Relazione Lavoro svolto Task 4 Terza Iterazione

Gruppo G15

1. Introduzione

Durante la terza iterazione (fase di sviluppo del prototipo) il team, nel corso di undici incontri della durata media di 2 ore, per un totale di circa 22 ore complessive, ha sviluppato un prototipo basato sugli artefatti prodotti e raffinati nella seconda iterazione. In *Figura1,* vi è una panoramica dell'architettura sviluppata finora, in seguito approfondiremo, qualora necessario, le modifiche apportate.

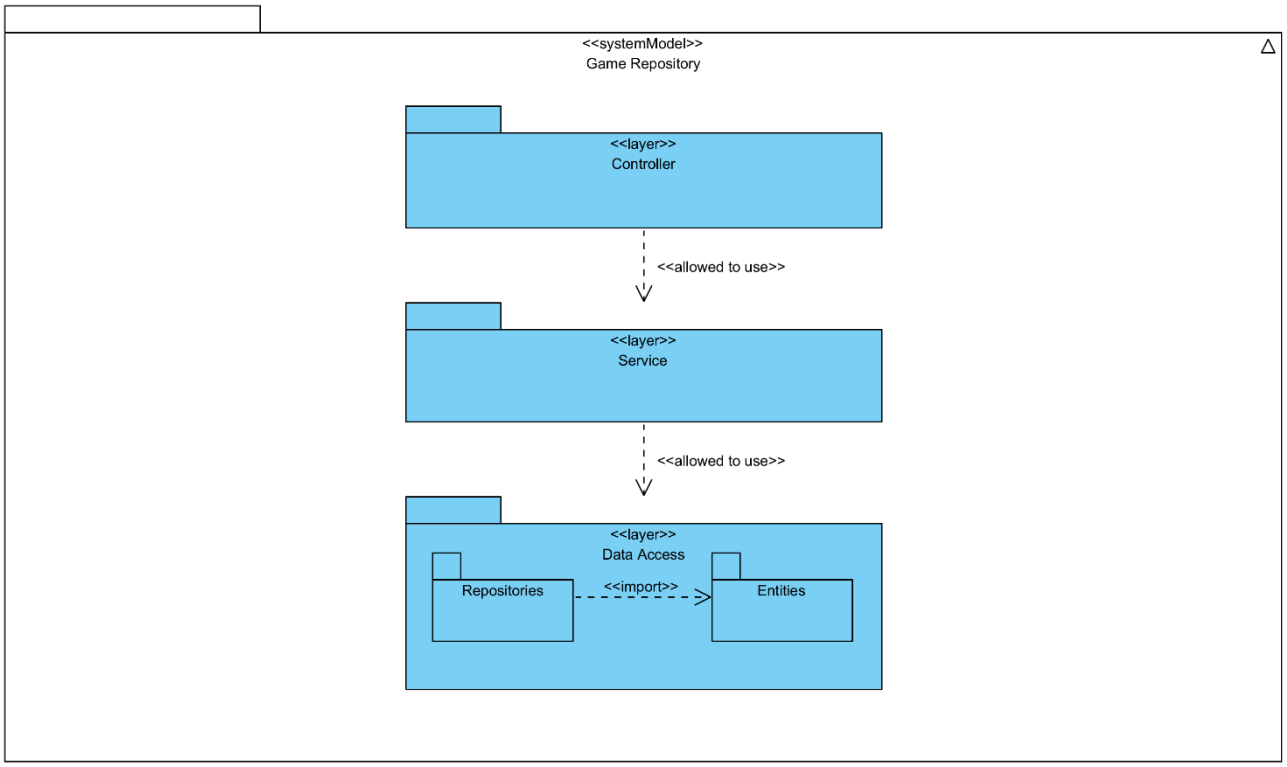


Figura 1: Architecture Diagram

2. *Class Diagram*

2.1 *Entity Diagram*

Durante lo sviluppo del software è emersa la necessità di raffinare il *Class Diagram* precedentemente prodotto, per meglio rappresentare l'architettura in oggetto.

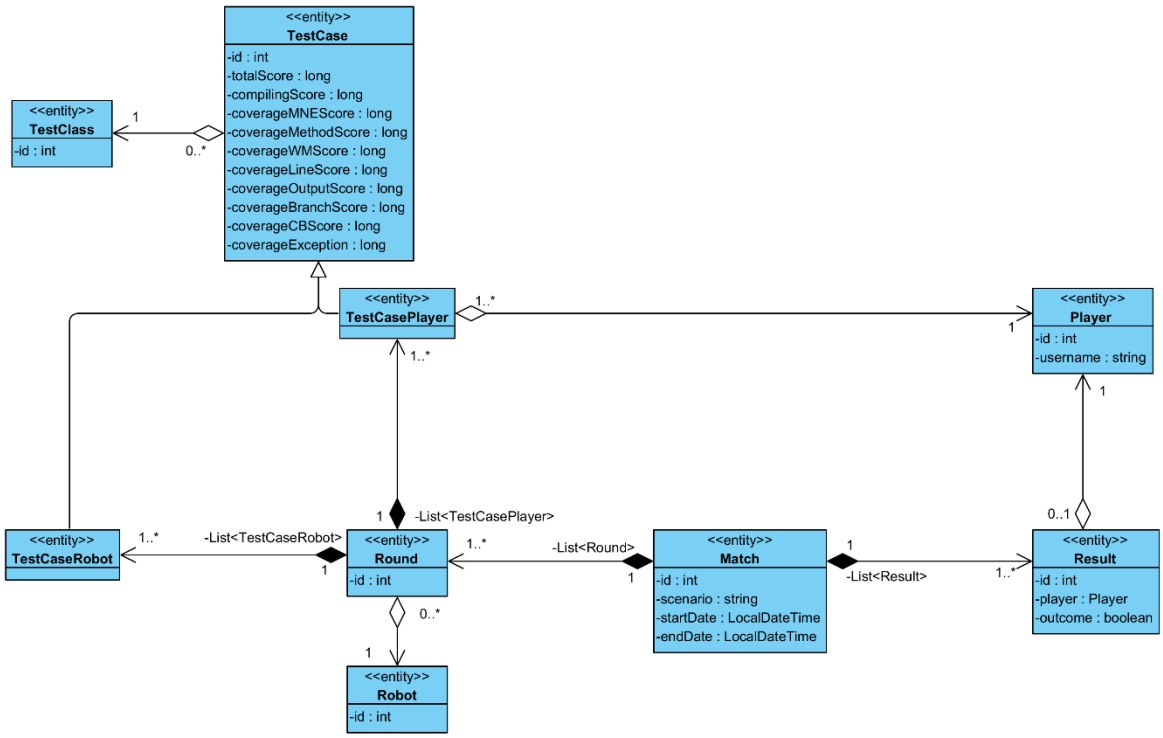


Figura 2: Class Diagram

La principale modifica apportata è quella relativa alla relazione dell'*entity* ***Result***, quest'ultima non è più associata ad un ***TestCase*** bensì ad un ***Match*** ed un ***Player***.

Grazie a questa modifica è stato risolta la difficoltà relativa all'associazione di un particolare risultato ad uno specifico giocatore.

2.2 *Repositories Class Diagram*

Per la permanenza dei dati, sfruttando un pattern *Facade,* abbiamo esposto i metodi CRUD delle diverse Repository facenti parte del nostro sistema. Quest'ultime implementano i metodi forniti dall'interfaccia *JPARepository.*

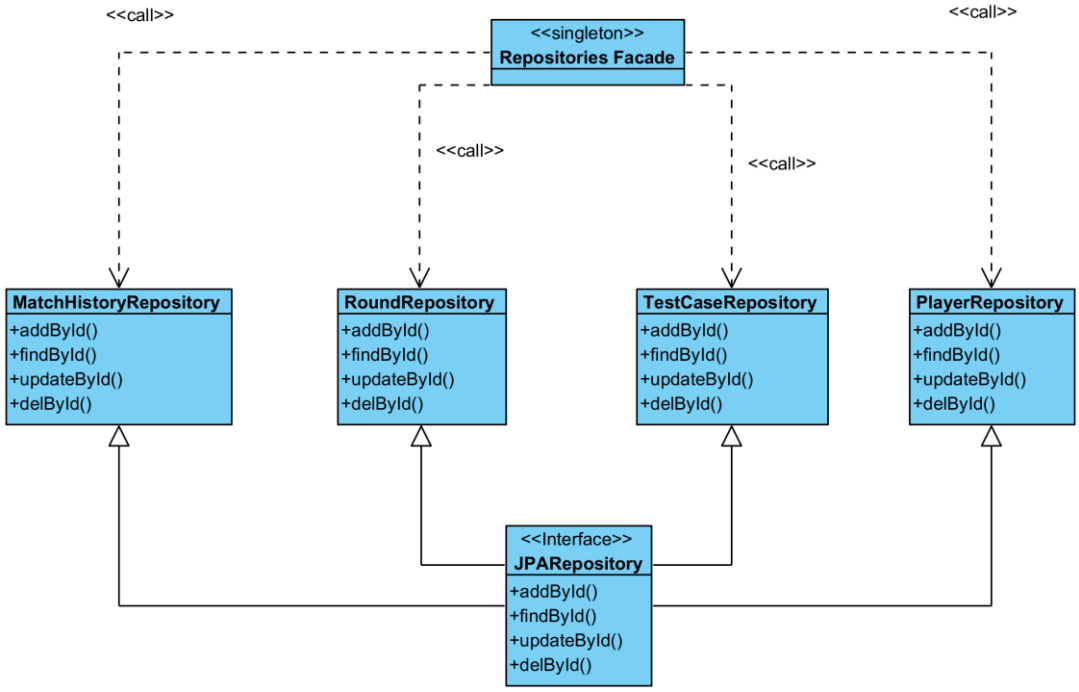


Figura 3: Repositories Class Diagram

2.3 *Service Layer Class Diagram*

Il *Service Layer*, sfruttando anch'esso un pattern di tipo *Facade,* contiene le più concrete implementazioni dei metodi CRUD sottostanti. Quindi, attraverso l'utilizzo delle *Repositories,* espone i servizi per la permanenza dei dati tenendo conto delle diverse relazioni che intercorrono tra le diverse entità.

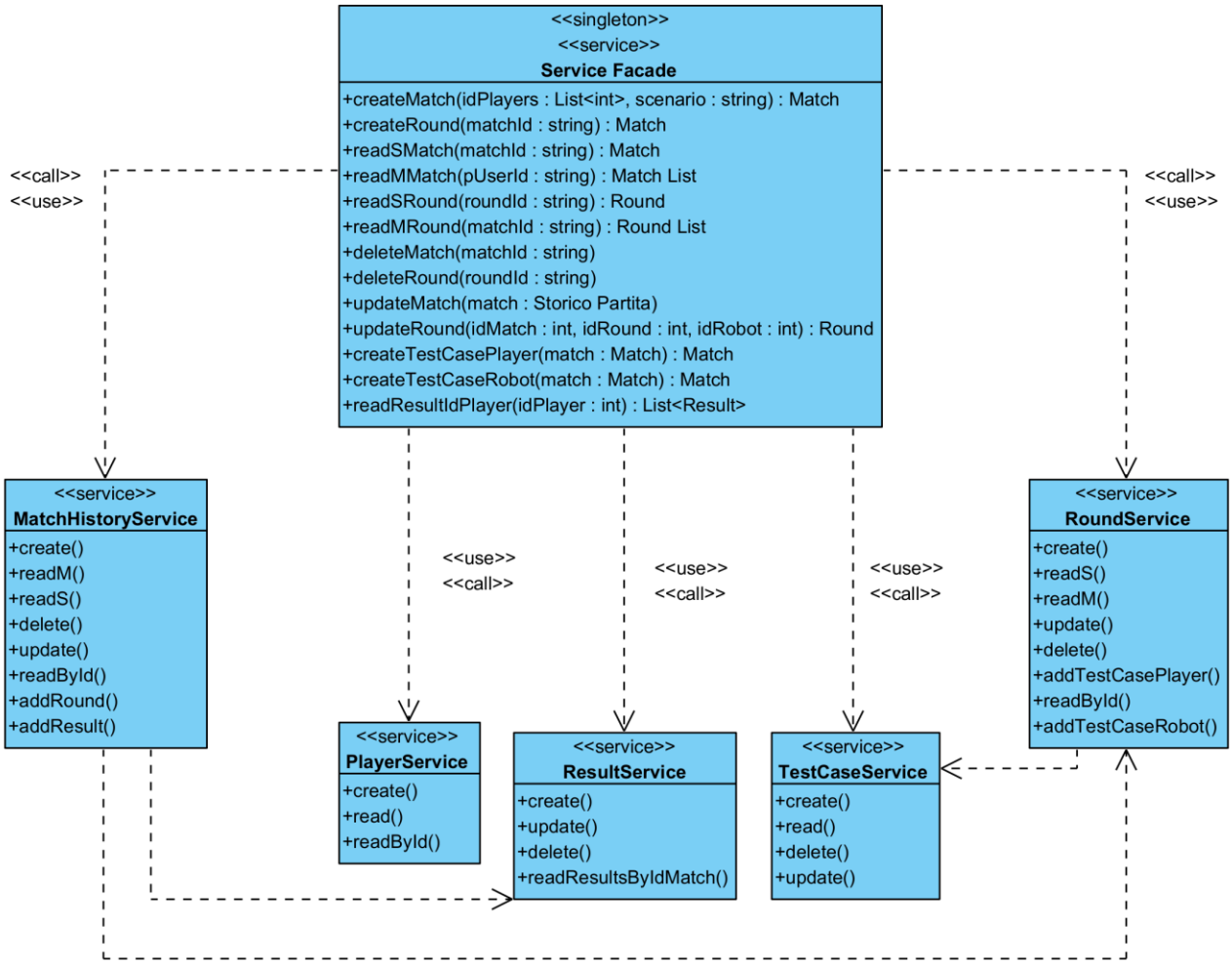
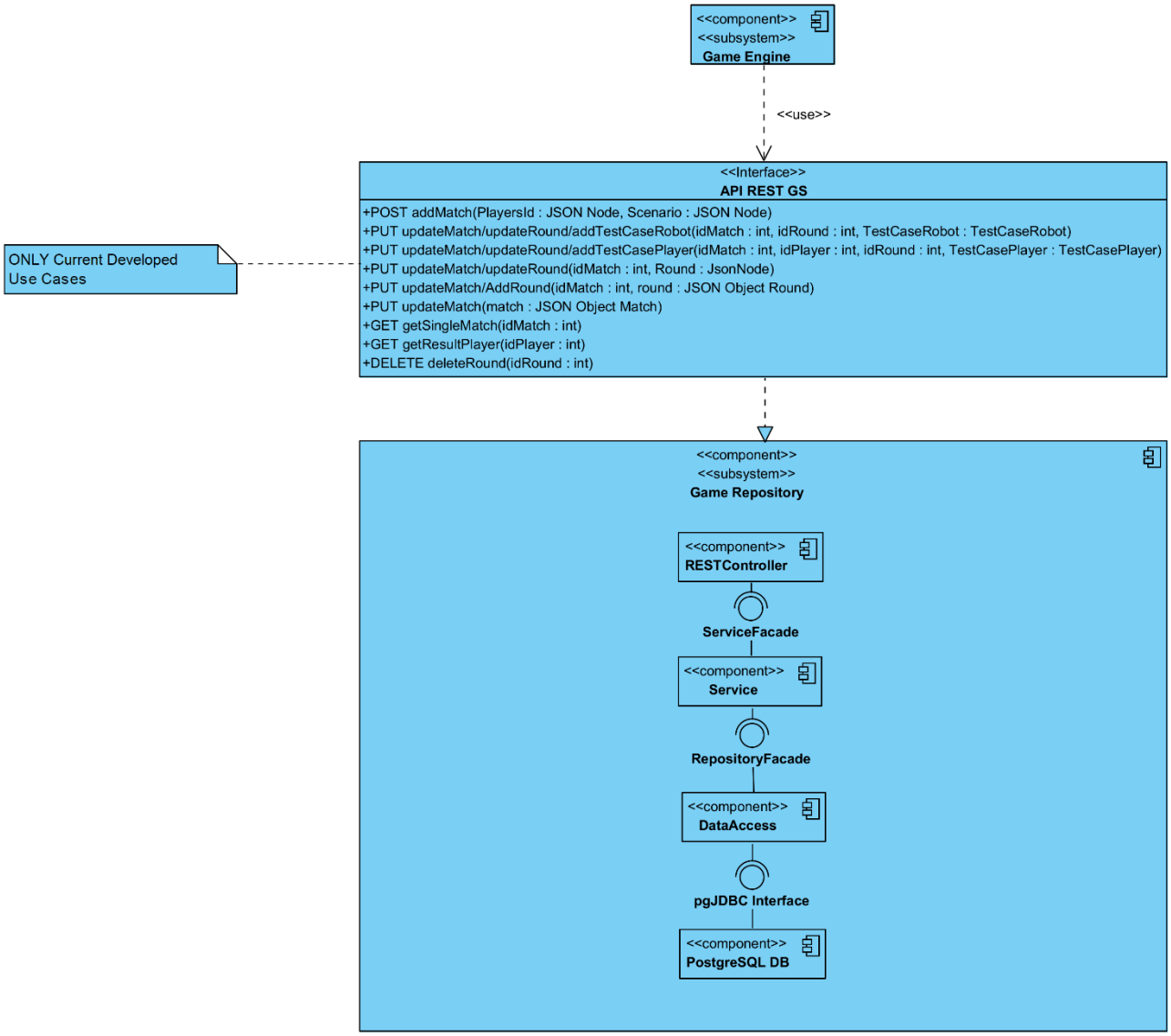


Figura 4: Service Layer Class Diagram

2.4 *Component Diagram*



3. *Sequence Diagram*

Nello sviluppo dei principali casi d'uso, sono state definite con maggiore precisione le diverse interazioni tra i componenti del sistema. Quindi, potendo comprendere al meglio l'architettura sono stati definiti e raffinati i seguenti *Sequence Diagrams:*

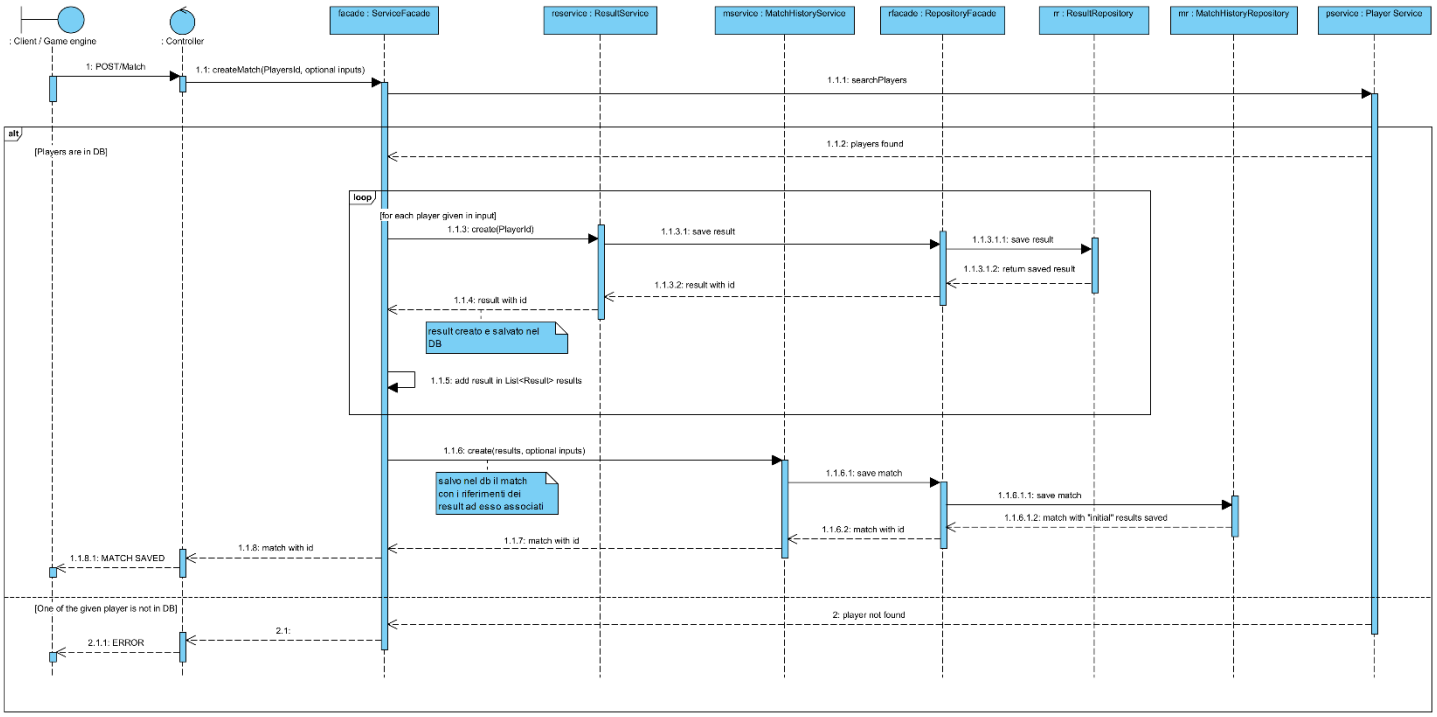


Figura 5: Sequence Diagram Crea Partita

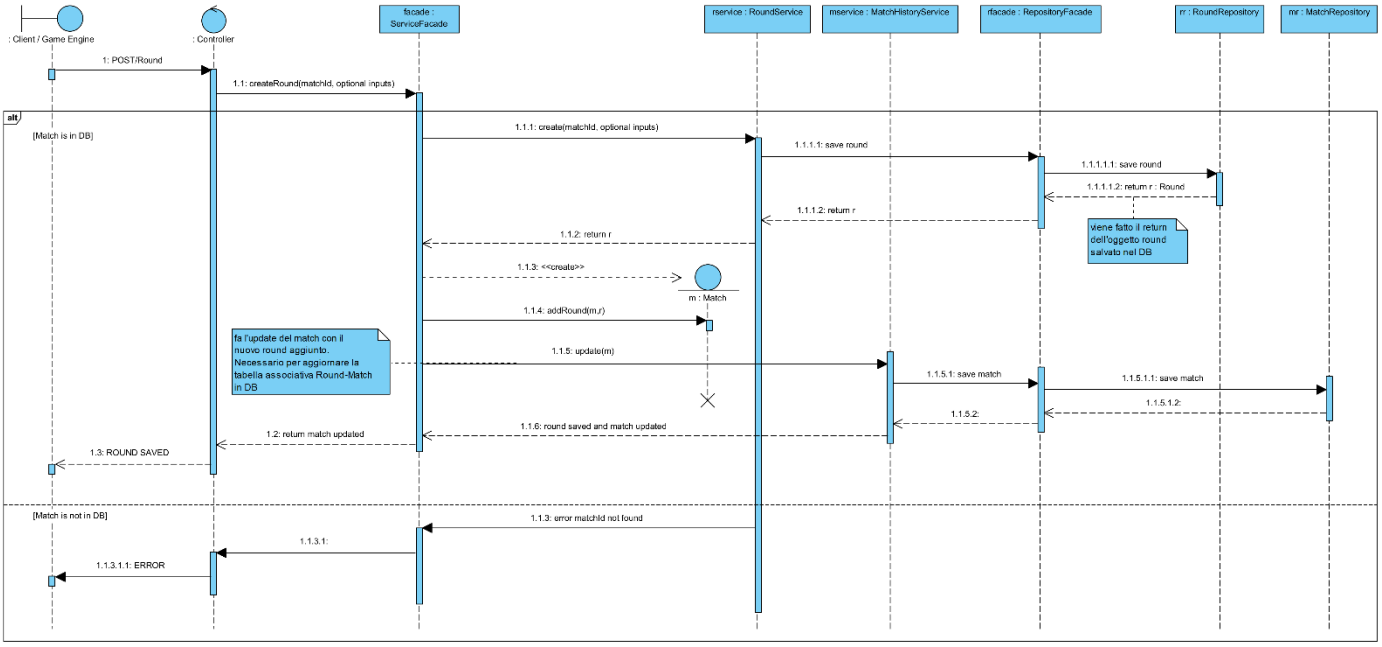


Figura 6: Sequence Diagram Crea Round

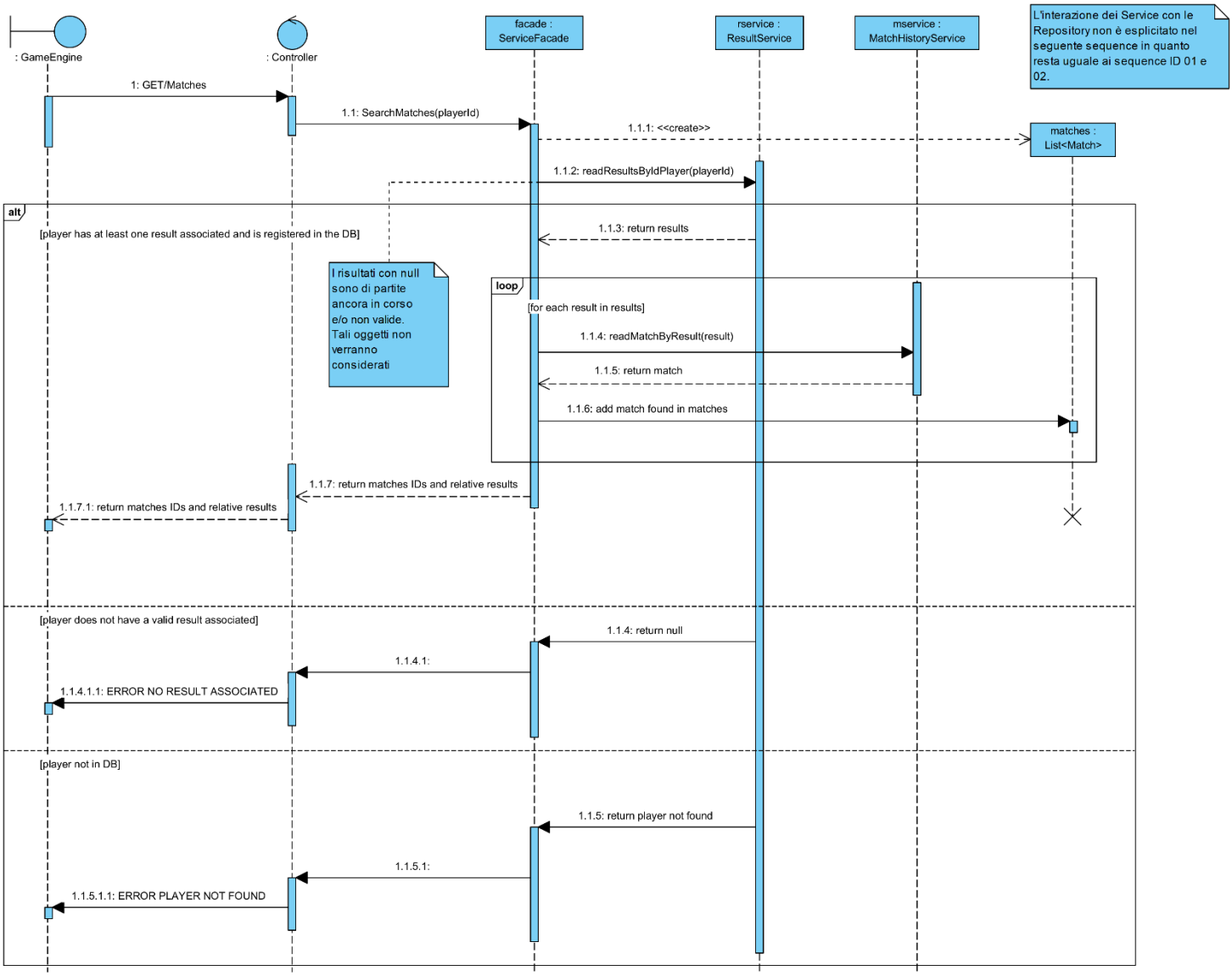


Figura 7: Sequence Diagram Ricerca Partite

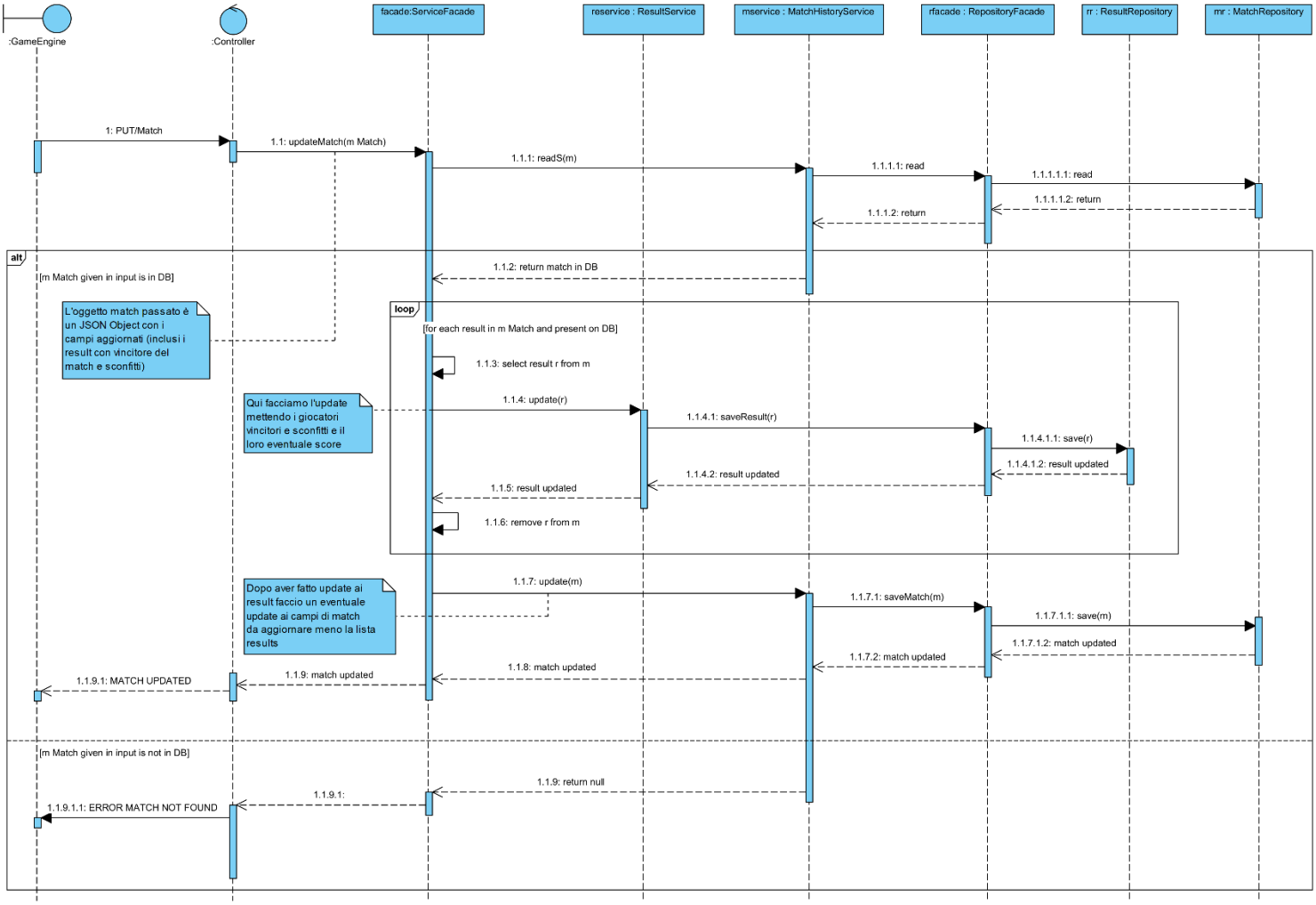


Figura 8: Sequence Diagram Modifica Partita

4. *Interface Document*

In questa sezione sono specificati i *servizi* e *tipi* definiti dall'interfaccia con dizionario. L'interfaccia esposta permette al *GameEngine* di effettuare operazioni di tipo CRUD per gli elementi *Match, Round, TestCase* e relativi.

INTERFACCIA CRUD SERVICES GAME REPOSITORY			
Funzione	**Parametri**	**Info Parametri**	**Eccezioni**
AddMatch	P1: JsonNode	Value Of IdPlayers and Values Of Match	
AddRound	P1: JsonObject	Value Of Round	E1: idMatch not found E2: idRobot not found E3: idRobot not specified
UpdateRound	P1: Integer P2: Integer P3: Integer	Value Of IdMatch Value Of IdRound Value Of IdRobot	E1: idMatch not found E2: idRobot not found E3: idRound not found
UpdateMatch	P1: Integer P2: JsonObject	Value Of IdMatch Value Of Match	E1: idMatch not found
AddTestCasePlayer	P1: Integer P2: Integer P3: Integer P4: JsonObject	Value Of IdMatch Value Of IdPlayer Value Of IdRound Value Of TestCasePlayer	E1: idMatch not found E2: idRound not found E3: idPlayer not found E4: Player not a partecipant
AddTestCaseRobot	P1: Integer P2: Integer P3: JsonObject	Value Of IdMatch Value Of IdRound Value Of TestCaseRobot	E1: idMatch not found E2: idRound not found
ReadResultIdPlayer	P1: Integer P2: Output	Value Of IdPlayer Return List Match	E1: idPlayer not found
ReadSMatch	P1: Integer P2: Output	Value Of IdMatch Return Match	E1: idMatch not found
ReadRound	P1: Integer P2: Output	Value Of IdRound Return Round	E1: idRound not found
ReadMRounds	P1: Integer P2: Output	Value Of IdMatch Return List Round	E1: idMatch not found E2: Match empty
ReadTestCase	P1: Integer P2: Output	Value Of IdTestCase Return TestCase	E1: idTestCase not found
ReadMTestCases	P1: Integer P2: Output	Value Of IdRound Return List TestCase	E1: idRound not found E2: Round empty
ReadTestCaseByIdPlayer	P1: Integer P2: Output	Value Of IdPlayer Return List TestCase	E1: idPlayer not found E2: No Test Case associated to the Player
ReadTestCaseByIdRobot	P1: Integer P2: Output	Value Of IdRobot Return List TestCase	E1: idRobot not found E2: No Test Case associated to the Robot
DeleteMatch	P1: Integer	Value Of IdMatch	E1: idMatch not found
DeleteRound	P1: Integer	Value Of IdRound	E1: idRound not found
DeleteResult	P1: Integer	Value Of IdResult	E1: idResult not found
DeleteTestCase	P1: Integer	Value Of IdTestCase	E1: idTestCase not found

LOCALLY DEFINED DATA TYPES			
Funzione	**Parametri**	**Definizione Parametri**	**Esempio**

AddMatch	JsonNode	Nel JsonNode deve essere specificata una lista di Studenti e uno Scenario	{
			“idStudents”: [value1, value2],
			“scenario”: “exampleScenario”
			}

AddRound	JsonObject	Nel JsonObject deve essere specificato un JsonObject della classe Round	{
			“idRobot”:value1,
			optional,
			}

UpdateMatch	Integer	Common Type	

	JsonObject	Deve essere specificato un JsonObject della classe Match	{
			"id": 1,
			"scenario": "scenario",
			"endDate": "2023-06-02T21:00:00",
			"results": [
			{
			"id": 1,
			"result": "sconfitta"
			}
			{
			"id": 2,
			"result": "vittoria"
			}
			]
			}

AddTestCasePlayer	Integer	Common Type	

	JsonObject	Deve essere specificato un JsonObject della classe TestCasePlayer con tutti i campi, meno id e Player, not null	{
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}

AddTestCaseRobot	Integer	Common Type	

	JsonObject	Deve essere specificato un JsonObject della classe TestCasePlayer con tutti i campi, meno id, not null	{
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}

ReadResultById	Integer	Common Type	

	Output	Viene restituita una lista di JsonObject Result con l’idMatch specificato	[
			{
			“idResult”: 1,
			“idMatch”: 3,
			“Outcome”: “vittoria”
			}
			{
			“idResult”: 2,
			“idMatch”: 6,
			“Outcome”: “sconfitta”
			}
			ecc
			]

ReadSMatch	Integer	CommonType	

	Output	Viene restituito un JsonObject Match	{
			"id": 1,
			"scenario": "scenario",
			"endDate": "2023-06-02T21:00:00",
			"results": [
			{
			"id": 1,
			"result": "sconfitta"
			}
			{
			"id": 2,
			"result": "vittoria"
			}
			]
			}

ReadRound	Integer	CommonType	

	Output	Viene restituito un JsonObject Round	{
			“idRobot”:value1,
			optional,
			}

ReadMRounds	Integer	Common Type	

	Output	Viene restituita una lista di JsonObject Rounds	{
			“id”: 1,
			“idRobot”:value1,
			optional
			}
			{
			“idRound: 2,
			“idRobot”:value2,
			optional
			}

ReadTestCase	Integer	Common Type	

	Output	Viene restituito un JsonObject TestCase	{
			“id”: 1,
			“idPlayer”: 4,
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}

ReadMTestCases	Integer	Common Type	

	Output	Viene restituita una lista di JsonObject TestCase	[
			{
			“id”: 1,
			“idPlayer”: 4,
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}
			{
			“id”: 2,
			“idPlayer”: 5,
			"totalResult" : 165568,
			"compilingResult" : 1212,
			ecc...
			}
			]

ReadTestCaseByIdPlayer	Integer	CommonType	

	Output	Viene restituita una lista di JsonObject TestCase	[
			{
			“id”: 1,
			“idPlayer”: 4,
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}
			{
			“id”: 2,
			“idPlayer”: 5,
			"totalResult" : 165568,
			"compilingResult" : 1212,
			ecc...
			}
			]

ReadTestCaseByIdRobot	Integer	CommonType	

		Viene restituita una lista di JsonObject TestCase	[
			{
			“id”: 1,
			“idPlayer”: 4,
			"totalResult" : 12568,
			"compilingResult" : 1212,
			ecc...
			}
			{
			“id”: 2,
			“idPlayer”: 5,
			"totalResult" : 165568,
			"compilingResult" : 1212,
			ecc...
			}
			]

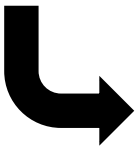
DIZIONARIO	
Funzione	**Descrizione**
AddMatch	Permette di aggiungere un Match ed inizializzare i relativi Risultati associati ad ogni Player partecipante. Tali risultati sono al momento null.
AddRound	Permette di aggiungere un Round ad un Match esistente specificato in input.
UpdateRound	Permette di modificare i campi di un round esistente specificato in input.
UpdateMatch	Permette di modificare i campi di un Match esistente specificato in input.
AddTestCasePlayer	Permette di aggiunger un TestCasePlayer ad un Round esistente spcificato in input.
AddTestCaseRobot	Permette di aggiunger un TestCaseRobot ad un Round esistente spcificato in input.
ReadResultIdPlayer	Permette di visualizzare lo storico di Risultati di un Player esistente specificato in input.
ReadSMatch	Permette di visualizzare un singolo Match esistente in base all'id specificato in input.
ReadRound	Permette di visualizzare un singolo Round esistente in base all'id specificato in input.
ReadMRounds	Permette di visualizzare tutti i Round associati ad un Match esistente specificato in input.
ReadTestCase	Permette di visualizzare un singolo TestCase esistente in base all'id specificato in input.
ReadMTestCases	Permette di visualizzare tutti i TestCase associati ad un Round esistente specificato in input.
ReadTestCaseByIdPlayer	Permette di visualizzare un singolo TestCase esistente in base all'id del Player specificato in input.
ReadTestCaseByIdRobot	Permette di visualizzare un singolo TestCase esistente in base all'id del Robot specificato in input.
DeleteMatch	Permette di eliminare un Match esistente specificato in input.
DeleteRound	Permette di eliminare un Round esistente specificato in input.
DeleteResult	Permette di eliminare un Result esistente specificato in input.
DeleteTestCase	Permette di eliminare un TestCase esistente specificato in input.

5. Esempi di funzionamento

In questa sezione saranno mostrati alcuni esempi di fruizione dei servizi esposti dal nostro sistema attraverso richieste HTTP alla REST API implementata.

- **Ricerca Singola Partita (GET)**

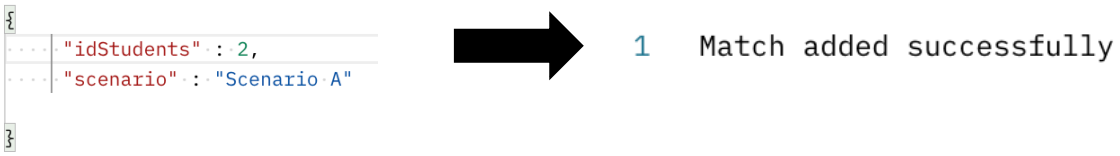
  host:port/getSingleMatch/idMatch




```
{
    "id": 1,
    "scenario": "test2",
    "startDate": "2023-06-02T12:41:56.64341",
    "endDate": null,
    "rounds": [
        {
            "id": 4,
            "robotId": 0,
            "testCasesPlayer": [],
            "testCasesRobot": []
        }
    ],
    "results": [
        {
            "id": 4,
            "player": {
                "id": 1,
                "username": "matteo"
            },
            "result": "f"
        },
        {
            "id": 5,
            "player": {
                "id": 2,
                "username": "mik"
            },
            "result": "f"
        }
    ]
```

- **Creazione Partita (POST)**

  host:port/addMatch



- **Modifica Partita (PUT)**

  host:port/updateMatch/idMatch