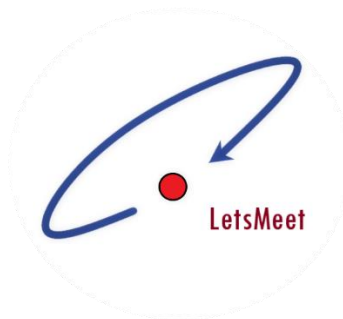


Università degli Studi di Salerno

Corso di Ingegneria del Software

LetsMeet SDD – Systema Design Document Versione 1.1



Docente:

Andrea De Lucia

Studenti:

Vittorio Aiello

Gerardo Benevento

Raffaele Sansone

Data: 24/03/2019

Progetto: LetsMeet	Versione: 1.2
Documento: SDD	Data: 24/03/2019

Coordinatore del progetto:

Nome	Matricola
Vittorio Aiello	0512104524

Partecipanti:

Nome	Matricola
Vittorio Aiello	0512104524
Gerardo Benevento	0512104584
Raffaele Sansone	0512104974

Scritto da:	Gerardo Benevento, Raffaele Sansone, Vittorio Aiello
--------------------	--

Revision History

Data	Versione	Descrizione	Autore
15/12/2018	1.0	Prima stesura del SDD	GB; VA; RS
20/01/2019	1.1	Revisione del SDD	GB; VA; RS
24/03/2019	1.2	Revisione TestPlan	GB; VA; RS

1.1 Scopo del sistema

La piattaforma LetsMeet si pone l'obiettivo di offrire l'opportunità, a qualsiasi tipologia di utente interessato, di poter creare eventi in real-time sul territorio attraverso una mappa interattiva.

Usualmente, chi interagisce con sistemi che offrono servizi di creazione e condivisione degli stessi eventi, mettono in secondo luogo l'interazione con la mappa, la quale funge solo come eventuale scorciatoia per accedere ad altre applicazioni, come Google Maps, per la visualizzazione del luogo d'incontro.

LetsMeet vuole concentrare l'attenzione proprio sull'utilizzo di una mappa dando anche la possibilità agli utenti di visionare in tempo reale anche il numero dei partecipanti effettivi agli eventi, cosa che effettivamente manca ad altre tipo di piattaforme in circolazione che si prefiggono l'obiettivo di creare eventi.

1.2 Obiettivi di Design

La piattaforma LetsMeet deve poter essere il più efficiente ed intuitiva possibile attraverso agevoli interfacce. Tale efficienza sarà costruita attraverso rapidi tempi di risposta ad ogni genere di input ma anche differenti politiche di tolleranza all'errore.

Per ottenere gli obiettivi finali vanno seguiti dei criteri di progettazione tenendo presente:

Performance, Affidabilità, Costi, Manutenzione e Utente Finale.

1.2.1 Criteri di Performance

Tempo di Risposta:

LetsMeet deve essere reattivo per tutte le operazioni più immediate come la creazione di un evento o la verifica della partecipazione ad un evento con un limite superiore di 10-20 secondi.

Per operazioni massive, come il caricamento di più eventi sulla mappa, il sistema deve garantire tempi di risposta ragionevoli nell'ordine delle decine di secondi, ma, essendo un sistema web, molto dipenderà dalla qualità della connessione e dalla congestione della rete sul sistema online.

Throughput:

I picchi di carico, fino a circa 100 (cento) utenti simultaneamente collegati, devono essere gestiti dal sistema senza rallentamenti significativi.

Memoria:

Il core del sistema non occuperà più di 10Gb. In questa stima non è incluso il database in quanto le sue dimensioni dipenderanno dal numero di utenti iscritti, eventi organizzati e utilizzo generale della piattaforma.

1.2.2 Criteri di affidabilità

Robustezza

Si tenta di mantenere consistenti i dati anche in caso di failure del sistema o dell'hardware ma non si garantisce la totale assenza di corruzioni o errori. Non vengono forniti strumenti per il backup automatico dei db, sarà compito dell'amministratore del sistema avere un piano di recupero

Disponibilità

LetsMeet sarà un'applicazione basata su un unico webserver, in quanto tale la disponibilità sarà la stessa di quella del provider del server a meno di problemi interni dell'applicazione

Tolleranza all'errore

Si cercherà di rendere l'applicazione più tollerante agli errori tramite controlli sui dati in ingresso sia lato server che lato client. Non è assicurata la totale tolleranza agli errori.

Sicurezza

La sicurezza sarà basata su quella offerta da Tomcat, come componenti valve e filtri. Le credenziali utente saranno mantenute in formato hash sha-256 per prevenire furti di dati

1.2.3 Criteri di costo

Sviluppo

I costi di sviluppo previsti riguardano l'affitto del server per la fase di sviluppo. Si utilizzerà come web Server un VPS fornito da OVH. Per quanto riguarda il DBMS verrà utilizzato MySQL che è open source.

Nei costi di sviluppo rientrano anche le licenze di Google per poter usufruire del servizio Maps e altri aggiuntivi di supporto come la reverse geolocalization.

Deployment

Il deployment avviene tramite war file. È necessario copiarlo all'interno della cartella corretta e Tomcat provvederà all'effettiva inizializzazione della web app.

1.2.4 Criteri di mantenimento

Estensibilità

La progettazione del sistema sarà condotta in modo da agevolare la facile introduzione di nuove funzionalità utilizzando il linguaggio di markup HTML5, i fogli di stile CSS3, JQuery e Java.

Si utilizzerà un approccio modulare, massivamente basato su interfacce in modo da facilitare l'estensione e la manutenibilità.

Portabilità

Il core del sistema sarà deployabile in un qualsiasi ambiente server che supporti Tomcat, sia Linux che Windows in quanto la web app sarà sviluppata in linguaggio Java.

Tracciabilità dei requisiti

Grazie alla tracciabilità dei requisiti, sarà possibile effettuare le modifiche necessarie al corretto funzionamento del sistema, valutando correttamente i costi e i rischi che le modifiche porteranno.

1.2.5 Criteri per l'utente finale

Usabilità

LetsMeet deve essere facilmente apprendibile (l'utente deve essere in grado di interagire con il sistema e padroneggiare le funzionalità in modo rapido) grazie ad una interfaccia pulita e minimale, deve essere flessibile e robusto (l'utente deve essere in grado di capire quando ha successo nel perseguire i suoi obiettivi nel sistema o quando sta sbagliando qualcosa per poter intervenire) utilizzando feedback informativi.

1.3 Definizione ed acronimi

- **LetsMeet:** La piattaforma di creazione eventi in questione.
- **Utente:** Utente generico registrato alla piattaforma che può creare, partecipare ad eventi ed utilizzare tutte le funzioni a disposizione della piattaforma.
- **Moderatore:** Utente generico registrato alla piattaforma che può visualizzare le segnalazioni fatte dagli Utenti e può sospenderli eventualmente.
- **Super-Admin:** Utente esterno che può indicare un Utente come Moderatore.
- **Login:** attività di accesso all'account;
- **Logout:** attività di uscita dell'account connesso;
- **DBMS:** Database Management System;
- **SQL:** Structured Query Language, ed è linguaggio di interrogazione (dei database) strutturato.
- **RAD:** Requirements Analysis Document

1.4 Riferimenti

L'insieme del materiale di riferimento utilizzato per la realizzazione del progetto e per la stesura di questo stesso documento comprende:

- Libro di testo: B.Bruegge, A.H. Dutoit, Object Oriented Software Engineering - Using UML, Patterns and Java, Prentice Hall, 3rd edition, 2009;
- Slide del docente, reperibili sulla piattaforma;
- Documento RAD del progetto LetsMeet;

1.5 Panoramica

Il documento si compone di una prima parte in cui vengono introdotti gli obiettivi di design. È doveroso fare un excursus alle attività di system design che compongono le basi per l'architettura software del sistema.

- **Decomposizione del sistema:** il sistema viene decomposto in sottosistemi allo scopo di poter assegnare ad un team parti di software semplici da sviluppare. Un sottosistema è una collezione di classi, associazioni, operazioni, eventi e vincoli che sono in relazione tra di loro. Un insieme di operazioni correlate forma un servizio. Ogni sottosistema quindi sarà caratterizzato dai servizi che offre ad altri sottosistemi; l'insieme dei servizi che un sottosistema espone sarà denominato Interfaccia(API).
- **Mapping Hardware/Software:** descrive come i sottosistemi vengono assegnati all'hardware e alle componenti "off-the-shelf".
- **Gestione dati persistenti:** descrive i dati persistenti memorizzati dal sistema e l'infrastruttura di gestione richiesta per essi.
- **Politiche di accesso e sicurezza:** descrive il modello utente del sistema in termini di una matrice degli accessi, stabilendo in modo più preciso le operazioni e le informazioni effettuabili da ogni singolo attore e come questi si autenticano al sistema.
- **Flusso di controllo globale:** descrive quali operazioni eseguire ed in che ordine, per garantire il corretto flusso di controllo del sistema.
- **Condizioni Limite:** descrive lo start-up, lo shutdown e i comportamenti errati del sistema.

2. Architettura del software corrente

Attualmente un sistema come LetsMeet non è in circolazione, ma esistono piattaforme che si prefiggono l'obiettivo di condividere eventi ed utilizzano una mappa interattiva unicamente per poter visionare il percorso oppure il luogo dell'evento interessato.

La maggior parte delle piattaforme offrono all'utente unicamente una pagina con le informazioni dell'evento ed eventualmente un numero di partecipanti, senza dare un'effettiva conferma del numero dei partecipanti effettivi ad un evento.

Google in parte offre un servizio servizi di visualizzazione di punti d'interesse su di una mappa, ma non ha le funzionalità del sistema LetsMeet proposto.

Per integrare la mappa interattiva verranno utilizzate le API messe a disposizione da Google: Maps e Geocoding. Le API possono essere trovare [qui](#) per Geocoding e [qui](#) per Maps.

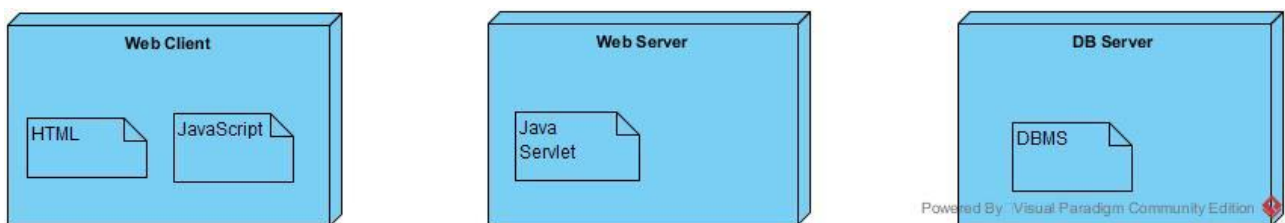
3. Architettura del software proposto

3.1 Panoramica

Il sistema proposto è un software web-based per la gestione dei propri eventi online. Le tipologie di utenti si dividono in: utente, moderatore e super-admin. L'utente usufruirà della quasi totalità dei servizi offerti dalla piattaforma; il moderatore accederà alle stesse funzionalità di un utente e in più potrà accedere alle funzionalità che si occupano di gestire gli utenti, dando la possibilità di poter visionare le eventuali segnalazione di commenti o eventi e poter sospendere dalla piattaforma gli utenti segnalati; infine il super-admin potrà gestire solamente l'eventuale trasformazione di un utente in moderatore, poi potrà accedere come utente alla piattaforma.

L'architettura del sistema si presenta come una architettura Client/Server "three-tier" per i seguenti motivi:

- **Portabilità:** il sistema (essendo web-based) potrà essere utilizzato su una varietà di macchine e sistemi operativi.
- **Trasparenza:** il sistema nonostante sia distribuito è in grado di fornire i propri servizi al singolo utente senza interferire con gli altri utenti del sistema.
- **Performance:** il sistema garantirà reattività per i task degli utenti collegati ma essendo web-based molto dipenderà dalla qualità della connessione online.
- **Scalabilità:** il sistema sarà in grado di gestire un grosso numero di utenti connessi contemporaneamente che effettuano le operazioni in contemporanea a patto di avere abbastanza risorse hardware e di banda.
- **Flessibilità:** Il sistema fornirà un'interfaccia grafica intuitiva e con funzionalità specifiche per il ruolo dell'utente che effettuerà l'accesso.
- **Affidabilità:** Il core del sistema non offre meccanismi di recupero automatici, sarà obbligo del sistemista provvedere al backup dei dati.



3.2 Decomposizione del sistema

Per realizzare la piattaforma LetsMeet è stato utilizzato lo stile architetturale three-tier in versione Client/server. L'architettura three-tier ("a tre strati") indica una particolare architettura software di tipo multi-tier per l'esecuzione di un'applicazione web che prevede la suddivisione dell'applicazione in tre strati dedicati rispettivamente alla interfaccia utente, alla logica funzionale e alla gestione dei dati persistenti.

In particolare, i tre strati si occupano di differenti funzionalità del sistema di seguito descritte:

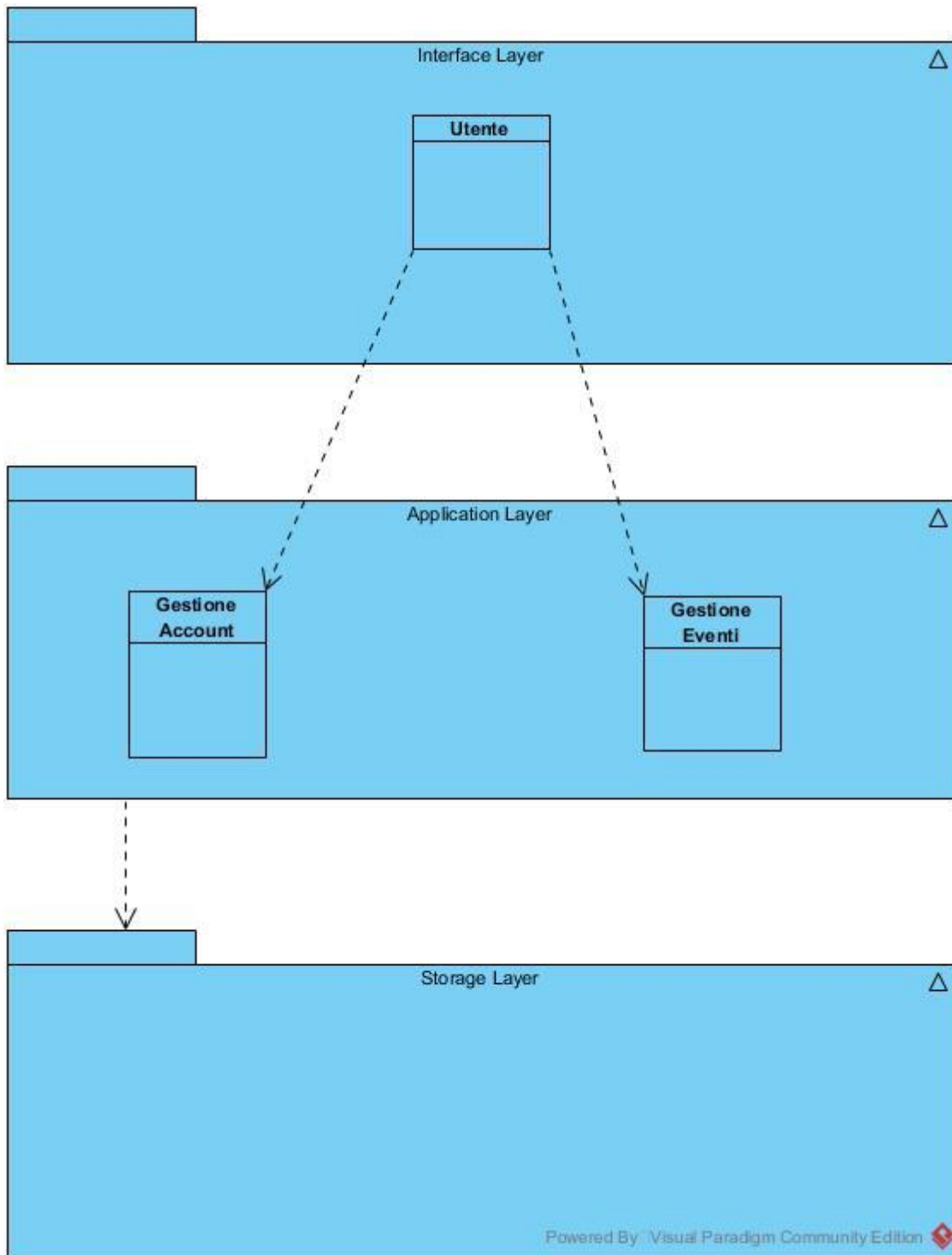
- **Interface Layer:** Include tutte le interfacce grafiche e in generale i *boundary objects*, come le form con cui interagisce l'utente. L'interfaccia verso l'utente è rappresentata da un Web server e da eventuali contenuti statici (es. pagine HTML).
- **Application Layer** include tutti gli oggetti relativi al controllo e all'elaborazione dei dati. Questo avviene interrogando il database tramite lo storage layer per generare contenuti dinamici e accedere a dati persistenti
- **Storage Layer** effettua la memorizzazione, il recupero e l'interrogazione degli oggetti persistenti. I dati, i quali possono essere acceduti dall'application layer, sono depositati in maniera persistente su un database tramite DBMS.

Riportiamo una breve descrizione delle gestioni del sistema:

- **Gestione account:** Questa funzionalità raccoglie tutte le informazioni per gestire l'autenticazione degli utenti su LetsMeet, la trasformazione di un profilo utente in moderatore e la ricerca/visualizzazione dei profili utenti.
- **Gestione Eventi:** Questa funzionalità raccoglie tutte le informazioni per la gestione degli eventi sulla piattaforma compresa la loro ricerca e l'invio dei commenti inerenti a quegli eventi.
- **Gestione Segnalazione:** Questa funzionalità raccoglie tutte le informazioni per la gestione delle segnalazioni degli utenti.

In seguito, verranno mostrati i singoli layer.

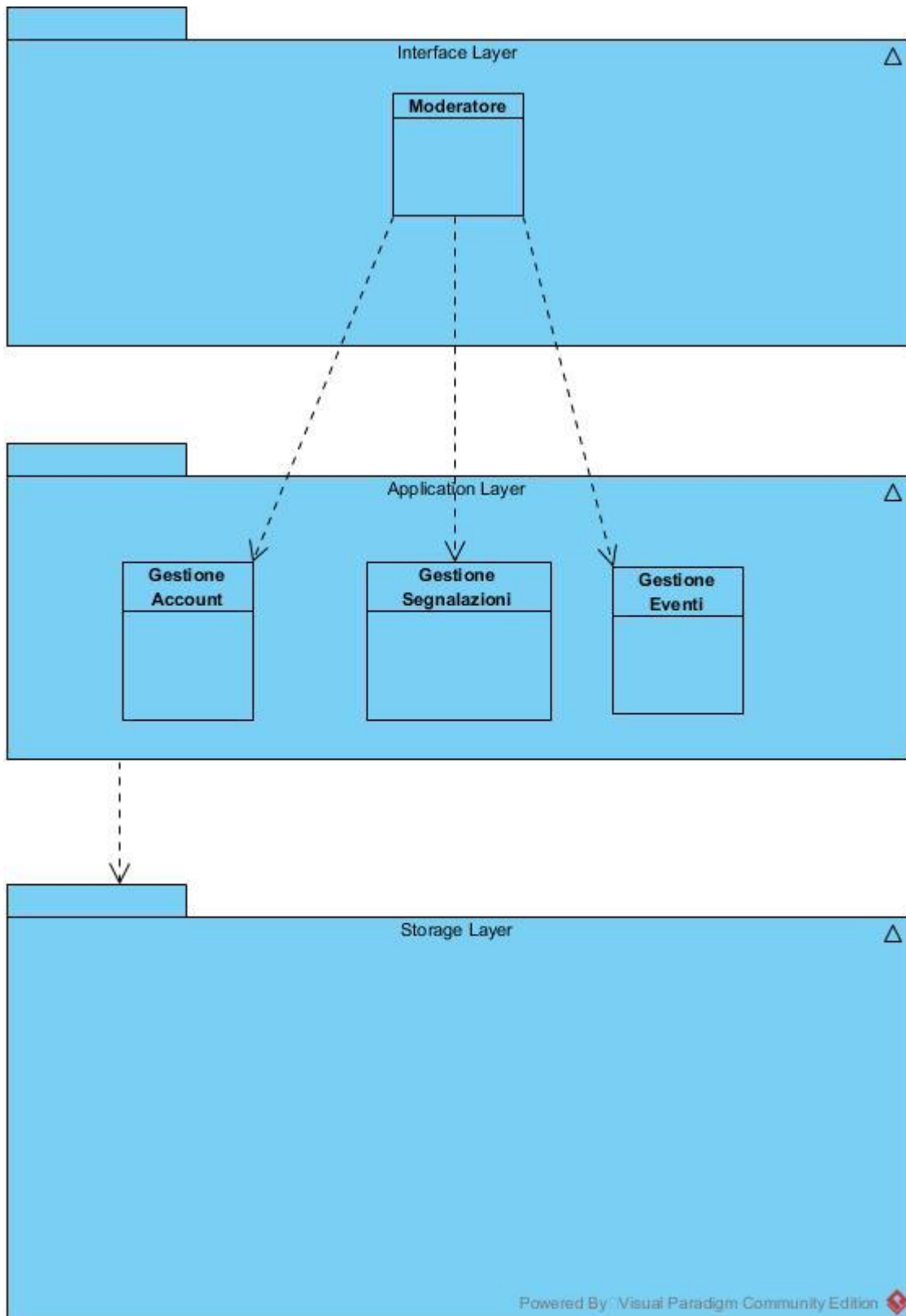
Utente



- **Gestione Account:** L'Utente potrà godere delle funzioni riguardanti la propria autenticazione all'interno della piattaforma, la visualizzazione del proprio profilo personale e quello degli altri utenti con eventuale ricerca.

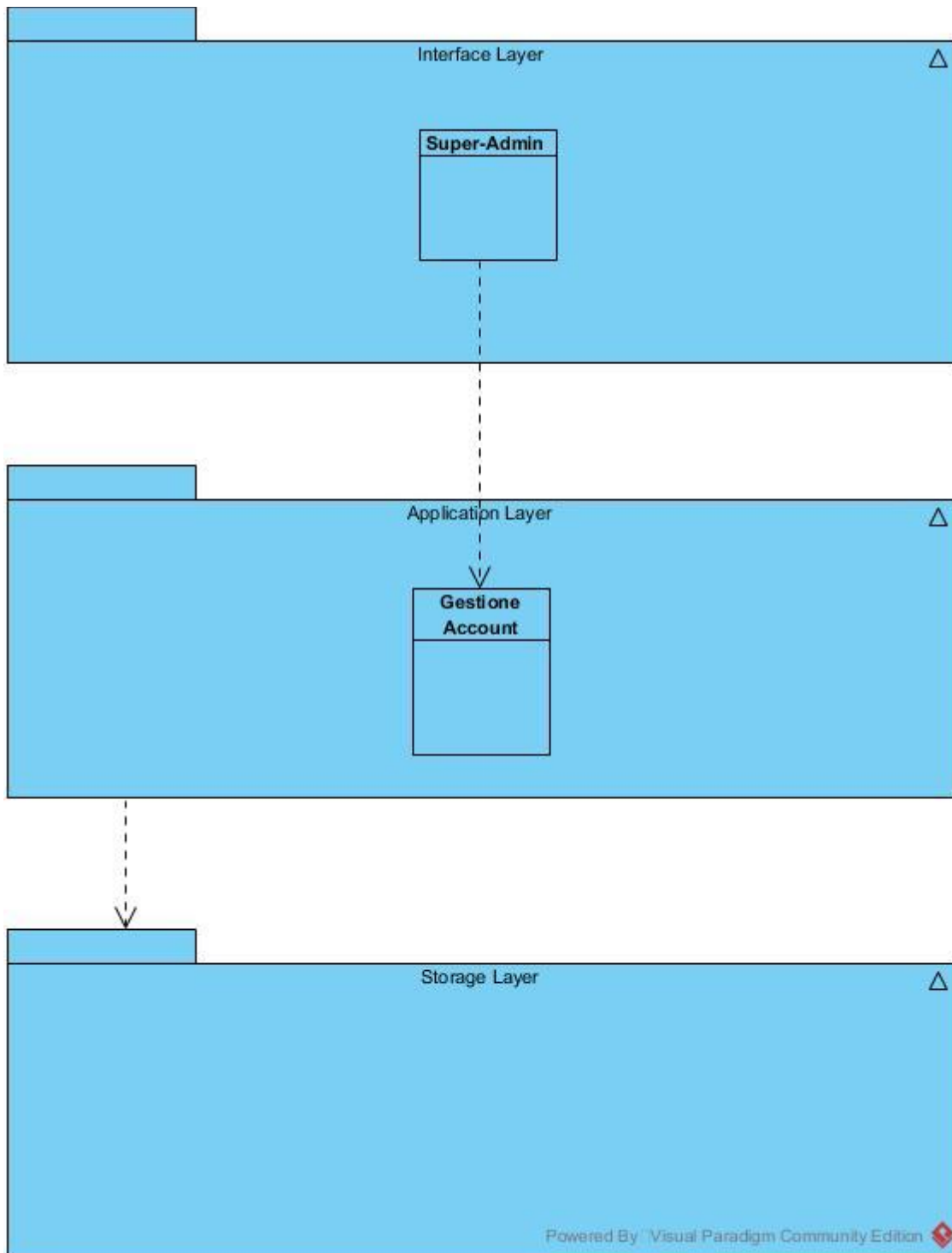
- **Gestione Eventi:** L'Utente potrà godere delle funzionalità di creazione, partecipazione, verifica partecipazione di un evento e la possibilità di ricerca degli eventi con visualizzazione delle eventuali pagine di informazioni.

Moderatore



- **Gestione Account:** Il Moderatore potrà godere delle funzioni riguardanti la propria autenticazione all'interno della piattaforma, la visualizzazione del proprio profilo personale e quello degli altri utenti con eventuale ricerca.
- **Gestione Eventi:** Il Moderatore potrà godere delle funzionalità di creazione, partecipazione, verifica partecipazione di un evento e la possibilità di ricerca degli eventi con visualizzazione delle eventuali pagine di informazioni.
- **Gestione Segnalazioni:** Il Moderatore potrà godere delle funzioni di visualizzazione e risoluzione di segnalazioni effettuate dagli altri profili della piattaforma, sospendendo gli eventuali utenti segnalati.

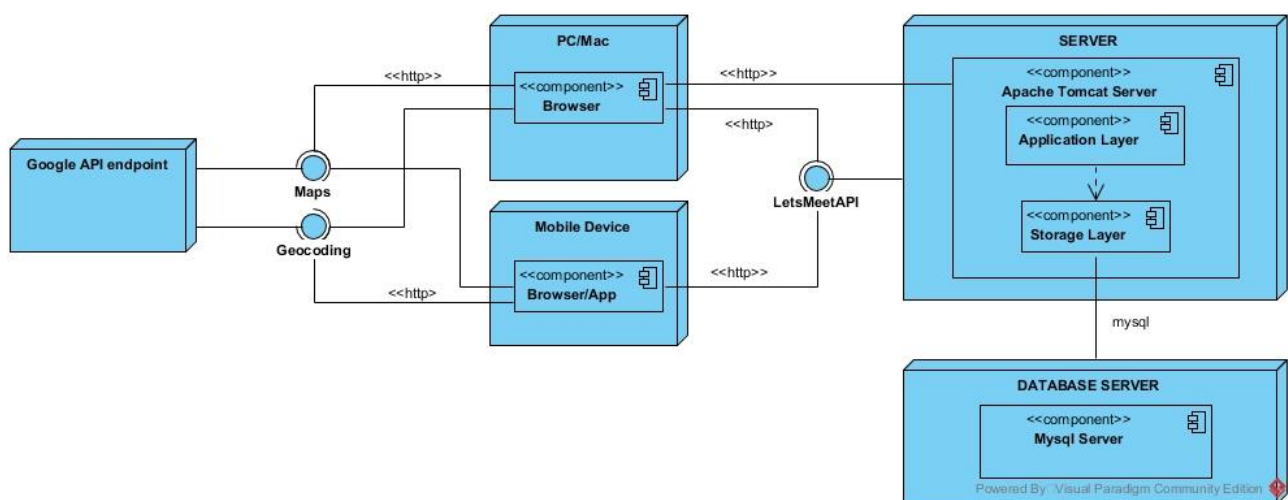
Super-Admin



- **Gestione Account:** Il Super-admin potrà godere della funzione di trasformazione degli utenti in moderatori.

3.3 Mapping Hardware /Software

La struttura hardware proposta è costituita da un server centrale e dai client cioè un qualsiasi computer al quale un utente può collegarsi per sfruttare il browser per accedere al sistema LetsMeet. Al server si collegano i client ed il database, il tipo di utente è determinato in fase di autenticazione controllando nel database il tipo di utente che corrisponde allo Username inserito in quel client. I client dovranno effettuare richieste al server per eseguire le operazioni legate alle loro rispettive funzionalità. Il client ed il server saranno connesse tramite una rete che utilizzerà il protocollo TCP/IP.



Interface Layer

L'utente utilizza il sistema mediante un Browser installato all'interno del suo PC/Mac (ad es. Opera, Firefox, Chrome) o da un device mobile tramite una possibile futura applicazione tramite API.

Application layer

Il sistema, e quindi le funzionalità, sono implementate in linguaggio JAVA. Il codice in JAVA produrrà codice in linguaggio HTML e il codice risultante viene inviato al browser del client.

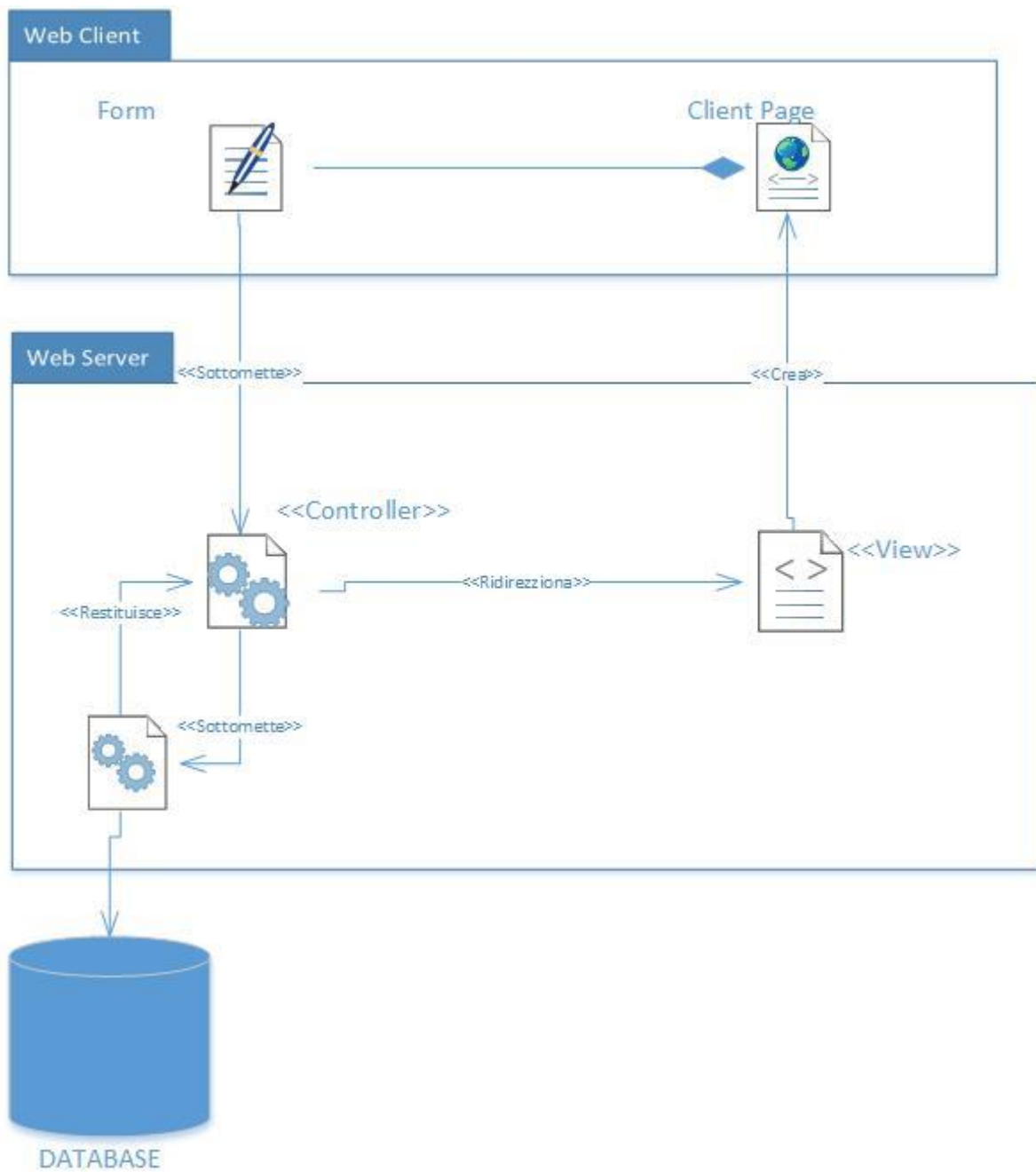
Inoltre sono previste funzioni esposte tramite API interrogabili tramite protocollo HTTP che non prevedono la restituzione di una pagina HTML ma di una risposta JSON.

Storage layer

Rappresenta il collegamento con il server da parte del sistema e si occupa di tutte le richieste di accesso e modifiche sui dati permanenti presenti nel database.

Database Server

Il DBMS usato è MySQL il quale presenta molte API che permettono l'interazione tra sistema e database.



3.4 Dati persistenti

Vedere il file (LetsMeet_Dati_Persistenti.pdf) allegato con la pubblicazione dei dati persistenti.

3.5 Controllo degli accessi e della sicurezza

Di seguito viene riportata la matrice degli accessi per i tre tipi di account

Oggetti	Entità	Utente	Moderatore	Super-Admin
Account		<ul style="list-style-type: none">• Login• Logout• Registrazione• Ricerca_Utenti• Visualizza_Profilo_PS	<ul style="list-style-type: none">• Login• Logout• Registrazione• Ricerca_Utenti• Visualizza_Profilo_PS	<ul style="list-style-type: none">• Registrazione_Mod
Eventi		<ul style="list-style-type: none">• Visualizza_Eventi• Creazione_Evento• Rating_Evento• Partecipa_Evento• Verifica_Partecipazione• Ricerca_Eventi• Visualizza_Evento• Scrittura_Commento• VisualizzaCommenti• Cancellazione Evento	<ul style="list-style-type: none">• Visualizza_Eventi• Creazione_Evento• Rating_Evento• Partecipa_Evento• Verifica_Partecipazione• Ricerca_Eventi• Visualizza_Evento• Scrittura_Commento• VisualizzaCommenti• Cancellazione Evento	
Segnalazioni		<ul style="list-style-type: none">• Segnala_Evento• Segnala_Commento	<ul style="list-style-type: none">• Segnala_Evento• Segnala_Commento• Accetta_Segnalazione• Rifiuta_Segnalazione	

3.6 Controllo del software globale

Il controllo del flusso software viene gestito da classi java che interagendo con il client, il quale si interfaccia tramite un web browser, svolgendo le varie operazioni. Il server smista ogni nuova richiesta alla classe java adeguata, inoltrando poi la risposta al client.

3.7 Condizioni Boundary

Le condizioni limite riguardano l'accensione e lo spegnimento del sistema per quanto riguarda il lato Server. Dal lato Client si riferiscono agli errori di connessione al server.

3.7.1 Avvio del sistema

È possibile gestire l'avvio del sistema tramite l'interfaccia di amministrazione di Apache Tomcat. Non sono previste particolari procedure di startup. Il sistema è in uno stato stabile come viene completata l'inizializzazione da parte di Tomcat.

Il sistema dopo essersi avviato presenta un'interfaccia ai client. Dopo aver effettuato l'autenticazione ogni utente può accedere alle funzionalità disponibili.

3.7.2 Terminazione del sistema

È possibile gestire la terminazione del sistema tramite l'interfaccia di amministrazione di Apache Tomcat. Non sono previste particolari procedure di shutdown.

La terminazione del sistema comporta la disconnessione di tutti i client connessi.

3.7.3 Fallimento del sistema

Nel caso si verifichi un errore dovuto all'hardware o al software si cercherà di ripristinare una configurazione del sistema precedente allo stato d'errore.

Non è da escludere la perdita dei dati, se si verifica un guasto al supporto di memorizzazione dei dati nel database-server o una corruzione dello stesso.

Per minimizzare questo rischio, si incoraggia il gestore ad eseguire periodicamente dei backup del database del sistema e sottoporre il sistema hardware a controlli.

4.0 Servizi dei sottosistemi

4.1 Gestione Account

Sottosistema	Gestione Account
Descrizione	Sottosistema che gestisce l'autenticazione degli utenti, tutte le operazioni riguardanti la registrazione degli Utenti o Moderatori e le operazioni di ricerca e visualizzazione di account
Servizi offerti	Descrizione
Login	Questa funzionalità permette di effettuare l'accesso al sistema, tramite le proprie credenziali, per poi sfruttare le funzionalità che esso offre
Logout	Questa funzionalità permette di uscire dal sistema.
Registrazione Utente	Questa funzionalità permette di registrare un nuovo account della piattaforma
Registrazione Moderatore	Questa funzionalità permette di registrare un Moderatore della piattaforma
Ricerca Utente	Questa funzionalità permette di poter cercare un Utente che abbia un profilo sulla piattaforma
Visualizza Profilo Utente	Questa funzionalità permette di visualizzare la pagina profilo di un Utente, che si la propria o di un altro Utente

4.2 Gestione Eventi

Sottosistema	Gestione Eventi
Descrizione	Sottosistema che gestisce tutte le operazioni riguardante gli eventi sulla piattaforma, come la ricerca, la visualizzazione, la partecipazione, verifica della partecipazione e la scrittura dei commenti sugli eventi
Servizi offerti	Descrizione
Visualizzazione Eventi	Questa funzionalità permette di visualizzare gli eventi su una mappa interattiva
Creazione Evento	Questa funzionalità permette di creare un evento
Rating Evento	Questa funzionalità permette di assegnare un voto ad un evento a cui si ha partecipato
Partecipa Evento	Questa funzionalità permette di indicare la propria partecipazione ad un evento
Verifica partecipazione Evento	Questa funzionalità permette di poter verificare la propria partecipazione ad un evento
Visualizza info Evento	Questa funzionalità permette di visualizzare le informazioni di un evento
Ricerca Evento	Questa funzionalità permette di ricercare un eventuale evento creato sulla piattaforma
Scrittura Commento	Questa funzionalità permette di scrivere un commento allegato ad un evento
Visualizzazione Evento	Questa funzionalità permette di visualizzare i commenti scritti inerenti ad un evento in precedenza

4.3 Gestione Segnalazione

Sottosistema	Gestione Segnalazione
Descrizione	Sottosistema che gestisce le segnalazioni inviate ai moderatori da parte degli utenti
Servizi offerti	Descrizione
Segnalazione Evento	Questa funzionalità permette di poter segnalare un evento creato sulla piattaforma
Segnalazione Commento	Questa funzionalità permette di poter segnalare un commento pubblicato sulla piattaforma
Accettazione segnalazione	Questa funzionalità permette ad un moderatore di accettare la segnalazione cancellando l'item segnalato e sospendendo l'utente
Rifiuto segnalazione	Questa funzionalità permette ad un moderatore di rifiutare una segnalazione

5.0 Glossario

LetsMeet: Piattaforma che verrà sviluppata.

Web-Based: Il termine identifica un sistema basato sul web, quindi accessibile simultaneamente da più postazioni.

Utente: Qualsiasi attore che può beneficiare dei servizi del sistema.

Moderatori: Utenti speciali che possono che possono godere delle funzioni legate alla gestione delle segnalazioni.

Amministratore: Un attore speciale che po' godere di funzionalità limitate come Startup e Shutdown del sistema.

RAD: Documento di Analisi dei Requisiti.

DBMS: Sistema di gestione delle basi di dati.

Database: Insieme organizzato di dati persistenti.