Sommario

[Revision History 3](#_Toc121301098)

[Team members 3](#_Toc121301099)

[1. Introduzione 4](#_Toc121301100)

[1.1 Obiettivo del sistema 4](#_Toc121301101)

[1.2 Ambito del sistema 5](#_Toc121301102)

[1.3 Obiettivi e criteri di successo 6](#_Toc121301103)

[1.4 Definizioni, acronimi e abbreviazioni 7](#_Toc121301104)

[1.5 Organizzazione del documento 7](#_Toc121301105)

[2. Sistema Attuale 8](#_Toc121301106)

[3. Sistema Proposto 8](#_Toc121301107)

[3.1 Sintesi della sezione 8](#_Toc121301108)

[3.2 Requisiti Funzionali 9](#_Toc121301109)

[3.3 Requisiti non funzionali 13](#_Toc121301110)

[3.4 Modello del Sistema 14](#_Toc121301111)

[3.4.1 Scenari 14](#_Toc121301112)

[3.4.2 Modello Casi D’Uso 30](#_Toc121301113)

[3.4.3 Modello a Oggetti 37](#_Toc121301114)

[3.4.4 Modello Dinamico 45](#_Toc121301115)

[3.4.4.1 Activity Diagram 45](#_Toc121301116)

[3.4.4.2 Sequence Diagram 46](#_Toc121301117)

[3.4.4.3 Statechart Diagram 48](#_Toc121301118)

[3.4.5 Interfacce Utente: Mock-ups 49](#_Toc121301120)

# 

# **Revision History**

Version 0: 29° novembre ’22;

Version 1: 1° dicembre ’22;

Version 2: 6° dicembre ’22;

# **Team members**

|  |  |
| --- | --- |
| Nome | Matricola |
| Carlo Tammaro | 0512112173 |
| Antonio Renzullo | 0512111906 |
| Annachiara Giugliano | 0512111849 |
| Mattia Domenico Garofalo | 0512110994 |

# **Introduzione**

## 1.1 Obiettivo del sistema

Il sistema che si vuole realizzare ha come obiettivo quello di facilitare l’organizzazione di eventi sportivi e quindi avvicinare più persone al mondo dello sport.

Il sistema facilita sia l’organizzazione utente e sia l’organizzazione gestore che, in un'unica piattaforma web ha, sia un calendario di prenotazioni attive, sia un modo semplice di gestire le nuove prenotazioni.

In particolare, il sistema permette:

* Di conoscere nuovi campetti da gioco nella propria zona
* Prenotazione di campetti a distanza
* Di organizzare partite con persone che non si conoscono, ma sono utenti del sistema

Quindi, il sistema permette di aumentare lo sport praticato, e di aumentare anche gli incassi di gestori di campetti.

## Ambito del sistema

L’obiettivo del sistema è il miglioramento dell’organizzazione di

sport. Questo dovrebbe avvicinare ancora di più le

persone agli sport di squadra.

Il sistema permetterà ai gestori di registrarsi e inserire i campetti di

loro proprietà in modo da essere conosciuti da un più vasto

pubblico.

Il sistema permette agli utenti di registrarsi e di scegliere o creare

gli eventi dello sport che preferiscono.

Il sistema permette anche la visualizzazione degli eventi attivi a cui

l’utente può iscriversi.

Nel dettaglio le funzionalità sono:

* Consente all’utente di registrarsi al sistema
* Consente al gestore di registrarsi al sistema
* Consente all’utente di creare e gestire un evento
* Consente al gestore di accettare o rifiutare un evento
* Consente all’utente di effettuare il pagamento
* Consente all’utente di visualizzare la bacheca eventi attivi
* Consente al gestore di visualizzare il calendario degli appuntamenti
* Consente all’utente di visualizzare la propria area utente
* Consente al gestore di visualizzare la sua area personale
* Consente all’utente di effettuare una ricerca filtrata per orari, disponibilità e posizione
* Consente la modifica di dati personali
* Consente il logout
* Consente l’eliminazione dell’account

## Obiettivi e criteri di successo

L’obiettivo del progetto è creare una piattaforma web che sarà di supporto all’organizzazione di sport di squadra.

I criteri di successo sono:

* Ottima qualità; Si intende realizzare un sistema con buona documentazione, al fine di poter essere utilizzato e manutenuto senza grosse difficoltà.
* Interfaccia user-frendly; Si intende realizzare un’interfaccia di facile utilizzo utente.
* Minimizzare i malfunzionamenti; Si intende realizzare un sistema che ha ridotto al minimo i malfunzionamenti.
* Completezza nell’implementazione; Si intende realizzare un sistema che soddisfi tutti i requisiti funzionali in base alle priorità del cliente.
* Rispetto scadenze; Si intende fornire il sistema finale al cliente soddisfando le scadenze stabilite.

## Definizioni, acronimi e abbreviazioni

RF: requisito funzionale

GU: gestione utente

GG: gestione gestore

GE: gestione evento

RNF: requisito non funzionale

SC: scenario

AD: activity diagram

SD: sequence diagram

SCD: statechart diagram

## Organizzazione del documento

1. Introduzione: la seguente sezione contiene l’obiettivo, l’ambito, i criteri di successo del sistema, e una raccolta di definizioni acronimi e abbreviazioni presenti nel documento.
2. Sistema Attuale: descrive il funzionamento manuale dell’organizzazione di una partita sportiva.
3. Sistema proposto: descrive il nuovo sistema, presentandone i requisiti funzionali e non funzionali. Tramite scenari e use case vengono descritti gli attori del sistema e come questi interagiscono con esso. Tramite il modello ad oggetti e il modello dinamico viene descritta la struttura del sistema. Tramite mock-ups e diagrammi navigazionali viene descritta l’interfaccia utente.
4. Glossario: descrive i termini presenti nel documento.

# **Sistema Attuale**

Attualmente risulta complicato per un gruppo di poche persone riuscire ad organizzare match, potrebbero infatti essere richiesti più membri di quanti disponibili e spesso non viene permesso di prenotare campi se non si arriva al numero minimo di giocatori richiesti dallo sport.

# **Sistema Proposto**

## 3.1 Sintesi della sezione

* Requisiti Funzionali: descrizione degli attori e dei requisiti funzionali, ovvero le interazioni tra gli attori e il sistema; questo ovviamente non tiene conto dell’implementazione.
* Requisiti Non Funzionali: descrive gli aspetti non funzionali del sistema, ad esempio usabilità, manutenzione e prestazioni.
* Modello del sistema:
* Scenari: descrizione informale dell’uso del sistema da parte dell’utente finale.
* Modello Casi D’Uso: descrizione completa delle interazioni che avvengono quando un attore utilizza il sistema.
* Modello ad Oggetti: descrizione tramite un class diagram dell’intero sistema, formato da tutti gli oggetti che lo compongono.
* Modello Dinamico: rappresenta la struttura dinamica del sistema.
* Mock-ups: rappresentazione dell’interfaccia grafica.

## Requisiti Funzionali

Nella presente sezione verranno elencati i requisiti funzionali del sistema proposto.

Si è deciso di divedere i requisiti in tre diverse gestioni:

* Gestione Utente (GU)
* Gestione Gestore (GG)
* Gestione Evento (GE)

Gli attori del sistema sono:



Ospite: utente non registrato al sistema e quindi può accedere solo a operazioni di visualizzazione.

Utente Registrato: utente registrato al sistema e può accedere all’intero sistema dedicato.

Gestore Registrato: gestore registrato al sistema e può accedere all’intero sistema dedicato.

RF\_GU: Gestione Utente

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NOME** | **DESCRIZIONE** | **ATTORE** | **PRIORITA’** |
| RF\_GU\_1 | Registrazione Utente | L’utente si iscrive alla piattaforma inserendo i suoi dati | Ospite | ALTA |
| RF\_GU\_2 | Login | L’utente effettua il login al sito tramite le sue credenziali | Utente | ALTA |
| RF\_GU\_3 | Logout | L’utente decide di uscire dalla piattaforma | Utente | ALTA |
| RF\_GU\_4 | Gestione Area Utente | L’utente vede e personalizza la sua area utente | Utente | ALTA |
| RF\_GU\_5 | Modifica Dati Utente | L’utente decide di modificare i propri dati personali | Utente | ALTA |
| RF\_GU\_6 | Visualizzazione Bacheca Eventi Attivi | L’utente visualizza la bacheca di tutti gli eventi attivi | Utente | MEDIA |
| RF\_GU\_7 | Cancellazione Account | L’utente decide di eliminare il proprio account dalla piattaforma | Utente | ALTA |

RF\_GG: Gestione Gestore

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NOME** | **DESCRIZIONE** | **ATTORE** | **PRIORITA’** |
| RF\_GG\_1 | Registrazione Gestore | Il gestore si iscrive alla piattaforma inserendo i suoi dati e i dati relativi ai campi a sua disposizione | Ospite | ALTA |
| RF\_GG\_2 | Visualizzazione Bacheca Gestore | Il gestore visualizza la propria bacheca in cui è visibile anche il calendario appuntamenti | Gestore | ALTA |
| RF\_GG\_3 | Modifica Campetti | Il gestore decide di modificare informazioni relative ai propri campi | Gestore | ALTA |
| RF\_GG\_4 | Accettazione o Rifiuto Evento | Il gestore decide se accettare o meno una data richiesta di prenotazione | Gestore | ALTA |

RF\_GE: Gestione Evento

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NOME** | **DESCRIZIONE** | **ATTORE** | **PRIORITA’** |
| RF\_GE\_1 | Creazione Evento | L’utente crea un nuovo evento di uno sport a piacere | Utente | ALTA |
| RF\_GE\_2 | Invito Evento | L’utente invita a partecipare ad un evento altri utenti del sistema | Utente | BASSA |
| RF\_GE\_3 | Adesione Evento | L’utente decide di aderire ad un evento attivo | Utente | MEDIA |
| RF\_GE\_4 | Eliminazione Evento | L’utente decide di eliminare l’evento precedentemente creato | Utente | BASSA |
| RF\_GE\_5 | Pagamento Prenotazione | L’utente, creatore dell'evento, effettua il pagamento | Utente | ALTA |
| RF\_GE\_6 | Ricerca per Orario, Disponibilità e Posizione | L’utente effettua una ricerca per capire la disponibilità di eventi | Utente | ALTA |

## Requisiti non funzionali

In questa sezione saranno elencati i requisiti non funzionali del sistema.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **ID** | **NOME** | **DESCRIZIONE** | **PRIORITA’** | **DIFFICOLTA’** |
| RNF\_1 | Prestazione\_1 | Risposta del  server in massimo  2 secondi | ALTA | MEDIA |
| RNF\_2 | Usabilità\_1 | Interfaccia grafica  intuitiva e di facile  utilizzo | ALTA | ALTA |
| RNF\_3 | Sicurezza\_1 | Crittografia  password e  informazioni  sensibili di  account utenti e  gestori | ALTA | ALTA |
| RNF\_4 | Implementazione\_1 | Utilizzo di Java | BASSA | BASSA |
| RNF\_5 | Implementazione\_2 | Il sistema verrà  sviluppato come  una piattaforma  Web | ALTA | MEDIA |
| RNF\_6 | Usabilità\_2 | Il sistema dovrà  essere responsive | ALTA | ALTA |
| RNF\_7 | Prestazione\_2 | Sistema  disponibile 24/7 | MEDIA | ALTA |
| RNF\_8 | Manutenibilità\_1 | Sistema  sviluppato in  modo da essere  ben manutenibile | ALTA | MEDIA |
| RNF\_9 | Affidabilità\_1 | Il sistema deve sapersi comportare in situazioni di fallimento notificando l’utente, tramite appositi messaggi. | MEDIA | ALTA |

## Modello del Sistema

Nella presente sezione vengono illustrati: gli scenari d’uso del sistema, il modello dei casi d’uso del sistema, il modello a oggetti e il modello dinamico.

## Scenari

Nella presente sezione sono elencati alcuni scenari relativi all’utilizzo del sistema, divisi per gestione.

**SC\_GU: Gestione Utente**

| NOME SCENARIO | SC\_GU\_1 Registrazione utente | |
| --- | --- | --- |
| ATTORI | Ospite: Valerio | |
| DESCRIZIONE | L’utente si iscrive alla piattaforma inserendo i suoi dati | |
| FLUSSO DEGLI EVENTI | Ospite | Sistema |
|  | Valerio, ragazzo di Roma, è un grande appassionato di Tennis. I suoi amici però non amano altrettanto questo sport e lui non riesce mai ad organizzare delle partite. Per soccombere al suo problema i suoi amici gli propongono Happy Fields.  Valerio così decide di registrarsi alla piattaforma come utente, inserendo tutte le informazioni, specificando il luogo dove vive e le sue preferenze in ambito sportivo. |  |
|  | Il sistema visualizza una pagina per l’inserimento dei dati. |
| Valerio controlla i dati richiesti e  inserisce “valerio38”come Username, [v.alerio38@studenti.unisa.it](mailto:v.alerio38@studenti.unisa.it) come Email, “happyfields80” come Password, reinserisce “happyfields80” come Conferma Password, “Valerio Mazzei” come Nome e Cognome, “18-07-1996” come Data di Nascita, “Roma, Fiumicino, Via delle Barche 56, 00054” come Provincia, Città, Via in cui risiede e CAP, “3669909768” come Numero di Telefono.  Valerio conferma la registrazione. |  |
|  | Il sistema controlla che tutti i campi siano stati inseriti e controlla che l’Username non sia associato ad un account già esistente e che le Password inserite coincidano. In seguito, il sistema comunica a Valerio che la registrazione ha avuto successo. |

| NOME SCENARIO | SC\_GU\_2 Gestione Area Utente | |
| --- | --- | --- |
| ATTORI | Utente: Pasquale | |
| DESCRIZIONE | Pasquale accede all’area utente per visionare lo storico delle proprie partite | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Pasquale, iscritto alla piattaforma HappyFields, decide di accedere alla propria area utente per visionare le partite effettuate nell’ultimo mese, selezionando la voce ‘storico partite’ |  |
|  | Il sistema restituisce a Pasquale una lista delle partite effettuate |

| NOME SCENARIO | SC\_GU\_3 Modifica Dati Utente | |
| --- | --- | --- |
| ATTORI | Utente: Giovanni | |
| DESCRIZIONE | L’utente decide di modificare i propri dati personali. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Giovanni ha cambiato residenza, vuole pertanto cercare partite più vicine alla sua attuale locazione. Per tale motivo provvede ad aggiornare i suoi dati personali sulla piattaforma HappyFields inserendo il nuovo indirizzo della sua abitazione. Giovanni per fare ciò clicca su gestisci profilo nella sua area utente. |  |
|  | Il sistema mostra a Giovanni la sua pagina di gestione del profilo. |
| Giovanni sceglie di modificare solo la residenza inserendo la nuova nel campo richiesto e cliccando sul bottone “Salva Modifiche”. |  |
|  | Il sistema salva le modiche apportate da Giovanni e notifica il successo dell’azione. |

| NOME SCENARIO | SC\_GU\_4 Visualizzazione Bacheca Eventi Attivi | |
| --- | --- | --- |
| ATTORI | Utente: Paolo | |
| DESCRIZIONE | L’utente visualizza la bacheca di tutti gli eventi attivi. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Paolo è uno sportivo di livello agonistico e sta cercando nuovi player contro i quali gareggiare. Apre la sua bacheca degli eventi attivi in modo da poter trovare quello più idoneo a lui per luogo dell’incontro e orario fissato. |  |
|  | Il sistema mostra tutti gli eventi aperti ancora non completati secondo le specifiche date da Paolo. |
| Paolo vede la lista degli eventi e clicca sul bottone di partecipazione degli eventi che più gli interessano. |  |
|  | Il sistema notifica agli admin degli eventi che c’è un nuovo partecipante e comunica il successo delle azioni. |

| NOME SCENARIO | SC\_GU\_5 Cancellazione Account | |
| --- | --- | --- |
| ATTORI | Utente: Pasquale | |
| DESCRIZIONE | L’utente decide di eliminare il proprio account dalla piattaforma. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Pasquale, utente della piattaforma HappyFields e giocatore di basket da oltre 30 anni, decide dopo una serie di sfortunati infortuni che è arrivata l’ora di appendere le scarpette al chiodo e per farlo decide di cancellare il suo account per non cadere in tentazione di tornare sul campetto e rischiare di provocarsi un infortunio ancora più grave. Per questo motivo va nelle impostazioni del proprio account di HappyFields e clicca sulla voce in rosso “Elimina Account”. |  |
|  | Il sistema mostra a schermo una finestra in cui si richiede all’ utente se si è sicuri della scelta. |
| Pasquale clicca sul si apparso nella finestra. |  |
|  | Il sistema accetta la richiesta di Pasquale ed elimina tutte le sue informazioni (UserName, Password, ecc.)  dalla piattaforma e restituisce la pagina iniziale. |

**SC\_GG: Gestione Gestore**

| NOME SCENARIO | SC\_GG\_1 Registrazione Gestore | |
| --- | --- | --- |
| ATTORI | Ospite: Aldo | |
| DESCRIZIONE | Il gestore si iscrive alla piattaforma inserendo i suoi dati e i dati relativi ai campi a sua disposizione. | |
| FLUSSO DEGLI EVENTI | Ospite | Sistema |
|  | Aldo, imprenditore e gestore dei campi sportivi Queens Garden decide di ampliare la sua clientela iscrivendosi alla piattaforma Happy Fields. In fase di iscrizione gli viene chiesto, oltre alle informazioni anagrafiche, di inserire i dati relativi agli orari di apertura dei propri campi, il costo ed altre informazioni circa il servizio offerto. Una volta terminata la fase di inserimento delle informazioni agli utenti sarà possibile visualizzare i suoi campi ed eventualmente prenotarli. |  |
|  | Il sistema visualizza una pagina per l’inserimento dei dati. |
| Aldo controlla i dati richiesti e inserisce “aldo64”come Username, c.aldo64@gmail.com come Email, “happyfields64” come Password, reinserisce “happyfields64” come Conferma Password, “Aldo Castaldo” come Nome e Cognome, “25-12-1964” come Data di Nascita, “Milano, Barona, Via Lago di Noemi 12, 20142” come Provincia, Città, Via in cui risiede e CAP, “3894567430” come Numero di Telefono,”09:00-22:00” come orari di apertura dei propri campi,”5 euro” come prezzo per persona e una rapida descrizione del servizio offerto .  Aldo conferma la registrazione. |  |
|  | Il sistema controlla che tutti i campi siano stati inseriti e controlla che l’Username non sia associato ad un account già esistente e che le Password inserite coincidano. In seguito, il sistema comunica ad Aldo che la registrazione ha avuto successo. |

| NOME SCENARIO | SC\_GG\_2 Visualizzazione bacheca gestore | |
| --- | --- | --- |
| ATTORI | Gestore: Gennaro | |
| DESCRIZIONE | Gennaro visualizza la bacheca per visionare il calendario degli appuntamenti | |
| FLUSSO DEGLI EVENTI | Gestore | Sistema |
|  | Gennaro, proprietario di tre campi da calcio nella zona di Marigliano, decide di effettuare un controllo sulle prenotazioni ricevute nell’ultimo periodo; dunque, accede alla sezione “calendario degli appuntamenti”. |  |
|  | Il sistema mostra a Gennaro la lista di tutte le prenotazioni ricevute, suddividendole per campo |
| Gennaro, soddisfatto, nota che grazie ad HappyFields le prenotazioni stanno aumentate drasticamente |  |

|  |  |  |
| --- | --- | --- |
| NOME SCENARIO | SC\_GG\_3 Modifica campetti | |
| ATTORI | Gestore: Marco | |
| DESCRIZIONE | Marco decide di modificare informazioni relative ai propri campi | |
| FLUSSO DEGLI EVENTI | Gestore | Sistema |
|  | Marco, gestore di campi della piattaforma HappyFields, decide di ampliare la sua struttura e di modificare la grandezza di uno dei suoi campi da calcio trasformandolo in campo regolamentare da calcio a undici. Dopo la restaurazione del campo Marco decide di modificare le informazioni anche sulla piattaforma per tenere aggiornati gli utenti. |  |
|  | Il sistema restituisce a Marco la lista dei suoi campi |
| Marco seleziona il campo del quale vuole modificare le informazioni |  |
|  | Il sistema mostra a Marco: nome del campo, dimensione e immagine allegata |
| Marco decide di modificare le dimensioni del campo |  |
|  | Il sistema mostra a Marco un modulo con tutti i dati del campo preinseriti |
|  | Marco sceglie di modificare le dimensioni del campo, modificando le credenziali  -‘lunghezza’ in 90 mt  -‘larghezza’ in 45 mt |  |
|  |  | Il sistema comunica a Marco che le credenziali sono state modificate correttamente |

| NOME SCENARIO | SC\_GG\_4 Accettazione o rifiuto evento | |
| --- | --- | --- |
| ATTORI | gestore: Mario | |
| DESCRIZIONE | Mario visualizzando la lista delle sue prenotazioni ai campi, decide di rifiutare un evento | |
| FLUSSO DEGLI EVENTI | Gestore | Sistema |
|  | Mario, proprietario del centro sportivo Golden Goal di Pontecagnano Faiano visualizza la lista delle richieste di prenotazioni dei suoi campi |  |
|  | Il sistema mostra a Mario la lista delle richieste di prenotazioni effettuate |
| Scorrendo la lista, Mario nota la richiesta di Pasquale per una partita di calcio a 11 alle ore 15.30 del sabato seguente, purtroppo però quel giorno ci sarà manutenzione; dunque, Mario decide di rifiutare l’evento selezionando la voce ‘rifiuta’ |  |
|  | Il sistema avverte subito Pasquale attraverso un’e-mail, invogliandolo ad effettuare la prenotazione in un’altra data disponibile |

**SC\_GE: Gestione Evento**

| NOME SCENARIO | SC\_GE\_1 Creazione Evento | |
| --- | --- | --- |
| ATTORI | Utente: Pasquale | |
| DESCRIZIONE | L’utente crea un nuovo evento di uno sport a piacere. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Pasquale, ragazzo di Salerno e amante di basket, vuole organizzare una partita ma non riesce a trovare componenti. Decide quindi di aprire HappyFields e di creare un evento. |  |
|  | Il sistema gli mostra la pagina di creazione di un di un evento dove viene richiesto di inserire i seguenti campi: lo sport, il campo, l’orario e la data. |
| Pasquale, quindi, inserisce “Basket” come sport, “M. Jordan” come campo, “19:00” come orario e “12-10-2022” come data. |  |
|  | Il sistema verifica che i dati siano validi e invia la richiesta di creazione dell’evento al gestore del campo. Il sistema, inoltre, notifica a Pasquale che la richiesta è stata effettuata con successo. |

| NOME SCENARIO | SC\_GE\_2 Invito Evento | |
| --- | --- | --- |
| ATTORI | Utente: Paolo | |
| DESCRIZIONE | L’utente invita a partecipare ad un evento altri utenti del sistema. | |
| FLUSSO DEGLI EVENTI | Ospite | Sistema |
|  | Paolo ha organizzato una partita di padel con i suoi amici e manca il quarto componente. Allora decide di entrare su HappyFields e invitare altri utenti della piattaforma. |  |
|  | Il sistema mostra a Paolo una lista di utenti, nelle sue vicinanze, che cercano una partita di padel. |
| Paolo decide di invitare Gennaro e Maria, entrambi amanti di padel. |  |
|  | Il sistema notifica a Gennaro e Maria dell’invito tramite una notifica |

| NOME SCENARIO | SC\_GE\_3 Adesione Evento | |
| --- | --- | --- |
| ATTORI | Utente: Gennaro | |
| DESCRIZIONE | L’utente decide di aderire ad un evento attivo. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Gennaro, ragazzo di Salerno, ama giocare a basket e il fine settimana vorrebbe partecipare a una partita, decide di controllare sulla piattaforma HappyFields se ci sono già eventi aperti. |  |
|  | Il sistema mostra a Gennaro gli eventi aperti che riguardano il basket per questo fine settimana. |
| Nota che Pasquale ha creato un evento per il sabato alle ore 7. Decide dunque di partecipare al suo evento. |  |
|  | Il sistema notifica a Gennaro che la partecipazione è avvenuta con successo e a Pasquale del nuovo ingresso nel suo evento. |

| NOME SCENARIO | SC\_GE\_4 Eliminazione Evento | |
| --- | --- | --- |
| ATTORI | Utente: Marco | |
| DESCRIZIONE | L’utente decide di eliminare l’evento precedentemente creato. | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Marco, utente della piattaforma, vede che purtroppo nessuno ha deciso di partecipare al suo evento in scadenza il giorno successivo e quindi per non far perdere tempo al gestore del campo decide di cancellare la sua prenotazione cliccando sul bottone “Elimina evento” all’interno dell’evento. |  |
|  | Il sistema mostra a video una finestra nella quale si chiede una descrizione del motivo della cancellazione. |
| Marco descrive la sua motivazione e invia la descrizione. |  |
|  | Il sistema fa apparire la cancellazione e la motivazione nell’area utente del gestore e notifica a Marco l’avvenuta cancellazione dell’evento. |

| NOME SCENARIO | SC\_GE\_5 Pagamento prenotazione | |
| --- | --- | --- |
| ATTORI | Utente: Pasquale | |
| DESCRIZIONE | Pasquale decide di effettuare il pagamento per il campo da basket prenotato | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Pasquale, una volta raggiunti i partecipanti necessari per l’organizzazione della partita che intende effettuare, procede alla transazione per la prenotazione del campo, cliccando sulla voce ‘pagamento’, decidendo di occuparsi in un secondo momento di riscuotere le somme di tutti gli altri partecipanti personalmente. |  |
|  | Il sistema mostra a Pasquale un modulo da compilare, in cui inserire i seguenti dati inerenti al pagamento:  -modalità di pagamento  -nome titolare del conto  -numero di carta  -data di scadenza della carta |
| Pasquale inserisce le credenziali richieste ed effettua il pagamento cliccando sulla voce ‘procedi alla prenotazione’ |  |
|  | Il sistema verifica la validità dei dati, e successivamente informa Pasquale che il pagamento è stato effettuato con successo |

| NOME SCENARIO | SC\_GE\_6 Ricerca per orario, luogo e disponibilità | |
| --- | --- | --- |
| ATTORI | Utente: Pasquale | |
| DESCRIZIONE | Pasquale decide di effettuare una ricerca per orario e posizione nella bacheca eventi attivi per capire la disponibilità di un evento a cui può partecipare | |
| FLUSSO DEGLI EVENTI | Utente | Sistema |
|  | Pasquale oggi vuole giocare a calcetto. I suoi amici non sono della sua stessa opinione e quindi decide di cercare un evento su HappyFields. Apre il sistema e clicca sulla bacheca eventi attivi, quindi effettua una ricerca con filtri, indicando posizione, orario e data. |  |
|  | Il sistema verifica la disponibilità e sottopone all’utente una pagina con gli eventi che rispettano i suoi filtri. |
| Pasquale adesso può scegliere tra gli eventi attivi quello a cui vuole aderire. |  |

## Modello Casi D’Uso

Nella presente sezione sono elencati i casi d’uso divisi per gestione.

**Gestione Utente**



|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UC\_GU\_1 | | Registrazione utente | Data | 08-11-22 |
| Vers. | 0.1 |
| Autore | Carlo Tammaro |
| **Descrizione** | | Lo UC fornisce la funzionalità di registrazione dell’utente al sistema | | |
| **Attore Principale** | | Ospite | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | L’ospite si reca nella sezione di registrazione | | |
| **Exit Condition**                         On success | | L’ospite visualizza “Registrazione completata” | | |
| **Exit Condition**                         On failure | | L’ospite riceve un messaggio di errore dovuto all’errato riempimento dei campi di registrazione | | |
| **Rilevanza/User Priority** | | Alta | | |
| **Frequenza stimata** | | 5/giorno per utente | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Ospite | L’ospite entra nella sezione dedicata alla registrazione. | | |
| 2. | Sistema | Il sistema mostra il form con i campi che l’ospite deve riempire per la registrazione. | | |
| 3. | Ospite | L’ospite inserisce i dati e conferma l’invio del form | | |
| 4. | Sistema | Il sistema verifica la correttezza dei dati inseriti dall’ospite conferma la registrazione | | |
|  | | | | |
| **Scenario/Flusso di eventi Alternativo: Uno dei campi inseriti non risulta valido** | | | | |
| **4.a1** | Sistema | Il sistema segnala al lettore la non correttezza di uno dei campi | | |
| **4.a2** | Sistema | Il sistema resta in attesa di un nuovo inserimento | | |

**Gestione Gestore**



**Gestione Evento**



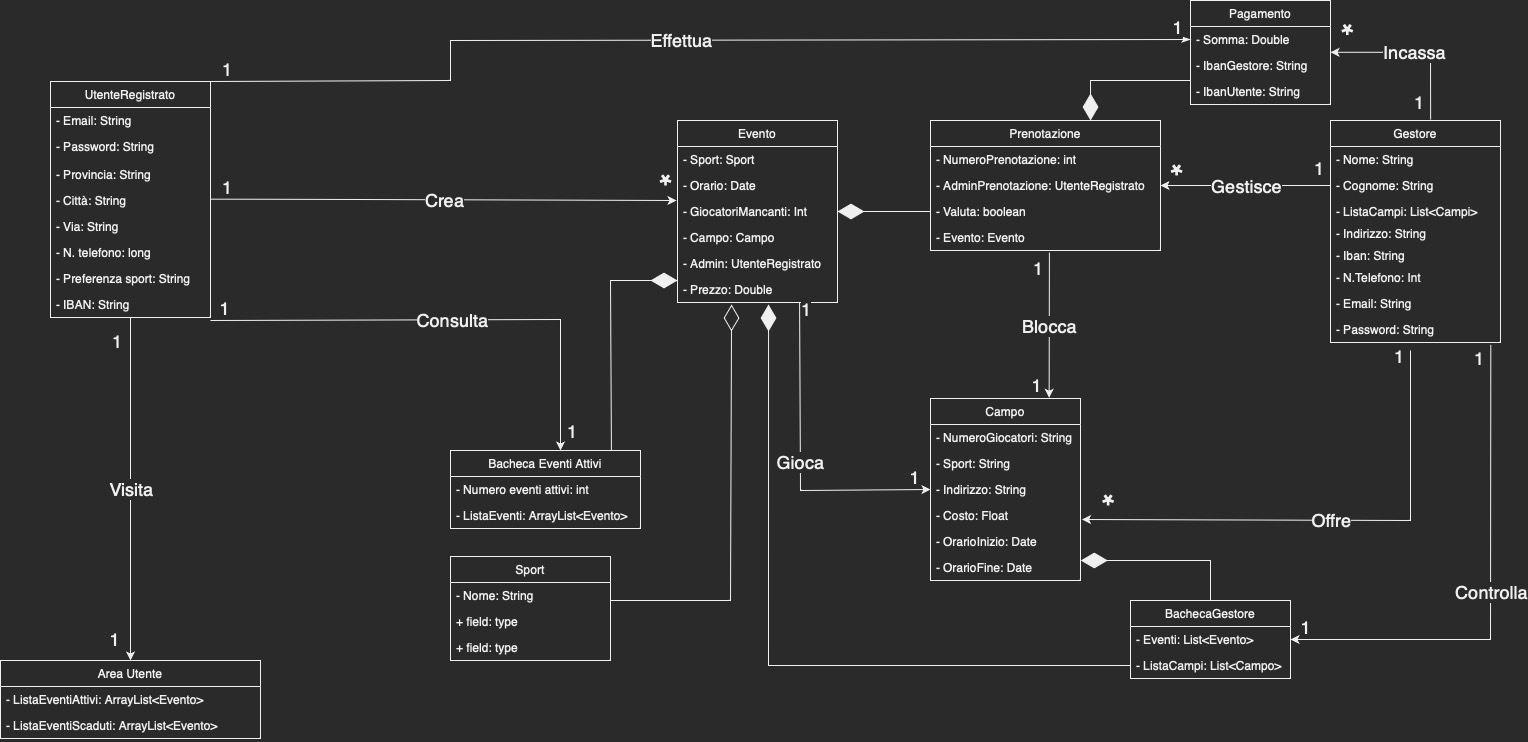
| **Identificativo**  UC\_GE\_1 | | Ricerca evento per posizione, orario e sport | Data | 07-11-22 |
| --- | --- | --- | --- | --- |
| Vers. | 0.1 |
| Autore | Antonio Renzullo |
| **Descrizione** | | Lo UC fornisce la funzionalità di creazione dell’evento per orario, posizione e disponibilità. | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | L’utente si trova nella bacheca eventi attivi. | | |
| **Exit Condition**  On success | | L’utente visualizza una lista dei campi dello sport selezionato disponibili, all’orario e prossimi alla locazione da lui inseriti. | | |
| **Exit Condition**  On failure | | L’utente riceve un messaggio di errore “Nessun campo disponibile”. | | |
| **Rilevanza/User Priority** | | Alta | | |
| **Frequenza stimata** | | 1/giorno per utente | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | L’utente effettua un ricerca in cui può filtrare sport, posizione e orario. | | |
| 2. | Sistema | Il sistema risponde con una lista di eventi attivi in base ai filtri selezionati dall’utente. | | |
| **Scenario/Flusso di eventi di ERRORE: Il sistema non riesce a trovare alcun campo disponibile** | | | | |
| **2.e1** | Sistema | Il sistema segnala all’utente che non è stato possibile trovare nessun campo disponibile | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UC\_GE\_2 | | Creazione di un evento | Data | 08-11-22 |
| Vers. | 0.1 |
| Autore | Garofalo Domenico Mattia |
| **Descrizione** | | Lo UC fornisce la funzionalità di creazione di un evento sportivo. | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | L’utente clicca sul bottone di creazione evento. | | |
| **Exit condition**                         On success | | La creazione è avvenuta con successo e gli altri utenti possono visualizzare e partecipare all’evento. | | |
| **Exit condition**                         On failure | | La creazione non è avvenuta con successo e l’evento non è stato creato per tanto il campo non è stato prenotato per giocare. | | |
| **Rilevanza/User Priority** | | Elevata | | |
| **Frequenza stimata** | | 1/giorno per utente | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | Richiede di creare un evento. | | |
| 2. | Sistema | Il sistema visualizza un form che richiede di inserire i dati sull’evento. | | |
| 3. | Utente | L’utente inserisce i dati e conferma l’invio del form. | | |
| 4. | Sistema | Il sistema verifica la correttezza dei dati inseriti dall’utente e conferma la creazione dell’evento | | |
|  | | | | |
| **Scenario/Flusso di eventi Alternativo: Uno dei campi inseriti non risulta valido** | | | | |
| **4.a1** | Sistema | Il sistema segnala al lettore la non correttezza di uno dei campi | | |
| **4.a2** | Sistema | Il sistema resta in attesa di un nuovo inserimento | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Identificativo**  UC\_GE\_3 | | Pagamento della prenotazione | Data | 08-11-22 |
| Vers. | 0.1 |
| Autore | Giugliano Annachiara |
| **Descrizione** | | Lo UC fornisce la funzionalità di pagamento per la prenotazione | | |
| **Attore Principale** | | Utente | | |
| **Attori secondari** | | NA | | |
| **Entry Condition** | | l’evento creato dall’utente ha raggiunto il numero di iscritti necessari AND l’utente clicca sul bottone di pagamento | | |
| **Exit Condition**                         On success | | il pagamento viene effettuato con successo | | |
| **Exit Condition**                         On failure | | l’utente visualizza un messaggio di errore con un invito a riprovare | | |
| **Rilevanza/User Priority** | | Elevata | | |
| **Frequenza stimata** | | 5/giorno per utente | | |
| **FLUSSO DI EVENTI PRINCIPALE/MAIN SCENARIO** | | | | |
| 1. | Utente | richiede di effettuare il pagamento | | |
| 2. | Sistema | Sottopone all’utente un form che richiede i dati di pagamento | | |
| 3. | Utente | inserisce i dati nel form e invia i dati | | |
| 4. | Sistema | Il sistema verifica la correttezza dei dati inseriti dall’utente e conferma il pagamento | | |
|  | | | | |
| **Scenario/Flusso di eventi Alternativo: Uno dei campi inseriti non risulta valido** | | | | |
| **4.a1** | Sistema | Il sistema segnala al lettore la non correttezza di uno dei campi | | |
| **4.a2** | Sistema | il sistema resta in attesa di un nuovo inserimento | | |

## Modello a Oggetti

Nella presente sezione è illustrato il class diagram che descrive l’intero sistema. Il diagramma è presentato molto ad alto livello, descrivendo al più tutti gli oggetti entity del nostro sistema. Successivamente, viene presentata la raccolta di oggetti divisa per oggetti.



Gestione Utente

|  |  |  |
| --- | --- | --- |
| Nome oggetto | Tipologia | Descrizione |
| Utente Registrato | Entity | Utente che si è registrato al sistema |
| RegistrazioneButton | Boundary | Pulsante che permette la registrazione |
| RegistrazioneForm | Boundary | Campi da riempire per effettuare la registrazione |
| ConfermaRegistrazioneButton | Boundary | Pulsante per confermare la rregistrazione |
| RegistrazioneConfermataAck | Boundary | Acknowledgement inviato dal sistema per confermare la registrazione |
| RegistrazioneErroreAck | Boundary | Acknowledgement inviato dal sistema per segnalare un errore nella registrazione |
| LoginButton | Boundary | Pulsante per accedere al sistema |
| LoginForm | Boundary | Campi da compilare per effettuare il login |
| ConfermaLoginButton | Boundary | Pulsante per confermare il login |
| LogoutButton | Boundary | Pulsante per effettuare il logout |
| VisualizzazioneAreaUtente | Boundary | Pulsante per visualizzare la propria area utente |
| ModificaDatiButton | Boundary | Pulsante per modificare i propri dati |
| ModificaDatiForm | Boundary | Form compilato con i dati attuali nel quale è possibile modificare i dati |
| VisualizzazioneEventiAttivi | Boundary | Pulsante per visualizzare gli eventi attualmente aperti e non pieni |
| CancellaDatiAccountButton | Boundary | Pulsante di eliminazione del proprio account |
| RegistrazioneUtenteControl | Control | Controlla l’esito della registrazione |
| LoginControl | Control | Controlla l’esito del login |
| LogoutControl | Control | Controlla l’esisto del logout |
| VisualizzazioneAreaUtenteControl | Control | Carica l’area utente |
| ModificaDatiControl | Control | Controlla l’esito della modifica dei dati |
| VisualizzaBachecaEventiAttivi | Control | Carica la bacheca degli eventi attivi |
| CancellazioneDatiAccountControl | Control | Controlla la corretta eliminazione dell’account |

Gestione Gestore

|  |  |  |
| --- | --- | --- |
| NOME | TIPO | DESCRIZIONE |
| GestoreRegistrato | Entity | Un gestore di campetti che ha effettuato la registrazione |
| Evento | Entity | Evento sportivo creato in un campetto |
| Campo | Entity | Luogo in cui si svolge l’evento |
| RegistrazioneButton | Boundary | Pulsante per aprire il form di registrazione |
| RegistrazioneForm | Boundary | Form da compilare per effettuare la registrazione |
| ConfermaRegistrazioneButton | Boundary | Pulsante per completare la registrazione |
| RegistrazioneConfermataAck | Boundary | Acknowledgement inviato dal sistema per confermare la registrazione |
| RegistrazioneErroreAck | Boundary | Acknowledgement inviato dal sistema per segnalare un errore nella registrazione |
| LoginButton | Boundary | Pulsante per effettuare il login |
| LoginForm | Boundary | Form da compilare per effettuare il login |
| ConfermaLoginButton | Boundary | Pulsante per confermare il login |
| LogoutButton | Boundary | Pulsante per effettuare il logout dell’account |
| VisualizzazioneBachecaButton | Boundary | Pulsante per visualizzare la propria area utente |
| ModificaDatiButton | Boundary | Pulsante per modificare i dati personali |
| ModificaDatiForm | Boundary | Form compilato con i dati attuali nel quale è possibile modificare i dati |
| CancellaDatiAccountButton | Boundary | Pulsante per l’eliminazione del proprio account |
| ModificaCampiButton | Boundary | Pulsante per modificare i dati relativi ai campi |
| ModificaCampiPage? | Boundary | Pagina con i campi attuali |
| VisualizzazioneCalendarioAppuntamentiButton | Boundary | Pulsante per la visualizzazione dei campi prenotati |
| VisualizzazioneCalendarioAppuntamentiPage? | Bondary | Pagina delle prenotazioni |
| RegistrazioneGestoreControl | Control | Controlla l’esito della registrazione |
| LoginControl | Control | Controlla l’esito del login |
| LogoutControl | Control | Controlla l’esito dell logout |
| VisualizzazioneBachecaControl | Control | Carica la bacheca gestore |
| ModificaDatiControl | Control | Controlla la modifica dei dati |
| ModificaCampiControl | Control | Controlla la modifica dei campi |
| VisualizzazioneCalendarioAppuntamentiControl | Control | Carica il calendario degli appuntamenti |
| CancellaDatiAccountControl | Control | Controlla la corretta eliminazione dell’account |

Gestione Evento

|  |  |  |
| --- | --- | --- |
| UtenteRegistrato | Entity | Un utente che ha effettuato una registrazione |
| GestoreRegistrato | Entity | Un gestore di campetti che ha effettuato la registrazione |
| BachecaGestore | Entity | Bacheca relativa all’account gestore (per accettare e rifiutare l’evento) |
| Evento | Entity | Entità che gestisce le partite |
| Campi | Entity | Luogo di svolgimento degli sport |
| Sport | Entity | Tipo di attività effettuata |
| Prenotazione | Entity | Prenotazione di un campo per effettuare un evento |
| PagamentoPrenotazione | Entity | Pagamento per completare la prenotazione |
| AreaUtente | Entity | Bacheca relativa all’utente registrato |
| CreazioneEventoButton | Boundary | Pulsante per la creazione di eventi |
| CreazioneEventoForm | Boundary | Form per l’inserimento dati relativi agli eventi |
| CreazioneConfermataAck | Boundary | Acknowledgement della creazione confermata dell’evento |
| CreazioneErroreAck | Boundary | Acknowledgement della creazione errata dell’evento |
| InvitoEventoButton | Boundary | Pulsante per effettuare un invito a un evento |
| InvitoConfermatoAck | Boundary | Acknowledgement dell’invito andato a buon fine |
| AderisciEventoButton | Boundary | Pulsante di adesione a un evento |
| AdesioneConfermataAck | Boundary | Acknowledgement dell’adesione andata a buon fine |
| EliminaEventoButton | Boundary | Pulsante per l’eliminazione dell’evento |
| EliminazioneEventoConfermataAck | Boundary | Acknowledgement dell’eliminazione avvenuta con successo |
| EliminazioneEventoErroreAck | Boundary | Acknowledgement di un errore durante l’eliminazine |
| PagamentoButton | Boundary | Pulsante per effettuare il pagamento |
| PagamentoForm | Boundary | Campi da compilare relativi al pagamento |
| PagamentoConfermatoAck | Boundary | Acknowledgement del pagamento avvenuto con successo |
| PagamentoErroreAck | Boundary | Acknowledgement di un errore durante il pagamento |
| RicercaConFiltri (orario, disponibilità e posizione) | Boundary | Sezione dedicata alla ricerca di campi o di eventi attivi |
| EffettuaRicercaButton | Boundary | Pulsante per effettuare la ricerca |
| AccettaEventoButton | Boundary | Pulsante per accettare l’evento da parte del gestore |
| RifiutaEventoButton | Boundary | Pulsante per rifiutare l’evento da parte del gestore |
| CreazioneEventoControl | Control | Controlla l’esito della creazione dell’evento |
| InvitoEventoControl | Control | Controlla l’esito dell’invito a un evento |
| AdesioneEventoControl | Control | Controlla l’esito dell’adesione a un evento |
| EliminazioneEventoControl | Control | Controlla l’esito dell’eliminazione di un evento |
| PagamentoPrenotazioneControl | Control | Controlla l’esito del pagamento |
| RicercaControl | Control | Controlla l’esito della ricerca |
| AccettaEventoControl | Control | Controlla l’esito della richiesta accettata |
| RifiutaEventoControl | Control | Controlla l’esito della richiesta rifiutata |

## Modello Dinamico

## Activity Diagram

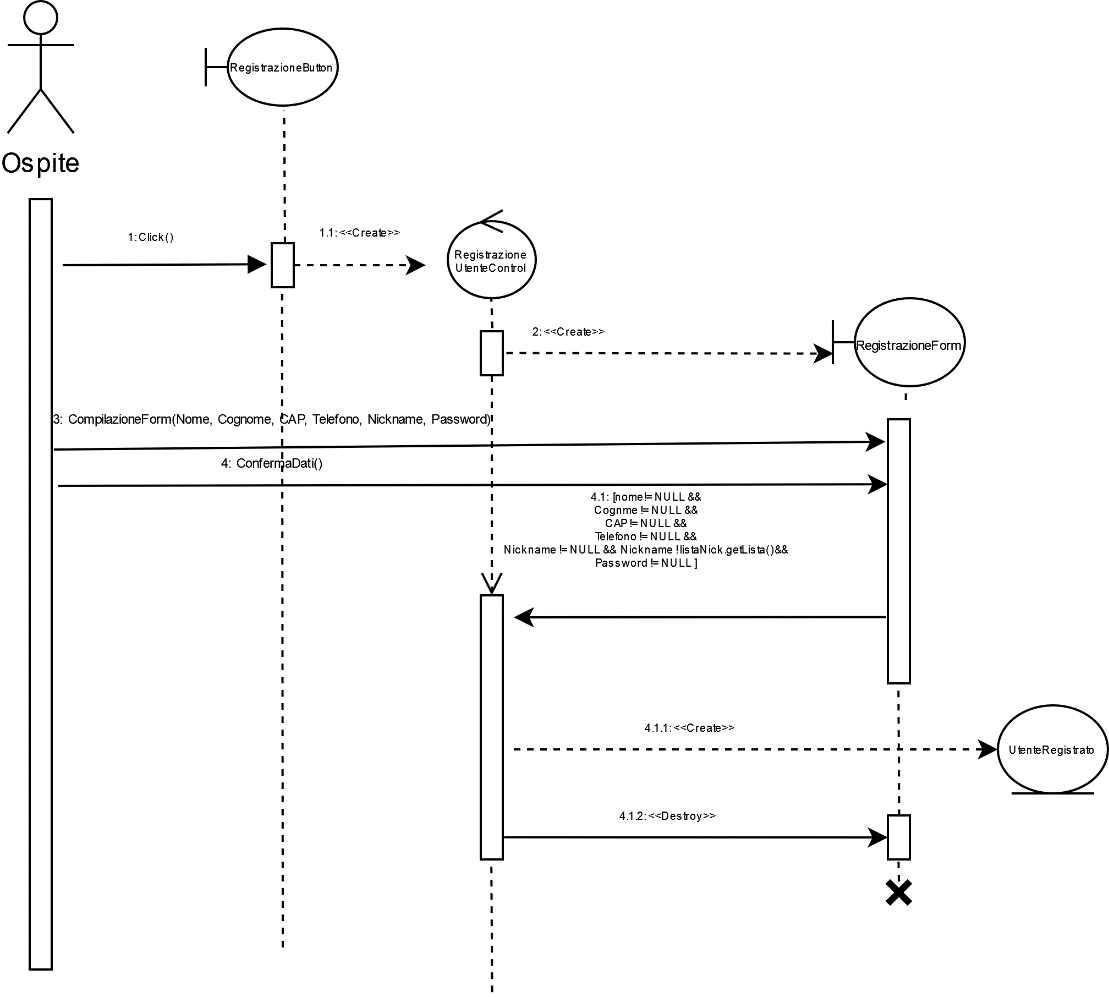
Nella presente sezione viene illustrato un activity diagram relativo al RF\_GE\_3: Adesione a Evento



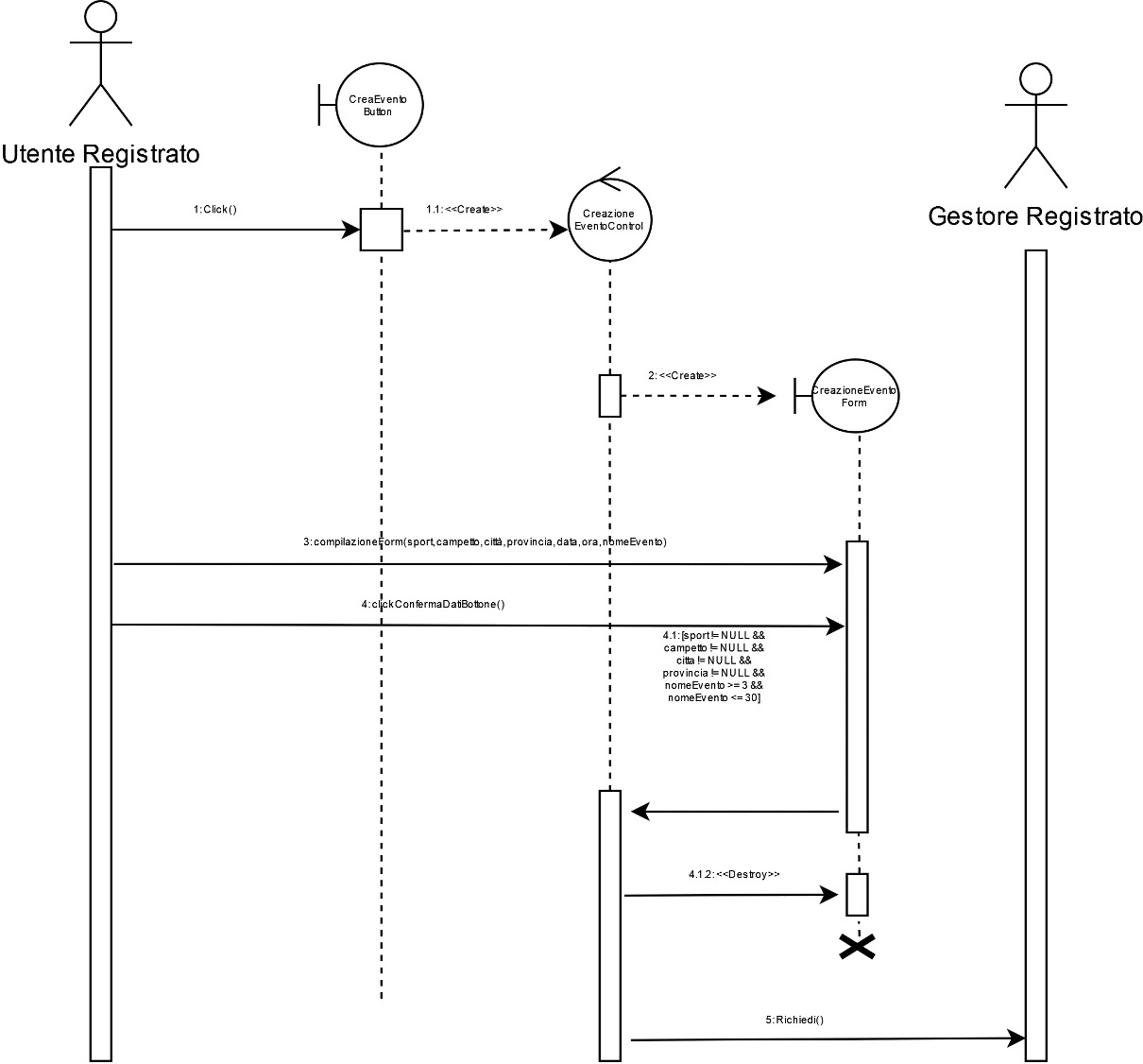
## Sequence Diagram

Di seguito vengono riportati alcuni sequence diagram relativi ai requisiti funzionali del sistema.

SD\_GU\_1



SD\_GE\_1



## 3.4.4.3 Statechart Diagram

Di seguito è riportato lo statechart diagram relativo al RF\_GU\_1:

Registrazione Utente

## 

## Interfacce Utente: Mock-ups

## 

Immagine che contiene testo

Descrizione generata automaticamente







