Architekturspezifikation

Take A Side

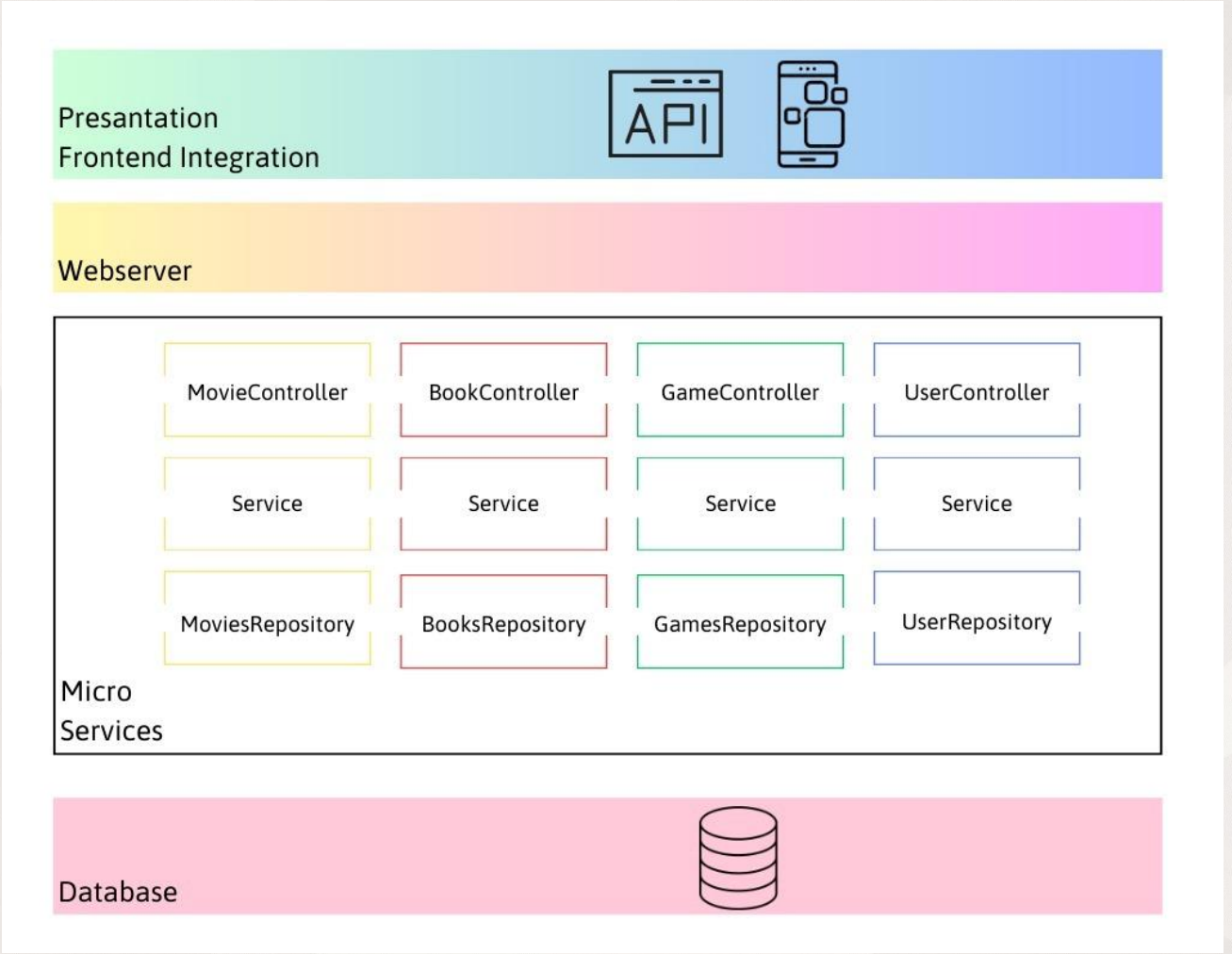
INF202 MS#3

Verantwortliche/r:
- Merve Damla İnce, e200503039@stud.tau.edu.tr
- Damla Ersoy, e200503005@stud.tau.edu.tr

Stakeholder:
- DI. Ömer Karacan, omer.karacan@tau.edu.tr

1) Architekturüberblick:

In diesem Abschnitt geben wir einen Überblick über die Architektur der Objekte Abstimmung Anwendung. Die Architektur wird in Bezug auf ihre Komponenten, Schnittstellen und Bereitstellung beschrieben.



1) Architekturüberblick:

Architektur Komponenten:

Benutzeroberfläche (UI): Die Benutzeroberfläche ist dafür verantwortlich, die Liste der Filme anzuzeigen, Benutzern die Möglichkeit zu geben, für Filme abzustimmen, und die aktuelle Stimmenzahl und den Prozentsatz für jeden Film anzuzeigen. Die Benutzeroberfläche wird mit JavaFX implementiert.

Controller-Klassen: Die Controller-Klassen behandeln die Geschäftslogik der Anwendung. Sie sind für die Verarbeitung von Benutzeranfragen, die Validierung von Eingaben und die Aktualisierung der Datenbank verantwortlich. Die Controller-Klassen werden mit dem Spring Framework implementiert.

Datenbankzugriffsschicht: Die Datenbankzugriffsschicht bietet eine Möglichkeit, auf die in der Datenbank gespeicherten Daten zuzugreifen. Es ist verantwortlich für das Abfragen der Datenbank, das Aktualisieren von Datensätzen und das Bereitstellen von Daten für die Controller-Klassen. Die Datenbankzugriffsschicht wird mit Spring Data JPA implementiert.

Datenbank: In der Datenbank speichert die Anwendung Informationen zu Filmen und Abstimmungen. Es wird mit Windows SQL implementiert.

1) Architekturüberblick:

Schnittstellen:

REST-API: Die REST-API bietet der Benutzeroberfläche eine Möglichkeit, mit den Controller-Klassen zu kommunizieren. Die API wird mit Spring MVC implementiert.

Datenbankverbindung: Die Datenbankverbindung Schnittstelle stellt eine Möglichkeit für die Datenbankzugriffsschicht bereit, sich mit der Datenbank zu verbinden. Die Schnittstelle wird mit JDBC implementiert.

Einsatz: Die Filmabstimmungsanwendung wird lokal auf einem Windows-Desktop- oder Laptop-Computer bereitgestellt. Auf die Benutzeroberfläche wird über die JavaFX-Anwendung zugegriffen, und auf die REST-API wird über HTTP-Anforderungen an einen lokalen Server zugegriffen. Die Datenbank wird auf einer lokalen Instanz von Windows SQL gehostet.

Wie Sie sehen können, ist die Architektur der Filmabstimmungsanwendung so konzipiert, dass sie in einer lokalen Umgebung und nicht auf einem Webserver funktioniert. Diese Architektur bietet eine skalierbare und wartbare Lösung zum Implementieren der Anforderungen der Anwendung auf einer einzelnen Maschine.

2) Beschreibung der „Controller" Klassen:

2.1 MovieController
Die MovieController-Klasse verarbeitet Anfragen im Zusammenhang mit Filmen. Es wird die folgenden Methoden haben:
•POST/addMovie - fügt dem System einen neuen Film hinzu.
•GET/getByYear/{year} - ruft die Details eines bestimmten Films ab, wobei {year} das Jahr des Films ist.
•GET/getByGenre/{genre} - ruft die Details eines bestimmten Films ab, wobei {genre} das Genre des Films ist.
•GET/getTopRated - ruft eine Liste der zehn am besten bewerteten Filme im System ab.
•GET/getByDirector/{director} - ruft die Details eines bestimmten Films ab, wobei {director} der Direktor des Films ist.
•GET/getByProducer/{producer} - ruft die Details eines bestimmten Films ab, wobei {producer} der Produzent des Films ist.
•GET/getByActor/{actor} - ruft die Details eines bestimmten Films ab, wobei {actor} der Schauspieler des Films ist.
•GET/getByScenarist/{scenarist} - ruft die Details eines bestimmten Films ab, wobei {scenarist} der Drehbuchautor des Films ist.
•GET/getByName/{name} - ruft die Details eines bestimmten Films ab, wobei {name} der Name des Films ist.
•PUT/updateVoteByID/{id} - aktualisiert die Abstimmungsrate eines bestimmten Films, wobei {id} die ID des Films ist.

2) Beschreibung der „Controller” Klassen:

2.2 BookController

Die BookController-Klasse verarbeitet Anfragen im Zusammenhang mit Bücher. Es wird die folgenden Methoden haben:

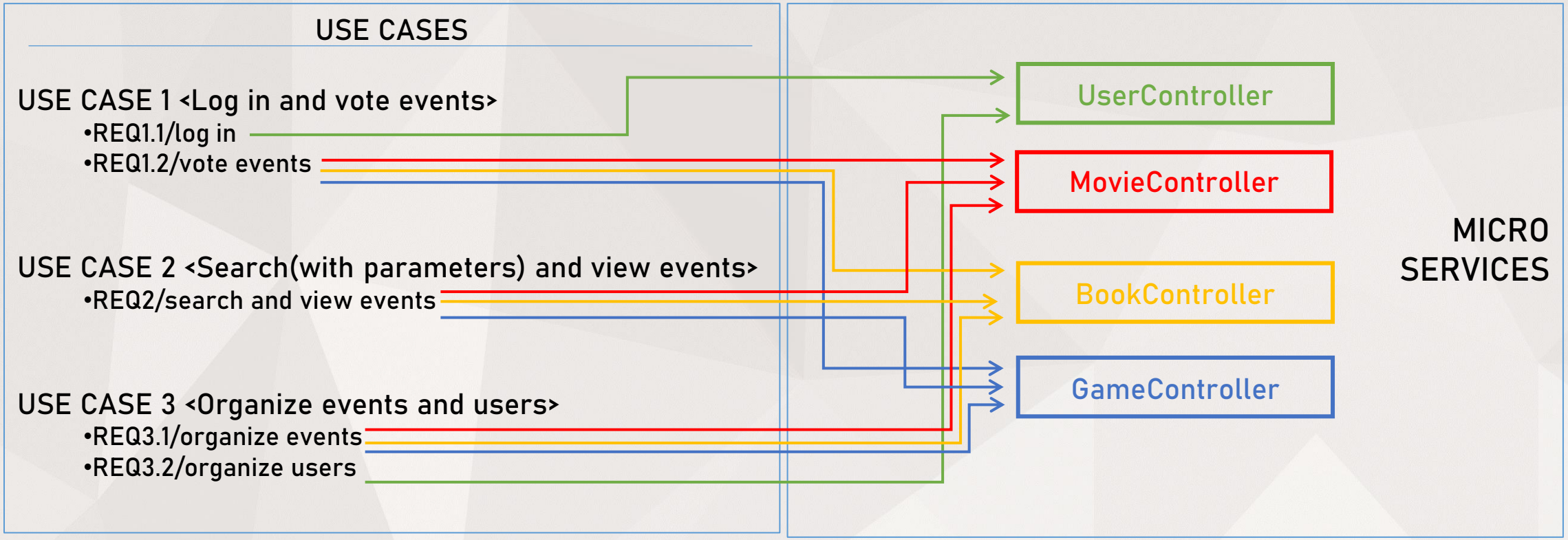
•POST/addBook – fügt dem System ein neues Buch hinzu.
•GET/getByYear/{year} – ruft die Details eines bestimmten Buchs ab, wobei {year} das Jahr des Buchs ist.
•GET/getByGenre/{genre} – ruft die Details eines bestimmten Buchs ab, wobei {genre} das Genre des Buchs ist.
•GET/getTopRated – ruft eine Liste der zehn am besten bewerteten Bücher im System ab.
•GET/getByAutor/{autor} – ruft die Details eines bestimmten Buchs ab, wobei {autor} der Autor des Buchs ist.
•GET/getByPagenumber/{page_number} – ruft die Details eines bestimmten Buchs ab, wobei {page_number} die Seitenzahl des Buchs ist.
•GET/getByPublisher/{publisher} – ruft die Details eines bestimmten Buchs ab, wobei {publisher} der Verlag des Buchs ist.
•GET/getByName/{name} – ruft die Details eines bestimmten Buchs ab, wobei {name} der Name des Buchs ist.
•PUT/updateVoteByID/{id} – aktualisiert die Abstimmungsrate eines bestimmten Buchs, wobei {id} die ID des Buchs ist.

2) Beschreibung der „Controller” Klassen:

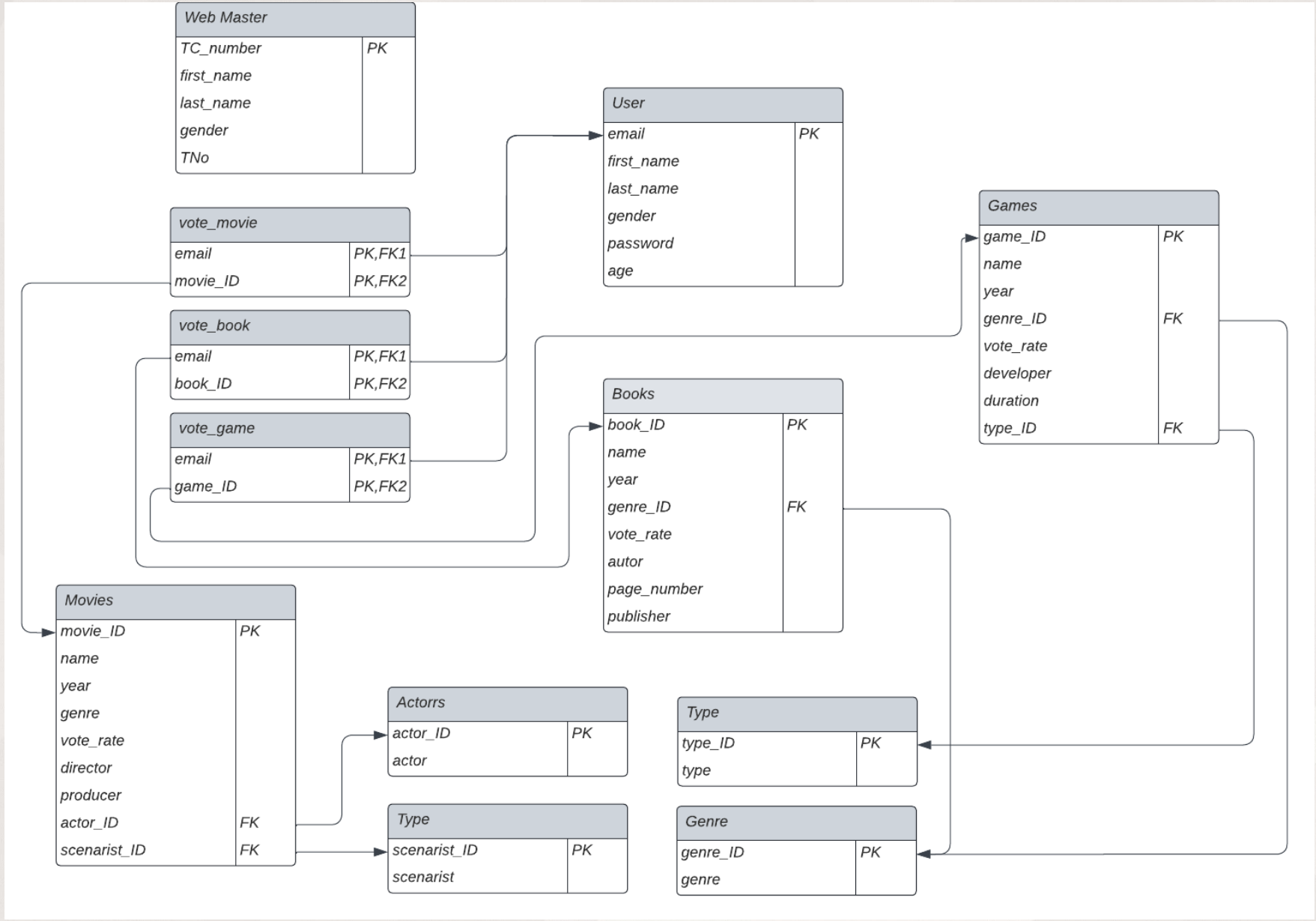
2.3 GameController
Die GameController-Klasse verarbeitet Anfragen im Zusammenhang mit Spieler. Es wird die folgenden Methoden haben:
•POST/addGame - fügt dem System ein neues Spiel hinzu.
•GET/getByYear/{year} - ruft die Details eines bestimmten Spiels ab, wobei {year} das Jahr des Spiels ist.
•GET/getByGenre/{genre} - ruft die Details eines bestimmten Spiels ab, wobei {genre} das Genre des Spiels ist.
•GET/getTopRated - ruft eine Liste der zehn am besten bewerteten Spieler im System ab.
•GET/getByDeveloper/{developer} - ruft die Details eines bestimmten Spiels ab, wobei {developer} der Entwickler des Spiels ist.
•GET/getByType/{type} - ruft die Details eines bestimmten Spiels ab, wobei {type} der Typ des Spiels ist.
•GET/getByName/{name} - ruft die Details eines bestimmten Spiels ab, wobei {name} der Name des Spiels ist.
PUT/updateVoteByID/{id} - aktualisiert die Abstimmungsrate eines bestimmten Spiels, wobei {id} die ID des Spiels ist.

2.4 UserController
Die UserController-Klasse verarbeitet Anfragen im Zusammenhang mit Benutzer. Es wird die folgenden Methoden haben:
•POST/addUser - fügt dem System einen neuen Benutzer hinzu.
•GET/ getByEmail/{email} - ruft die Details eines bestimmten Benutzers ab, wobei {email} die E-Mail-Adresse des Benutzers ist.
•PUT/ updateByEmail/{email} - aktualisiert die Benutzerinformation eines bestimmten Benutzers, wobei {email} die E-Mail-Adresse des Benutzers ist.
•DELETE/ deleteByEmail/{email} - löscht den angegebenen Benutzer, wobei {email} die E-Mail-Adresse des Benutzers ist.

3) Rückverfolgbarkeit der Anforderungen:

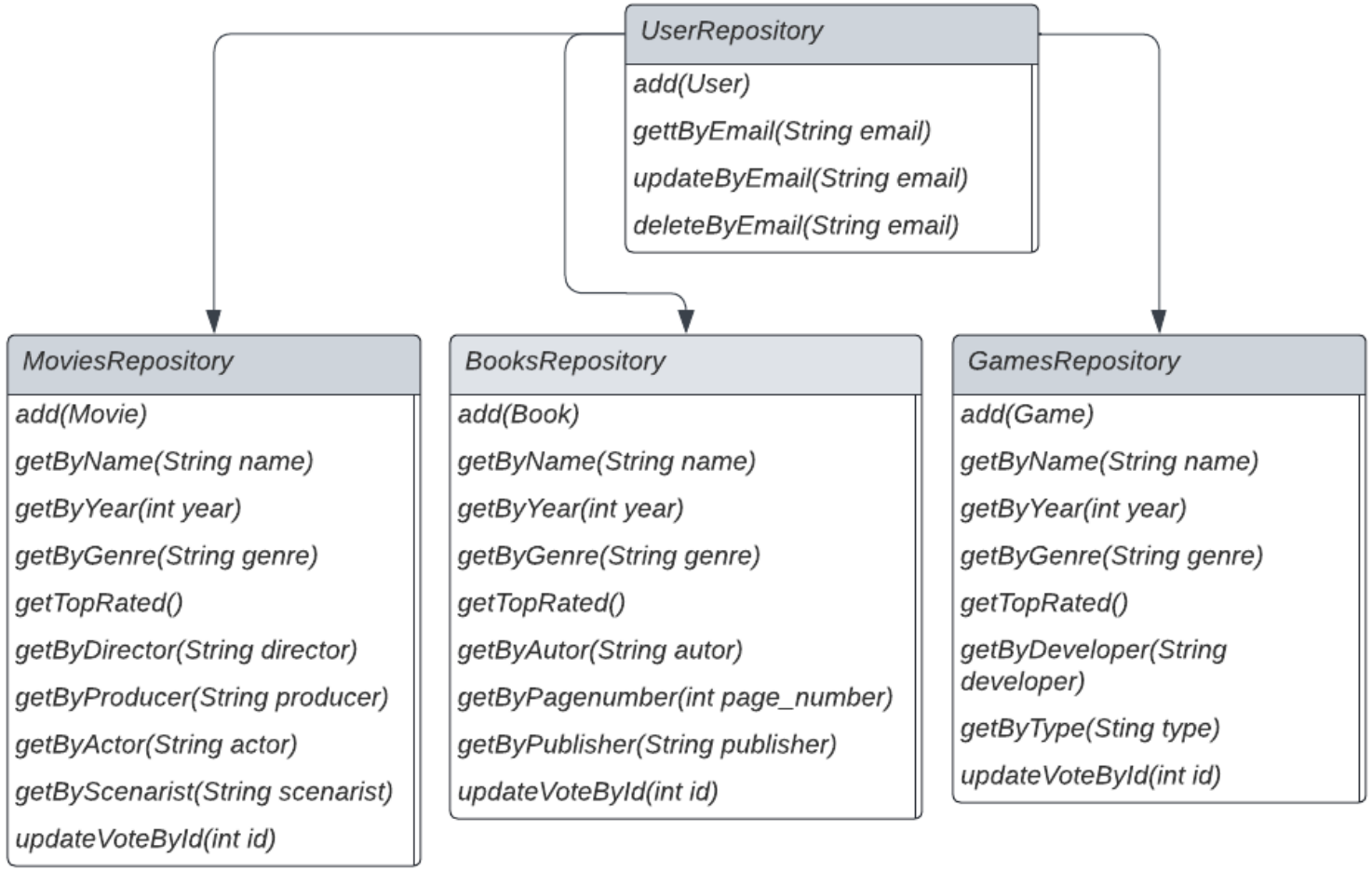


4) Beschreibung der DB-Zugriffsschicht(Daten-Modelle):



Datenbank Relationsmodell

4) Beschreibung der DB-Zugriffsschicht(Daten-Modelle):



Repository Klassen