Sprawozdanie z projektu – Program *G4B Game Store*

Amadeusz Żyła
Nr indeksu 43907

Sebastian Wróblewski
Nr indeksu 43902

Adrian Jurak
Nr indeksu 43847

PWSZ w Legnicy – Wydział Nauk Technicznych i Ekonomicznych
Informatyka, semestr IV – Legnica 2025

Przedmiot: Projektowanie i Programowanie Obiektowe (PIPO)

Rok akademicki 2024/2025

3 czerwca 2025

Spis treści

Rozdział 1

Ogólny zarys programu

"MVC Game Store"to pełnowartościowa aplikacja webowa oparta o architekturę Model–View–Controller (MVC), przeznaczona do sprzedaży cyfrowych gier komputerowych. System pozwala użytkownikom na przeglądanie oferty, zakup gier, zarządzanie koszykiem, subskrypcjami oraz historią zamówień. Kluczowe cechy:

- **Wielojęzyczność** – dynamiczne przełączanie języka interfejsu użytkownika (i18n) bez odświeżania sesji.
- **Paginacja** – dzielenie wyników na strony z możliwością przechodzenia między nimi bez przeładowywania całej strony.
- **Filtrowanie wyników** – interaktywne zawężanie listy gier na podstawie wybranych kryteriów, z natychmiastową aktualizacją widoku.
- **Przeglądanie gier** – dynamiczne wyświetlanie listy produktów z możliwością filtrowania po gatunku, platformie i przedziale cenowym.

Repozytorium projektu: `https://github.com/AdrianJurak/GameKeySite-Project-PSBD`

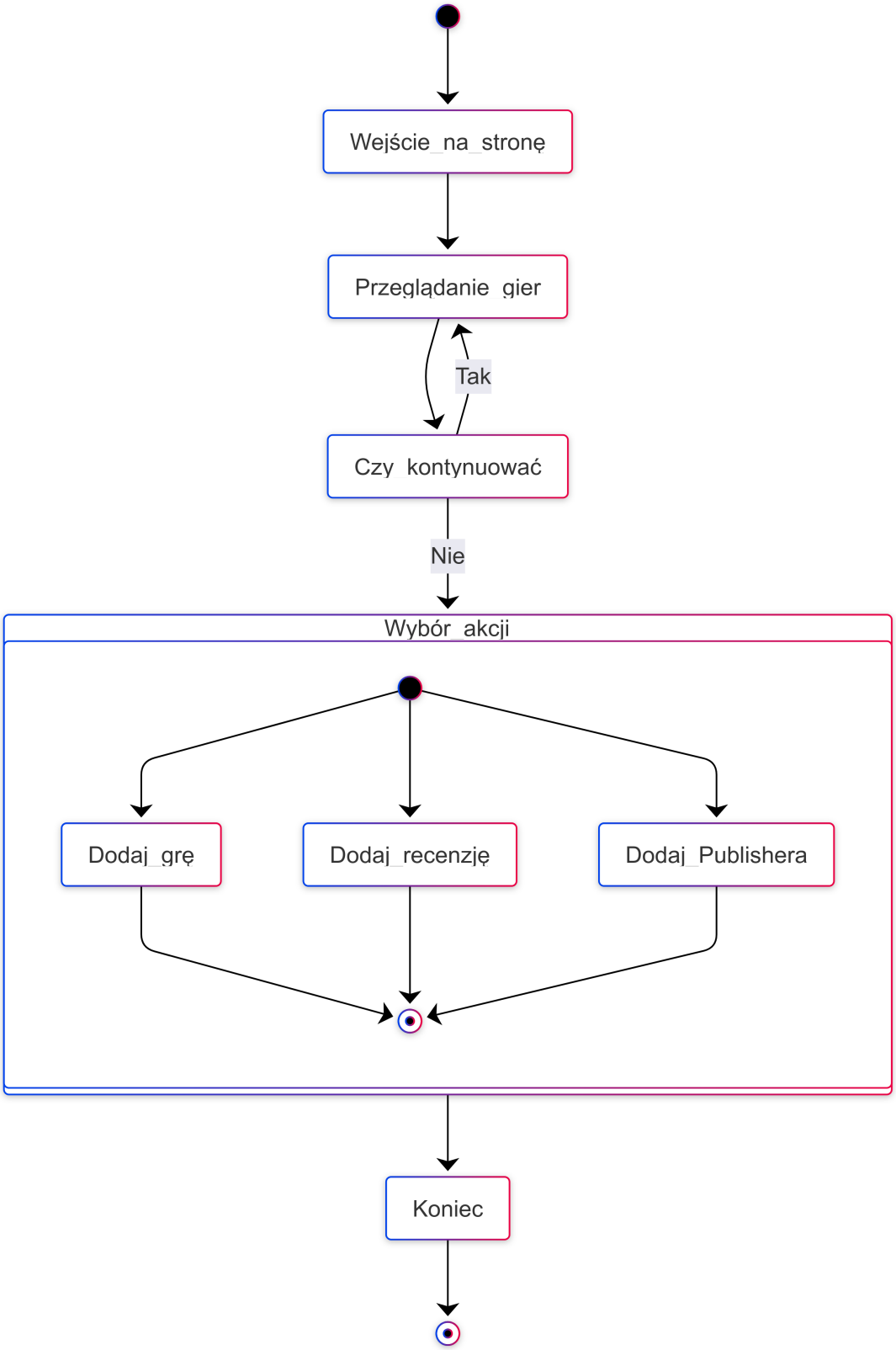
Baza danych MySQL przechowuje informacje o grach, wydawcach i recenzjach, a aplikacja wykorzystuje JPA/Hibernate do odwzorowania relacyjno–obiektowego.

Rozdział 2

Podstawowe funkcje programu w diagramach UML

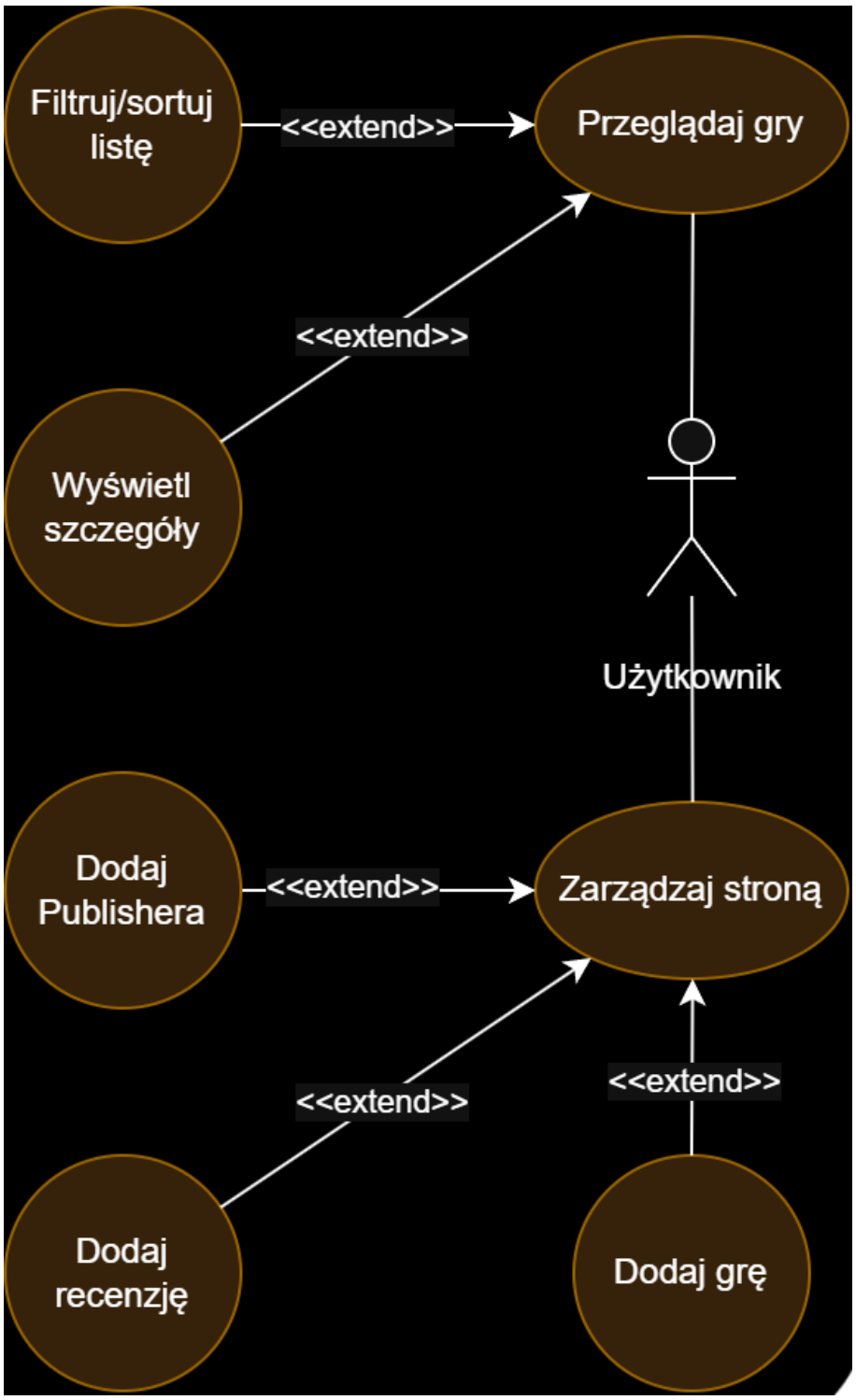
W tym rozdziale zaprezentowano cztery główne diagramy UML przygotowane w narzędziu `diagrams.net` oraz `Mermaid.js`.

2.1 Diagram czynności (Activity Diagram)



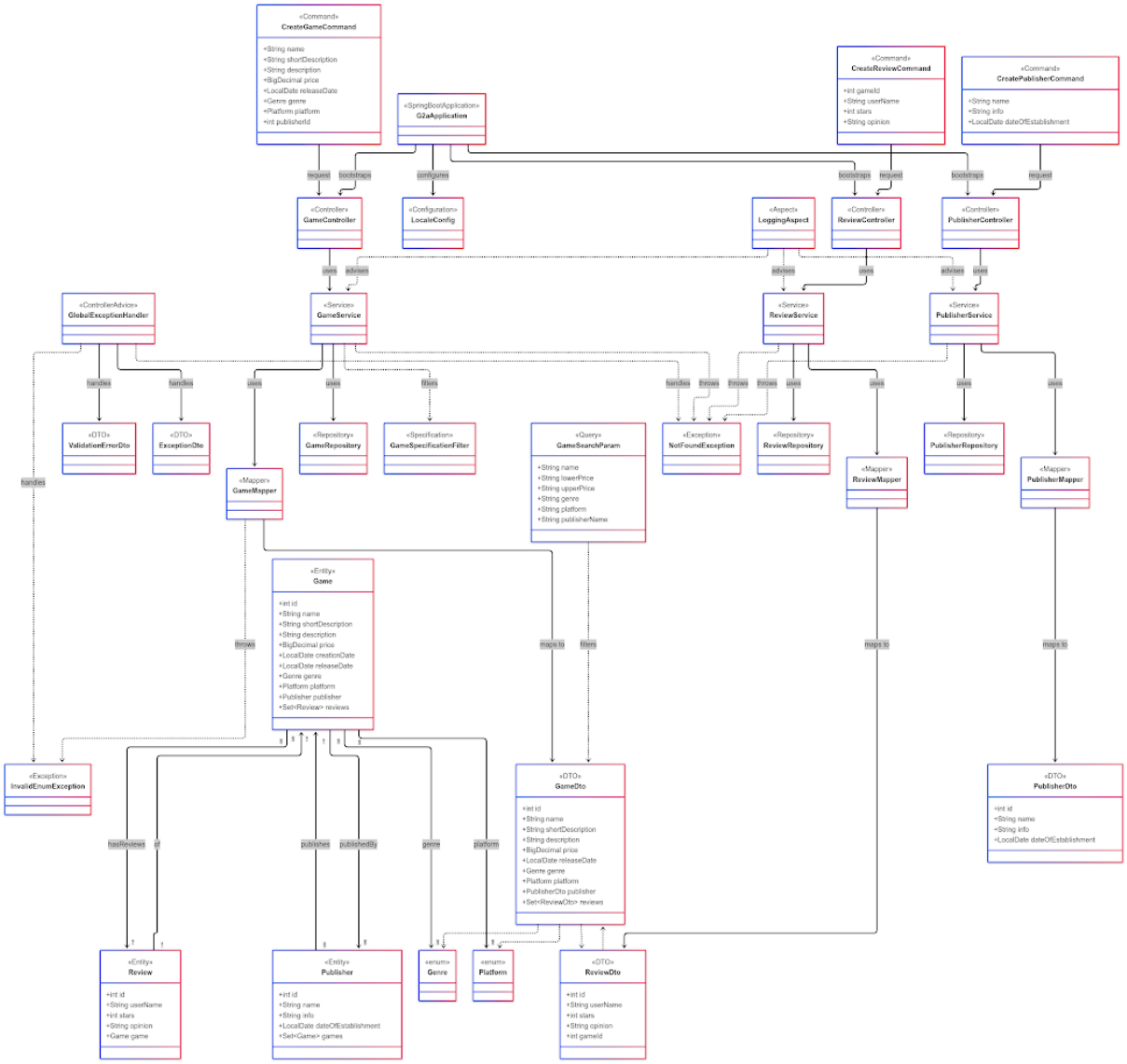
Rysunek 2.1: Przepływ czynności użytkownika w aplikacji *MVC Game Store*.

2.2 Diagram przypadków użycia (Use Case)



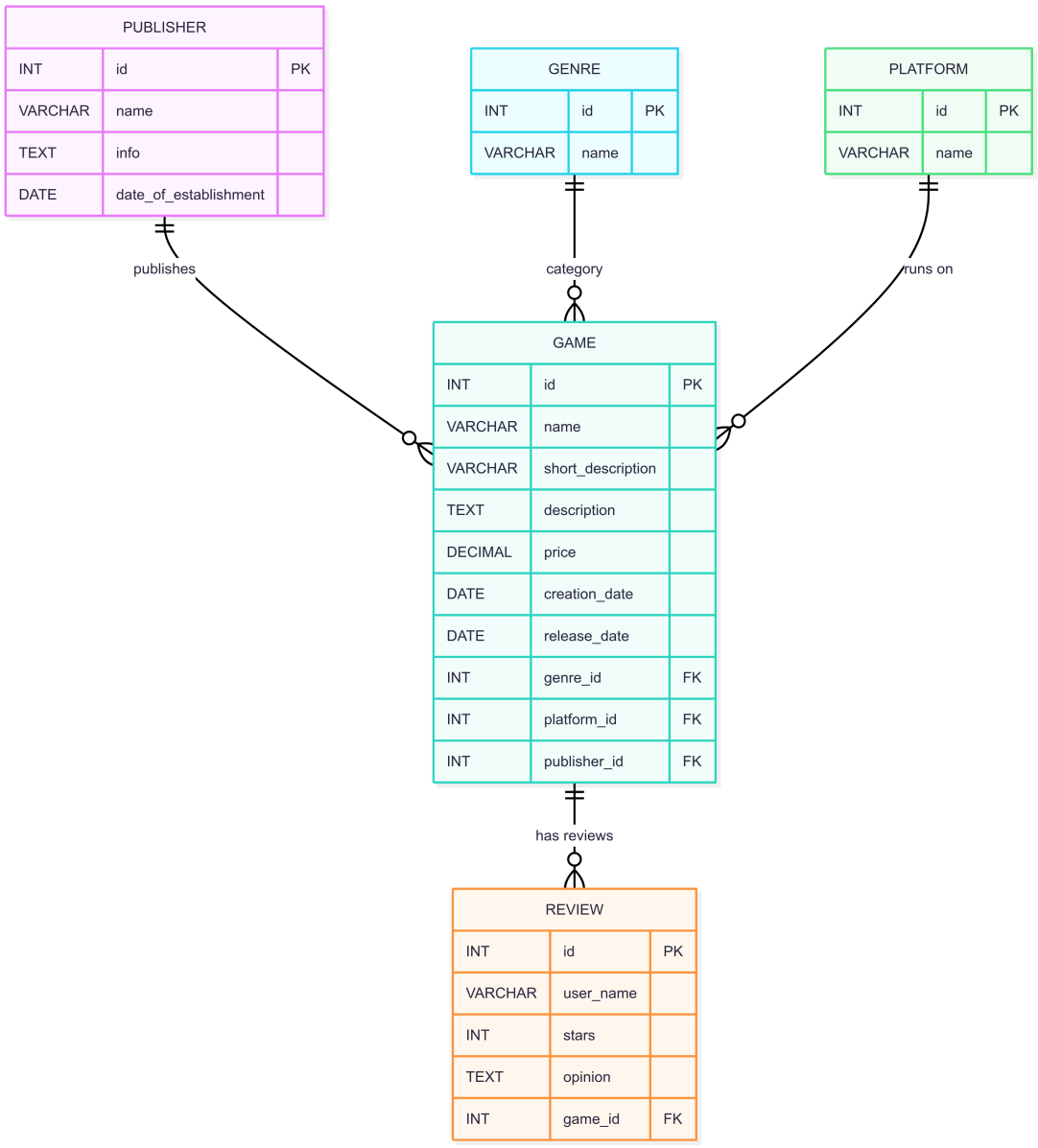
Rysunek 2.2: Przypadki użycia oraz aktorzy systemu.

2.3 Diagram klas (Class Diagram)



Rysunek 2.3: Struktura klas domenowych i serwisów.

2.4 Diagram schematu bazy danych



Rysunek 2.4: Model relacyjny bazy danych MySQL.

Rozdział 3

Wbudowane funkcje i klasy

Klasa / komponent	**Zadanie**
`GameService`	Logika biznesowa związana z wyszukiwaniem, filtrowaniem i dodawaniem gier.
`GameSpecificationFilter`	Dynamiczne budowanie kryteriów wyszukiwania (Spring Data JPA `Specification`).
`GlobalExceptionHandler`	Przechwytywanie i serializacja wyjątków do formatu JSON.

Tabela 3.1: Wybrane klasy i ich przeznaczenie.

Klasy encji oznaczone są adnotacją `@Entity` oraz mapowane na tabele MySQL, a relacje odzwierciedlają zależności `many to one` / `one to many` między grami, wydawcami i recenzjami.

Rozdział 4

Dalsze możliwości rozwoju programu

- **Konteneryzacja** – przygotowanie obrazów Docker oraz orkiestracja Kubernetes.
- **Użytkownicy** – rejestracja, logowanie, zarządzanie profilem, uwierzytelnianie JWT.
- **Koszyk** – dynamiczne dodawanie i aktualizacja produktów, lokalne przechowywanie (`localStorage`), synchronizacja z bazą danych.
- **Płatności** – integracja z bramkami płatniczymi (Stripe, PayPal), obsługa transakcji i walidacja danych.
- **Historia zakupów** – filtrowanie wg daty i statusu, generowanie faktur w formacie PDF.

Rozdział 5

Opis użytych narzędzi i bibliotek

- **IntelliJ IDEA 2024.1.2** – główne IDE z obsługą Spring Boot oraz Maven.
- **Spring Boot 3.2.1** – szkielet aplikacji, wbudowany kontener Tomcat, autokonfiguracja.
- **Hibernate 6.5** – ORM z mapowaniem JPA i migracjami Flyway.
- **Thymeleaf 3.1** – silnik szablonów HTML.
- **Lombok 1.18** – generowanie kodu boilerplate (gettery, konstruktory).
- **Maven 3.9** – system budowania i zarządzania zależnościami.
- **MySQL 8.4** – relacyjna baza danych.
- **diagrams.net** – tworzenie diagramów UML (wersja 25.04.2025).
- **Mermaid.js** – generowanie diagramów w przeglądarce.

Rozdział 6

Uproszczona instrukcja obsługi i FAQ

6.1 Krótka instrukcja obsługi

6.1.1 Instalacja i konfiguracja

1. Sklonuj repozytorium:

   ```
   git clone https://github.com/AdrianJurak/GameKeySite-Project-PSBD.git
   cd GameKeySite-Project-PSBD
   ```

2. Skonfiguruj bazę danych w pliku `application.yml`.

3. Uruchom aplikację:

   ```
   mvn spring-boot:run
   ```

4. Aplikacja będzie dostępna pod adresem `http://localhost:8080`.

6.2 Najczęstsze pytania (FAQ)

Aplikacja nie uruchamia się po wpisaniu `mvn spring-boot:run`. Sprawdź, czy plik `application.yml` został poprawnie skonfigurowany (dane do bazy danych) oraz czy port 3306 nie jest zajęty.

Jak zmienić język interfejsu? Kliknij ikonę języka w górnym menu aplikacji. UI przełączy się natychmiast, bez odświeżania strony.

Dlaczego nie widzę wszystkich gier na liście? Włączona jest paginacja. Użyj przycisków *Poprzednia*/*Następna* na dole strony, aby przejść do kolejnych wyników.

Jak usunąć zastosowane filtry? Kliknij przycisk *Wyczyść filtry* znajdujący się nad listą gier.

Czy mogę filtrować wiele kategorii jednocześnie (np. platforma i gatunek)? Tak. System pozwala na nakładanie wielu filtrów równocześnie – ich efekty są łączone i stosowane w czasie rzeczywistym.