

Easy Fut5al



Requirements Analysis Document

Sommario

| | |
|--|----|
| 1. Introduzione | 2 |
| 1.1 Obiettivi del Sistema | 2 |
| 1.2 Scopo del Sistema | 2 |
| 1.3 Obiettivi e criteri di successo del progetto | 2 |
| 1.4 Definizioni, acronimi e abbreviazioni | 3 |
| 1.5 Riferimenti | 3 |
| 1.6 Panoramica | 4 |
| 2. Sistema Corrente | 4 |
| 3. Sistema Proposto | 4 |
| 3.1 Requisiti funzionali | 4 |
| 3.2 Requisiti non funzionali | 6 |
| 3.2.1 Usabilità | 6 |
| 3.2.2 Affidabilità | 6 |
| 3.2.3 Prestazioni | 6 |
| 3.2.4 Sostenibilità | 6 |
| 3.2.5 Implementazione | 6 |
| 3.2.6 Interfacce | 6 |
| 3.2.7 Packaging | 7 |
| 3.2.8 Legal | 7 |
| 3.3 Modello di Sistema | 7 |
| 3.3.1 Modello Casi d'Uso | 7 |
| 3.3.2 Object Model | 16 |
| 3.3.3 Dynamic Model | 21 |

1. Introduzione

1.1 Obiettivi del Sistema

L'attività si interessa per lo sviluppo di un sistema che possa fornire una piattaforma che faciliti l'organizzazione di partite di Calcio a 5 sia per giocatori che per i gestori dei campetti. Gli utenti potranno svolgere tale pratica in modo semplice, pratico e immediato. Per tal motivo si è decisi a realizzare un App Android scaricabile che sarà punto cardine del sistema assieme al server che smisterà le richieste HTTP per accedere al DB che terrà traccia di partite, campetti, prenotazioni, etc.

1.2 Scopo del Sistema

Il cliente dovrà installare il sistema presso la propria sede e rilascerà l'App Android prodotta su Play Store. Essa sarà scaricabile solo da utenti in Italia poiché il sistema prevede la registrazione esclusiva di campi di calcio a 5 sul territorio nazionale.

1.3 Obiettivi e criteri di successo del progetto

La buona riuscita del progetto sarà conciliata con il raggiungimento dei seguenti obiettivi:

- L'automatizzazione della procedura di prenotazione di campetti da parte del cliente e la gestione di esse da parte dell'amministratore
- Facilitare la formazione di gruppi di 10 persone pronte per giocare partite.
- Ampliare la conoscenza del cliente sui vari campi disponibili sul territorio con relativi prezzi, condizioni e disponibilità.
- Delineare un profilo personale per tutti gli utenti.
- Delineare un profilo di ogni campo, informando la comunità su affidabilità, manutenzione e gestione dello stesso.

1.4 Definizioni, acronimi e abbreviazioni

EF5: acronimo che fa riferimento al sistema “Easy Fut5al”

Gestore: responsabile dei campetti da calcio a 5, piazza tariffe e disponibilità del campo tramite l’ app

Atleta: utente del sistema che ha intenzione di giocare.

Prenotazione automatizzata: Prenotazione di una partita di calcetto che avviene via App.

Atleta Manager: colui che crea la partita (va di default ad essere Giocatore “sicuro”).

Giocatore “sicuro”: l’ atleta che accetta di partecipare ad una partita.

Giocatore “riservato”: l’ atleta che è stato invitato ad una partita e deve accettare o declinare l’ invito.

Amministratore: il proprietario del Sistema.

1.5 Riferimenti

Bernd Bruegge & Allen H. Dutoit, Object-Oriented Software Engineering: Using UML, Patterns and Java, (3rd edition), Prentice-Hall.

1.6 Panoramica

In questo documento al secondo punto verrà presentato il Sistema Corrente mentre nel terzo punto sarà esposto il Sistema Proposto con i suoi requisiti (funzionali e non funzionali), la tecnologia che sarà usata per implementarlo, il System model con i casi d'uso, l'object model e il design model.

2. Sistema Corrente

Il sistema proposto non poggia su basi preesistenti, perciò la Requirements Elicitation nasce solo dal meeting con cliente. (Fase “Greenfield Engineering”).

3. Sistema Proposto

3.1 Requisiti funzionali

Registrazione

Gli utenti finali che useranno il Sistema devono avere la possibilità di registrarsi scegliendo tra due modalità differenti: Atleta o Gestore. In entrambi i casi essi compileranno un form adattato alla modalità. Per il Gestore deve esserci la possibilità di registrare anche il suo Campetto. In questo caso la registrazione andrà a buon fine soltanto dopo che l'amministratore del Sistema abbia verificato l'esistenza concreta e l'agibilità del Campetto.

Crea Nuova Partita

“Easy Fut5al” permette di creare nuove partite per un certo campo in un dato orario (a meno che non sia già prenotato) e posizionarle nella coda delle partite aperte. Appena si raggiunge un totale di

10 giocatori “sicuri”, la prenotazione viene ufficialmente convalidata e registrata correttamente sul Database. Nel caso in cui ad un’ ora prima della partita, i 10 giocatori non sono “sicuri”, la partita viene cancellata.

Invita giocatori e partecipazione a partite

La partita deve apparire a tutti gli utenti che potranno successivamente unirsi selezionandola dalla propria home. Inoltre colui che crea la partita ha la possibilità di invitare altri utenti, che riceveranno una notifica e la visualizzeranno nella propria casella postale. Fino a quando essi non accettano o rifiutano, la partita conserverà un posto almeno fino ad un’ ora prima dell’ inizio della partita (Giocatore “riservato”). Nel momento in cui qualcuno accetta, passa nello stato di Giocatore “sicuro”. Nel caso in cui qualcuno rifiuta, nella partita viene liberato un posto.

Visualizza Profilo

Ogni utente ha un profilo personale. Se trattasi di un “Gestore” in più ci saranno anche i dati relativi al suo campo.

Valutazione Campo

A fine partita si può dare una valutazione personale al Campo (da 1 a 5).

Visualizza inviti

Sia il gestore che l’ atleta hanno un servizio di posta. L’ atleta deve visualizzare gli inviti ricevuti mentre il gestore deve visualizzare le “prenotazioni complete” che ha ricevuto.

Calendario (Ricerca per Nome Campo)

L’ atleta ha la possibilità di visualizzare un calendario relativo ad un campo da lui scelto e ricercare per ogni giornata, quali sono le partite organizzate in suddetto campo.

Ricerca per Città

L’ atleta ha la possibilità di ricercare per città quali partite si terranno e in quali campi.

3.2 Requisiti non funzionali

3.2.1 Usabilità

EF5 è intuitivo e facile da usare. Il menu in basso permette la navigazione immediata verso “Calendario”, “Cerca”, “Profilo”, “Home”. Il tasto centrale “+” porta direttamente alla creazione di una nuova partita, la quale sarà definita grazie alla form.

3.2.2 Affidabilità

Il sistema deve essere accessibile in qualunque momento onde evitare rallentamenti nell’organizzazione delle partite. Easy Fut5al deve supportare un bacino minimo di 100 profili online.

3.2.3 Prestazioni

E’ importante che il sistema riesca a convalidare più prenotazioni possibili in un tempo accettabile per l’utente. Garantire l’aggiornamento delle informazioni in tempo reale è essenziale per aggiudicarsi il Campetto.

3.2.4 Sostenibilità

Il Sistema è dedicato a utenti in suolo italiano, poiché si vogliono registrare solo Campetti italiani. Non si esclude un’espansione del sistema al di fuori dei confini nazionali. Al momento il sistema necessita di un unico server.

3.2.5 Implementazione

Il sistema dovrà essere implementato con tecnologia Java Servlet per il lato Server. L’app per l’utente sarà sviluppata con Android Studio.

3.2.6 Interfacce

Il Sistema deve offrire le funzionalità attraverso un’interfaccia familiare anche per utenti non esperti della tecnologia con presentazione di form standard per poter sottomettere facilmente i dati.

3.2.7 Packaging

E' prevista una sola installazione di un server da parte del cliente. Gli utenti si preoccuperanno di installare l' App da Play Store.

3.2.8 Legal

I dati forniti dagli utenti non dovranno essere divulgati a terze parti secondo gli Artt. 1175 e 1375 e seguenti, artt. 2105, 2125 codice civile; artt. 98 e 99 d. lgs. n. 30/2005.

3.3 Modello di Sistema

3.3.1 Modello Casi d'Uso

| Nome | <u>RegistraAtleta</u> | |
|---|--|---|
| Attori partecipanti | <i>Inizializzato da Cliente</i> | |
| Flusso di eventi | <i>1. Il cliente richiede di potersi registrare al sistema.</i> | |
| | | 2. Il sistema visualizza una form che richiede l'inserimento obbligatorio di: nome, cognome, username, sesso, data di nascita, indirizzo e-mail, città, password e conferma password. |
| | <i>3. Il Cliente riempie i campi indicati e sottomette la form compilata</i> | |
| | | 4. Il sistema verifica che: -non ci siano campi obbligatori mancanti. -la password coincida con conferma password -l'email inserita non sia stata già registrata. Il sistema salva i dati dell'utente, visualizza la schermata principale e avvisa che l'operazione è terminata con successo. |
| Flusso alternativo: "password non corrisponde" | | 4.b1. Il sistema avvisa all'utente che la password inserita non corrisponde con il campo "Conferma Password". |
| Flusso alternativo: "e-mail già registrata" | | 4.c1. Il sistema avvisa all'utente che l'email inserita è già |

| | | |
|---|---|--|
| | | registrata nel sistema. |
| Flusso di eventi di errore: “Il sistema fallisce nel salvataggio dei dati” | | 4.d1. Il sistema visualizza una segnalazione di errore all’utente informandolo del mancato salvataggio dei dati. Si invita di riprovare più tardi. Il sistema riporta l’applicazione alla schermata iniziale di registrazione. |
| Condizioni di entrata | <i>L’utente preme sulla voce “Registrati”.</i> | |
| Condizioni di uscita | <i>Il sistema ha registrato l’utente come Atleta.</i> | |
| Requisiti speciali | | |

| Nome | <u>RegistraGestore</u> | |
|--|---|--|
| Attori partecipanti | <i>Inizializzato da Utente</i> | |
| Flusso di eventi | 1. <i>Il cliente richiede di potersi registrare al sistema.</i> | |
| | | 2. Il sistema visualizza una form che richiede l’inserimento obbligatorio di: nome, cognome, username, indirizzo e-mail, password, conferma password, nome campetto, città e tariffa oraria. |
| | 3. <i>Il Cliente riempie i campi indicati e sottomette la form compilata.</i> | |
| | | 4. Il sistema verifica che: -non ci siano campi obbligatori mancanti. -la password coincida con conferma password -l’indirizzo e-mail non sia già registrato. Il sistema salva i dati dell’utente. Successivamente avvisa l’utente specificando che la registrazione non terminerà fin a quando non sarà verificata l’agibilità del campetto indicato. |
| Flusso alternativo: “Campo obbligatorio mancante” | | 4.a1. Il sistema avvisa l’utente del mancato inserimento dati. Attende una nuova sottomissione corretta della form. |

| | | |
|---|--|--|
| Flusso alternativo: “password non corrisponde” | | 4.b1. Il sistema avvisa all’utente che la password inserita non corrisponde con il campo “Conferma Password”. |
| Flusso alternativo: “e-mail già registrata” | | 4.c1. Il sistema avvisa all’utente che l’email inserita è già registrata nel sistema. |
| Flusso di eventi di errore: “Il sistema fallisce nel salvataggio dei dati” | | 4.d1. Il sistema visualizza una segnalazione di errore all’utente informandolo del mancato salvataggio dei dati. Si invita di riprovare più tardi. Il sistema riporta l’applicazione alla sua schermata iniziale di registrazione. |
| Condizioni di entrata | <i>L’utente preme sulla voce “Registrati come gestore”.</i> | |
| Condizioni di uscita | <i>Il sistema ha registrato un nuovo campetto da verificare dall’amministratore.</i> | |
| Requisiti speciali | | |

| Nome | VerificaCampetto | |
|---|---|--|
| Attori partecipanti | <i>Inizializzato da Amministratore</i> | |
| Flusso di eventi | 1. <i>L’amministratore del sistema, dopo aver verificato l’esistenza del campetto, accetta la registrazione dal suo pannello di controllo.</i> | |
| | | 2. Il sistema visualizza un messaggio dichiarando che l’operazione è terminata con successo. |
| Flusso alternativo: “Registrazione rifiutata per mancata agibilità del campetto” | 1.a. <i>L’amministratore rifiuta la registrazione del gestore.</i> 1.b. <i>E’ compito dell’amministratore comunicare al gestore la mancata registrazione</i> | |
| Flusso di eventi di errore: “Il sistema fallisce nel salvataggio dei dati” | | 2.a1. Il sistema visualizza una segnalazione di errore all’amministratore informandolo del mancato salvataggio dei dati. Si invita di riprovare più tardi. |
| Condizioni di entrata | <i>Il nuovo gestore deve aver registrato il suo campetto sul database di sistema.</i> | |
| Condizioni di uscita | <i>Il campetto viene dichiarato agibile.</i> | |
| Requisiti speciali | | |

| Nome | EffettuaLogin | |
|---|--|---|
| Attori partecipanti | <i>Inizializzato da Guest</i> | |
| Flusso di eventi | 1. <i>L'utente preme su "Login"</i> | |
| | | 2. Il sistema richiede l'inserimento di e-mail e password. |
| | 3. <i>L'utente compila la form.</i> | |
| | | 4. Il sistema verifica che: -non ci siano campi obbligatori mancanti. -l'indirizzo e-mail specificato sia già registrato dal sistema. Si accerta del login come "Atleta" e visualizza la schermata principale. |
| Flusso alternativo: "Campo obbligatorio mancante" | | 4.a1. Il sistema avvisa l'utente del mancato inserimento dati. Attende una nuova sottomissione corretta della form. |
| Flusso alternativo: "e-mail o password errata" | | 4.b1. Il sistema avvisa all'utente che l'e-mail o la password sono errate. Attende una correzione dei campi obbligatori ed una nuova sottomissione. |
| Flusso alternativo: "Il sistema verifica il login come "Gestore" | | 4.c2 Il sistema verifica il login come "Gestore" e visualizza la schermata principale dedicata al Gestore. |
| Condizioni di entrata | <i>E' visualizzata la schermata iniziale contenente la voce "Login".</i> | |
| Condizioni di uscita | <i>Il guest si è loggato correttamente.</i> | |
| Requisiti speciali | | |

| Nome | CreaPartita | |
|---------------------|---|--|
| Attori partecipanti | Inizializzato da Atleta-Manager | |
| Flusso di eventi | 1. L'Atleta-Manager ha intenzione di creare una nuova partita. | |
| | | 2. Il sistema visualizza la form contenente ora, nome campetto e data, nella sezione "Nuova Partita". |
| | 3. L'Atleta-Manager compila la form. | |
| | 4. L'Atleta-Manager ha la possibilità di inviare inviti ai suoi amici iscritti all'app ed inserisce X username. | |
| | | 5. Il sistema riconosce ogni username. |
| | 6. L'atleta-Manager sottomette la form. | |
| | | 7. Il sistema salva i dati ottenuti dal form; crea una nuova partita "in sospeso", visibile sulla home principale dell'App. Il sistema aggiunge l'Atleta-Manager come primo partecipante alla partita, dichiarandolo con la dicitura "sicuro". Il sistema invia una notifica sulla casella postale di ogni invitato. Tutti gli invitati sono effettivamente partecipanti con lo stato "riservato" della partita "in sospeso". [Partecipanti=(1+ X) su 10)]. |

| | | |
|------------------------------|--|--|
| | <p>8. <i>Una parte Y degli X atleti (che al momento sono in stato “riservato”), rifiutano l’invito. I partecipanti “riservati” invitati ora sono $X = X - Y$.</i></p> <p><i>Una parte Z degli invitati accetta l’invito. Quindi i Z partecipanti passano dallo stato “riservato” allo stato “sicuro”.</i></p> | |
| | | <p>9. Il sistema resta in attesa di ulteriori aggiornamenti riguardanti di X finché la partita non passa da “in sospeso” a “da giocare”.</p> |
| Condizioni di entrata | <i>L’atleta-manager deve essere loggato e pigia il tasto “+” centrale sul menù in basso.</i> | |
| Condizioni di uscita | <i>Deve essere creata una nuova partita in sospeso con X invitati.</i> | |
| Requisiti speciali | <i>Il sistema fa scegliere a priori solamente date e orari che sono potenzialmente disponibili, ossia dove non vi sono partite la cui prenotazione è “Confermata”.</i> | |

| Nome | Unisciti a partita | |
|--|--|--|
| Attori partecipanti | <i>Inizializzato da Atleta</i> | |
| Flusso di eventi | 1. <i>L'atleta si trova sulla home. riceve una notifica sulla sua posta: la apre.</i> | |
| | | 2. <i>Il sistema visualizza un invito per una partita ricevuto da un suo amico.</i> |
| | 3. <i>L'atleta decide di accettare l'invito.</i> | |
| | | 4. <i>Il sistema visualizza la partita "in sospeso" (nella sezione "Le mie partite") al quale l'atleta è stato invitato.</i> |
| | 5. <i>L'atleta invitato(partecipante) passa dallo stato "riservato" allo stato "sicuro" per la partita in sospeso.</i> | |
| Flusso alternativo: "L'atleta si unisce ad una partita in sospeso libera" | 1a-2a-3a-4a. <i>L'atleta visualizza dalla home le partite pubbliche "in sospeso" dove può unirsi.</i> | |
| Condizioni di entrata | <i>L'utente deve essere loggato ed è necessario che ci sia una nuova partita "in sospeso"</i> | |
| Condizioni di uscita | <i>L'atleta deve essersi unito ad una partita.</i> | |
| Requisiti speciali | | |

| Nome | Valuta Campetto | |
|------------------------------|---|---|
| Attori partecipanti | <i>Inizializzato da Atleta</i> | |
| Flusso di eventi | 1. <i>Il cliente effettua l'accesso al sistema e va sulla sezione "Le mie partite".</i> | |
| | | 2. <i>Il sistema visualizza un range di stelle di valutazione [da 1 a 5] per ogni campetto dove si è giocato la partita.</i> |
| | 3. <i>Il cliente fa le sue valutazioni e sottomette la form.</i> | |
| | | 4. <i>Il sistema salva i dati appena caricati dall'utente poi aggiorna le medie delle valutazioni di ogni singolo campetto.</i> |
| Condizioni di entrata | <i>La partita è terminata.</i> | |
| Condizioni di uscita | <i>Il sistema ha aggiornato le valutazioni di ogni campetto.</i> | |
| Requisiti speciali | | |

| Nome | Visualizza Profilo | |
|------------------------------|---|--|
| Attori partecipanti | <i>Inizializzato da Atleta, Gestore</i> | |
| Flusso di eventi | 1. <i>L'utente clicca sul bottone "Profilo".</i> | |
| | | 2. <i>Il sistema visualizza il profilo dell'utente con tutti i dati personali che ha specificato alla registrazione.</i> |
| Condizioni di entrata | <i>L'utente deve essere loggato e deve aver cliccato sul bottone "Profilo".</i> | |
| Condizioni di uscita | <i>L'utente si disconnette dal sistema, dopo essersi informato sulle sue credenziali.</i> | |
| Requisiti speciali | | |

| Nome | Logout | |
|------------------------------|---|--|
| Attori partecipanti | <i>Inizializzato da Atleta, Gestore</i> | |
| Flusso di eventi | 1. <i>L'utente preme sul pulsante in alto a destra.</i> | |
| | | 2. Il sistema risponde aprendo una tendina con l'opzione "Disconnetti" |
| | 3. <i>L'utente sceglie "Disconnetti".</i> | |
| | | 4. Il sistema effettua il logout e visualizza la schermata iniziale.(di Login/Registrazione) |
| Condizioni di entrata | <i>L'utente deve essersi loggato al sistema.</i> | |
| Condizioni di uscita | <i>L'utente si è disconnesso dal sistema correttamente.</i> | |
| Requisiti speciali | | |

3.3.2 Object Model

- **RegistraAtleta/RegistraGestore**

Entity:

Cliente: L'utente anonimo che si registra al sistema.

Atleta: lo si diventa dopo essere stati registrati come "Atleta".

Gestore: lo si diventa dopo essere stati registrati come "Gestore".

Boundary:

FormRegistrazioneAtleta: Form usata da un atleta per inserire i dati personali.

FormRegistrazioneGestore: Form usata da un Gestore per inserire i dati personali, includendo la registrazione del campo.

Control:

RegistrazioneControl: Verifica la form, salva i dati dell'utente.

- **Verifica Campo**

Entity:

Amministratore: Colui che verifica l'agibilità del campo.

Boundary:

ConfermaCampoButton: Serve all'amministratore per decidere se convalidare il campo.

Control:

ConvalidaCampoControl: Si occupa della procedura della convalida del campo.

- EffettuaLogin

Entity:

Utente: L'utente anonimo che si registra al sistema.

Boundary:

LoginButton: Inizia il caso d'uso Login.

LoginForm: Form usata per e-mail e password.

Control:

LoginControl: verifica la correttezza del login.

- CreaPartita

Entity:

Atleta-Manager: Colui che crea la partita.

Partita: Rappresenta la partita da svolgersi, può assumere gli stati "in sospeso", "da giocare" o "Annullata".

Partecipante: Membro che giocherà la partita, può essere "sicuro" o "riservato".

Boundary:

TastoPlus: Bottone usato dall'atleta per iniziare il caso d'uso CreaPartita.

FormDetails: Contiene ora, data, campo e eventuali inviti alla partita.

ListMembers: Lista dei players a giocare la partita tramite invito.

Control:

CreaPartitaControl: Avvia il caso d'uso CreaParitta

- **Unisciti a partita**

Entity:

Atleta: lo si diventa dopo essere stati registrati come “Atleta” .

Partita: rappresenta la partita da svolgersi, può assumere gli stati “in sospeso” , “da giocare” o “annullata” .

Boundary:

UniscitiButton: Tasto che permette ad un atleta di diventare un partecipante di una partita.

PostaButton: Tasto che permette di accedere alla posta

NotificaInvito: Avvisa l' atleta dell' arrivo di un invito a partecipare ad una partita.

Control:

UniscitiPartitaControl: Gestisce il raggruppamento dei membri per una partita.

PostaControl: Gestisce la posta.

- **Valuta Campetto**

Entity:

Atleta: Colui che valuta il campetto.

Partecipanti: Coloro che hanno giocato la partita.

Campetto: Può avere una valutazione da parte dei suoi partecipanti.

Boundary:

ValutaButton: Serve per far assegnare successivamente una valutazione.

Control:

ValutaPartitaControl: Si preoccupa di gestire le varie valutazioni e registrarle sul sistema, aggiornando tutti i relativi dati.

- **Visualizza Profilo**

Entity:

Atleta, Gestore: visualizza il proprio profilo.

Boundary:

ProfileButton: Permette di accedere alla schermata dei propri dati personali.

Control:

EditControl: Si preoccupa di registrare eventuali modifiche del profilo e rifiutarle qualora non siano valide.

- **Logout**

Entity:

Atleta/Gestore: Coloro che effettuano il logout.

Boundary:

ExitButton: Permette di avviare il logout.

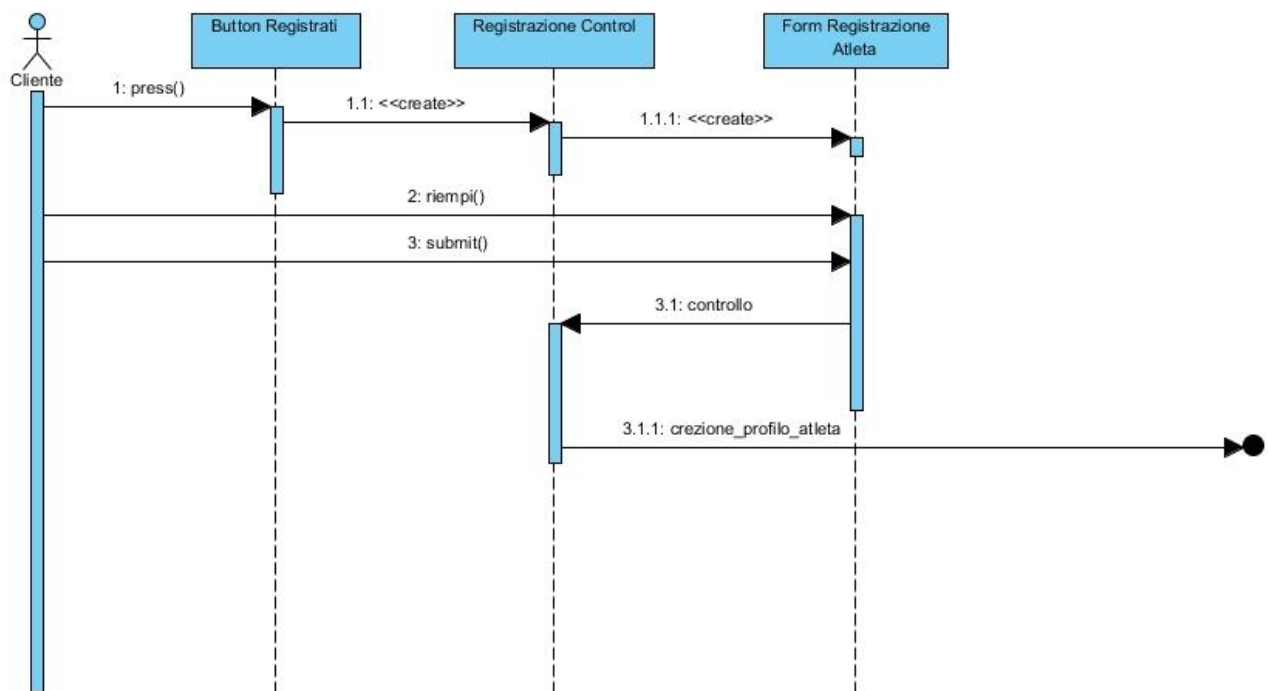
Control:

LogoutControl: Gestisce la disconnessione dal sistema.

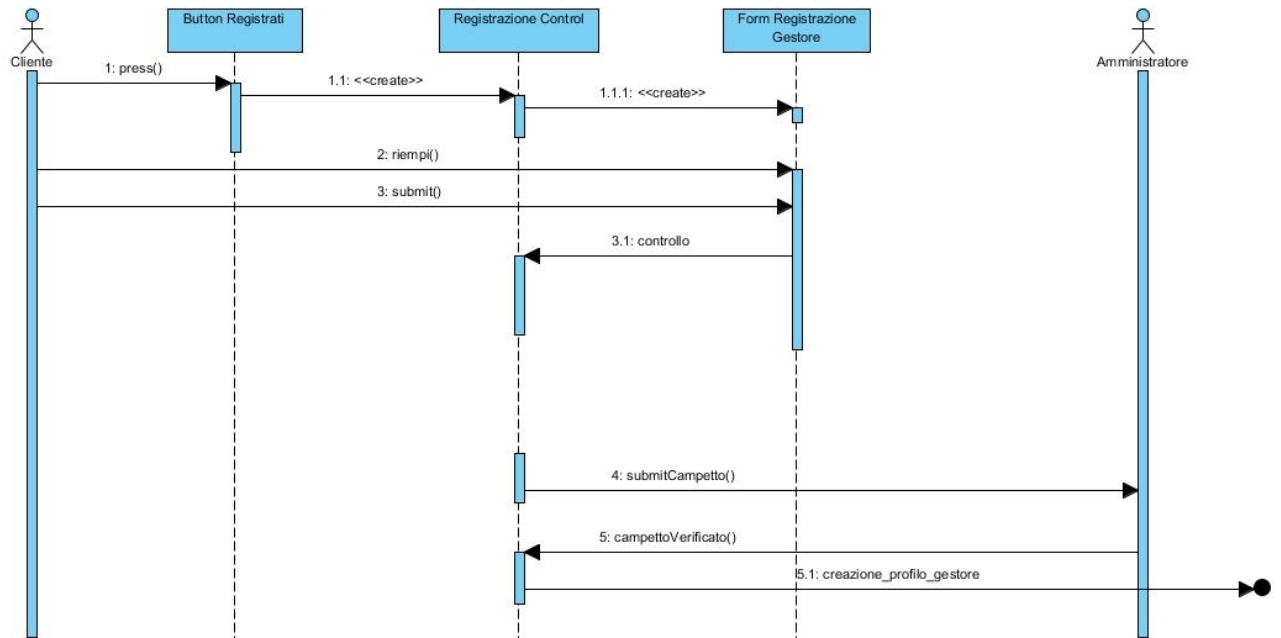
3.3.3 Dynamic Model

- **Sequence Diagram**

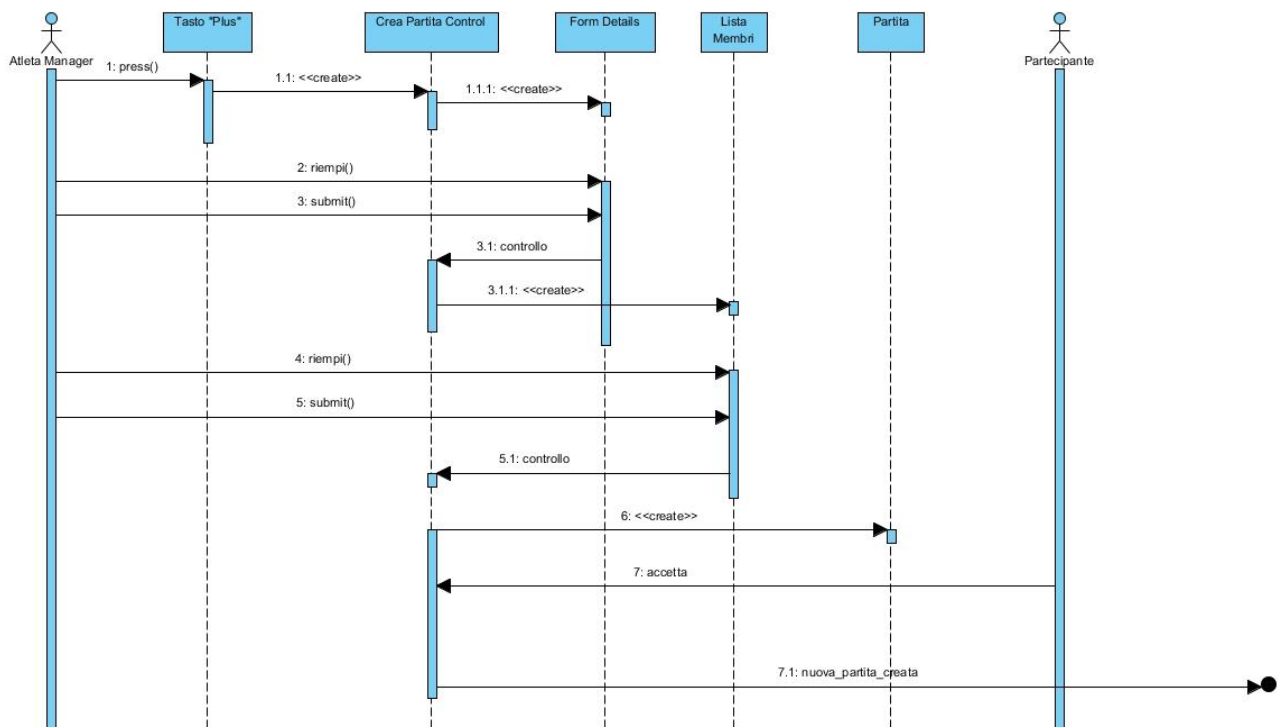
- **Registrazione Atleta**



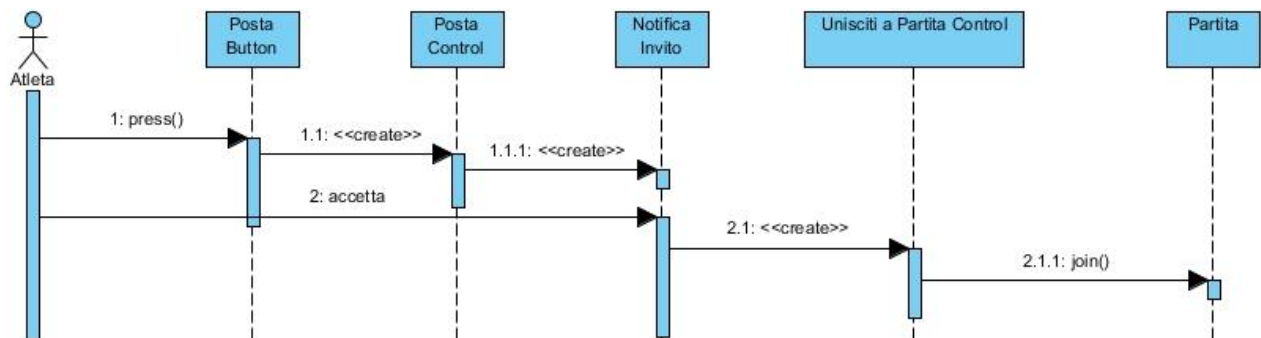
- Registrazione Gestore



- Crea Partita



- Unisciti a partita



- Verifica Campetto

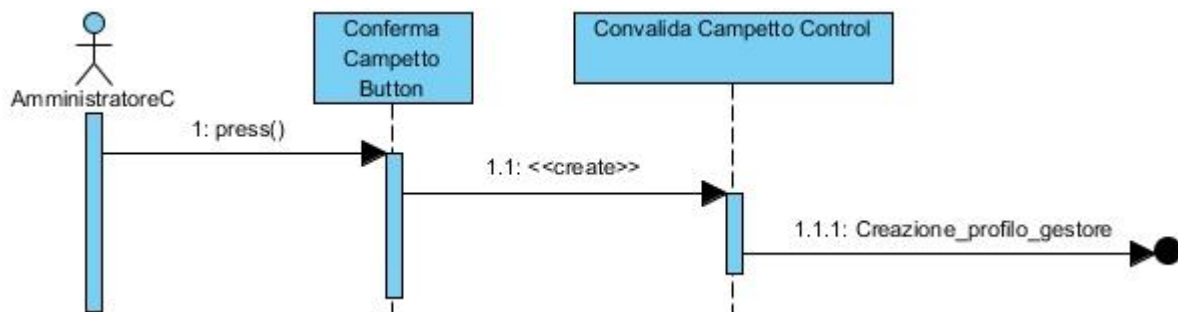
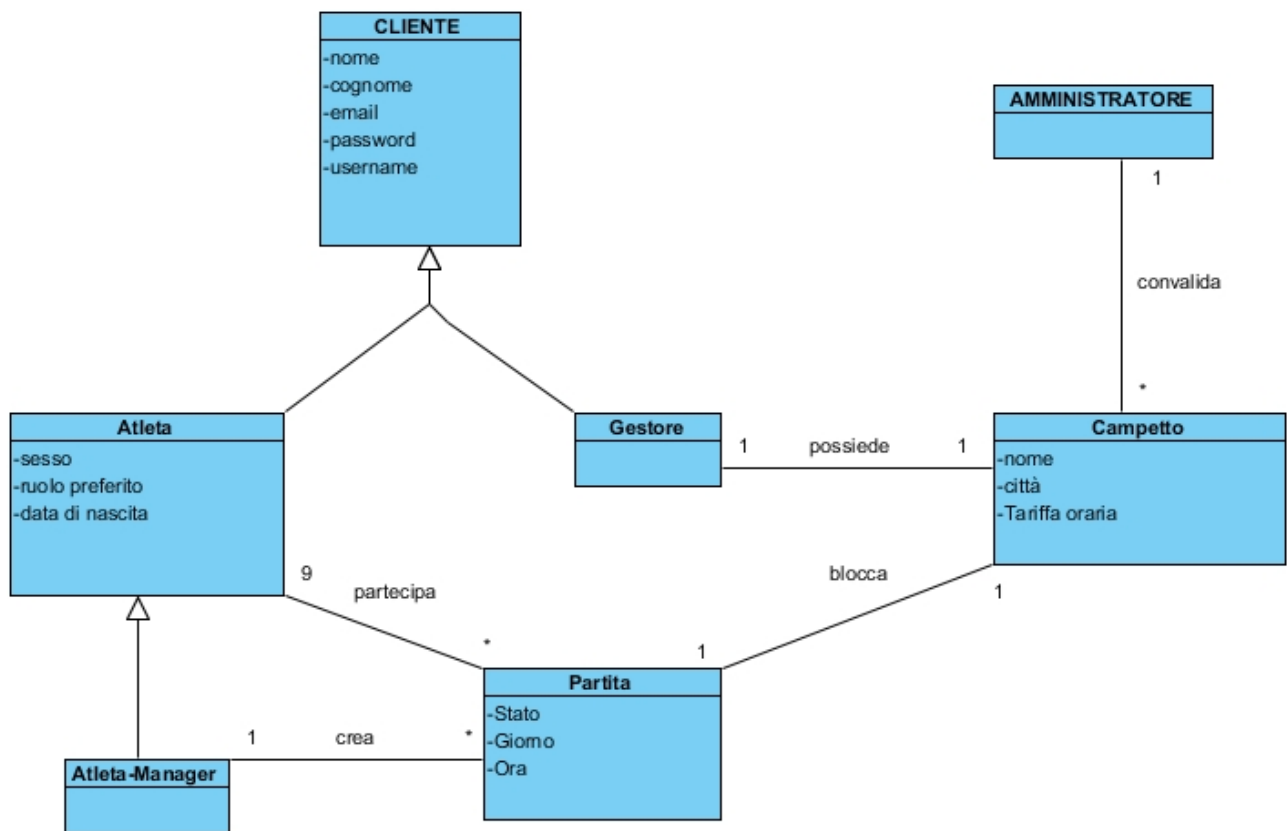
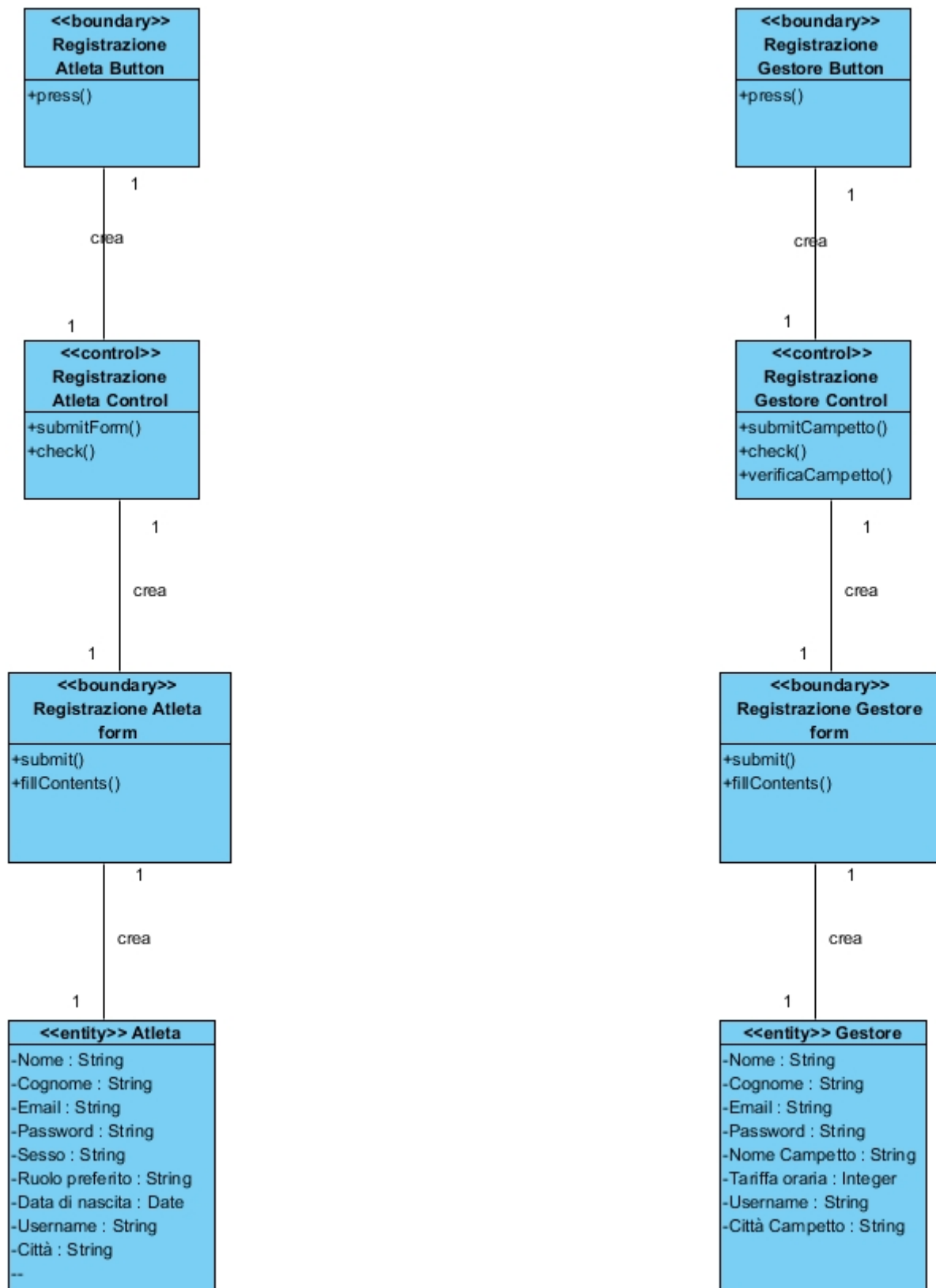
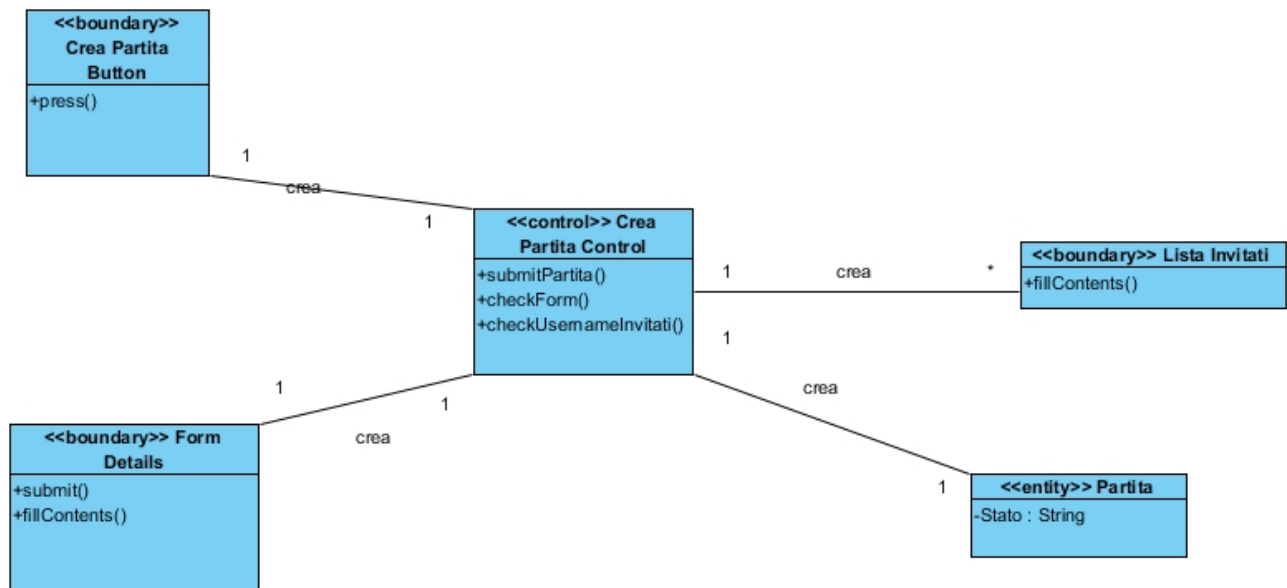


DIAGRAMMA GENERALE



CLASS DIAGRAMS





STATE CHART DIAGRAMS

