

Ingegneria del Software - Concorso di Foto

Antonio Marini
559200

Riccardo Pacioni
551337

Emanuele De Luca
587537

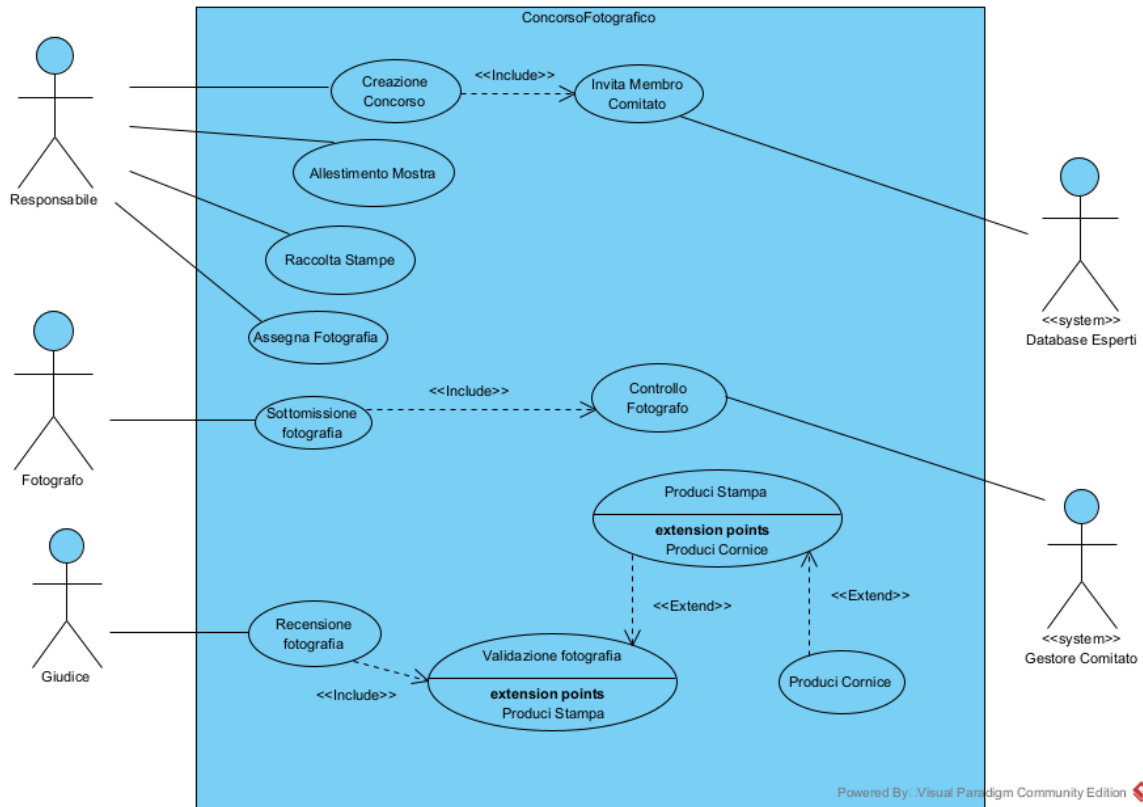
Lorenzo, Battistello
580240

30 agosto 2021

Indice

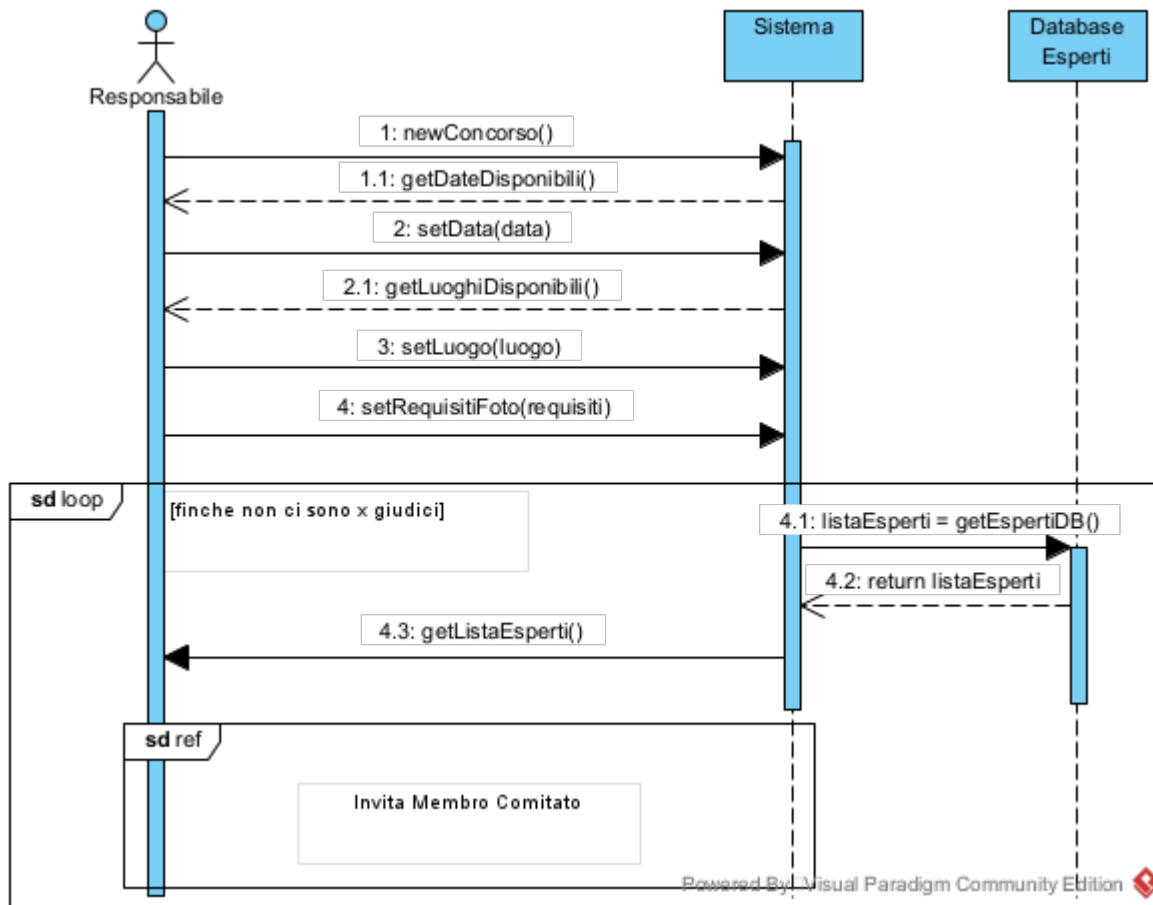
1	Casi d'uso e narrativa.	3
2	Diagramma di sequenza.	5
3	Diagramma delle classi.	6
4	Diagramma macchina a stati.	7
5	Diagramma attività.	8
5.1	Decisione Comitati	9
5.2	Sottomissione foto	10
6	Architettura in vista C&C e dislocazione.	11
6.1	Vista C&C	11
6.2	Diagramma di dislocazione	12
7	Test obligations	13

1 Casi d'uso e narrativa.

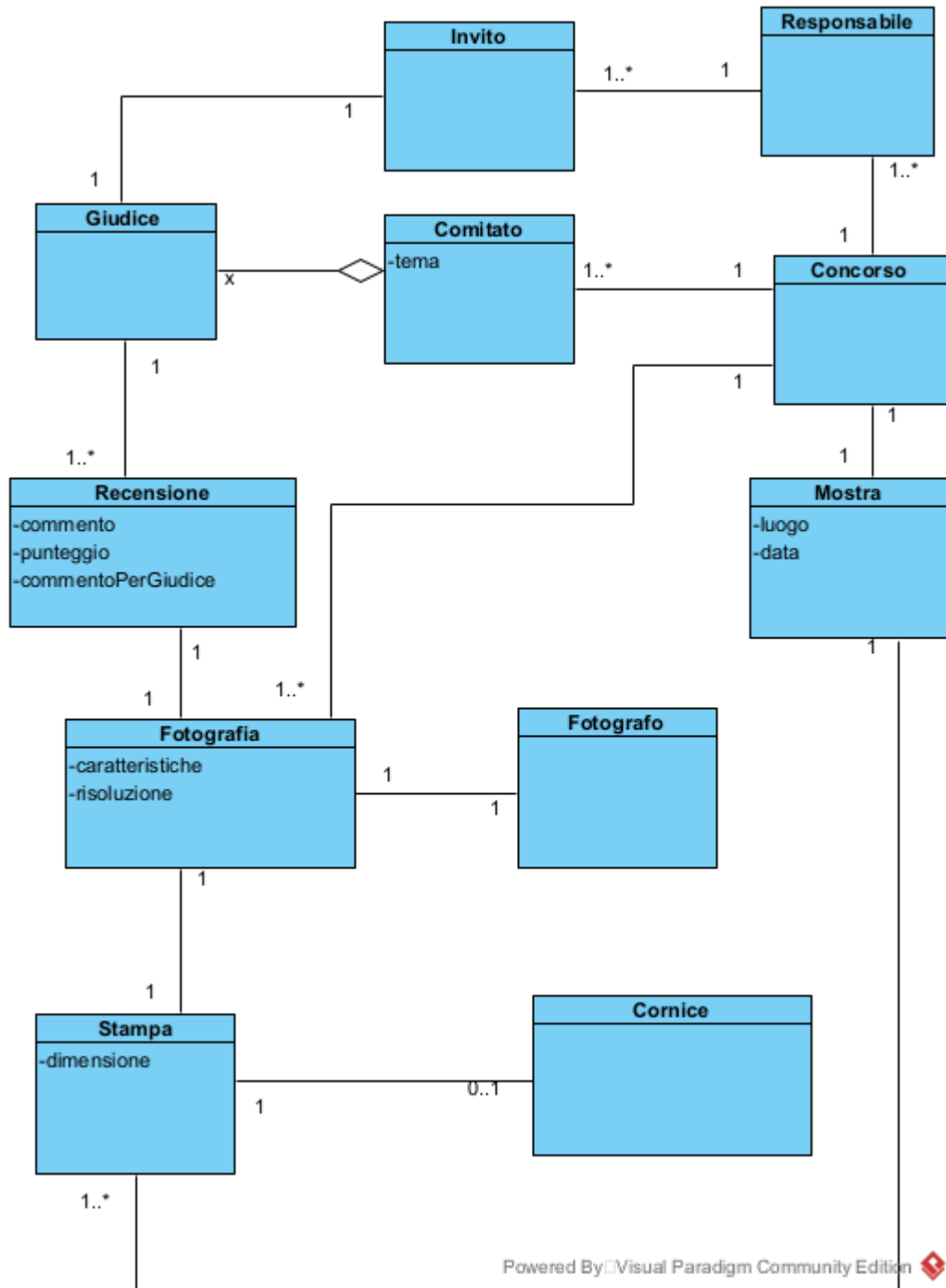


Nome	Creazione concorso
Descrizione	Azioni necessarie per creare un concorso fotografico e decidere il comitato.
Attori primari	Responsabile
Attori secondari	Nessuno
Precondizioni	Nessuna
Sequenza principale degli eventi	<ol style="list-style-type: none"> 1. Il responsabile richiede al sistema di creare un concorso 2. Il sistema chiede la data e il luogo in cui tenere l'evento 3. Il sistema chiede i requisiti delle foto e numero massimo di foto 4. While(num giudici scelti $< x$) <ol style="list-style-type: none"> (a) Mostra lista esperti disponibili ottenuta dal database degli esperti (b) Include Invita Membro Comitato
Postcondizioni	<ul style="list-style-type: none"> • Almeno un tema scelto, • Requisiti fotografie scelti. • Esattamente x giudici scelti. • Concorso creato con successo.
Sequenze alternative degli eventi	<ul style="list-style-type: none"> • Ci sono meno di x giudici nella base di dati • Non ci sono date o luoghi disponibili per tenere l'evento

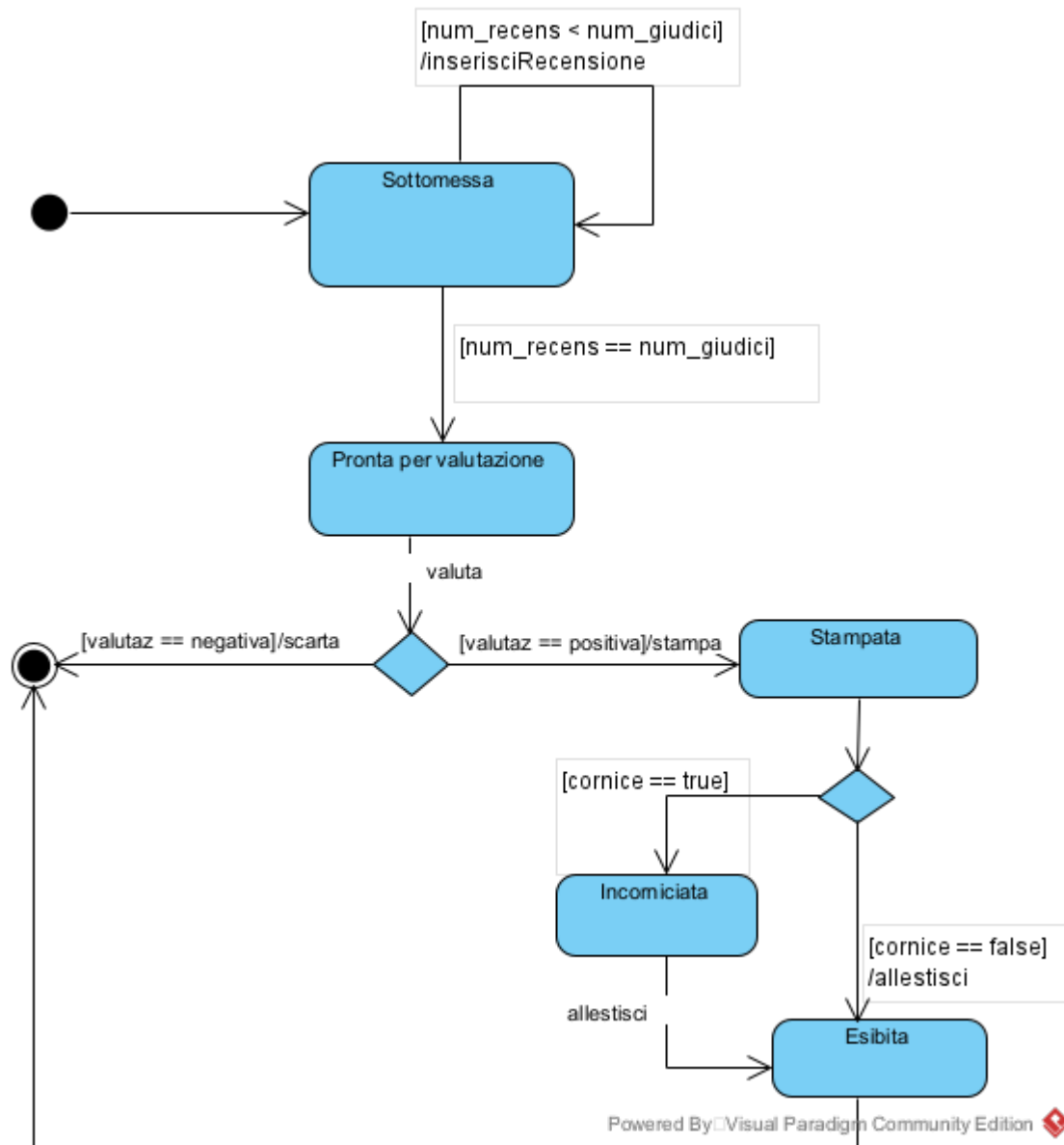
2 Diagramma di sequenza.



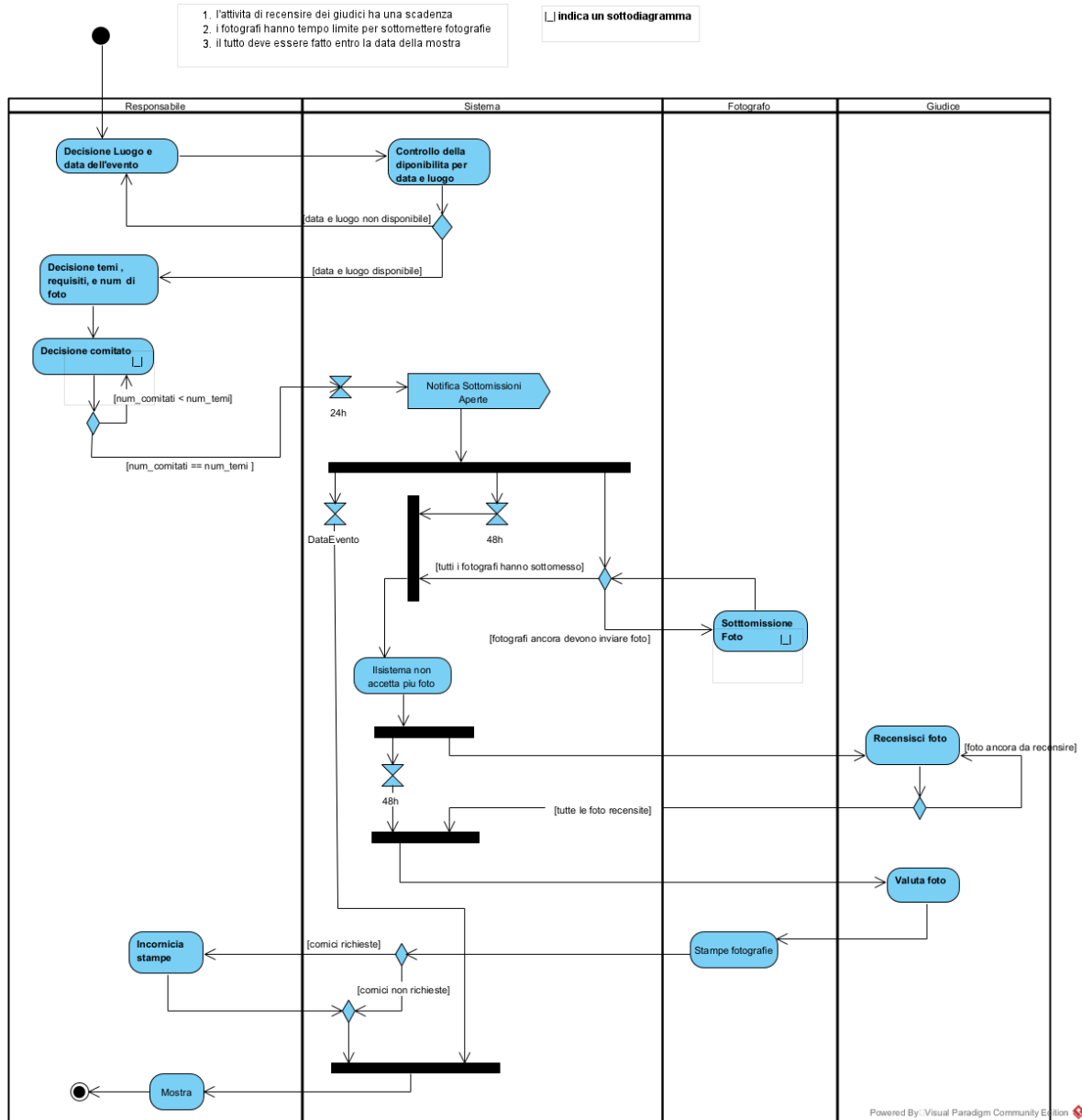
3 Diagramma delle classi.



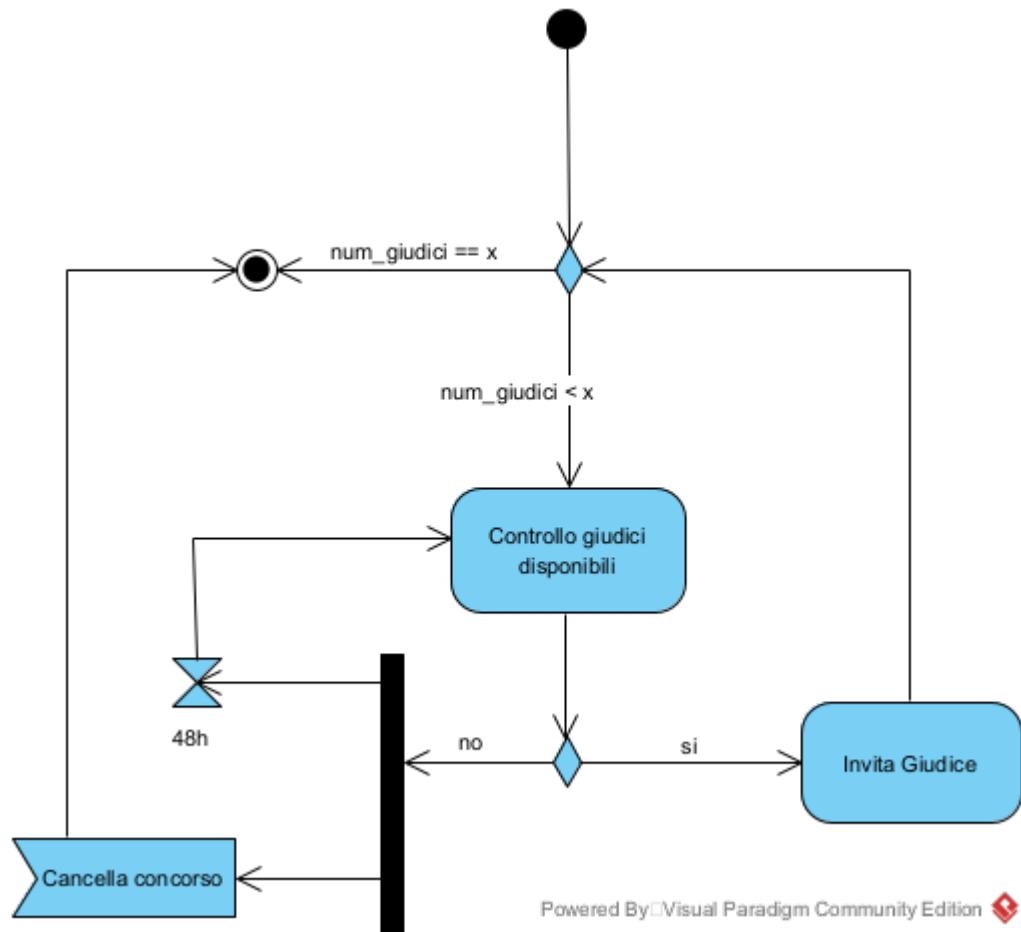
4 Diagramma macchina a stati.



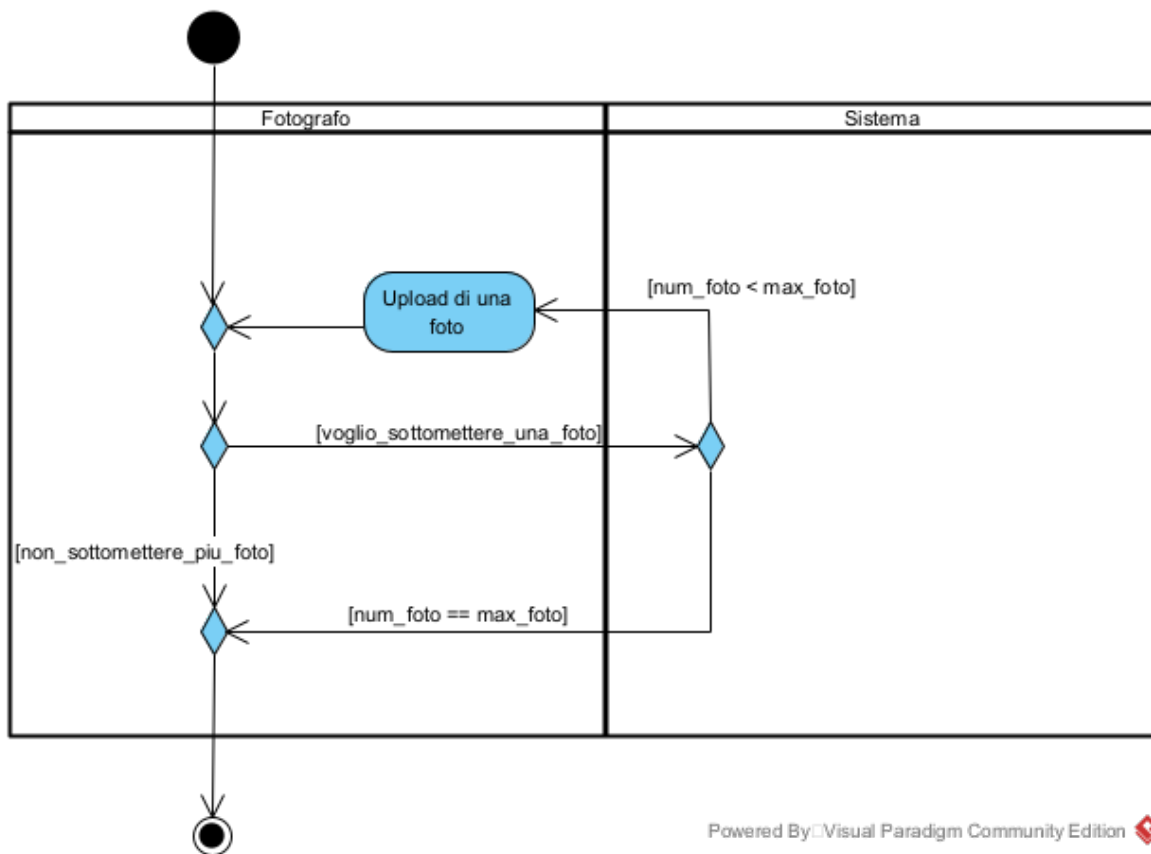
5 Diagramma attività.



5.1 Decisione Comitati

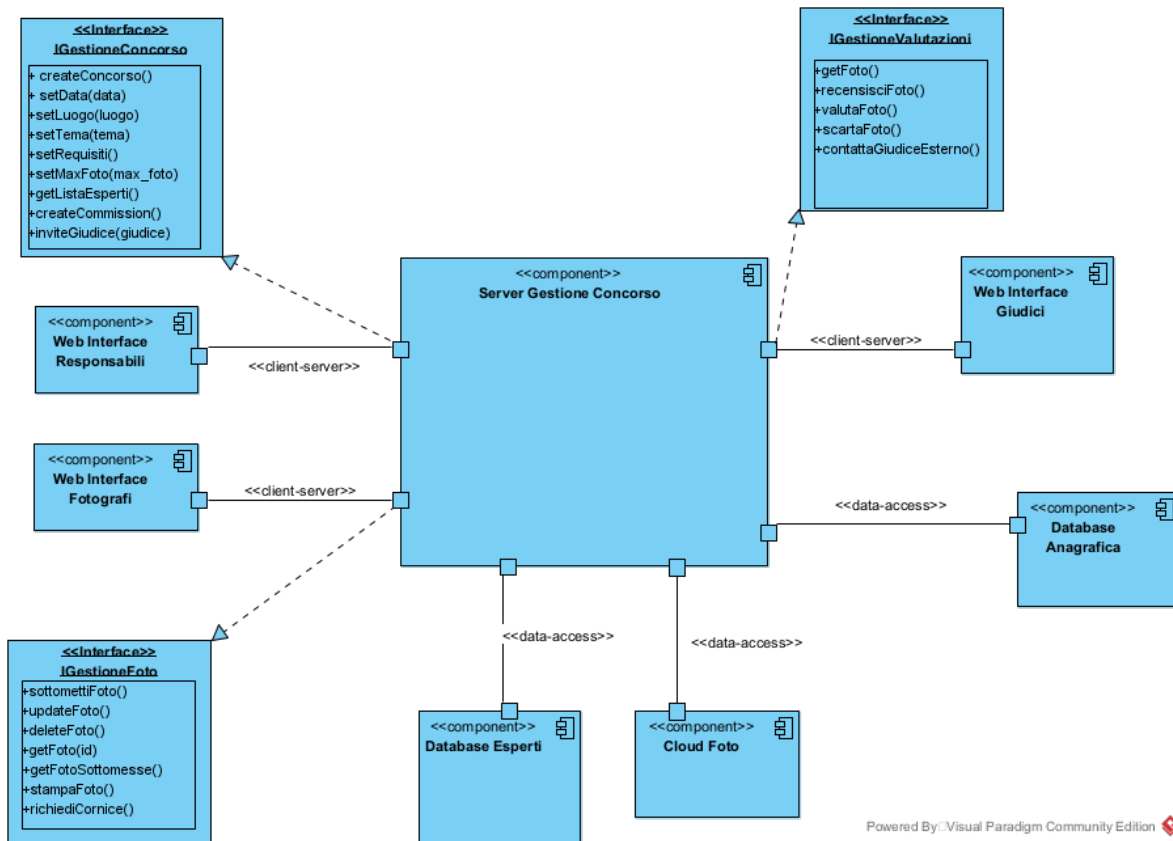


5.2 Sottomissione foto

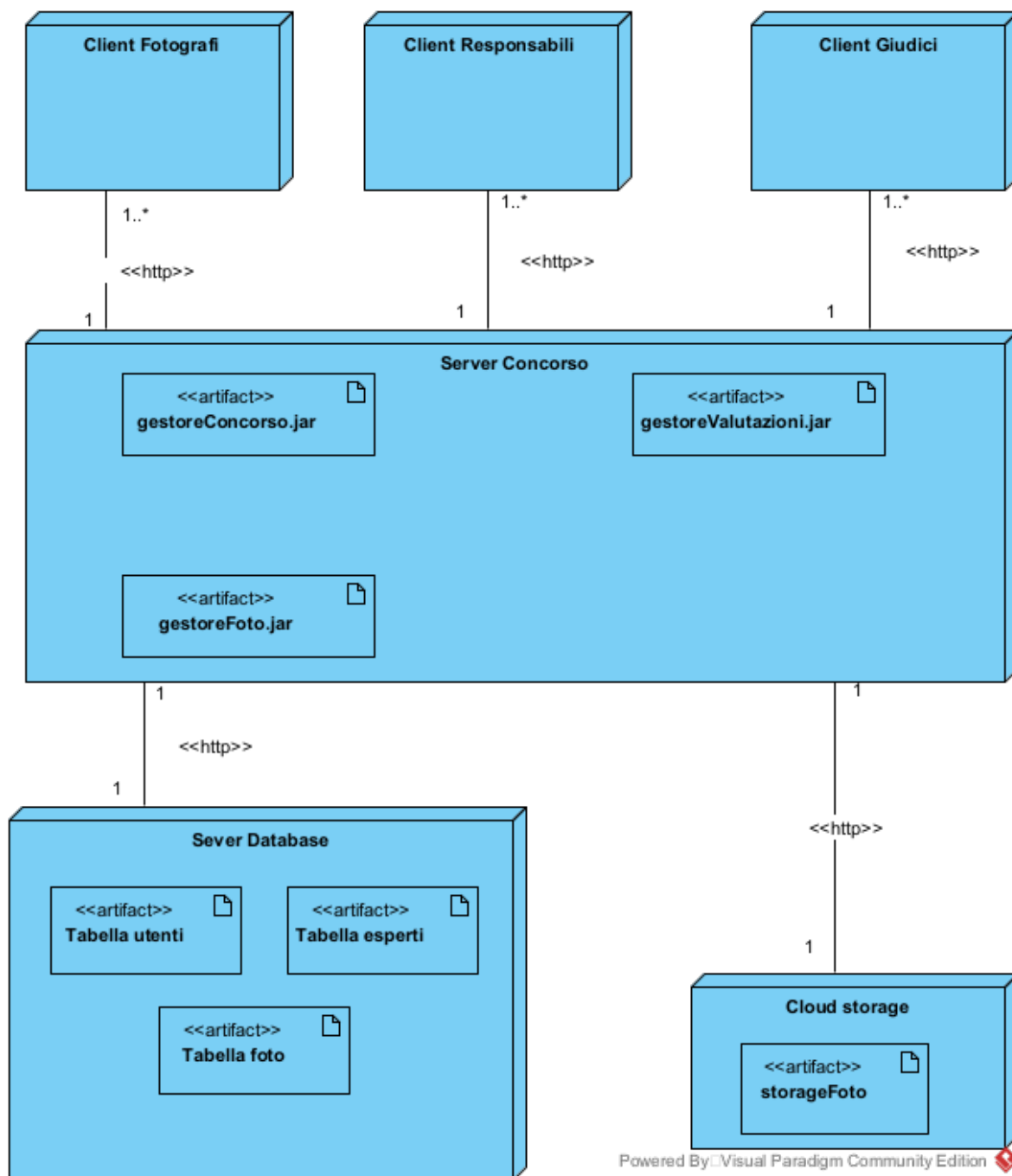


6 Architettura in vista C&C e dislocazione.

6.1 Vista C&C



6.2 Diagramma di dislocazione



7 Test obligations

```
public boolean assegnaFoto(  
    List<Fotografia> fotografie,  
    List<Giudice> giudici,  
    int numGiudiciPerFoto){  
  
    if(fotografie != null || giudici != null)  
        return false;  
    if(numGiudiciPerFoto <= 1)  
        return false;  
    if(giudici.size() <= 0)  
        return false;  
    if(giudici.size() <= numGiudiciPerFoto)  
        return false;  
  
    for(Fotografia foto : fotografie){  
        for(int i = 0; i < numGiudiciPerFoto; i++){  
            foto.assegnaGiudice(giudici.getRandom());  
        }  
    }  
  
    return true;  
}
```

numFotografie	numGiudici	numGiudiciPerFoto	Output	Giustificazione
0	10	3	true	Il test è andato a buon fine.
6	15	3	true	Il test è andato a buon fine.
2	15	1	false	Il numero di giudici per foto è ≤ 1 .
7	0	5	false	Il numero di giudici è ≤ 0 .
11	2	3	false	Il numero di giudici è \leq del numero di giudici per foto.