# Contratti delle Operazioni

## UC Giocare a UnoXTutti

|  |  |
| --- | --- |
| apriStanza(nomeStanza, portaStanza) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione. L’attore fornisce un testo *nomeStanza* e un numero *portaStanza* |
| **Postcondizioni** | Se fra le Stanza che *gioc gestisce* non esiste alcuna Stanza con *nome=nomeStanza* e *porta=portaStanza,* viene creata una Stanza *s* con *proprietario=gioc* e con i seguenti attributi:   * *nome* è pari a *nomeStanza*. * *porta* è pari al valore di *portaStanza* * *indirizzo* è pari all’indirizzo della macchina su cui viene creata   Altrimenti fallisce senza nessun effetto. |

|  |  |
| --- | --- |
| chiudiStanza(stanza) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione. L’attore identifica una Stanza *stanza* con proprietario *gioc*. |
| **Postcondizioni** | Viene eliminata l’associazione fra *stanza* e *gioc* di tipo *gestisce/proprietario.* Vengono eliminate tutte le associazioni di tipo *partecipa/partecipanti* associate a *stanza*. Viene eliminata *stanza*. |

|  |  |
| --- | --- |
| entraInStanza(nomeStanza, indirizzoStanza, portaStanza) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione; *gioc* non *partecipa* in nessuna Stanza. L’attore fornisce un testo *nomeStanza*, un indirizzo IP *indirizzoStanza*, e un numero *portaStanza* |
| **Postcondizioni** | Se esiste una Stanza *s* con *indirizzo*=*indirizzoStanza, nome=nomeStanza* e *porta=portaStanza*, allora viene creata un’associazione *partecipa* fra *gioc* e *s*. |

|  |  |
| --- | --- |
| esciDaStanza()[[1]](#footnote-0) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione e *gioc* *partecipa* in una Stanza *stanza*. |
| **Postcondizioni** | L’associazione *partecipa* fra *gioc* e *stanza* viene rimossa. |

|  |  |
| --- | --- |
| esciDalGioco() | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione. Non esiste nessuna Stanza con *proprietario=gioc*. |
| **Postcondizioni** | Viene eliminato *gioc* (con tutte le sue associazioni) |

## UC Autenticarsi

|  |  |
| --- | --- |
| accedi(email, pass) | |
| **Precondizioni** | L’attore fornisce un testo *email* e un testo *pass*. |
| **Postcondizioni** | Se esiste un Giocatore Registrato *reg* con *nome account=email* e *password* = *pass*, allora crea un Giocatore *profiloGiocatore* con *nome* pari al *nome utente* di *reg*, associato a *reg*. Altrimenti fallisce senza alcun effetto. |

|  |  |
| --- | --- |
| registra(nome, email, pass) | |
| **Precondizioni** | L’attore fornisce i seguenti testi: *nome, email, pass* |
| **Postcondizioni** | Se non esiste alcun Giocatore Registrato con *nome account* = *email*, allora crea un Giocatore Registrato *reg* con *nome account=email*, *password* = pass, *nome utente=nome.* Altrimenti fallisce senza alcun effetto. |

## UC Gestire Squadre

|  |  |
| --- | --- |
| ottieniInfoSquadre() | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione. |
| **Postcondizioni** | Crea un elenco di tutte le *Squadre* esistenti. |

|  |  |
| --- | --- |
| ottieniIscrittiSquadra(squa) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione.  L’attore identifica una Squadra *squa*. |
| **Postcondizioni** | Crea un elenco di tutti i Giocatori associati a *squa* tramite *iscrizione*. |

|  |  |
| --- | --- |
| iscriviInSquadra(squa) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione, con Giocatore Registrato associato *reg,* e *reg* non è *iscritto a* nessuna Squadra.  L’attore identifica una Squadra *squa* con *aperta*=**vero**. |
| **Postcondizioni** | Crea una associazione di tipo *iscrizione* fra *reg* e *squa*. |

|  |  |
| --- | --- |
| creaSquadra(nomeSquadra) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione, con Giocatore Registrato associato *reg*, e *reg* non ha alcuna associazione di tipo *iscritto a*.  L’attore fornisce un testo *nomeSquadra*. |
| **Postcondizioni** | Se non esiste alcuna Squadra con *nome=nomeSquadra,* crea una nuova Squadra *s* con *nome=nomeSquadra* e *aperta*=**vero**, nonché una associazione di tipo *proprietario* fra *reg* e *s,* e una associazione di tipo *iscritto a* sempre fra *reg* e *s*. Altrimenti fallisce senza effetto. |

|  |  |
| --- | --- |
| disiscriviDaSquadra(squa) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione, con Giocatore Registrato associato *reg*, ed esiste una associazione di tipo *iscritto a* fra *reg* e la squadra *squa* (che quindi è *aperta*). Il Giocatore *reg* non è *proprietario* di *squa*. |
| **Postcondizioni** | Rimuove l’associazione di tipo *iscritto a* fra *reg* e *squa*. |

|  |  |
| --- | --- |
| chiudiSquadra(squa) | |
| **Precondizioni** | E’ univocamente identificato il Giocatore *gioc* che compie l’operazione, con Giocatore Registrato associato *reg*, ed esiste una associazione di tipo *proprietario* fra *reg* e la Squadra *squa*. La Squadra *squa* è *aperta*. |
| **Postcondizioni** | Rimuove tutte le associazioni di tipo *iscritto a/ha iscritti* fra Giocatori Registrati e la Squadra *squa*. L’attributo *aperta* di *squa* diventa **falso**. |

1. Attenzione: è modificato rispetto a quanto riportato nell’UC dettagliato (era esciDaStanza(stanza)). Dal modello di dominio emerge che una volta identificato il Giocatore, è anche univocamente identificata - se esistente - la Stanza in cui *partecipa*. Per cui il parametro *stanza* non è necessario. [↑](#footnote-ref-0)