ПУЕ-2021, а також інших чинних нормативних актів України.

Робота з комп'ютером належить до категорії «1а» за ДСанПіН, що характеризується мінімальними фізичними зусиллями, але водночас супроводжується значним зоровим навантаженням, статичним положенням тіла та психоемоційним напруженням.

Класифікація небезпечних і шкідливих факторів:

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		57

1. Фізичні фактори

Параметри мікроклімату:

- Температура повітря у робочому приміщенні може коливатися за межами оптимальних значень (22-24 С взимку, 23-25 С влітку);
- Відносна вологість повітря може не відповідати нормативним 40-60%;
- Швидкість руху повітря може перевищувати допустимі 0,1 м/с;

Освітлення:

- Недостатня освітленість робочого місця (норма - не менше 500 лк при змішаному освітленні);
- Нерівномірність освітлення та наявність відблисків на екрані монітора;
- Невідповідність коефіцієнта природної освітленості (КПО $\geq 1,5\%$);

Шум та вібрація:

- Шум від системних блоків, вентиляторів охолодження, принтерів та іншого обладнання;
- Вібрація від працюючого обладнання, особливо при щільному розміщенні техніки;

Електромагнітні поля:

- Електромагнітне випромінювання від моніторів, системних блоків, мережевого обладнання;
- Електростатичні поля, що виникають на поверхні екранів

2. Хімічні фактори

- Виділення озону та оксидів азоту від лазерних принтерів;
- Випаровування шкідливих речовин з пластикових корпусів обладнання при нагріванні;
- Пари та аерозолі від засобів очищення техніки;
- Можливе перевищення концентрації шкідливих речовин у повітрі робочої зони;

3. Біологічні фактори:

- Накопичення мікроорганізмів у системах вентиляції та

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		58

кондиціювання;

- Бактеріальне забруднення клавіатури, миші та інших засобів введення інформації;
- Можливе перевищення концентрації мікроорганізмів у повітрі (норма < 500 КУО/м³)

4. Психофізіологічні та ергономічні фактори:

Фізичні навантаження:

- Статичне навантаження на м'язи шийі, спини та верхніх кінцівок;
- Тривале перебування в одній позі;
- Монотонні рухи при роботі з клавіатурою та мишею;

Зорове навантаження:

- Тривала робота з екраном монітора;
- Необхідність фокусування зору на близькій відстані;
- Різка зміна яскравості при переведенні погляду з екрана на документи;

Нервово-психічне навантаження:

- Висока відповідальність за результати роботи;
- Дефіцит часу при виконанні завдань;
- Необхідність прийняття рішень в умовах обмеженого часу;
- Монотонність трудового процесу;

5. Електротехнічні фактори

- Ураження електричним струмом при дотику до струмоведучих частин обладнання;
- Коротке замикання в електромережі або обладнанні;
- Перевантаження електромережі;
- Відсутність або несправність засобів захисту (заземлення, занулення, УЗО);

6. Пожежні фактори

- Перегрів блоків живлення та іншого електронного обладнання;
- Займання кабель-каналів та проводки;;
- Накопичення пилу в системних блоках

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		59

— Використання горючих матеріалів у конструкції меблів та оздоблення приміщень;

Для забезпечення безпечних умов праці встановлено наступні нормативні значення:

- Мікроклімат: температура 22-24 С (зима), 23-25 С (літо), вологість 40-60%, швидкість повітря $\leq 0,1$ м/с;
- Освітлення: мінімальна освітленість 500 лк, КПО $\geq 1,5\%$;
- Шум: не більше 50 дБА для робочих місць категорії Пб;
- ЕМП: не більше 1 мкТл у діапазоні 50 Гц;
- Повітрообмін: не менше 60 м³/год на одне робоче місце;

За результатами аналізу можна виділити наступні рівні ризику:

Високий ризик:

- Зорове перенапруження при тривалій роботі з монітором;
- Порушення постави внаслідок статичного навантаження;

Середній ризик:

- Ураження електричним струмом при несправності обладнання;
- Пожежна небезпека від перегріву електронних компонентів;
- Психоемоційне перенапруження;

Низький ризик:

- Вплив електромагнітних полів;
- Шумове забруднення;
- Хімічні забруднювачі повітря;

Для мінімізації виявлених ризиків необхідно розробити комплекс технічних та організаційних заходів, спрямованих на створення безпечних і здорових умов праці користувачів персональних комп'ютерів.

3.2 Гігієнічні вимоги до виробничого середовища

Цей розділ визначає гігієнічні вимоги до створення безпечного та здорового виробничого середовища для користувачів персональних комп'ютерів, охоплюючи такі аспекти, як приміщення, освітлення, шум,

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		60

організація робочого місця, мікроклімат та електробезпека. Дотримання цих вимог є обов'язковою умовою для забезпечення нормальних умов праці та збереження здоров'я.

3.2.1 Освітлення

Правильно організоване освітлення робочих місць є одним з найважливіших факторів створення комфортних умов праці та збереження зору користувачів ПК.

Основні вимоги до виробничого освітлення:

- Рівень освітленості робочих поверхонь повинен відповідати гігієнічним нормам для певного виду робіт;
- Забезпечення рівномірності та часової стабільності рівня освітленості у приміщенні;
- Відсутність різких контрастів між освітленістю робочої поверхні та навколишнього простору;
- Відсутність на робочій поверхні різких тіней (особливо рухомих);
- Недопущення створення сліпучого блиску предметів;
- Штучне світло за своїм спектральним складом повинно бути наближеним до природного;
- Джерела світла не повинні створювати додаткових небезпечних та шкідливих факторів;
- Енергозбереження, надійність та простота в експлуатації;

Нормативні показники освітлення:

- Мінімальна освітленість робочих поверхонь при роботі з ПК - 500 лк при змішаному освітленні;
- Коефіцієнт природної освітленості (КПО) - не менше 1,5%;
- Співвідношення яскравостей у полі зору - не більше 3:1;
- Нерівномірність освітлення - не більше 3:1;

Організація природного освітлення:

- Природне освітлення організовується через віконні прорізи;
- Світловий коефіцієнт (відношення площі вікон до площі підлоги) - 1:4;

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		61

- Робочі місця розташовують перпендикулярно до вікон;
- Екрани моніторів повинні бути орієнтовані під кутом 90 до світлових прорізів;

Штучне освітлення:

- Використання світлодіодних панелей з температурою кольору 4000 К;
- Індекс передачі кольору $Ra \geq 80$;
- Коефіцієнт пульсацій менше 5%;
- Світильники розташовуються паралельно до лінії погляду оператора;
- Застосування місцевого освітлення (настільні лампи) потужністю 200-300 лк;

3.2.2 Шум

Шум є одним із основних шкідливих факторів у виробничому середовищі, який може негативно впливати на здоров'я працівників, знижувати їхню працездатність та ефективність роботи. Тривалий вплив шуму може призводити до порушень слуху, нервової системи, серцево-судинних захворювань та зниження концентрації уваги.

Відповідно до санітарних норм, допустимий рівень еквівалентного шуму в приміщеннях для роботи з персональними комп'ютерами не повинен перевищувати 50 дБА протягом робочої зміни.

Основними джерелами акустичного навантаження в приміщеннях з комп'ютерним обладнанням є:

- системні блоки персональних комп'ютерів;
- серверні стійки та сервери;
- системи вентиляції та кондиціонування повітря;
- принтери та інше периферійне обладнання;
- зовнішні джерела шуму (транспорт, будівельні роботи тощо);

Технічні заходи:

- використання корпусів з низькошумними вентиляторами (рівень шуму ≤ 25 дБА);
- встановлення гумових антивібраційних опор під системні блоки;

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		62

- застосування шумогасних вставок в каналах вентиляційної системи;
- звукоізоляція стін, стелі та підлоги приміщення;
- використання звукопоглинальних матеріалів;

Контроль шумового навантаження здійснюється за допомогою класичного шуміра відповідно до методики ГОСТ 12.1.003-83. Вимірювання проводяться у п'яти точках робочої зони на рівні вух працівника (1,5 м від підлоги).

3.2.3 Вимоги до організація робочого місця

При організації робочого місця, яке передбачає роботу з персональним комп'ютером (ПК) та периферійними пристроями (клавіатура, маніпулятор «миша», дискова система, модем, принтер, сканер тощо), необхідно дотримуватися вимог «Державних санітарних норм і правил роботи з візуальними дисплейними терміналами (ВДТ) електронно-обчислювальних машин» ДСанПіН 3.3.2.007-98.

При плануванні робочого місця з ПК слід передбачити:

Достатній простір для працівника – площа робочого місця повинна становити не менше 4,5 м² на одного користувача, а об'єм приміщення – не менше 20 м³. Це забезпечує комфортні умови роботи та можливість вільного переміщення працівника.

Правильне розміщення робочого місця відносно світлових прорізів – екран монітора повинен розташовуватися перпендикулярно до лінії «вікно-працівник», щоб уникнути відблисків на екрані та засліплення природним світлом. Відстань від вікна до робочого місця має бути не менше 1,5 м.

Оптимальні відстані між робочими столами з ПК:

- відстань між бічними поверхнями відеотерміналів – не менше 1,2 м;
- відстань між екраном одного відеотерміналу та тильною частиною іншого – не менше 2,0 м;

Раціональне розташування основних елементів робочого місця:

- відстань від очей до екрана – 500-700 мм;

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		63

- верхня межа екрана повинна знаходитися на рівні очей або нижче;
- кут зору на екран – 15-20 вниз від горизонталі;
- клавіатура розташовується на відстані 100-300 мм від краю стола;

Організація робочого місця повинна регулярно перевірятися на відповідність встановленим нормам. При виявленні невідповідностей необхідно вжити заходів щодо їх усунення для забезпечення безпечних та комфортних умов праці користувачів ПК.

3.2.4 Мікроклімат

Мікроклімат виробничого приміщення є одним з найважливіших факторів, що впливає на працездатність, комфорт та здоров'я користувачів персональних комп'ютерів. Виробниче повітряне середовище характеризується певними фізичними властивостями, які визначаються параметрами мікроклімату: температурою, відносною вологістю, швидкістю руху повітря та барометричним тиском.

Відповідно до державних санітарних норм і правил роботи з візуальними дисплейними терміналами ДСанПіП 3.3.2.007-98, для приміщень з персональними комп'ютерами встановлені оптимальні та допустимі значення параметрів мікроклімату.

Для категорії робіт «1а» (мінімальні фізичні навантаження), до якої відноситься робота з комп'ютерною технікою, оптимальні значення параметрів мікроклімату становлять:

Таблиця 3.1 Оптимальні параметри мікроклімату для роботи з ПК

Період року	Температура повітря, С	Відносна вологість, %	Швидкість руху повітря, м/с
Холодний	22-24	40-60	≤ 0,1
Теплий	23-25	40-60	≤ 0,1

Порушення оптимальних параметрів мікроклімату може призвести до:

- зниження працездатності та концентрації уваги;

- появи синдрому "сухого ока" при низькій вологості повітря;
- дискомфорту та швидкої втомлюваності;
- погіршення загального самопочуття користувачів;

Особливо важливим є контроль вологості повітря, оскільки пересушене повітря (менше 40%) сприяє накопиченню електростатичного заряду на поверхнях обладнання та може викликати офтальмологічні проблеми.

Дотримання оптимальних параметрів мікроклімату є обов'язковою умовою для забезпечення здорових та безпечних умов праці користувачів персональних комп'ютерів, підвищення їх працездатності та запобігання професійним захворюванням.

3.2.5 Електробезпека

Електробезпека на робочому місці з персональним комп'ютером є критично важливим аспектом охорони праці, оскільки електричне обладнання може становити потенційну загрозу для здоров'я та життя користувача.

Класифікація приміщення за електробезпекою

Згідно з ПУЕ (Правила улаштування електроустановок), приміщення для роботи з персональними комп'ютерами класифікується як приміщення без підвищеної небезпеки, оскільки:

- відносна вологість повітря не перевищує 75%;
- температура повітря не перевищує +35 С;
- відсутні хімічно активні або органічні пари;
- відсутній струмопровідний пил;
- підлоги не є струмопровідними;

Основні джерела електричної небезпеки

На робочому місці з ПК потенційні джерела електричної небезпеки включають:

- системний блок комп'ютера (блок живлення 220-240 В);
- монітор (високовольтні кола до 25 кВ у ЕПТ-моніторах);
- периферійні пристрої (принтер, сканер, модем);

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		65

- мережеві кабелі та з'єднання;
- розетки електроживлення;

Дотримання вимог електробезпеки на робочому місці з ПК забезпечується комплексом технічних та організаційних заходів. Основними елементами системи електробезпеки є якісна ізоляція, надійне заземлення, автоматичний захист від струмів витоку та регулярний контроль технічного стану електрообладнання.

3.3 Пожежна безпека

У цьому підрозділі розглянуто організаційно-технічні заходи, що забезпечують належний рівень пожежної безпеки в приміщенні з персональними комп'ютерами. Аналіз проведено відповідно до Правил пожежної безпеки в Україні, затверджених наказом Міністерства внутрішніх справ України, НПАОП 60.1-1.01-06, ДБН В.1.1-7:2022 «Пожежна безпека об'єктів».

Основними вибухо- та пожежонебезпечними речовинами і матеріалами, що знаходяться в приміщенні з персональними комп'ютерами, є:

Таблиця 3.2 Характеристики горючих матеріалів приміщення

Найменування	Характеристики	Місце розташування
Корпуси системних блоків (АБС-пластик)	Температура займання 340 С, група горючості Г4	Робочі столи
Кабельна продукція (ПВХ ізоляція)	Температура займання 280 С, виділення токсичних газів	Електропроводка
Меблі з ДСП	Температура займання 250 С, група горючості Г4	По всьому приміщенню
Папір та картон	Температура займання 230 С, група горючості Г4	Документація, упаковка
Тонер для принтерів	Пилоподібна речовина, здатна до статичної електризації	Картриджі принтерів

Найбільш вірогідними причинами виникнення пожежі в приміщенні з ПК є:

- Короткі замикання в електропроводці та електронному обладнанні;
- Перевантаження електричних мереж;
- Перегрів блоків живлення комп'ютерів;
- Несправність периферійного обладнання (принтери, сканери);
- Накопичення статичної електрики при роботі з тонерами;
- Необережне поводження з електроприладами;

Розрахунок питомої пожежної навантаженості:

- Загальна площа приміщення: 40 м²;
- Пожежна навантага від обладнання: 8000 МДж;
- Питома пожежна навантага: $8000/40 = 200$ МДж/м²;

За ДБН В.1.1-7:2022 приміщення відноситься до категорії В4 (речовини та матеріали здатні горіти, але кількість горючих речовин невелика).

Клас зони за ПУЕ: П-ІІа (приміщення з горючим пилом від тонерів та можливістю утворення статичної електрики).

Таблиця 3.3 Первинні засоби пожежогасіння

Тип вогнегасника	Марка	Кількість	Призначення
Вуглекислотний	ВВК-3	2 шт.	Гасіння електрообладнання під напругою до 1000 В
Порошковий	ВП-4	1 шт.	Гасіння твердих горючих речовин (меблі, папір)

Розміщення вогнегасників забезпечує доступність з будь-якої точки приміщення на відстані не більше 15 м. Технічне обслуговування проводиться щорічно згідно з НПАОП 0.00-5.13-12.

Евакуація персоналу передбачена через:

- Основний вихід: двері шириною 0,9 м, що відчиняються назовні;
- Запасний вихід: через суміжне приміщення в евакуаційний коридор;

Максимальна відстань від робочого місця до найближчого виходу не

перевищує 20 м, що відповідає нормативним вимогам для приміщень категорії В4 (не більше 25 м).

План евакуації у форматі А3 розміщено біля входу в приміщення. Евакуаційні виходи позначено світловими покажчиками з автономним живленням.

Проведений аналіз показує, що приміщення з персональними комп'ютерами відноситься до категорії В4 за вибухопожежною небезпекою. Основними джерелами пожежної небезпеки є електричне обладнання та горючі матеріали корпусів приладів і меблів.

Для забезпечення пожежної безпеки передбачено комплекс технічних та організаційних заходів: автоматична пожежна сигналізація, первинні засоби пожежогасіння, організовані шляхи евакуації та встановлений протипожежний режим. Реалізовані заходи забезпечують допустимий рівень пожежного ризику та відповідають чинним нормативним вимогам.

					<i>РП 08. 20 003. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		68

ВИСНОВКИ

У результаті виконання дипломного проекту було успішно розроблено та впроваджено інноваційну веб-платформу для гри в техаський холдем, яка працює у веб-браузері та забезпечує інтерактивну взаємодію між користувачем і ботами в реальному часі.

Система реалізована на основі архітектури клієнт-сервер із використанням REST API для обміну даними між фронтендом (React) та бекендом (Node.js/Express), а також інтеграцією з хмарною базою даних MySQL через ORM Sequelize.

У процесі роботи проведено аналіз сучасних онлайн-покерних платформ, досліджено технології real-time взаємодії та збереження стану гри. Обґрунтовано вибір технологічного стеку, що включає React для клієнтської частини, Node.js/Express для серверної логіки, MySQL/Aiven.io для зберігання даних, а також Socket.IO для синхронізації ігрового стану між учасниками.

Реалізовано повний цикл ігрової логіки техаського покеру: автоматичну роздачу карт, підрахунок покерних комбінацій, визначення переможця, обробку ставок (blinds, call, raise, fold), а також систему ботів, які приймають рішення на основі ймовірнісних алгоритмів. Додатково інтегровано гру Blackjack, що стало можливим завдяки універсальному API та модульній структурі бекенду.

Особливу увагу приділено розробці користувацького інтерфейсу, який забезпечує зручну взаємодію з ігровим столом, відображенням карт, фішок та забезпечує зручну взаємодію з ігровим столом. Система реєстрації та автентифікації користувачів забезпечує безпеку та персоналізацію ігрового досвіду.

Проект підтвердив ефективність обраних технічних рішень та архітектурних підходів. Проведене комплексне тестування підтвердило стабільність роботи системи, коректність ігрової логіки та надійність збереження даних.

					<i>РП 08. 20 000. 00 ДП ПЗ</i>	Арк.
Зм.	Арк.	№ докум.	Підпис	Дата		69

ПЕРЕЛІК ВИКОРИСТАНИХ ІНФОРМАЦІЙНИХ ДЖЕРЕЛ

1. Дакетт Дж. JavaScript и jQuery: Інтерактивна розробка веб-інтерфейсів. – Х.: Видавництво "Диалектика", 2021. – 640 с.
 2. Ерманн С. Програмування ігор на JavaScript. – Львів: Видавництво "Новий курс", 2021. – 512 с.
 3. Замоткін Д. JavaScript для новачків. – К.: Видавництво "Фактор", 2020. – 320 с.
 4. Зеллманн Т. Алгоритми на JavaScript. – К.: Видавництво "Фабула", 2021. – 352 с.
 5. Макдональд К. Вивчаємо Node.js. – К.: Видавництво "Діалектика", 2022. – 480 с.
 6. Фріман Е., Робсон Е. Вивчаємо JavaScript. – Львів: Видавництво "Артбук", 2021. – 700 с.
 7. Aiven.io – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://aiven.io/docs/products/mysql/> – Дата звернення: 20.04.2025.
 8. Express.js – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://expressjs.com/> – Дата звернення: 27.04.2025.
 9. Fisher-Yates Shuffle Algorithm [Електронний ресурс]. – Режим доступу: https://en.wikipedia.org/wiki/Fisher%E2%80%93Yates_shuffle – Дата звернення: 03.05.2025.
 10. MySQL – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://dev.mysql.com/doc/> – Дата звернення: 20.03.2025.
 11. Node.js – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://nodejs.org/> – Дата звернення: 10.05.2025.
 12. React – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://react.dev/> – Дата звернення: 22.05.2025.
- Socket.IO – Офіційна документація [Електронний ресурс]. – Режим доступу: <https://socket.io/docs/> – Дата звернення: 28.05.2025.

ДОДАТОК А. Лістинги основних класів та модулів

```
/**
Клас моделі користувача (User.js)
*/
const { DataTypes } = require('sequelize');
const bcrypt = require('bcryptjs');

module.exports = (sequelize) => {
  const User = sequelize.define('User', {
    id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
    username: { type: DataTypes.STRING, unique: true, allowNull: false },
    email: { type: DataTypes.STRING, unique: true, allowNull: false },
    password: { type: DataTypes.STRING, allowNull: false },
    chips: { type: DataTypes.INTEGER, defaultValue: 1000 }
  }, {
    tableName: 'users',
    timestamps: false
  });

  User.prototype.validPassword = function(password) {
    return bcrypt.compareSync(password, this.password);
  };

  return User;
};
/**
Клас моделі гри в покер (PokerGame.js)
*/
const { DataTypes } = require('sequelize');

module.exports = (sequelize) => {
  const PokerGame = sequelize.define('PokerGame', {
    id: { type: DataTypes.INTEGER, primaryKey: true, autoIncrement: true },
    type: { type: DataTypes.STRING, defaultValue: 'poker', allowNull: false },
    players: { type: DataTypes.JSON, defaultValue: [] },
    pot: { type: DataTypes.INTEGER, defaultValue: 0 },
    deck: { type: DataTypes.JSON, defaultValue: [] },
    status: { type: DataTypes.ENUM('waiting', 'playing', 'finished',
    'eliminated', 'replaced'), defaultValue: 'waiting' },
    settings: { type: DataTypes.JSON, defaultValue: { maxPlayers: 4,
    smallBlind: 10, bigBlind: 20 } },
    winner: { type: DataTypes.STRING, allowNull: true, defaultValue: null },
    showdown: { type: DataTypes.BOOLEAN, defaultValue: false },
    userId: { type: DataTypes.INTEGER, allowNull: true, field: 'user_id' }
  }, {
    tableName: 'poker_games',
    timestamps: false
  });

  return PokerGame;
};
/**
Основний контролер для створення гри (game.js)
*/
const express = require('express');
const router = express.Router();
const pokerController = require('../controllers/pokerController');
const blackjackController = require('../controllers/blackjackController');

// Створення нової гри
router.post('/create', async (req, res) => {
  try {
```

```

    const { type, userId, username } = req.body;
    if (type === 'blackjack') {
      return await blackjackController.createGame(req, res);
    } else {
      return await pokerController.createGame(req, res);
    }
  } catch (error) {
    res.status(500).json({ message: 'Помилка при створенні гри' });
  }
});

module.exports = router;/**
 * Основний компонент клієнта для гри в покер (PokerGame.js)

import React, { useState, useEffect } from 'react';
import { useParams, useNavigate } from 'react-router-dom';
import { useAuth } from '../contexts/AuthContext';
import axios from 'axios';
import { io } from 'socket.io-client';
import defaultAvatar from '../assets/default_avatar.png';
import { API_URL } from '../config/api';

function PokerGame() {
  const { gameId } = useParams();
  const { user } = useAuth();
  const navigate = useNavigate();
  const [socket, setSocket] = useState(null);
  const [gameData, setGameData] = useState(null);
  useEffect(() => {
    const fetchGameData = async () => {
      try {
        const response = await axios.get(`${API_URL}/api/poker/${gameId}`);
        setGameData(response.data);
      } catch (error) {
        setError('Не вдалося завантажити дані гри');
      }
    };
    if (gameId) fetchGameData();
  }, [gameId, user]);

  // ... решта логіки компонента ...
}
export default PokerGame;

```

Додаток Б. Слайди мультимедійної презентації

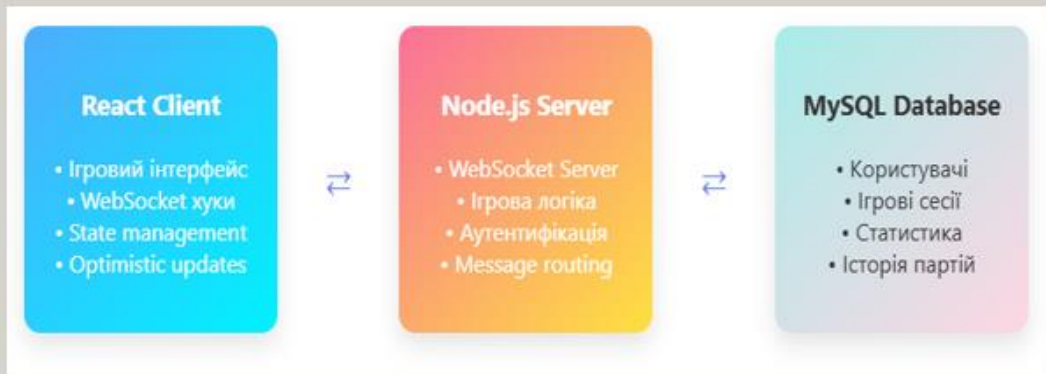
Розробка комп'ютерної гри "Техаський холдем" у web-середовищі

*Виконав
студент 4 курсу
Групи РП-08
Трохимчук Я.С.*

Одеса-2025

Методологічна схема комплексного технічного аналізу онлайн- платформ





Трьохрівнева архітектура веб-додатка



Схема бази даних



Блок-схема визначення переможця



Схема алгоритму карткової гри

Блок-схема алгоритму реалізації системи ставок

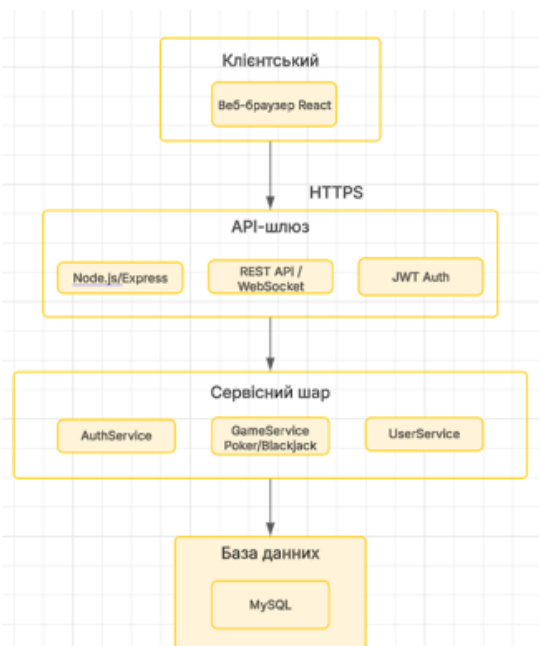


Схема архітектури веб-платформи

Скріншоти з проекту



РЕЦЕНЗІЯ

на дипломний проект (роботу) здобувача (здобувачки) освіти
відділення комп'ютерних систем

Трохимчук Ярослав Сергійович

(прізвище, ім'я та по батькові)

Спеціальність 121 "Інженерія програмного забезпечення"

Освітня програма «Розробка програмного забезпечення»

Керівник дипломного проекту (роботи) Залапін Олексій Ігорович

(прізвище, ім'я та по батькові)

Тема дипломного проекту (роботи) Розробка комп'ютерної гри «Техаський холдем» у web-середовищі

Обсяг розрахунково-пояснювальної записки 74 сторінок

Обсяг графічної (презентаційної) частини 14 аркушів (слайдів)

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ (РОБОТИ)

а) заключення про ступінь відповідності виконаного дипломного проекту (роботи) завданню
Представлений на рецензію дипломний проект повністю відповідає меті
проекткування та технічному завданню. Тематика дипломного проекту є
актуальною для своєї галузі та присвячена питанням створення комп'ютерної
гри у web-середовищі.

б) характеристика виконання кожного розділу дипломного проекту (роботи) _____
Дипломний проект складається зі вступу, трьох розділів, висновків, переліку
використаних джерел. У основному розділі розглянуті питання розробки гри
засобами Node.js для бекенду та React для фронтенду, сформовано концепцію гри
згідно до теми дипломного проекту та завданню, виконано проектування
основних аспектів гри, що розробляється. За допомогою відповідного програмного
забезпечення реалізовані всі намічені роботи з ігровим процесом.

в) оцінка якості виконання пояснювальної записки та графічної частини дипломного проекту
(роботи) Графічна частина виконана на достатньо високому рівні у вигляді
презентації із використанням офісного пакету Microsoft PowerPoint та Visio.
Пояснювальна записка виконана акуратно та у відповідності до норм оформлення
документів із використанням офісного пакету Microsoft Word. Загальна якість
виконання документації – добра, академічного плагіату у роботі не виявлено

г) перелік позитивних якостей дипломного проекту (роботи) _____

Детально описано процес виконання розробки гри;

Виконано проектування елементів гри із поясненнями на схемах та за допомогою коду;

Реалізовані архітектура системи та структура бази даних

д) основні недоліки дипломного проекту (роботи) _____

Гра має обмежену візуальну складову з точки зору ефектів, алгоритм роздачі карт та управління ігровими раундами є передбачуваним. Попри наявність численних графічних схем і архітектурних описів, деякі елементи користувацького інтерфейсу й взаємодії могли б бути розроблені детальніше.

Наявні деякі недоліки оформлення пояснювальної записки

Оцінка розрахункової частини _____ *Відмінно*

Оцінка графічної частини _____ *Добре*

Загальна оцінка _____ *Відмінно*

Прізвище, ім'я, по батькові рецензента *к.т.н. Шibaєва Наталя Олегівна*

Місце роботи і посада рецензента *Національний університет «Одеська політехніка»,
доцент кафедри інформаційних технологій*

Підпис _____

« 20 » *червня* 2025 р.



ВІДГУК

керівника на дипломний проект здобувача (здобувачки) освіти
відділення комп'ютерних систем

Трохимчука Ярослава Сергійовича

(прізвище, ім'я та по батькові)

Спеціальність: 121 "Інженерія програмного забезпечення"

Освітньо-професійна програма: «Розробка програмного забезпечення»

Тема дипломного проекту: Розробка комп'ютерної гри «Техаський холдем»
у web-середовищі

ХАРАКТЕРИСТИКА ДИПЛОМНОГО ПРОЕКТУ

а) обсяг і якість виконання проекту (графічного матеріалу і розрахунково-пояснювальної записки) Дипломний проект виконано відповідно технічному завданню. Пояснювальна записка містить 76 сторінок. У пояснювальній записці наведено етапи розробки програмного забезпечення для комп'ютерної гри «Техаський холдем» у web-середовищі. Графічна частина складається з 10 слайдів мультимедійної презентації, які також містять схеми, ER-діаграми та скріншоти веб-інтерфейсу, які передбачені технічним завданням. Якість виконання пояснювальної записки та графічної частини добра, розробку виконано в повному обсязі.

б) самостійність роботи над проектом: Протягом всього строку дипломного проектування та переддипломної практики здобувач освіти Трохимчук Я.С. поступово та послідовно виконував всі етапи розробки. Всі роботи здобувач освіти виконував самостійно, з оглядом на рекомендації керівника

в) теоретична підготовка випускника (випускниці): Здобувач освіти Трохимчук Я.С. під час роботи над дипломним проектом вивчив достатню кількість літературних джерел та матеріалів за даною тематикою. Вважаю, що теоретична підготовка дипломника добра і він готовий до захисту дипломного проекту

г) вміння розв'язувати виробничі та конструкторські питання Під час дипломного проектування здобувач освіти Трохимчук Я.С. мав змогу самостійно приймати окремі рішення з реалізації програмної та графічної частини комп'ютерної гри «Техаський холдем» у web-середовищі та показав вміння організовано працювати над поставленим завданням, розробляти структурні схеми та програмні зв'язки з застосуванням сучасних комп'ютерних програмних засобів, таких Visual Studio Code, HTML, CSS, JavaScript, а також готувати презентаційні та звітні матеріали.

Оцінка розрахункової частини Відмінно
Оцінка графічної частини Відмінно
Загальна оцінка Відмінно

Прізвище, ім'я, по батькові керівника дипломного проекту Залатін Олексій Ігорович

Місце роботи і посада керівника дипломного проекту ВСП "Одеський технічний фаховий коледж ОНТУ", викладач спеціалізації комісії комп'ютерних технологій та програмної інженерії

Підпис 

« 16 » червень 2025 р.

**ДОЗВІЛ
НА РОЗМІЩЕННЯ
ВИПУСКНОЇ КВАЛІФІКАЦІЙНОЇ РОБОТИ
(ДИПЛОМНОГО ПРОЕКТУ)
В ЕЛЕКТРОННОМУ РЕПОЗИТАРІЇ ВСП «ОТФК ОНТУ»**

Ми, що нижче підписалися,

Трохимчук Ярослав Сергійович,
здобувач освіти гр. 4РП-08, та

Залапін Олексій Ігорович,
керівник дипломного проекту,

не заперечуємо щодо розміщення електронного варіанту пояснювальної записки до дипломного проекту фахового молодшого бакалавра на тему:

«Розробка комп'ютерної гри «Техаський холдем» у web-середовищі» (автор роботи – Трохимчук Я.С., керівник роботи – Залапін О.І.)

виконаного у ВСП «Одеський технічний фаховий коледж Одеського національного технологічного університету» в 2025 році, у повному обсязі в електронному репозитарії ВСП «ОТФК ОНТУ» для вільного доступу через мережу Інтернет.

Несемо відповідальність за ідентичність електронного та друкованого варіантів випускної кваліфікаційної роботи і даємо згоду на обробку персональних даних.

Виконавець



/ Трохимчук Я.С. /

Керівник



/ Залапін О.І. /

«16» червня 2025 р.

ДОВІДКА


циклової комісії КТ та ПІ
про допуск до захисту дипломного проєкту
здобувача (здобувачки) освіти IV курсу
відділення комп'ютерних систем групи 4РП-08

Трохимчука Ярослава Сергійовича

на тему *Розробка комп'ютерної гри "Техаський холдем"*
у web-середовищі

Висновок відповідальної особи за проведення нормоконтролю:

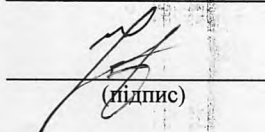
пояснювальна записка до дипломного проєкту виконана з некритичними
порушеннями ДСТУ та оформлена відповідно до вимог Положення про
дипломне проєктування


(підпис)

16.06.2025
(дата)

Петрашова В.І.
(П.І.Б.)

Висновок відповідальної особи за перевірку роботи на наявність академічного
плагіату *згідно звіту про перевірку від 15.06.2025 р. значення коефіцієнту*
подібності в роботі становить 4,34%, коефіцієнт цитування – 1,15%.


(підпис)

16.06.2025
(дата)

Краснокутська К.Г.
(П.І.Б.)

Попередня експертиза (малий захист) дипломного проєкту

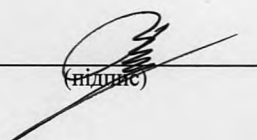
здобувача (здобувачки) освіти

Трохимчука Я.С.
(П.І.Б.)

проведена « *16* » *червня* *2025* р.

Висновки *Пояснювальна записка до дипломного проєкту виконана у повному*
обсязі. Випускна кваліфікаційна робота (дипломний проєкт) відповідає
вимогам Положення про дипломне проєктування та рекомендована до
захисту.

Голова ЦК КТ та ПІ


(підпис)

Кривченко Ю.В.
(П.І.Б.)

Звіт подібності

метадані

Назва організації

Odesa Technical Professional College of Odesa National University of Technology

Заголовок

Розробка комп'ютерної гри "Техаський холдем" у web-середовищі

Автор

Науковий керівник / Експерт

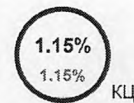
Трохимчук Ярослав Сергійович Залапін Олексій Ігорович

підрозділ

Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету"

Обсяг знайдених подібностей

Коефіцієнт подібності визначає, який відсоток тексту по відношенню до загального обсягу тексту було знайдено в різних джерелах. Зверніть увагу, що високі значення коефіцієнта не автоматично означають плагіат. Звіт має аналізувати компетентна / уповноважена особа.



25

Довжина фрази для коефіцієнта подібності 2

11597

Кількість слів

93350

Кількість символів

Тривога

У цьому розділі ви знайдете інформацію щодо текстових спотворень. Ці спотворення в тексті можуть говорити про МОЖЛИВІ маніпуляції в тексті. Спотворення в тексті можуть мати навмисний характер, але частіше характер технічних помилок при конвертації документа та його збереженні, тому ми рекомендуємо вам підходити до аналізу цього модуля відповідально. У разі виникнення запитань, просимо звертатися до нашої служби підтримки.

Заміна букв		20
Інтервали		4
Мікропробіли		6
Білі знаки		0
Парафрази (SmartMarks)		25

Подібності за списком джерел

Нижче наведений список джерел. В цьому списку є джерела із різних баз даних. Колір тексту означає в якому джерелі він був знайдений. Ці джерела і значення Коефіцієнту Подібності не відображають прямого плагіату. Необхідно відкрити кожне джерело і проаналізувати зміст і правильність оформлення джерела.

10 найдовших фраз

Колір тексту

ПОРЯДКОВИЙ НОМЕР	НАЗВА ТА АДРЕСА ДЖЕРЕЛА URL (НАЗВА БАЗИ)	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	32 0.28 %
2	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	25 0.22 %
3	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	23 0.20 %
4	https://card-file.ontu.edu.ua/bitstreams/5240e379-7721-49f0-8ee8-27140b0b473a/download	23 0.20 %
5	https://mylektsii.su/9-86674.html	22 0.19 %

6	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	21 0.18 %
7	https://card-file.ontu.edu.ua/bitstreams/34a6756b-592f-4b77-a805-183aa03a6a26/download	21 0.18 %
8	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	20 0.17 %
9	Розробка програмного забезпечення для відстеження ступеня готовності виконання проекту 6/15/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	19 0.16 %
10	2016_6030601_Rudyk_Luliia_Mykhailivna_6026 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	19 0.16 %

з домашньої бази даних (0.40 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	Розробка програмного забезпечення для відстеження ступеня готовності виконання проекту 6/15/2025 Odesa Technical Professional College of Odesa National University of Technology (Відокремлений структурний підрозділ "Одеський технічний фаховий коледж Одеського національного технологічного університету")	46 (4) 0.40 %

з програми обміну базами даних (0.49 %)

ПОРЯДКОВИЙ НОМЕР	ЗАГОЛОВОК	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	2016_6030601_Rudyk_Luliia_Mykhailivna_6026 10/25/2024 National University "Lviv Politechnika" (National University Lviv Politechnika)	19 (1) 0.16 %
2	КР Хміль 0 6/13/2025 Sambir Applied College of Economics and Information Technologies (Sambir Applied College of Economics and Information Technologies)	15 (1) 0.13 %
3	MP на друк.docx 12/9/2021 Ivano-Frankivsk National Technical University of Oil and Gas (Каф. ВНГ)	11 (1) 0.09 %
4	Спесівцев Кирило Андрійович.pdf 6/22/2021 Odessa National Polytechnic University (IEE, Каф. електропостачання та енергетичного менеджменту)	7 (1) 0.06 %
5	bitstream_ec6f1d5f-2c2e-448f-aae9-c54a387c6526 12/8/2024 National Technical University "Kharkiv Polytechnic Institute" students papers (National Technical University "Kharkiv Polytechnic Institute" students papers)	5 (1) 0.04 %

з Інтернету (3.45 %)

ПОРЯДКОВИЙ НОМЕР	ДЖЕРЕЛО URL	КІЛЬКІСТЬ ІДЕНТИЧНИХ СЛІВ (ФРАГМЕНТІВ)
1	https://card-file.ontu.edu.ua/bitstreams/bbaf3f38-16a8-4070-bead-5562769b7c71/download	74 (5) 0.64 %
2	https://card-file.ontu.edu.ua/server/api/core/bitstreams/a141b658-5fa7-4f90-b0bd-7f0ccaed21e5/content	50 (3) 0.43 %
3	https://card-file.ontu.edu.ua/bitstreams/29489599-0581-4ce6-8890-c3b13d9f2e0e/download	44 (3) 0.38 %

