

Розробка кросплатформеної ігрової системи з динамічною моделлю винагород у середовищі Testnet

Вінницький національний технічний університет

Анотація

У даній статті розглянуто процес проектування кросплатформеної ігрової системи «Lucky Money» з динамічною моделлю винагород у середовищі Testnet. Проаналізовано особливості реалізації клієнт-серверної архітектури із використанням Flutter та .NET, а також інтеграцію із зовнішнім сервісом WhiteBit Testnet для проведення тестових фінансових операцій. Описано механізм реалізації ігрового процесу, що базується на ризик-орієнтованій моделі винагород, JWT-авторизації та взаємодії із REST API. Практичне застосування UML-моделювання та CASE-засобу Visual Paradigm дозволило формалізувати функціональні можливості системи та спростити процес її проектування.

Ключові слова: Flutter, Testnet, WhiteBit, UML, JWT, REST API, кросплатформений застосунок, ігрова система, динамічна модель винагород

Abstract

This article examines the design process of the cross-platform gaming system “Lucky Money” with a dynamic reward model in the Testnet environment. The study analyzes the implementation features of a client-server architecture using Flutter and .NET, as well as integration with the WhiteBit Testnet service for performing test financial transactions. The paper describes the implementation of the game process based on a risk-oriented reward model, JWT authorization, and REST API interaction. The practical application of UML modeling and the Visual Paradigm CASE tool made it possible to formalize the system functionality and simplify the software design process.

Keywords: Flutter, Testnet, WhiteBit, UML, JWT, REST API, cross-platform application, gaming system, dynamic reward model.

Вступ

У сучасних умовах стрімкого розвитку цифрових технологій особливої актуальності набуває розробка кросплатформених програмних систем, що поєднують інтерактивний користувацький інтерфейс, безпечну взаємодію із серверною частиною та підтримку фінансових операцій у середовищах тестових мереж. Одним із перспективних напрямів є створення ігрових систем із динамічною моделлю винагород, у яких механіка ризику та винагороди реалізується за допомогою клієнт-серверної архітектури та інтеграції із зовнішніми API [1].

Використання кросплатформених фреймворків, зокрема Flutter, дозволяє забезпечити єдину кодову базу для мобільних платформ, що спрощує процес розробки та підтримки програмного забезпечення. Водночас застосування технологій JWT-авторизації та REST API забезпечує захищену взаємодію між клієнтською та серверною частинами системи [2].

Особливий інтерес становить інтеграція ігрових систем із блокчейн-орієнтованими тестовими середовищами, такими як WhiteBIT Testnet, що дозволяє моделювати фінансові операції без використання реальних цифрових активів. Це створює умови для дослідження механізмів управління балансом, обробки транзакцій та реалізації динамічних моделей винагород у безпечному тестовому середовищі [3].

Результати дослідження

Процес проектування будь-якої програмної системи розпочинається з детального аналізу функціональних можливостей, які вона повинна надавати кінцевим користувачам та зовнішнім системам. У контексті розробки ігрової системи «Lucky Money» предметна область охоплює механіки ризику, управління цифровими активами в тестовій мережі WhiteBit та забезпечення безпечного доступу до даних. Основним інструментом для опису цих взаємодій у нотації UML є діаграма прецедентів (Use Case Diagram), яка дозволяє відокремити зовнішніх суб'єктів (акторів) від внутрішніх функцій системи.

Під час аналізу системи було ідентифіковано наступних дійових осіб:

- а) Користувач (Гравець) – основний актор, який безпосередньо взаємодіє з інтерфейсом Flutter-застосунку для здійснення ігрових дій;
- б) WhiteBit API (Testnet) – зовнішня система (вторинний актор), що забезпечує проведення транзакцій поповнення та підтвердження наявності коштів на балансі;
- в) Адміністратор – актор, відповідальний за моніторинг ігрових сесій та перегляд загальної статистики системи.

Центральним елементом діаграми є прецеденти, що відображають конкретні сценарії використання. Для ігрової системи ключовими функціями є авторизація, управління балансом та безпосередньо ігровий процес. Оскільки система використовує JWT-токени, прецедент «Авторизація у системі» є базовим і включається (відношення «include») у більшість інших операцій, що потребують ідентифікації користувача.

Ігровий цикл описується через прецедент «Проведення ігрової сесії», який передбачає вибір кількості бомб та розміру ставки. Динамічна модель винагород реалізується через ітераційне відкриття клітинок, де кожна успішна дія збільшує коефіцієнт виплати. Важливою особливістю є можливість прецеденту «Здійснити Cashout», який дозволяє гравцю зупинити гру та зафіксувати прибуток до моменту потрапляння на міну.

Взаємодія з блокчейн-середовищем винесена в окремий блок прецедентів, що стосуються фінансових операцій. Прецедент «Поповнити ігровий баланс» ініціює запит до WhiteBit API, забезпечуючи перевірку успішності транзакції в мережі Testnet перед зарахуванням віртуальних коштів у внутрішню базу даних .NET бекенду.

На рисунку 1 наведено структуру діаграми, яку створено у середовищі Visual Paradigm. Вона демонструє ієрархію акторів та логічне групування функцій

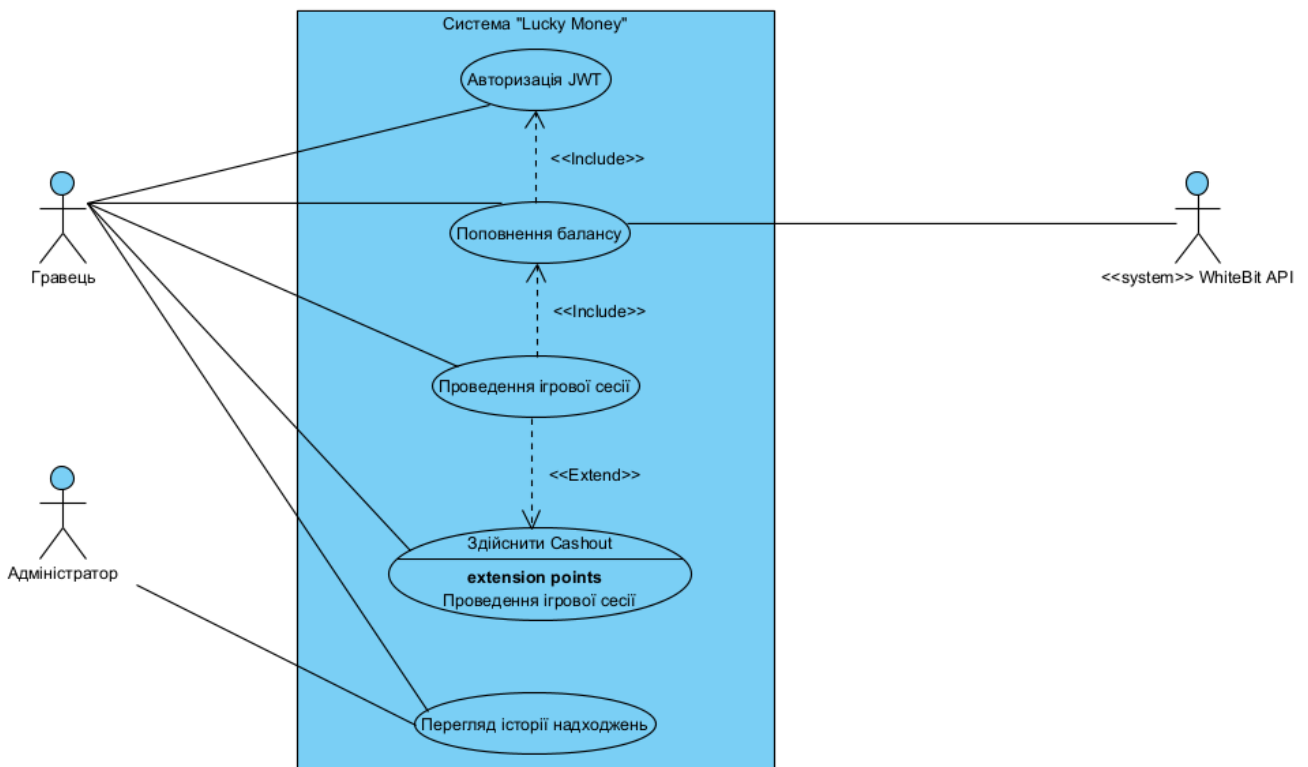


Рисунок 1 – Діаграма прецедентів ігрової системи

Для деталізації логіки обрано прецедент «Проведення ігрової сесії». Потік подій описує взаємодію користувача з інтерфейсом Flutter та обробку даних на сервері .NET.

Актори:

1. Гравець – авторизована фізична особа, яка має доступ до ігрового інтерфейсу, здійснює ставки та керує власним балансом.
2. Адміністратор – співробітник, що має розширені права для аналізу ігрової активності та технічного стану системи.

3. WhiteBit API – зовнішній програмний інтерфейс блокчейн-біржі, що використовується для обробки транзакцій у мережі Testnet.

Прецеденти:

1. Авторизація (JWT) – процес перевірки облікових даних користувача та видачі токена доступу для подальшої взаємодії з API.

2. Проведення ігрової сесії – основний процес, що включає генерацію мінного поля, обробку вибору клітинок та розрахунок проміжних коефіцієнтів.

3. Здійснити Cashout – дія, що дозволяє гравцю припинити гру та перевести накопичений виграш на основний баланс.

4. Поповнення балансу – фінансова операція, яка пов'язує гаманець користувача у Testnet із внутрішнім балансом системи.

5. Перегляд історії надходжень – забезпечує доступ до архівних даних про проведені ігрові сесії та фінансові операції

Основний потік:

а) Гравець відкриває екран гри та вказує суму ставки;

б) система перевіряє наявність достатнього балансу через API;

в) Гравець обирає кількість мін (динамічна складність) та підтверджує початок гри;

г) сервер генерує поле та повертає ідентифікатор сесії;

д) Гравець по черзі відкриває клітинки;

е) при кожному успішному відкритті система перераховує поточний коефіцієнт винагороди;

ж) Гравець натискає «Cashout», і сума виграшу зараховується на баланс.

Альтернативний потік (Програш):

а) Якщо Гравець відкриває клітинку з міною, сесія завершується;

б) ставка анулюється, дані про програш зберігаються в історії.

Висновки

У результаті дослідження було проаналізовано процес проектування кросплатформеної ігрової системи «Lucky Money» із динамічною моделлю винагород у середовищі Testnet. Побудована UML-діаграма прецедентів дозволила формалізувати взаємодію між основними акторами системи та визначити ключові функціональні можливості застосунку.

Встановлено, що використання Flutter для реалізації клієнтської частини забезпечує кросплатформеність та зручну інтеграцію із серверним API, а застосування JWT-авторизації підвищує рівень безпеки доступу до ресурсів системи. Інтеграція з WhiteBit Testnet дозволила реалізувати механізми поповнення балансу та перевірки транзакцій у тестовому середовищі без використання реальних цифрових активів.

Отримані результати підтверджують доцільність використання UML-моделювання та CASE-засобів під час проектування сучасних ігрових систем із фінансовими механіками та клієнт-серверною архітектурою.

СПИСОК ВИКОРИСТАНОЇ ЛІТЕРАТУРИ

1. Flutter Documentation. Flutter documentation : вебсайт. URL: <https://docs.flutter.dev>
2. JWT Introduction – Auth0. JSON Web Tokens (JWT) : [Електронний ресурс] – Режим доступу до ресурсу: <https://auth0.com/docs/secure/tokens/json-web-tokens>
3. WhiteBIT API Documentation. WhiteBIT API Documentation : вебсайт. [Електронний ресурс] – Режим доступу до ресурсу:: <https://docs.whitebit.com>

Stoliar Vladislav Vasilyovich – студент четвертого курсу групи ІПІ-25м, ФІТКІ, Вінницький національний технічний університет, Вінниця, e-mail: vlad100vntu@gmail.com

Ліщинська Людмила Броніславівна – д-р техн. наук, професор, професор кафедри програмного забезпечення, Вінницький національний технічний університет, м. Вінниця, e-mail: llb@vntu.edu.ua

Stoliar Vladislav Vasi— fourth-year student of group ІPI-25m, FITKI, Vinnytsia National Technical University, Vinnytsia, e-mail: vlad100vntu@gmail.com.

Lishchynska Lyudmyla Bronislavivna – Dr. Sc. (Eng.), Full Professor, Professor of Program Engineering, Vinnytsia National Technical University, Vinnytsia, e-mail: llb@vntu.edu.ua