

Task 4 – Terza Iterazione

Gruppo G4: Arena Letizia

M63001513

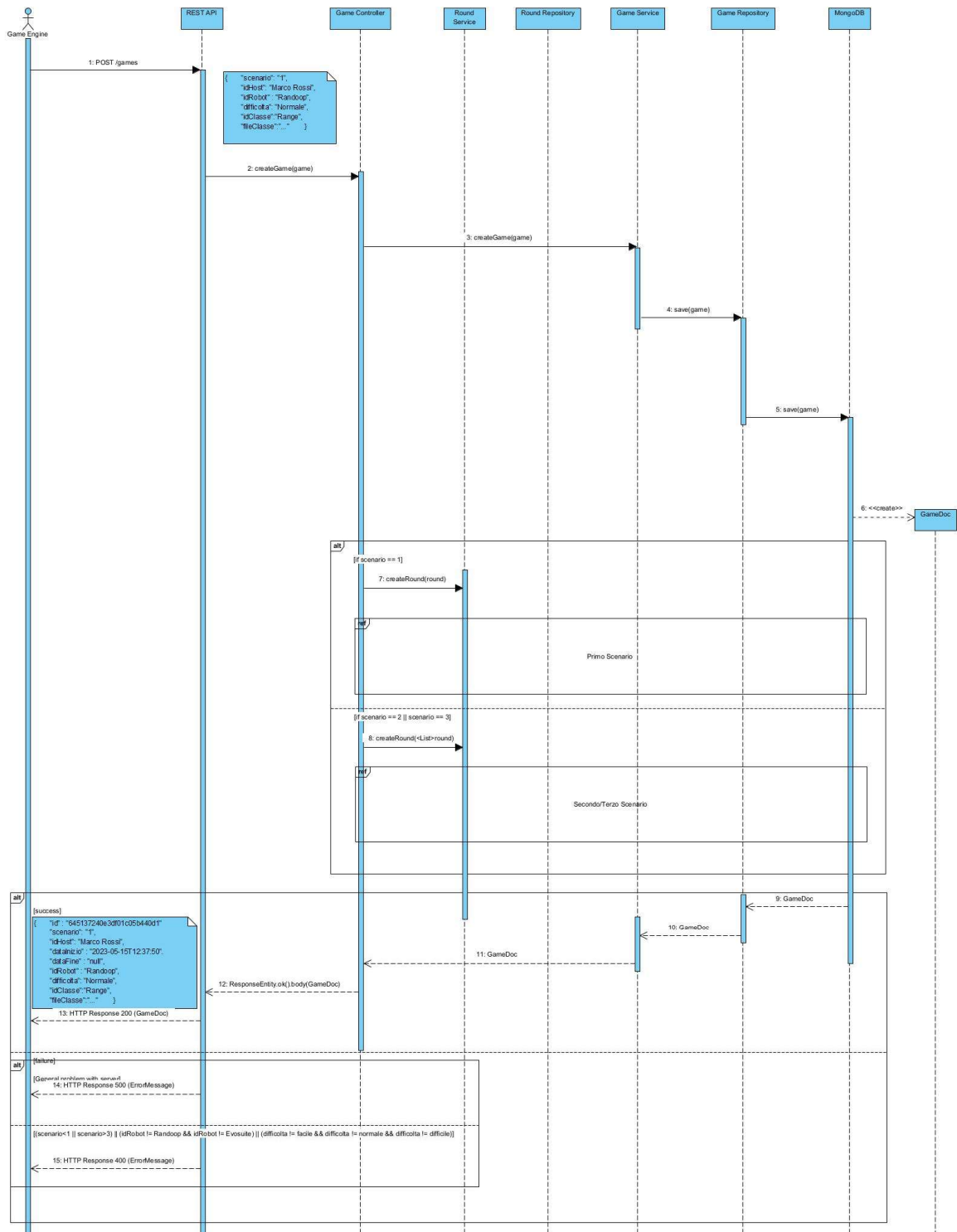
Ferrara Leonardo

M63001517

1. Sequence Diagram:

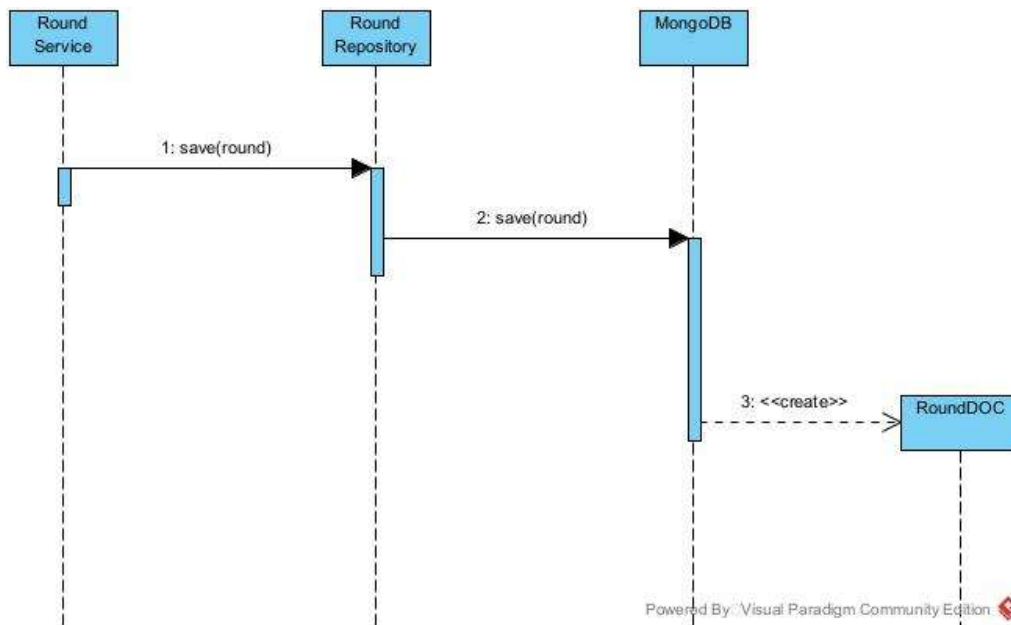
Il primo passo per questa iterazione è stata l'implementazione dei sequence diagram. Avendo ormai chiara l'architettura del software, possiamo infatti procedere al raffinamento delle interfacce e ad una loro prima implementazione.

1.1 Create Game:

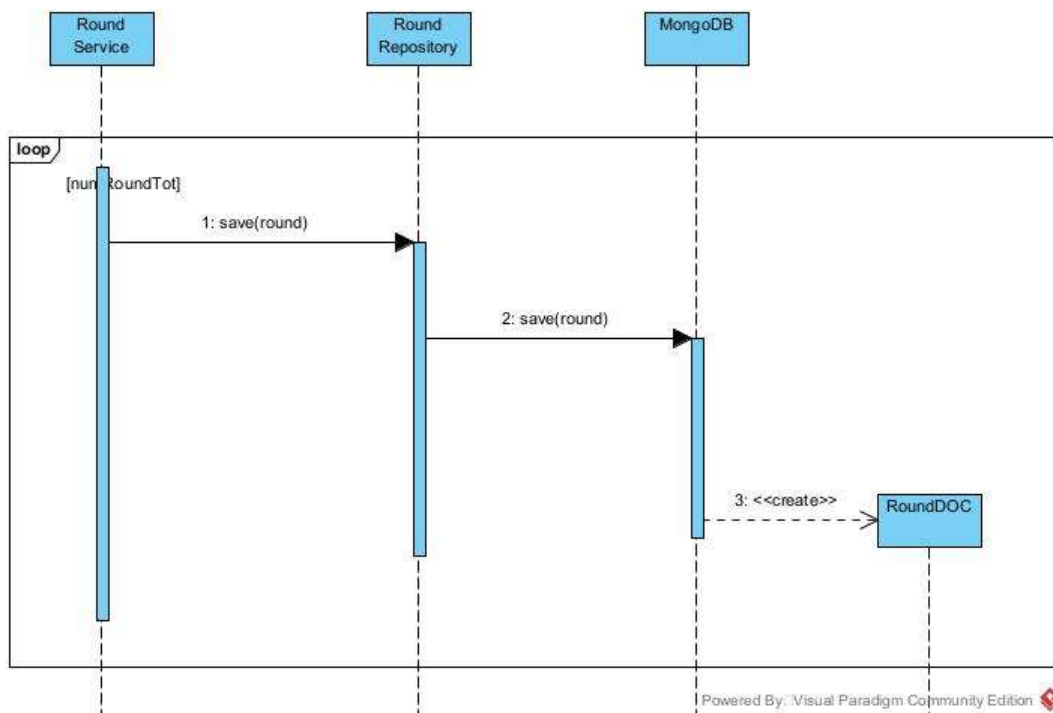


La creazione dei round all'interno della Create Game dipende dallo scenario scelto dal giocatore: se ha scelto il primo scenario si dovrà creare un singolo round; se invece ha scelto il secondo si dovrà creare un numero di round pari a NumRoundTot. Per mostrare questa differenza di implementazione senza rendere troppo lungo e complesso il diagramma di Create Game abbiamo creato altri due diagrammi secondari:

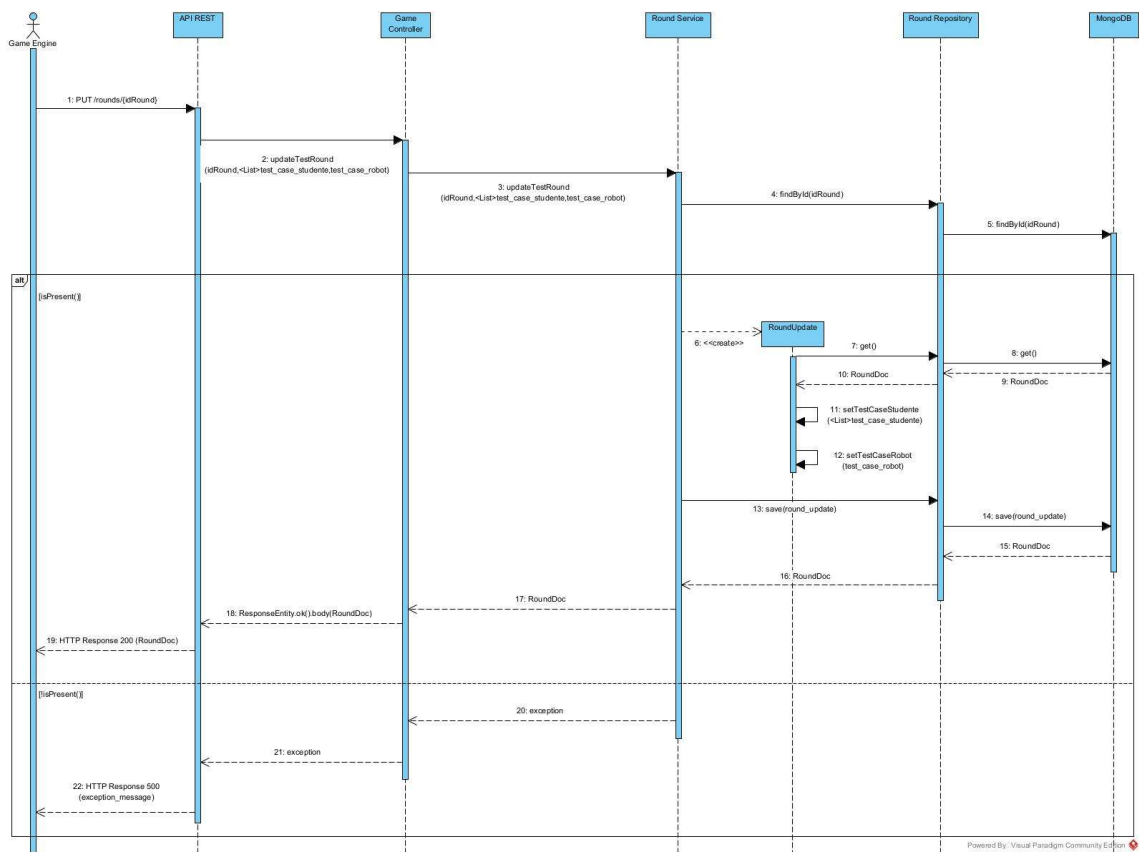
1.1.1 Primo scenario:



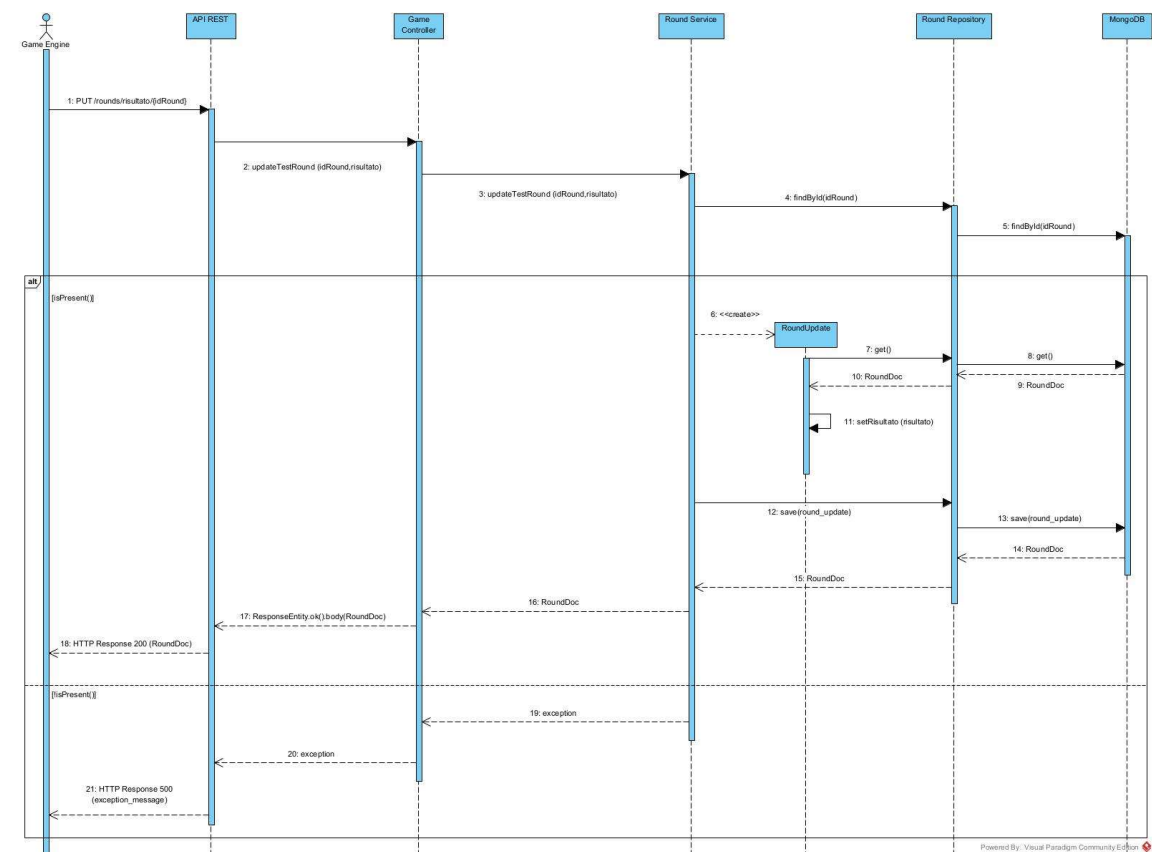
1.1.2 Secondo/Terzo Scenario:



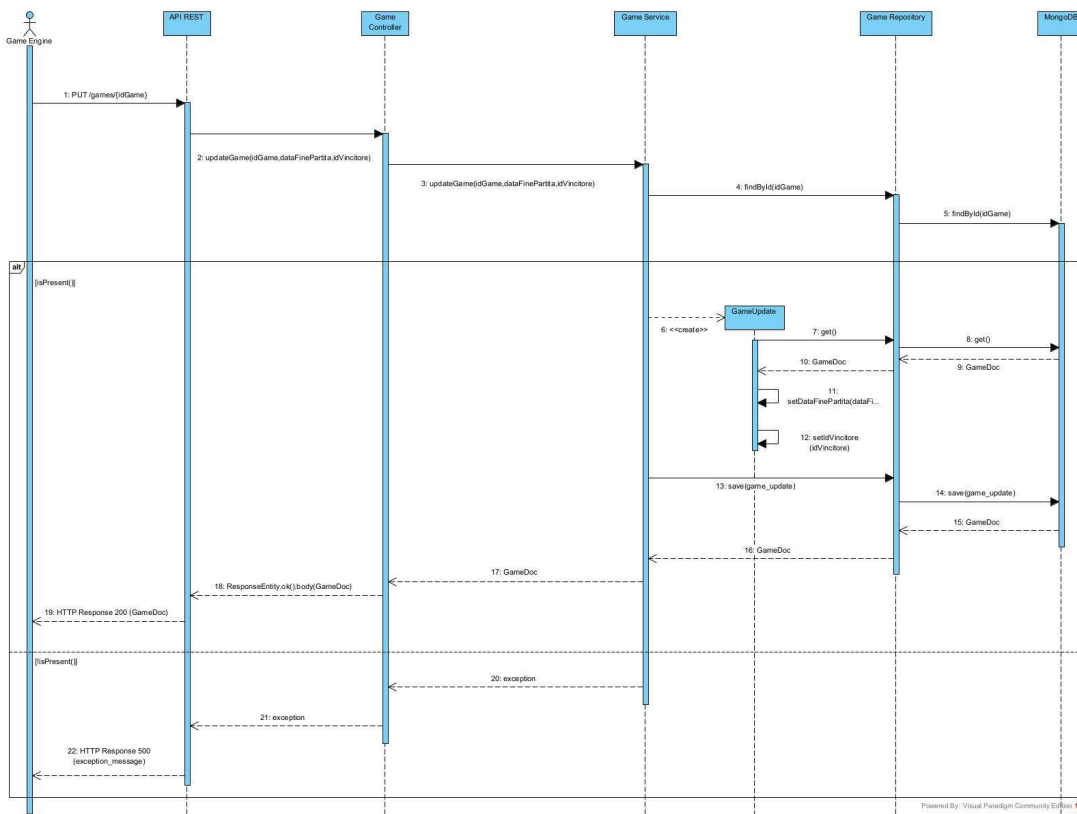
1.2 Update Test Round:



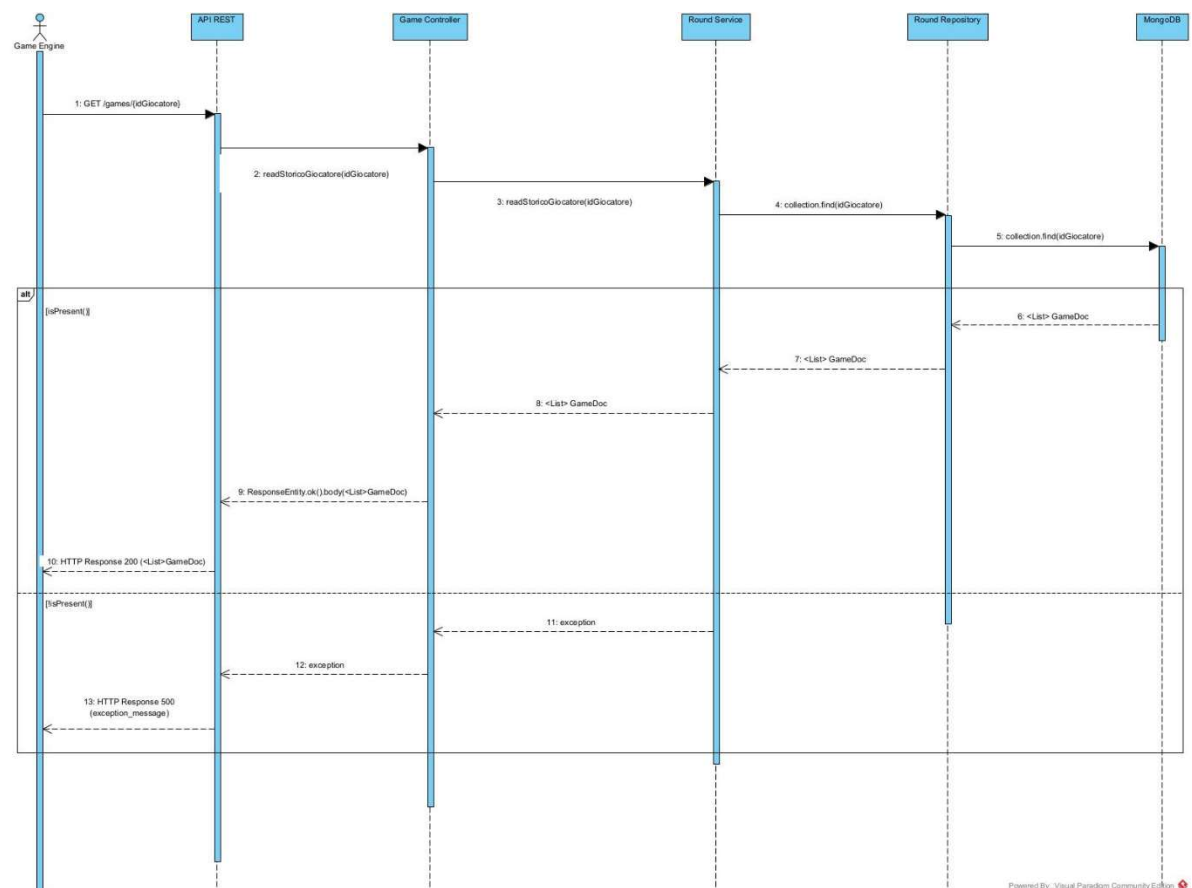
1.3 Update Risultato Round:



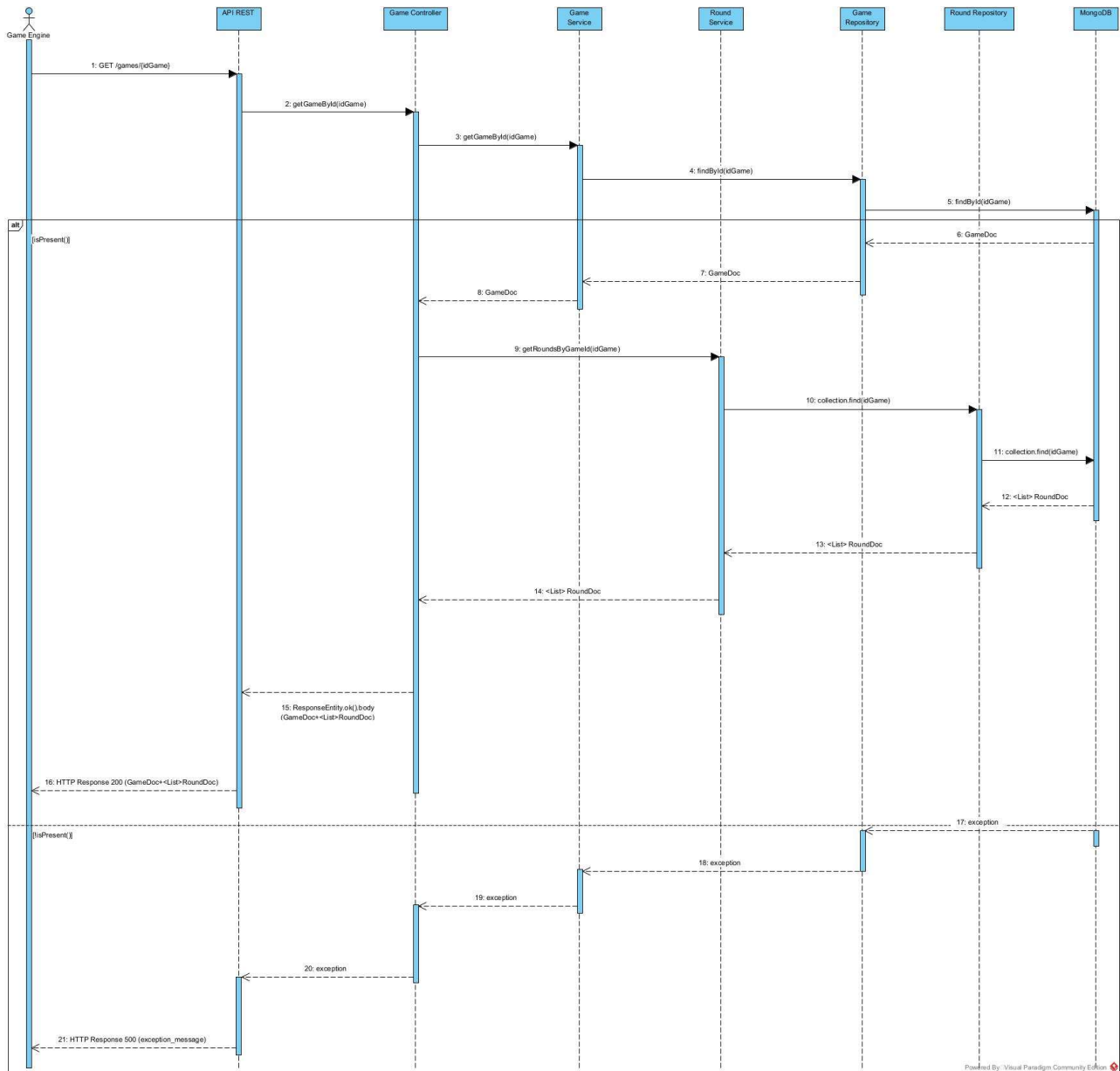
1.4 End Game:



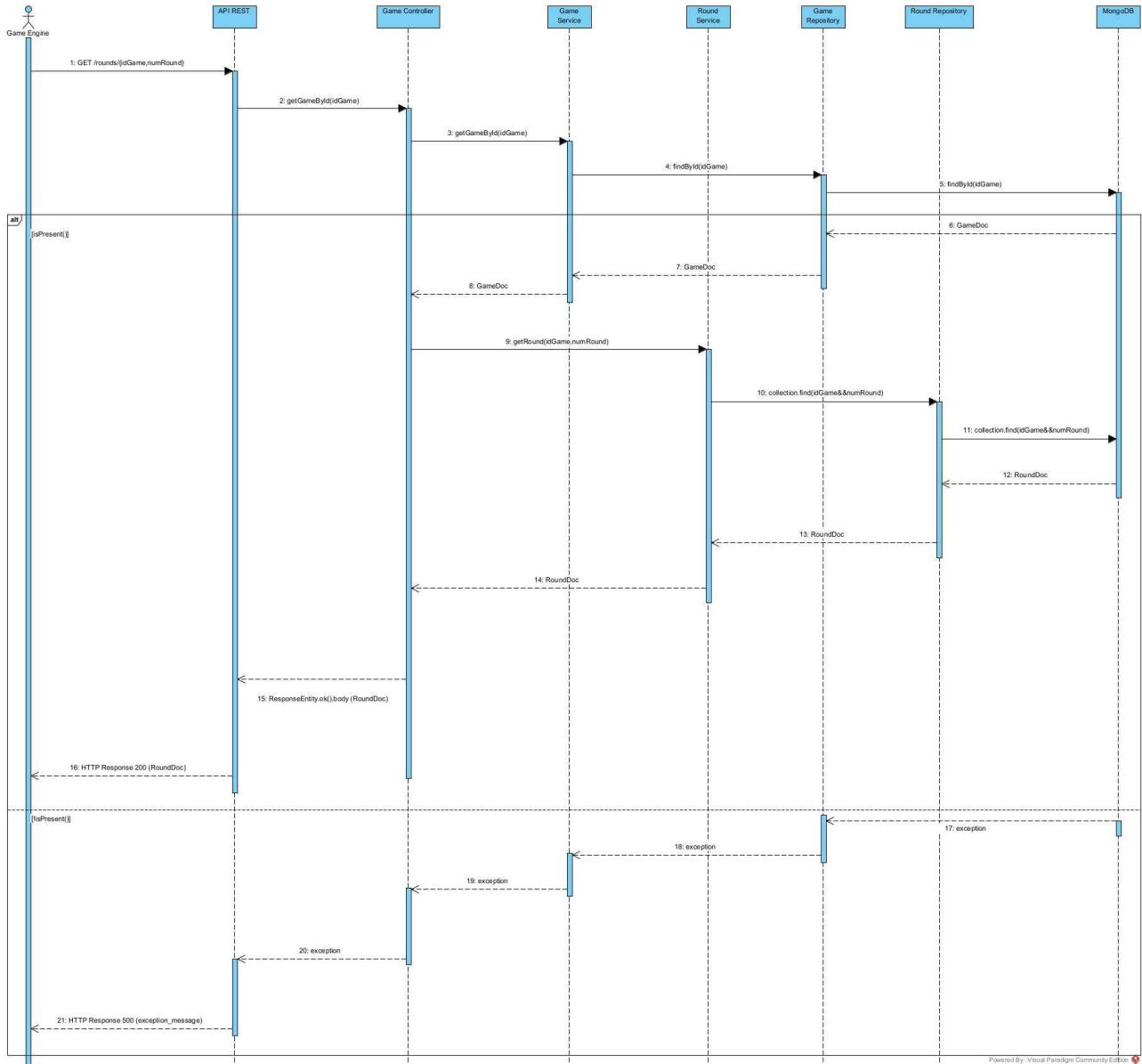
1.5 Read Storico Giocatore:



1.6 Read Game:



1.7 Read Round:



2. Modifiche all'Interfaccia:

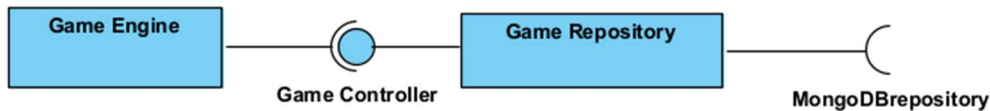
A seguito della stesura dei sequence diagram e dell'implementazione di parte del codice ci siamo accorti che era necessario apportare alcune modifiche alla tabella delle interfacce proposta nella scorsa iterazione. Presentiamo quindi di seguito la versione aggiornata:

Operazione	Descrizione Operazione	Input	Output
createGame	Il game engine richiede il salvataggio dei dati della partita avviata dal giocatore	idGiocatore, scenario, idRobot, difficoltaRobot, idClasse, fileClasse	idPartita
updateTestRound	Il game engine permette di salvare i file di testo creati (sia dai giocatori che dal robot) durante la scrittura dei test case	idRound, testStudenti, testRobot	Round
updateRisultatoRound	Il game engine permette di salvare i risultati relativi ai test case (sia dei giocatori che del robot) relativi al round in corso	idRound, esitoCompilazione, punteggioStudente, punteggioRobot	Round
endGame	Il game engine deve salvare i dati di conclusione della partita	idPartita, risultato	Partita
readStoricoGiocatore	Su richiesta il database deve fornire i dati relativi a tutte le partite giocate dallo specifico giocatore	idGiocatore	List<Partita>
readGame	Su richiesta il database deve fornire lo storico della partita selezionata e dei suoi round	idPartita	Partita, List<Round>
readRound	Su richiesta il database deve fornire le informazioni relative al round selezionato nella partita indicata	idPartita, numeroRound	Round

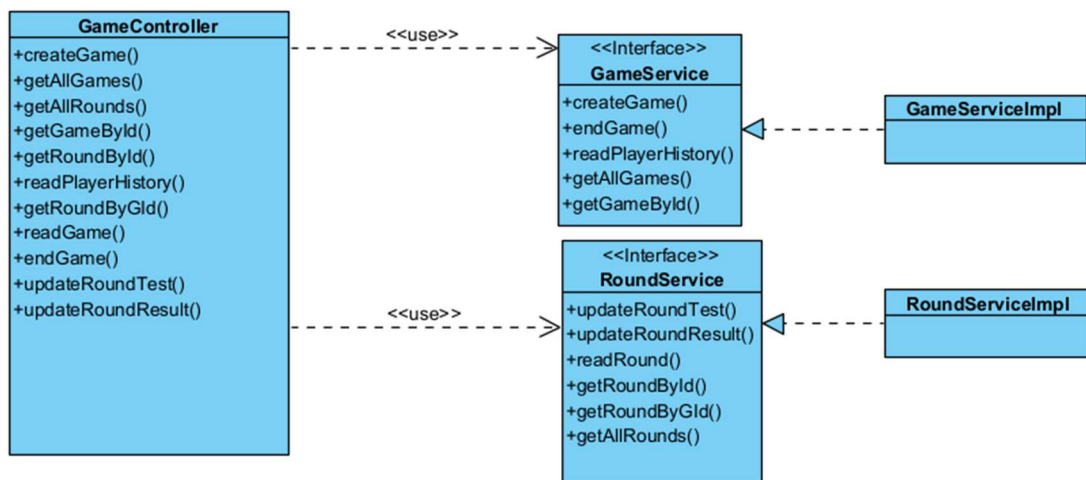
3. Diagrammi di Interfacce:

Per rendere più immediata la visualizzazione dei servizi che andiamo ad offrire abbiamo deciso di sviluppare alcuni diagrammi delle Interfacce.

Il primo rappresenta la struttura basilare del sistema, con il collegamento agli altri componenti con cui il Game Repository interagisce:



Il secondo rappresenta in maniera più esplicita il comportamento del sistema e le operazioni che il sistema mette a disposizione:

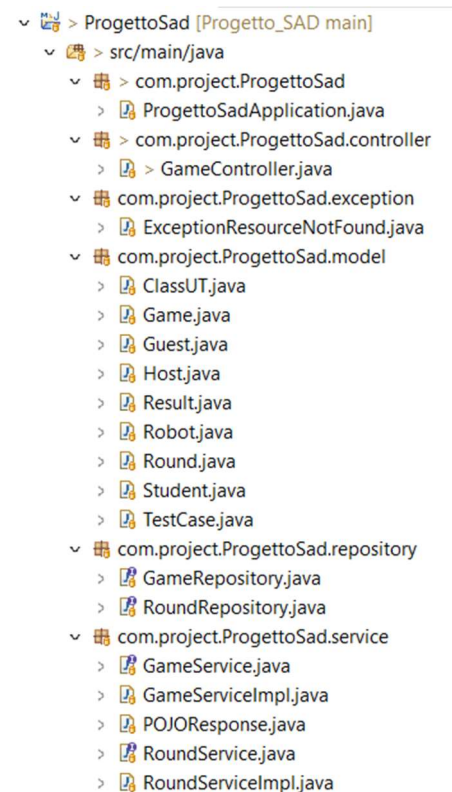


4. Implementazione del codice:

Una porzione consistente di questa iterazione è stata deputata alla stesura del codice: a partire dalla backbone definita dal prototipo implementato nella scorsa iterazione sono state infatti implementate tutte le funzioni di base richieste al Game Repository.

Al momento abbiamo deciso di concentrare il nostro effort sull'implementazione effettiva delle funzioni, e di lasciare alla prossima iterazione l'implementazione delle eccezioni e dei vincoli sui dati.

Di lato mostriamo la struttura interna del progetto.



5. Verifica del funzionamento:

Di seguito mostriamo il funzionamento di alcuni servizi, indicandone l'URL ed il body da inserire, ed il risultato che viene riportato. Per non appesantire la relazione abbiamo deciso di mostrare soltanto alcuni dei servizi relativi alla partita, aggiungendo le altre informazioni necessarie senza mostrarlo.

5.1 Create Game:

URL richiesta: <http://localhost:8080/games>

Body:

```
1  {
2    "host": "Mario Rossi",
3    "scenario": 3,
4    "totalRoundNum": 5,
5    "guest": [
6      "Luca Giordano",
7      "Leonardo Da Vinci"
8    ],
9    "robot": {
10     "robotId": "Randoop",
11     "difficulty": "Normal"
12   }
13 }
```

Output:

```
1  647b58b84790de55405b44e1
```

L'output atteso per la createGame è l'Id della partita.

Nel database è però possibile osservare l'effettiva creazione del documento relativo alla partita, contenente tutti i dati inseriti nel Body. Sono stati inoltre creati allo stesso tempo tutti i round associati alla partita stessa (in questo caso 5).

```
_id: ObjectId('647b58b84790de55405b44e1')
startDate: 2023-06-03T15:14:00.659+00:00
▼ host: Object
  studentId: "Mario Rossi"
▼ guest: Array
  ▼ 0: Object
    studentId: "Luca Giordano"
  ▼ 1: Object
    studentId: "Leonardo Da Vinci"
☑ robot: Object
  robotId: "Randoop"
  difficulty: "Normal"
scenario: 3
totalRoundNumber: 0
_class: "com.project.ProgettoSad.model.Game"
```

5.2 End Game

URL richiesta: <http://localhost:8080/games/end/647b58b84790de55405b44e1>

Body:

```
1 Luca Giordano
```

Output:

```
1  {}
2  "_id": "647b58b84790de55405b44e1",
3  "startDate": "2023-06-03T17:14:00.659",
4  "endDate": "2023-06-03T17:21:56.442742",
5  "host": {
6    "studentId": "Mario Rossi"
7  },
8  "guest": [
9    {
10     "studentId": "Luca Giordano"
11   },
12   {
13     "studentId": "Leonardo Da Vinci"
14   }
15 ],
16 "robot": {
17   "robotId": "Randoop",
18   "difficulty": "Normal"
19 },
20 "scenario": 3,
21 "totalRoundNumber": 0,
22 "classUt": null,
23 "winner": "Luca Giordano",
24 "id": {
25   "timestamp": 1685805240,
26   "date": "2023-06-03T15:14:00.000+00:00"
27 },
28 "studente": {
29   "studentId": "Mario Rossi"
30 }
31 {}
```

5.3 Read Game

URL richiesta: <http://localhost:8080/games/rounds/647b5c4d4790de55405b44e2>

Output:

```
1  {
2    "game": {
3      "_id": "647b5c4d4790de55405b44e2",
4      "startDate": "2023-06-03T17:29:17.807",
5      "endDate": null,
6      "host": {
7        "studentId": "Mario Rossi"
8      },
9      "guest": [
10       {
11         "studentId": "Luca Giordano"
12       },
13       {
14         "studentId": "Leonardo Da Vinci"
15       }
16     ],
17     "robot": {
18       "robotId": "Randoop",
19       "difficulty": "Normal"
20     },
21     "scenario": 3,
22     "totalRoundNumber": 5,
23     "classUt": null,
24     "winner": null,
25     "id": {
26       "timestamp": 1685806157,
27       "date": "2023-06-03T15:29:17.000+00:00"
28     },
29     "studente": {
30       "studentId": "Mario Rossi"
31     }
32   },
33   "rounds": [
34     {
35       "roundId": "647b5c4d4790de55405b44e3",
36       "gameId": "647b5c4d4790de55405b44e2",
37       "roundNumber": 1,
38       "testCase": {
39         "studentTest": [
40           "Esempio di test case ",
41           "Esempio di test case ",
42           "Esempio di test case "
43         ],

```

Continua allo stesso modo mostrando le informazioni relative agli altri round della partita.