

Doodle: Pianificazione eventi

Giovanni Ricucci

11 dicembre 2014

Indice

1	Requisiti Funzionali	1
2	Scenari	3
3	Diagramma dei casi d'uso	5
3.1	Registrazione	5
3.2	Login	6
3.3	Crea Evento	7
3.4	Partecipazione ad evento – Utente senza accesso	8
3.5	Partecipazione ad evento – Utente con accesso	9
3.6	Chiudere evento	10
3.7	Eliminazione evento	11
3.8	Diagramma casi d'uso completo	12
4	Diagramma delle attività	13
4.1	Registrazione	14
4.2	Login	15
4.3	Crea Evento	16
4.4	Partecipazione ad evento – Utente senza accesso	17
4.5	Partecipazione ad evento – Utente con accesso	18
4.6	Chiudere evento	19
4.7	Eliminare evento	20
5	Diagramma delle classi	21
5.1	Cardinalità	21
5.2	Diagramma totale delle classi Doodle	22
5.3	Diagramma totale delle classi Doodle 2.0	23
6	Diagramma di sequenza	25
6.1	Diagramma di sequenza	25

7	Diario	27
7.1	Diario Doodle 1.0	27
7.2	Diario Doodle 2.0	29
8	Ciclo di sviluppo	31
8.1	Ciclo di sviluppo	31

Capitolo 1

Requisiti Funzionali

L'applicazione deve permettere di creare eventi e di gestire le disponibilità espresse dagli utenti relativamente agli eventi stessi. Nello specifico, l'applicazione dovrà permettere agli utenti di:

1. Creare e gestire un evento (funzionalità riservate agli utenti registrati);
2. Esprimere le proprie disponibilità per un dato evento (funzionalità disponibile sia per utenti registrati sia per utenti non registrati).

Capitolo 2

Scenari

1. L'applicazione deve poter registrare nuovi utenti ,oltre a quelli registrati, e fare delle ricerche su utenti già registrati:
 - Il sistema richiede le credenziali d accesso all'utente se questo è registrato;
 - Se l'utente è registrato può procedere con l'accesso;
 - Se l'utente vuole registrarsi all'applicazione fornisce i dati personali;
 - I dati personali richiesti sono: nome, nickname, password ed email (opzionale);
 - Una volta registrato l'utente non potrà più modificare i propri dati;
 - Non è previsto nessun meccanismo di recupero della password;
2. L'applicazione permetterà un meccanismo di autenticazione degli utenti:
 - L'utente fornisce le proprie credenziali d'accesso (nickname e password) al sistema;
 - Il sistema concede l'accesso, se l'utente è stato riconosciuto;
 - Dopo aver effettuato l'accesso, l'utente/amministratore vede elencati tutti gli eventi da lui creati;
 - L'utente/amministratore può in ogni momento chiudere l'evento da lui creato eventualmente fornendo i motivi;
 - L'utente/amministratore può in ogni momento cancellare il sondaggio da lui creato rendendolo inaccessibile.

3. L'utente registrato all'applicazione deve poter creare e gestire eventi :
 - L'utente fornisce i dati necessari e opzionali alla creazione dell'evento;
 - I dati necessari sono: nome dell'evento e opzioni di scelta (giorni e orari);
 - I dati opzionali sono: luogo dell'evento e descrizione dell'evento;
 - Durante la creazione di un evento, ogni opzione di scelta deve essere cancellabile o modificabile;
 - Una volta che l'evento è stato creato, non è più possibile modificarlo;
 - Al termine della fase di creazione il sistema comunica un identificatore univoco dell'evento creato. Tale identificatore verrà usato in seguito per gestire l'evento;
 - Dopo averlo creato, il sistema iscrive automaticamente l'utente all'evento;
 - Al termine della creazione l'evento diventa automaticamente disponibile per qualunque utente volesse esprimere la sua disponibilità;
4. L'applicazione deve permettere agli utenti di poter accedere, visualizzare le informazioni, poter iscriversi ed esprimere le proprie disponibilità per ogni evento:
 - L'utente che accede a un evento può visualizzare le disponibilità di tutti gli altri utenti e i commenti già inseriti;
 - L'utente che accede a un evento può inserire il proprio nome e la disponibilità (si/no);
 - Nel caso l'evento è stato chiuso può visualizzare i motivi che sono stati eventualmente scelti dall'utente/amministratore di quel evento;
 - Se l'utente accede all'evento dopo aver effettuato l'accesso, il suo nome e il suo nickname saranno inseriti nell'apposito modulo in automatico;
 - Se l'utente accede all'evento dopo aver effettuato l'accesso può lasciare un commento e modificare le proprie disponibilità;

Capitolo 3

Diagramma dei casi d'uso

3.1 Registrazione

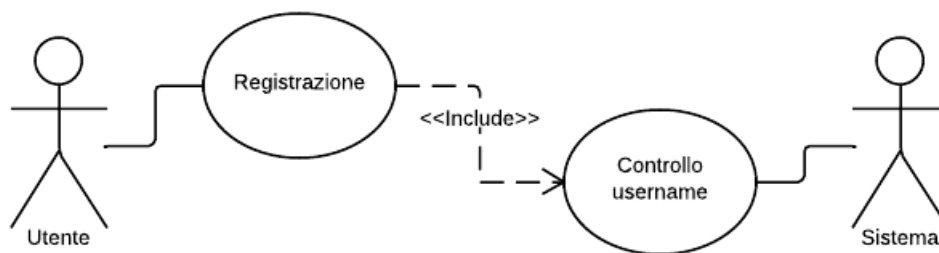


Figura 3.1: Registrazione

NOME	Registrazione;
OBBIETTIVO:	Registrare l'utente nella applicazione;
PRE-CONDIZIONE	L'utente non deve essere registrato nel sistema;
SUCCESSO:	La registrazione avviene con successo;
FALLIMENTO:	Non è possibile effettuare la registrazione;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.1: Registrazione Utente

FLUSSO PRINCIPALE:

1. L'utente compila e inoltra il modulo d'iscrizione;
 - (a) Controllo dei campi non opzionali e necessari alla registrazione;

- (b) Doppio inserimento password;
- 2. Controllo del username;
- 3. Ricerca del username;
 - (a) Il sistema verifica che l'utente (username univoco) non sia già registrato;
- 4. Avviene la registrazione del utente;
- 5. La registrazione è avvenuta con successo;

3.2 Login

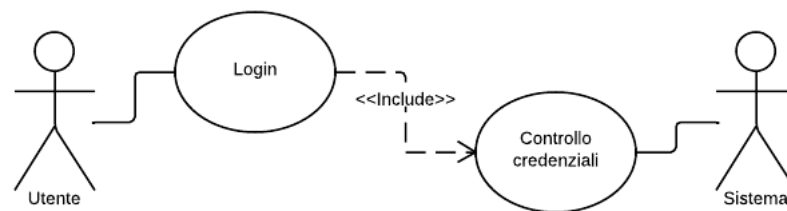


Figura 3.2: Login

NOME	Login;
OBBIETTIVO:	Ottenere accesso all'applicazione;
PRE-CONDIZIONE	L'utente deve essere registrato all'applicazione;
SUCCESSO:	L'utente ottiene accesso all'applicazione;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.2: Login Utente

FLUSSO PRINCIPALE:

- 1. L'utente immette le proprie credenziali;
- 2. L'utente inoltra la richiesta di accesso;
 - (a) Il sistema verifica la corrispondenza delle credenziali;
- 3. L'utente ottiene accesso all'applicazione;

3.3 Crea Evento

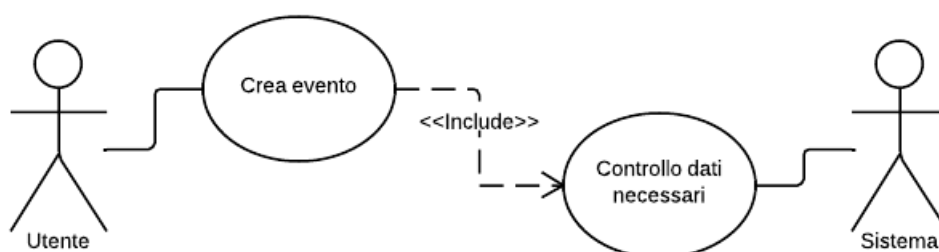


Figura 3.3: Crea evento

NOME	Crea evento;
OBBIETTIVO:	Creare un evento;
PRE-CONDIZIONE	L'utente deve essere registrato all'applicazione; L'utente deve aver eseguito l'accesso all'applicazione;
SUCCESSO:	L'utente crea l'evento;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.3: Crea evento

FLUSSO PRINCIPALE:

1. L'utente immette i dati necessari alla creazione dell'evento;
2. L'utente inoltra la richiesta di creazione;
 - (a) Il sistema verifica i dati inseriti;
3. L'utente ha creato l'evento;

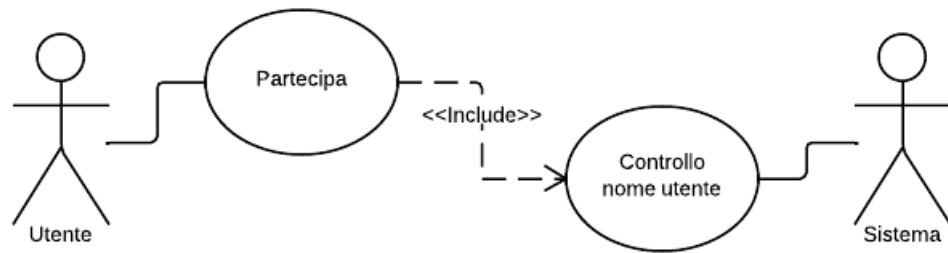


Figura 3.4: Partecipazione senza accesso

3.4 Partecipazione ad evento – Utente senza accesso

NOME	Partecipazione ad evento senza accesso;
OBBIETTIVO:	L'utente rende la propria disponibilità per un evento senza essere registrato all'applicazione;
PRE-CONDIZIONE	Nessuna;
SUCCESSO:	L'utente si rende disponibile per tale evento;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.4: Partecipazione ad evento senza accesso

FLUSSO PRINCIPALE:

1. L'utente seleziona l'evento desiderato e visualizza tutte le informazioni riguardanti l'evento;
 - (a) L'utente visualizza la lista di tutti gli utenti iscritti all'evento e il username dell'utente che l'ha creato;
 - (b) L'utente visualizza la lista di tutti i commenti per l'evento selezionato;
2. L'utente inserisce il proprio nome nel apposito spazio;
 - (a) Il sistema verifica che l'utente ha inserito correttamente il suo nome;
3. L'utente inoltra la richiesta d'iscrizione;
4. L'Utente si rende disponibile per tale evento;

3.5 Partecipazione ad evento – Utente con accesso

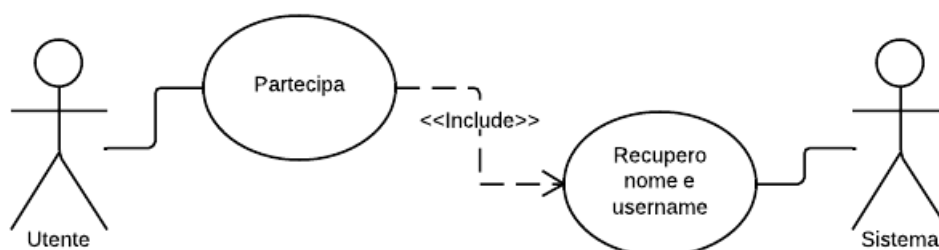


Figura 3.5: Partecipazione con accesso

NOME	Partecipazione ad evento con accesso;
OBBIETTIVO:	L'utente rende la propria disponibilità per un evento dopo aver eseguito l'accesso all'applicazione;
PRE-CONDIZIONE	L'utente deve essere registrato all'applicazione;
SUCCESSO:	L'utente si rende disponibile per tale evento;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.5: Partecipazione ad evento con accesso

FLUSSO PRINCIPALE:

1. L'utente seleziona l'evento desiderato e visualizza tutte le informazioni riguardanti l'evento;
 - (a) L'utente visualizza la lista di tutti gli utenti iscritti all'evento e il username dell'utente che l ha creato;
 - (b) L'Utente visualizza la lista di tutti i commenti per l evento selezionato;
2. L'utente inoltra la richiesta d iscrizione;
 - (a) Il Sistema recupera il nome del utente che ha eseguito l accesso;
 - (b) Il Sistema recupera il username del utente che ha eseguito l accesso;

3. L'utente si rende disponibile per tale evento;
 - (a) L'utente può lasciare commenti per l'evento selezionato;

3.6 Chiudere evento

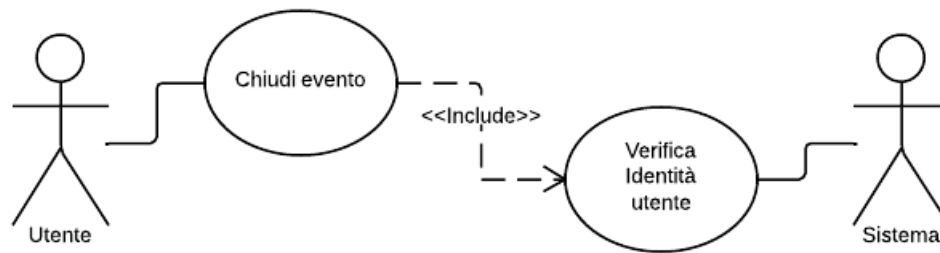


Figura 3.6: Chiudere un evento

NOME	Chiusura di un evento;
OBBIETTIVO:	L utente chiude un evento;
PRE-CONDIZIONE	L'utente deve essere registrato all'applicazione; L'utente è il creatore dell'evento;
SUCCESSO:	L utente chiude correttamente l'evento;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.6: Chiudere un evento

FLUSSO PRINCIPALE:

1. L'utente seleziona l'evento desiderato;
2. L'utente inserisce le eventuali cause che hanno portato alla chiusura dell'evento;
3. L'utente inoltra la richiesta di chiusura dell'evento;
 - (a) Il Sistema controlla che l'utente che vuole chiudere l'evento sia il creatore dello stesso;
4. L'utente ha chiuso l'evento;

3.7 Eliminazione evento

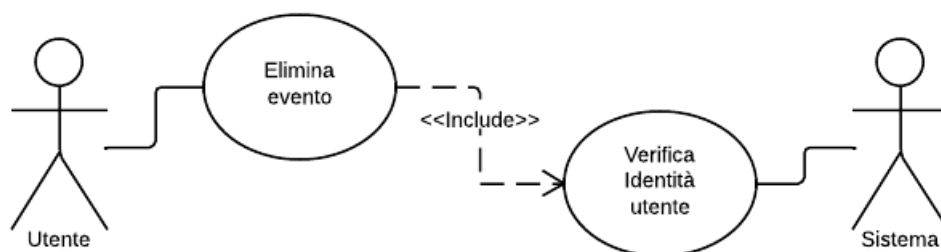


Figura 3.7: Eliminare un evento

NOME	Eliminazione di un evento;
OBIETTIVO:	L'utente elimina un evento;
PRE-CONDIZIONE	L'utente deve essere registrato all'applicazione; L'utente è il creatore dell'evento;
SUCCESSO:	L'utente elimina correttamente l'evento;
FALLIMENTO:	La procedura non va a buon fine;
ATTORE PRIMARIO:	Utente;
ATTORI SECONDARI:	Sistema;

Tabella 3.7: Eliminare di un evento

FLUSSO PRINCIPALE:

1. L'utente seleziona l'evento desiderato;
2. L'utente inoltra la richiesta di eliminazione dell'evento;
 - (a) Il sistema controlla che l'utente che vuole eliminare l'evento sia il creatore dello stesso;
3. L'utente ha eliminato l'evento;

3.8 Diagramma casi d'uso completo

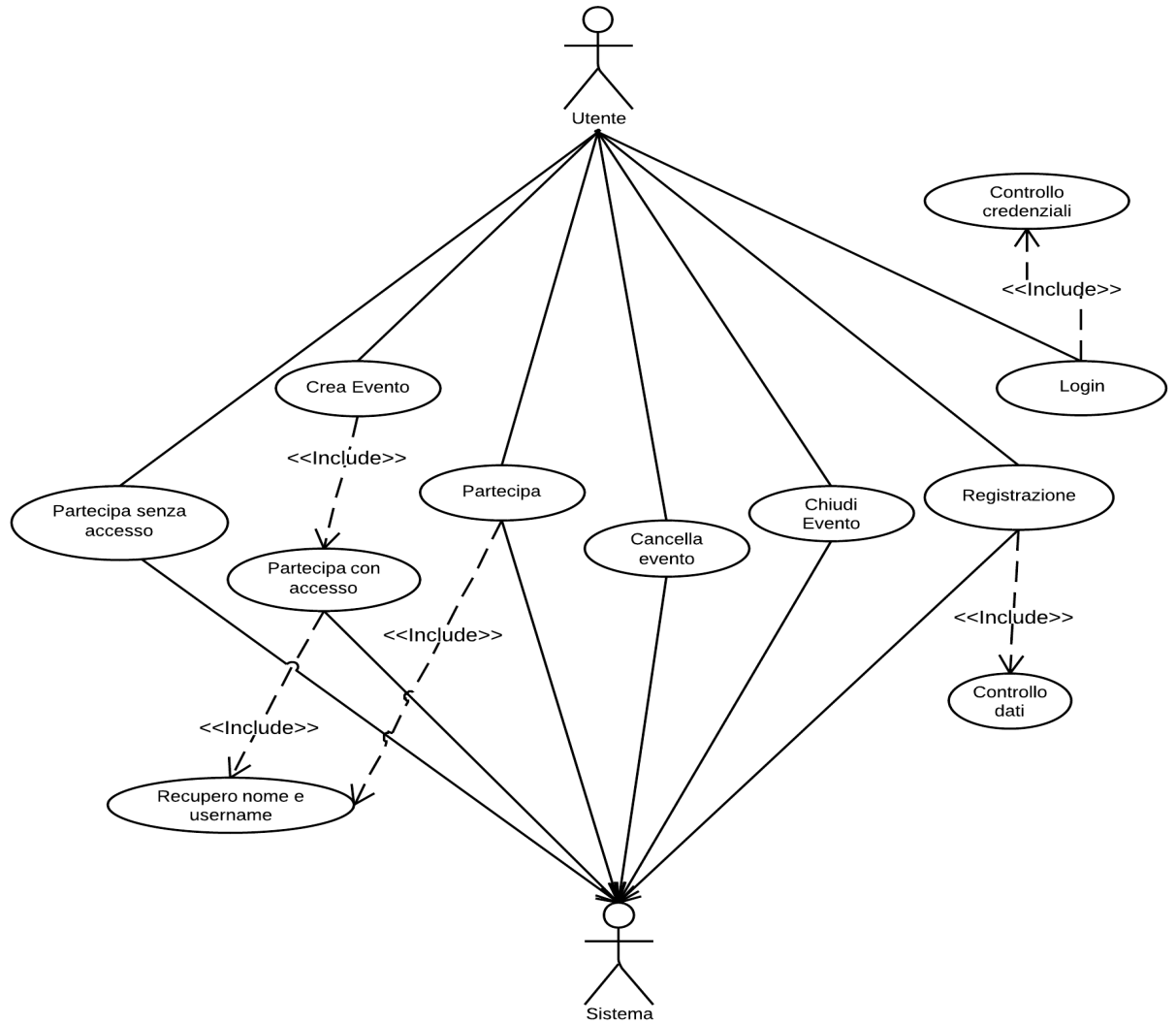


Figura 3.8: Diagramma casi d'uso

Capitolo 4

Diagramma delle attività

4.1 Registrazione

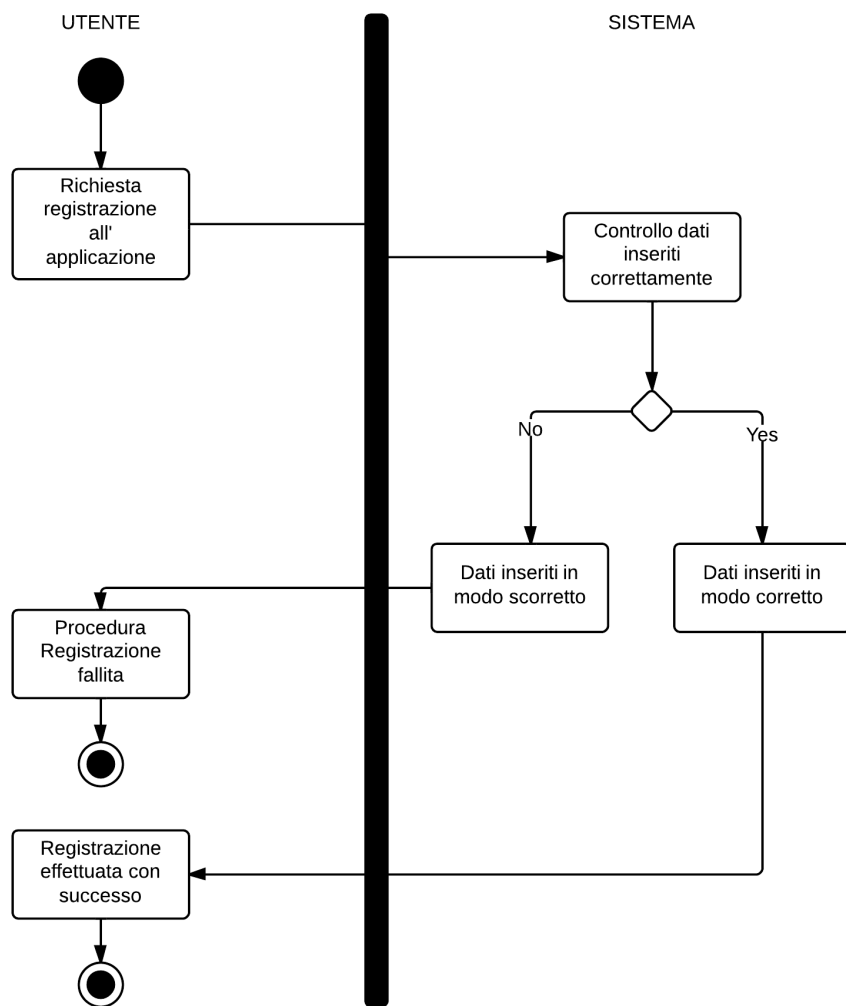


Figura 4.1: Diagramma Attività: Registrazione

4.2 Login

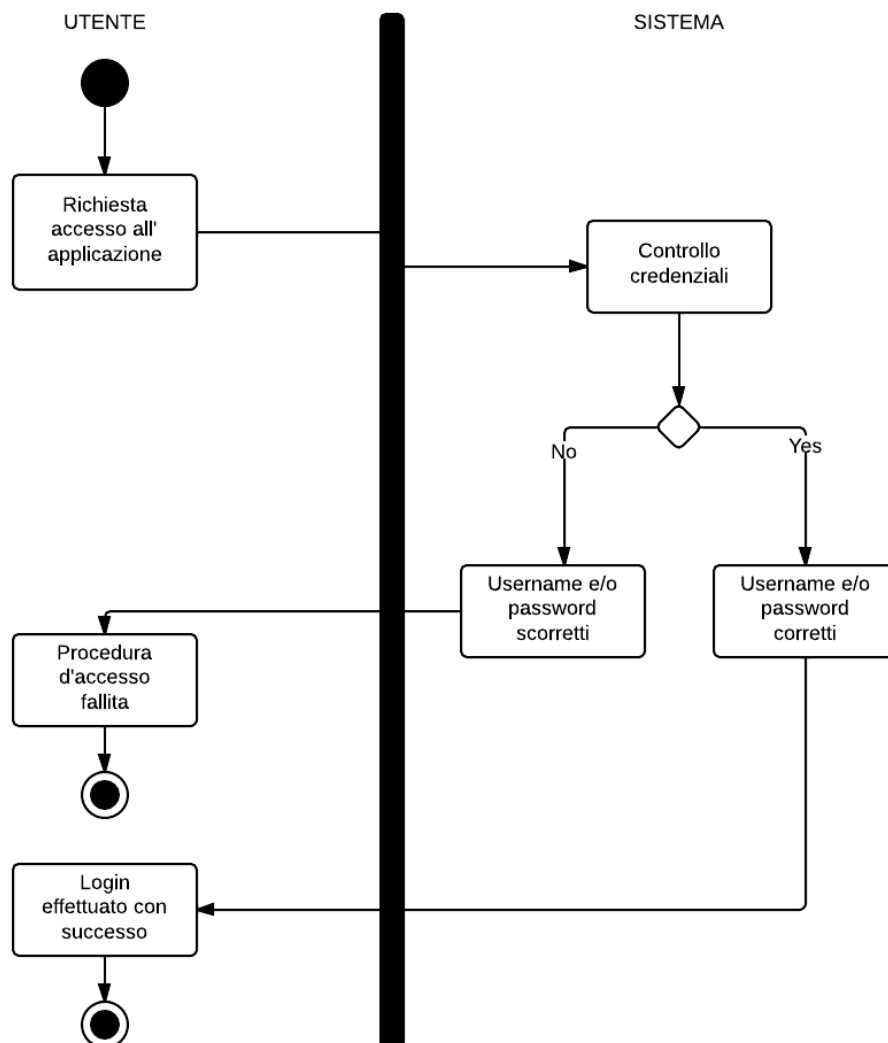


Figura 4.2: Diagramma Attività: Login

4.3 Crea Evento

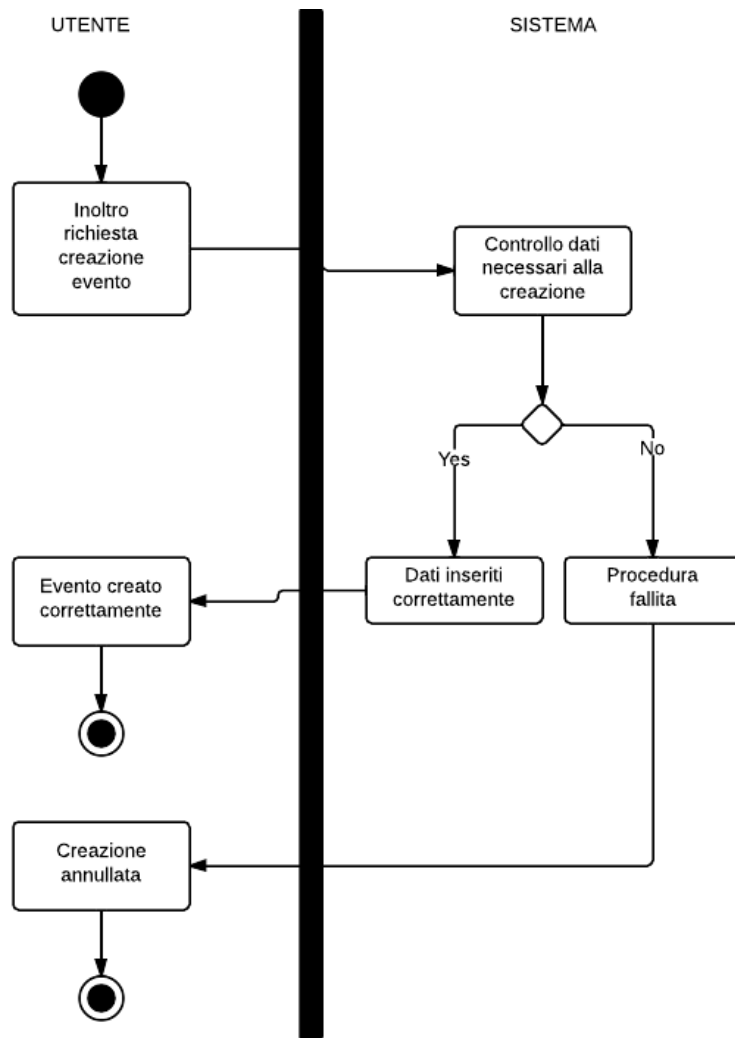


Figura 4.3: Diagramma Attività: Crea evento

4.4 Partecipazione ad evento – Utente senza accesso

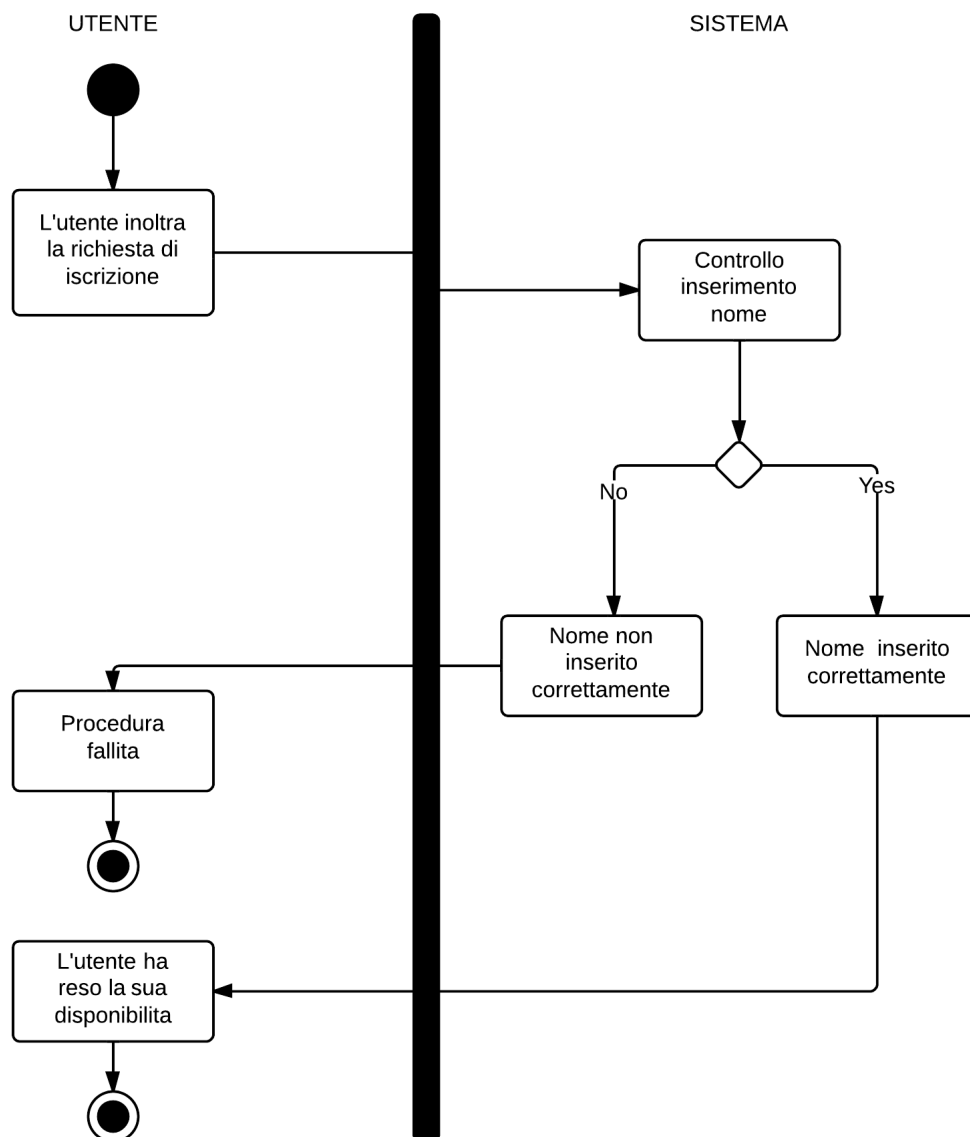


Figura 4.4: Diagramma Attività: Partecipazione senza accesso

4.5 Partecipazione ad evento – Utente con accesso

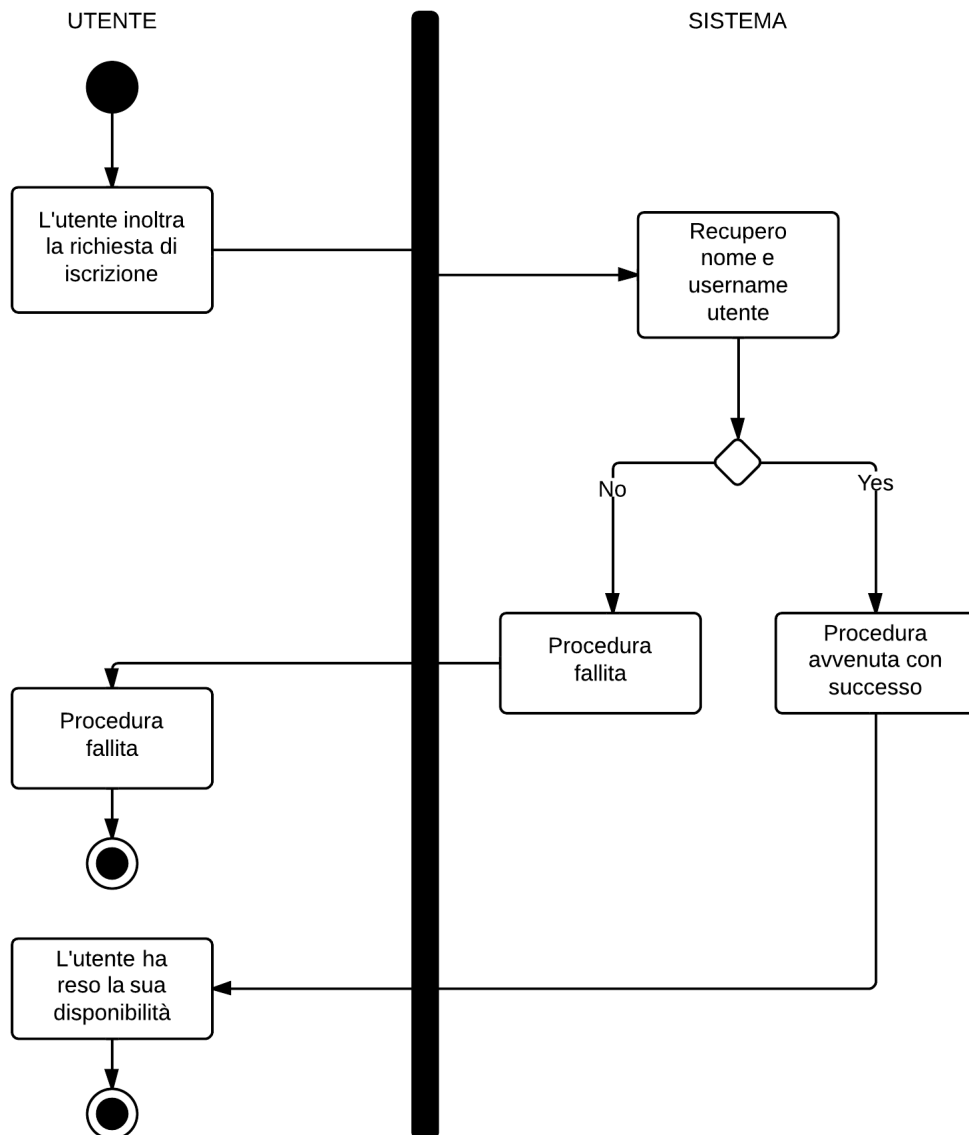


Figura 4.5: Diagramma Attività: Partecipazione con accesso

4.6 Chiudere evento

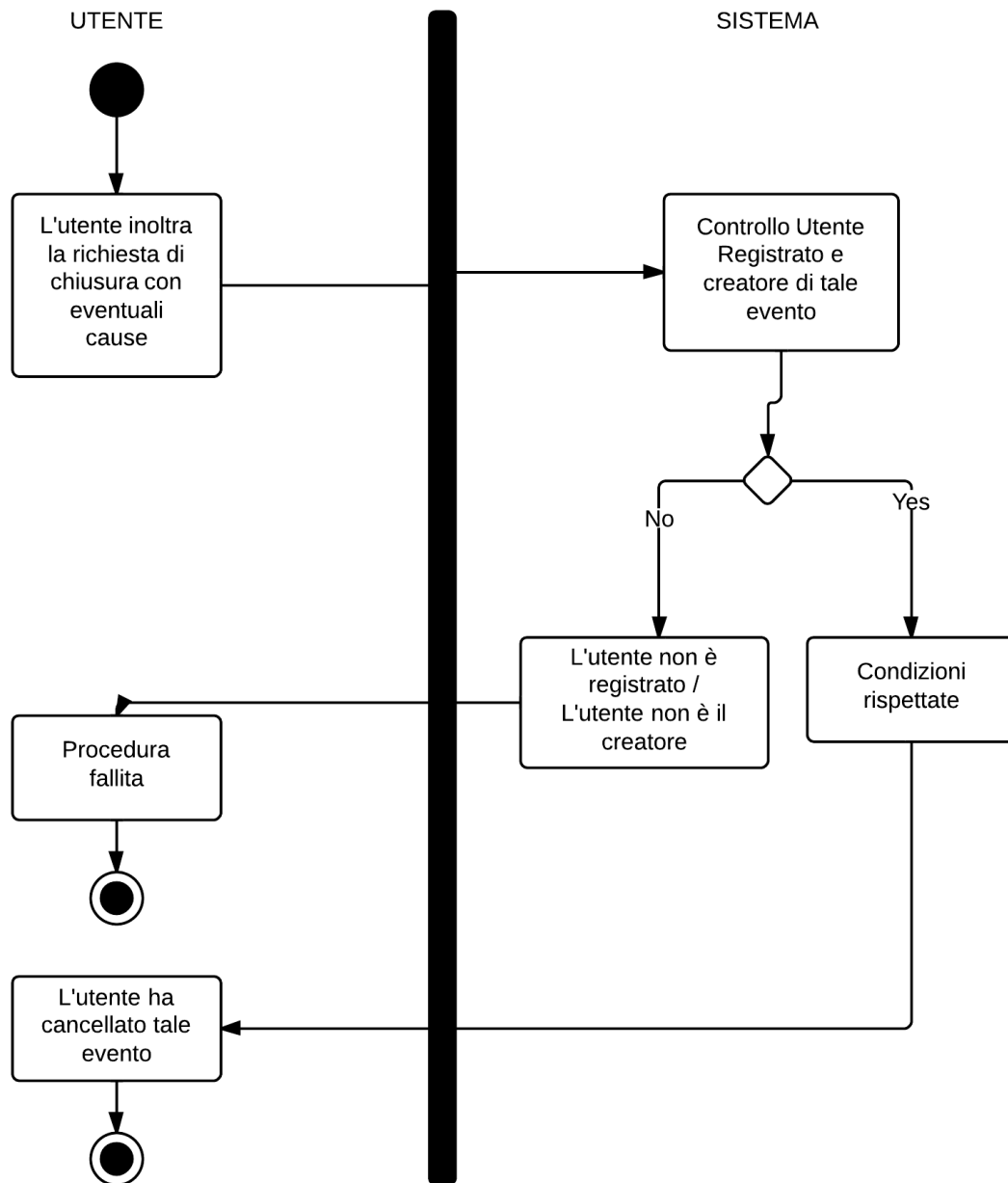


Figura 4.6: Chiudere un evento

4.7 Eliminare evento

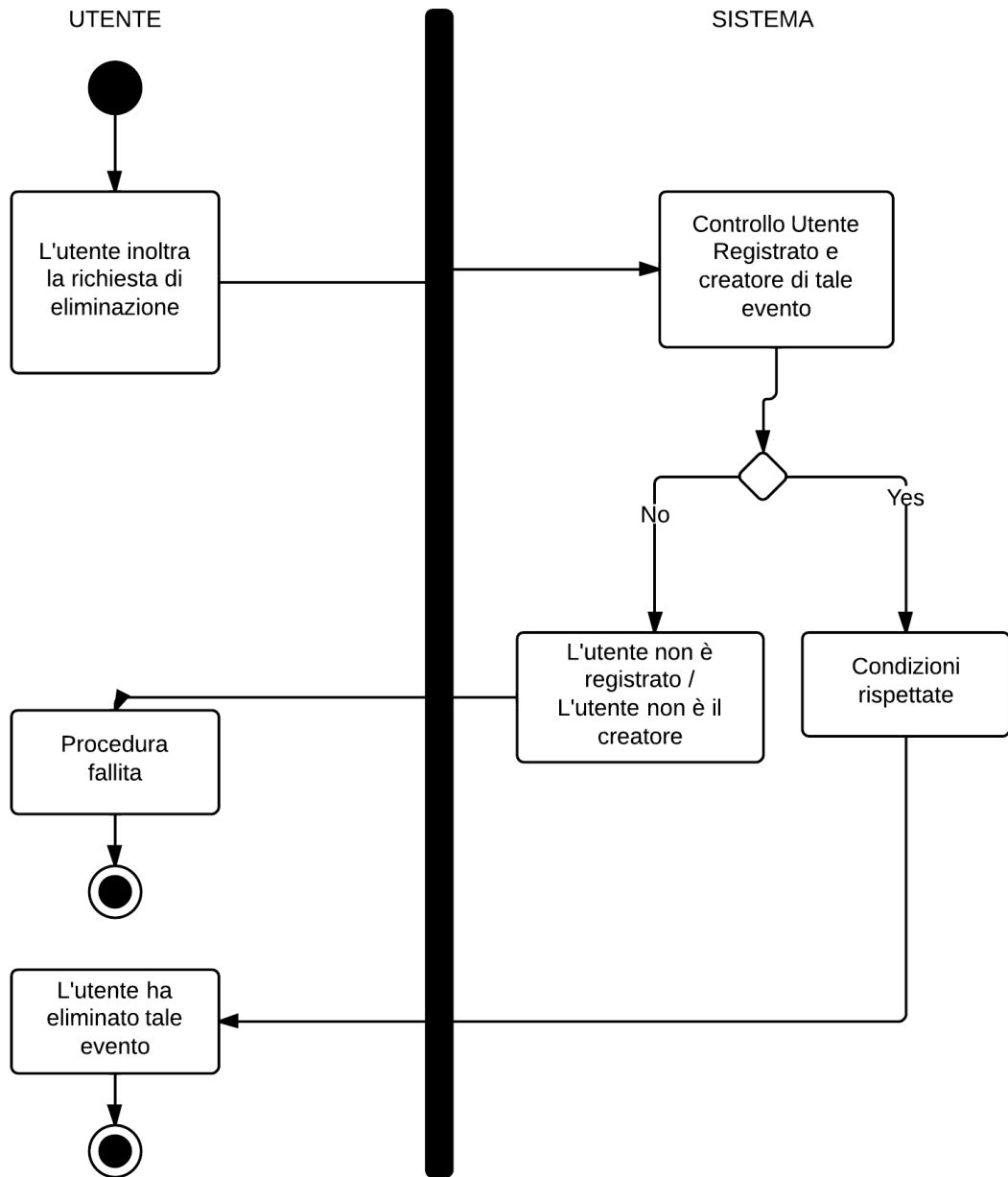


Figura 4.7: Eliminare un evento

Capitolo 5

Diagramma delle classi

5.1 Cardinalità

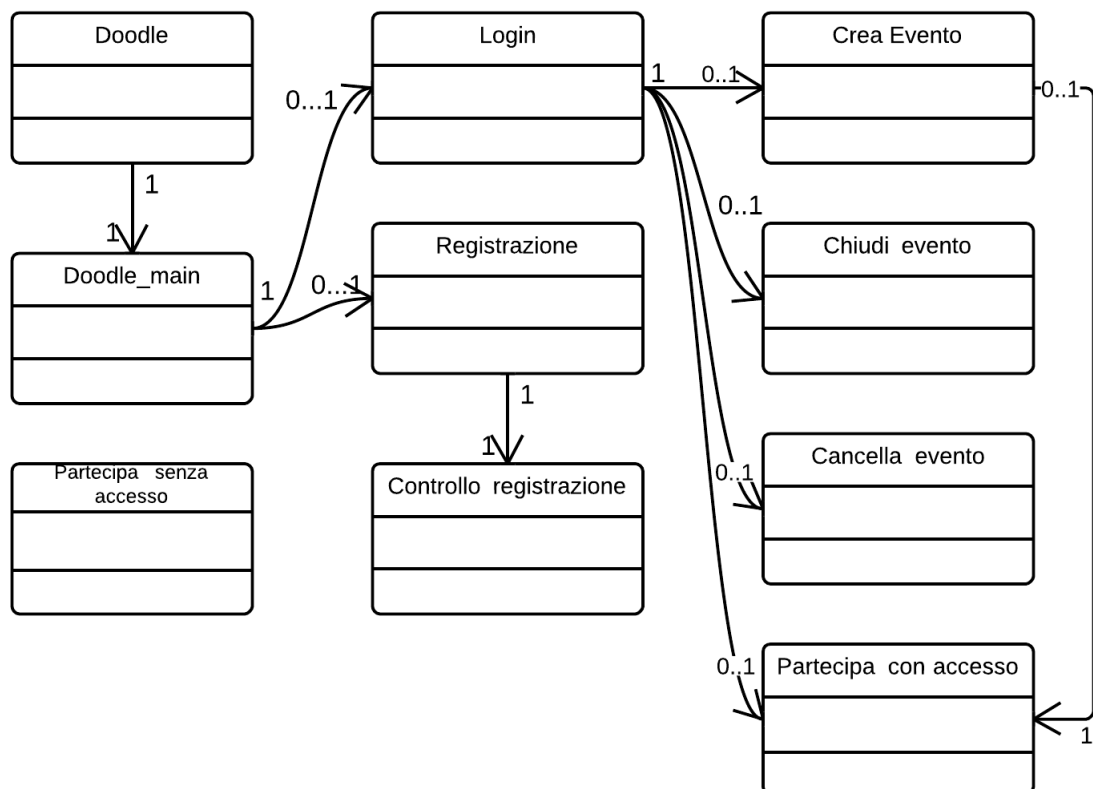


Figura 5.1: Diagramma delle classi

5.2 Diagramma totale delle classi Doodle

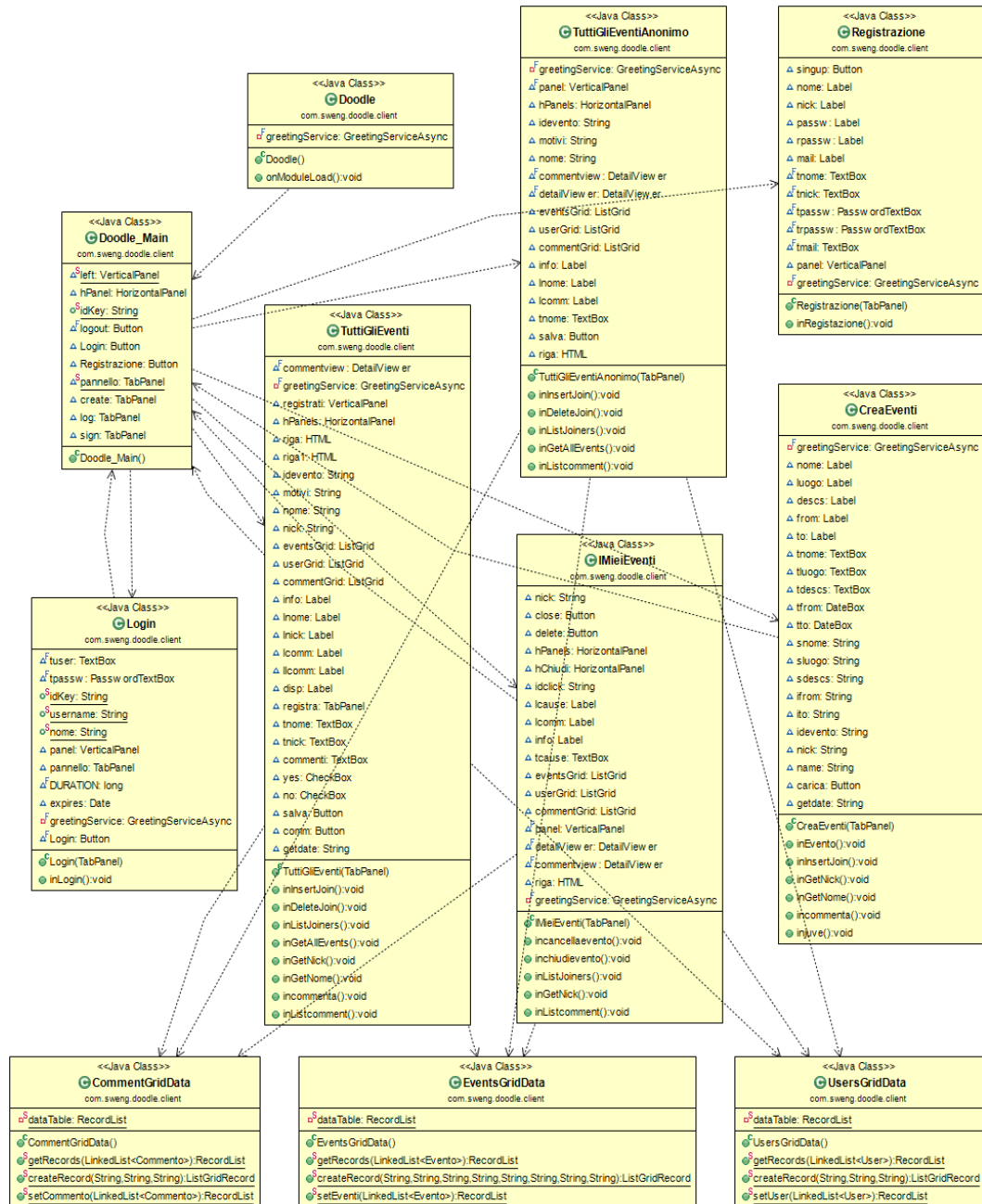


Figura 5.2: Diagramma delle classi

5.3 Diagramma totale delle classi Doodle 2.0

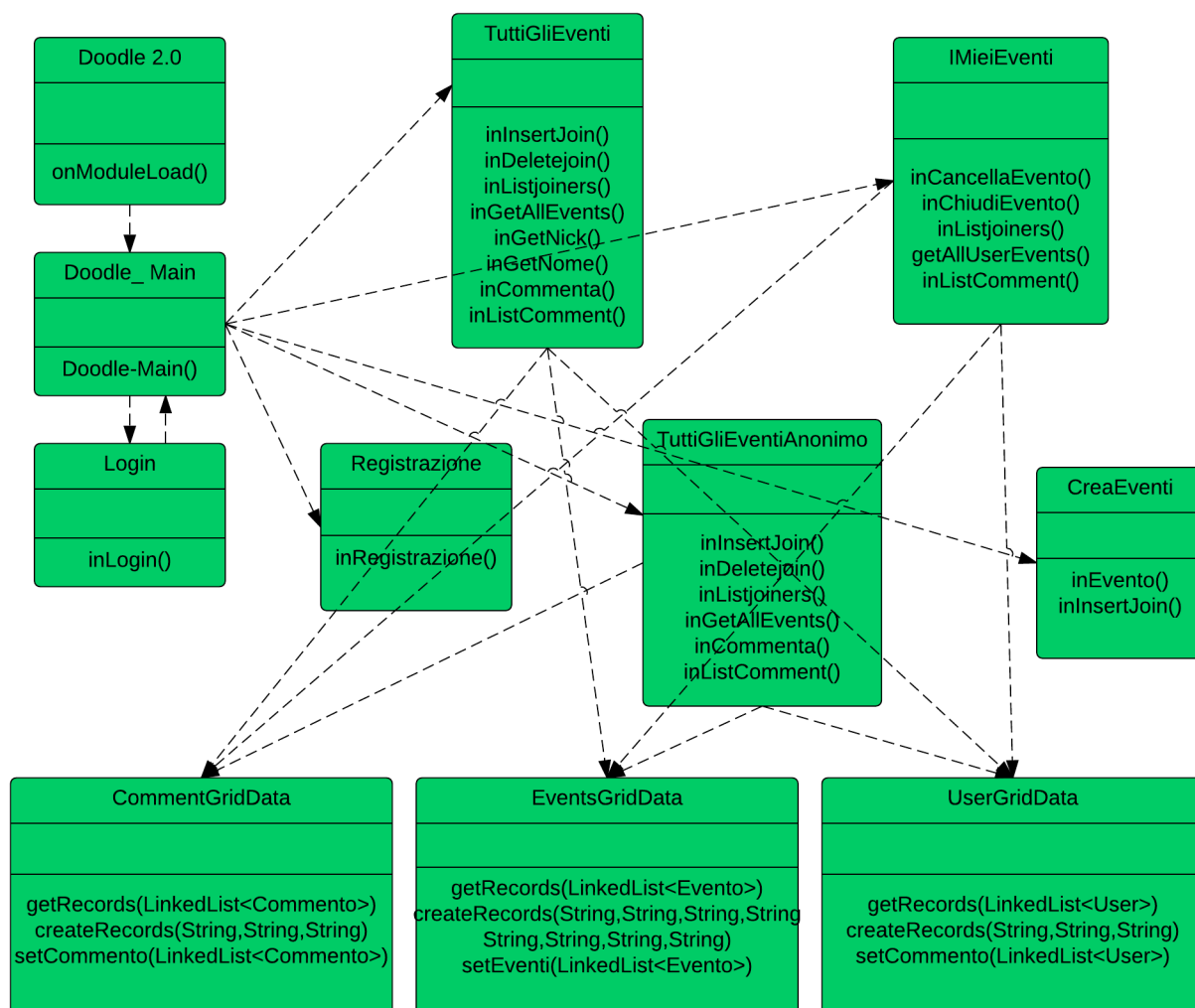


Figura 5.3: Diagramma delle classi Doodle 2.0

Capitolo 6

Diagramma di sequenza

6.1 Diagramma di sequenza

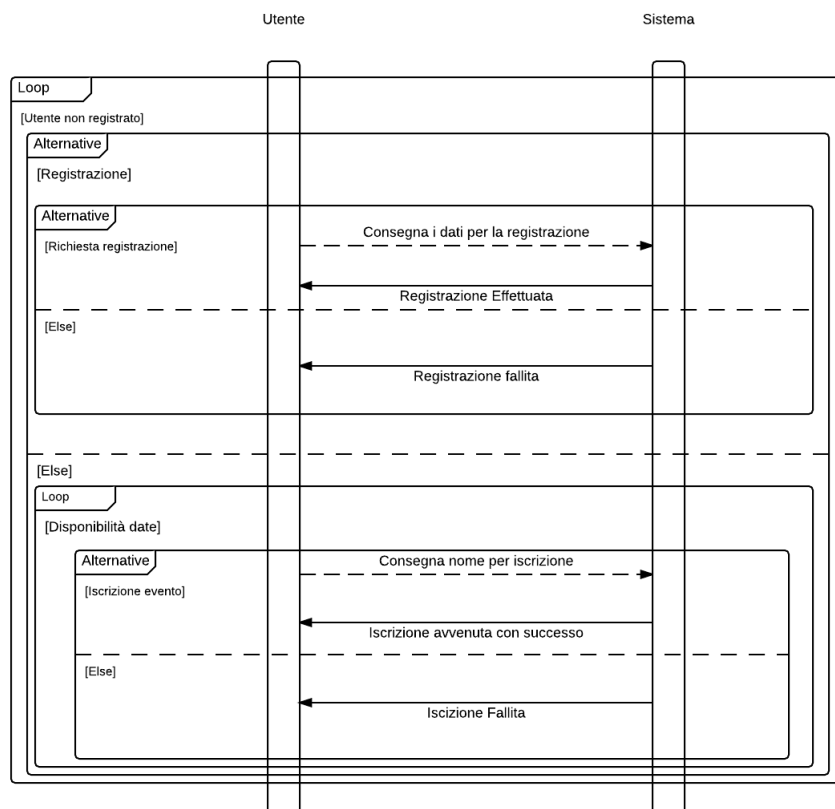


Figura 6.1: Diagramma di Sequenza

Capitolo 7

Diario

7.1 Diario Doodle 1.0

- 3 Settembre:
 - Identificazione dei primi dubbi sulle specifiche;
 - Identificazione e cenni all'utilizzo dei tool:

Lucidchart.com per i diagrammi;
- 4 Settembre:
 - Individuazione dei vari scenari possibili.
 - Prima stesura dell' Use Case diagram.
- 10 settembre:
 - Prodotte le bozze per il Glossario di progetto e diagrammi dei casi d'uso.
 - Revisione dei diagrammi dei casi d'uso
 - Inizio dei diagrammi delle attività.
- 14 settembre:
 - Diagrammi delle attività.
- 16 settembre:
 - Sviluppo dei diagrammi delle attività.
 - Effettuate delle piccole correzioni sui diagrammi dei casi d'uso.

- 18 settembre:
 - Fine dei diagrammi delle attività ed inizio il diagramma delle classi.
- 20 settembre:
 - Diagramma di sequenza.
- 21 settembre:
 - Analisi dei requisiti funzionali.
- 25 settembre:
 - Ultima revisione della documentazione , sono stati corretti gli ultimi errori.
- 28 settembre:
 - Prima stesura del progetto.
 - Inizio stesura codice e progettazione dell'interfaccia.
- 2 ottobre:
 - Scelta della gestione del db. In un primo momento scelto MYSQL.
 - Inizio Stesura codice server.
 - Inizio stesura query (file querymethods -> Doodle 1.0)
 - Inizio Stesura codice client.
- 3..5..8 ottobre:
 - Continuo la stesura del codice server e client.
 - Aggiunte funzioni di controllo.
- 10 Ottobre
 - Primi test dell'applicazione.
 - Correzione dei bug.
- 11 Ottobre
 - Stesura codice del client.
 - Nuovi test.

- 15 Ottobre
 - Stesura codice del client e server.
 - Nuovi test.
- 15 Ottobre
 - Debug.
 - Nuovi test.
- 16 Ottobre
 - Ultimate le ultime modiche sulla parte server e client del progetto.
 - Test della applicazione.
 - Revisione della documentazione.

7.2 Diario Doodle 2.0

Dopo aver scoperto che si consigliava usare MapDb per la gestione del DB ho ripreso Doodle 1.0 per apportare le modifiche richieste. Le modifiche sono state apportate facendo attenzione a non modificare , se non nei casi di obbligo, la parte client side. In seguito il diario del porting a mapdb:

- 1 Novembre:
 - Inizio a pensare a come introdurre MapDB.
 - Identificazione di tutte le funzioni necessarie alla gestione del db (openDb(), commit(), rollback(), etc.)
- 2...3...4 Novembre:
 - Modifiche alla documentazione.
 - Inizio modifiche lato server.
 - Creazione ServerDoodle.
- 5...9 Novembre:
 - Scrittura ServerDoodle.
- 10 Novembre:

- Scrittura ServerDoodle.
 - Piccole modifiche lato client.
- 11 Novembre
 - Modifca ServerDoodle.
 - Introdotta nuova grafica.
- ...15 Novembre
 - Debug ServerDoodle.
 - Nuova grafica.
 - Test junit.
- 17 Novembre
 - Test junit.
 - Rimozione codice inutilizzato.

Capitolo 8

Ciclo di sviluppo

8.1 Ciclo di sviluppo

La metodologia suggerita prevede una fase di analisi monolitica, che produca come artefatti una modellazione dei casi d'uso, attività, classi e sequenza e una di realizzazione iterativa ed incrementale ispirata ad una delle varie metodologie agili. In prima fase, quindi, ho sviluppato il Documento di Analisi con i vari diagrammi. In seguito, poi, mi sono dedicato alla progettazione dell'applicazione. Dopodiché mi sono dedicato allo sviluppo della prima release dell'applicazione andando a eliminare i problemi e le incongruenze sorte, risolvendo i problemi ad alto rischio prima e tralasciando quelli banali. Così facendo ho avuto modo di consentire un rapido sviluppo di versioni man mano più complete dell'applicazione. Una volta testato il lato server, mediante utilizzo di Junit mi sono dedicato ai test. Infine ho dato spazio all'aspetto grafico curandone i dettagli. Il tutto è stato frequentemente "committato" su github Github e Bitbucket. Questa relazione (pdf) è stata scritta in Latex. Caricati i file .tex sui git.