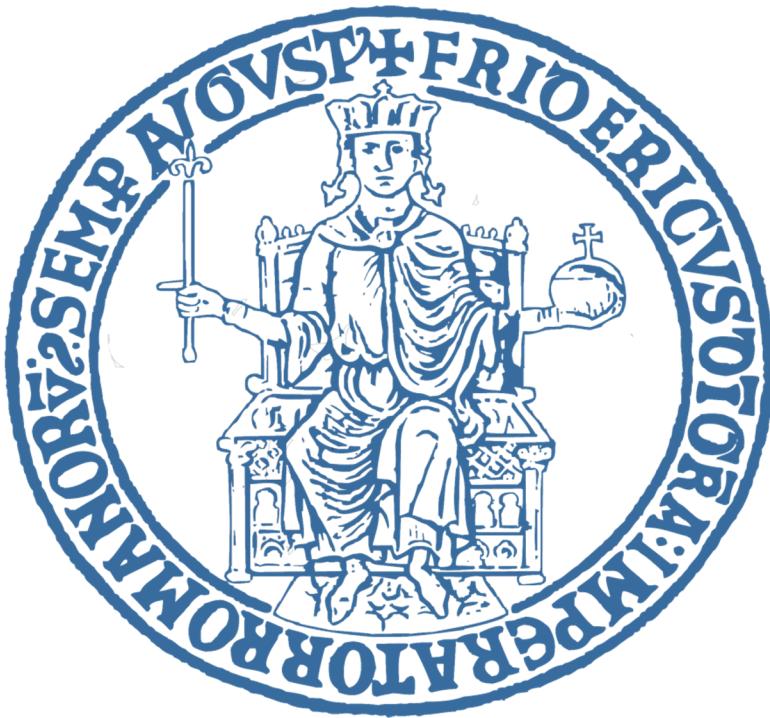


UNIVERSITA' DEGLI STUDI FEDERICO II DI NAPOLI

Facoltà di Scienze Informatiche



Tecnologie Web

Anno Accademico:2012/2013

“Progetto Agenda”

Studenti:

*Bove Luigi N86/75
Email: luig.bove@gmail.com*

*Fattoruso Michele N86/299
Email: mi.fattoruso@studenti.unina.it*

*Grazioso Marco N86/391
Email: mar.grazioso@libero.it*

Indice

Parte Prima-Proposta di soluzione

1. Descrizione del problema	3
2. Tipologie di utenti	3
3. Funzionalità del sistema	3
4. Metodi di Session Tracking.....	9
5. Tecnologie utilizzate	11
5.1 Tecnologie lato server.....	11
5.2 Tecnologie lato client	11
5.3 Base di dati	11
5.4 Strumenti di supporto.....	11

Parte Seconda-Relazione Finale

6. Introduzione	12
7. Funzionalità da implementare con relative modifiche alle sitemap.....	12
8. Architettura del sistema.....	19
9. Tecnologie Utilizzate.....	19
10. Tecnologie di Session tracking.....	20
11. Ipotesi Aggiuntive	21

1. Descrizione del problema

Si vuole realizzare un versione online di una comune agenda cartacea. Il problema richiede che l'applicazione consenta all'utente di programmare eventi (appuntamenti, riunioni, meeting,...) che avvengono in determinate fasce orarie o per l'intera giornata. Si richiede inoltre che sia possibile programmare attività con scadenze che saranno ricordate all'utente. Sarà messa a disposizione dell'utente una rubrica atta a contenere i contatti abituali o comunque coinvolti nelle attività dell'utente. Quest'ultimo, inoltre, avrà la possibilità di inviare email ai contatti contenuti nella rubrica. L'agenda permetterà mediante le API di Google Maps di visualizzare il luogo dell'evento e la distanza dalla posizione attuale dell'utente. L'applicazione prevede una fase di autenticazione che consentirà all'utente di accedere alla propria area dove potrà disporre di tutte le funzionalità sopra elencate. L'idea è quella di sviluppare un'applicazione utilizzabile mediante browser web senza l'installazione di nessun software client sulla macchina.

2. Tipologie Utente

Il primo passo da eseguire è quello di individuare la tipologia di utente che utilizzerà il sistema. Il sistema in esame prevede un'unica tipologia di utente, rappresentata dall'utilizzatore dell'agenda, il quale avrà a disposizione tutte le funzioni sopra citate.

3. Funzionalità del sistema

All'interno di questo paragrafo verrà delineata la struttura dell'applicazione web. Inoltre verrà presentata la sitemap dell'applicazione, suddividendola in più parti, al fine di dare una maggiore comprensione di ogni singola parte.

La prima fase presa in esame si compone delle seguenti funzionalità:

- Login
- Registrazione
- Recupero Password

esse sono descritte nella seguente Sitemap.

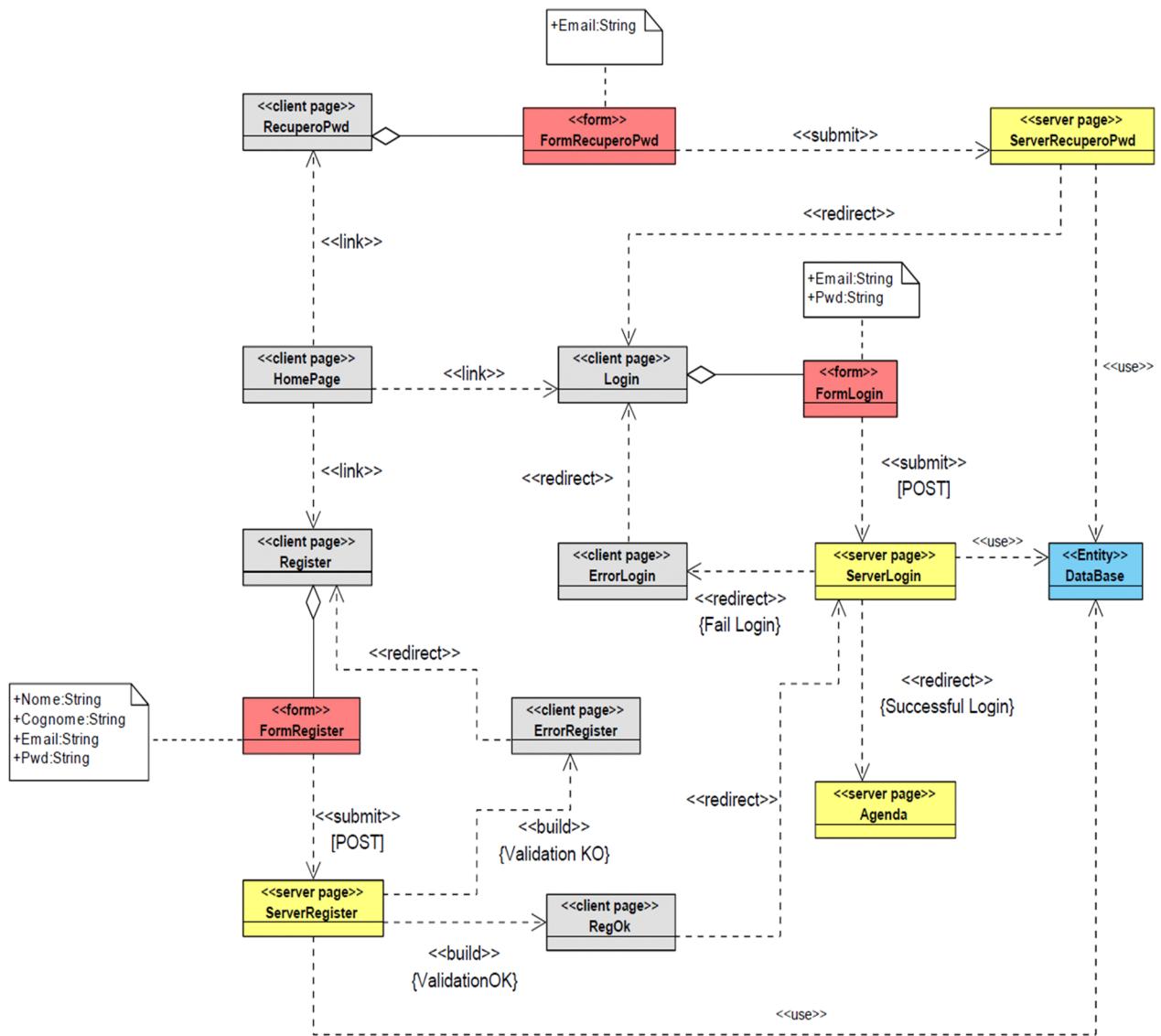


Figura 1 - Sitemap Login

Una volta effettuata l'autenticazione l'utente si troverà nella schermata Home Agenda. Di seguito, andremo ad esporre una Sitemap di tipo generale che descriverà i vari collegamenti tra le funzionalità del sistema che esporremo successivamente in maniera più approfondita.

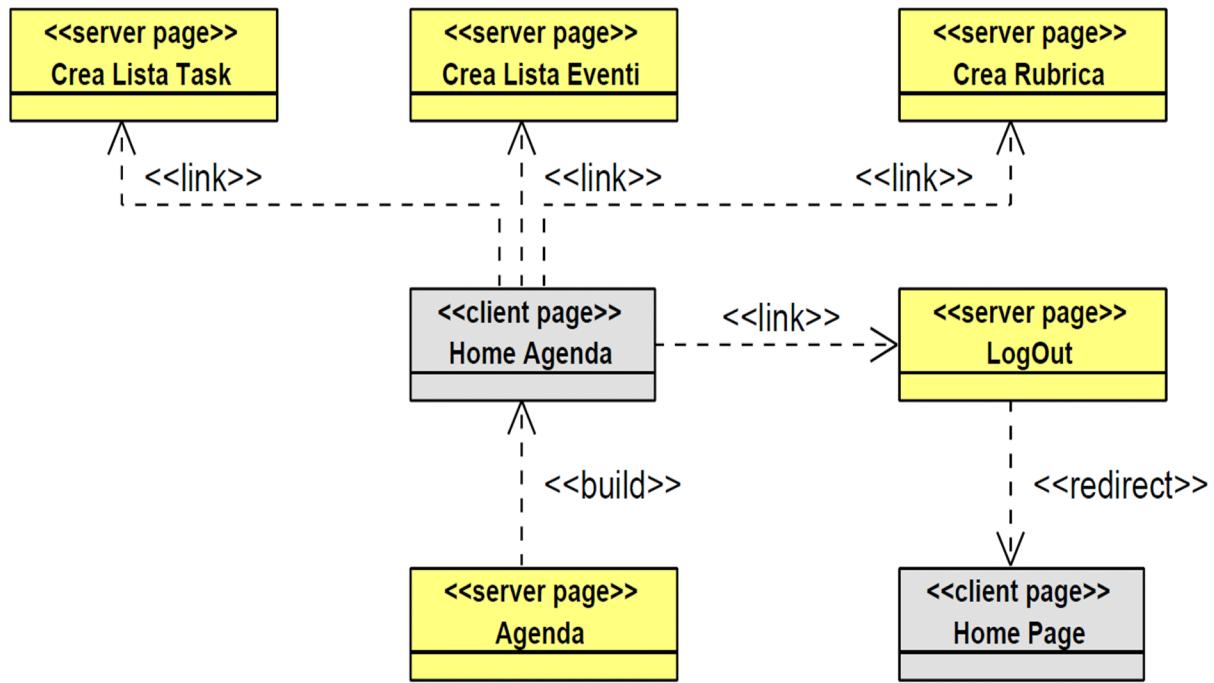


Figura 2 - Sitemap Generale

L'utente a questo punto si troverà nella sua home page e da lì potrà gestire :

- Gli eventi;
- I task ;
- La rubrica;

Da questo punto in poi le Sitemap saranno descritte a partire dalla Home Agenda.

Gestione degli eventi

In questa sezione descriveremo le operazioni che è possibile effettuare per quanto riguarda gli eventi.

Esse si suddividono in:

- Modifica
- Cancellazione
- Visualizzazione dell'evento e del luogo in cui si terrà quest'ultimo
- Aggiunta

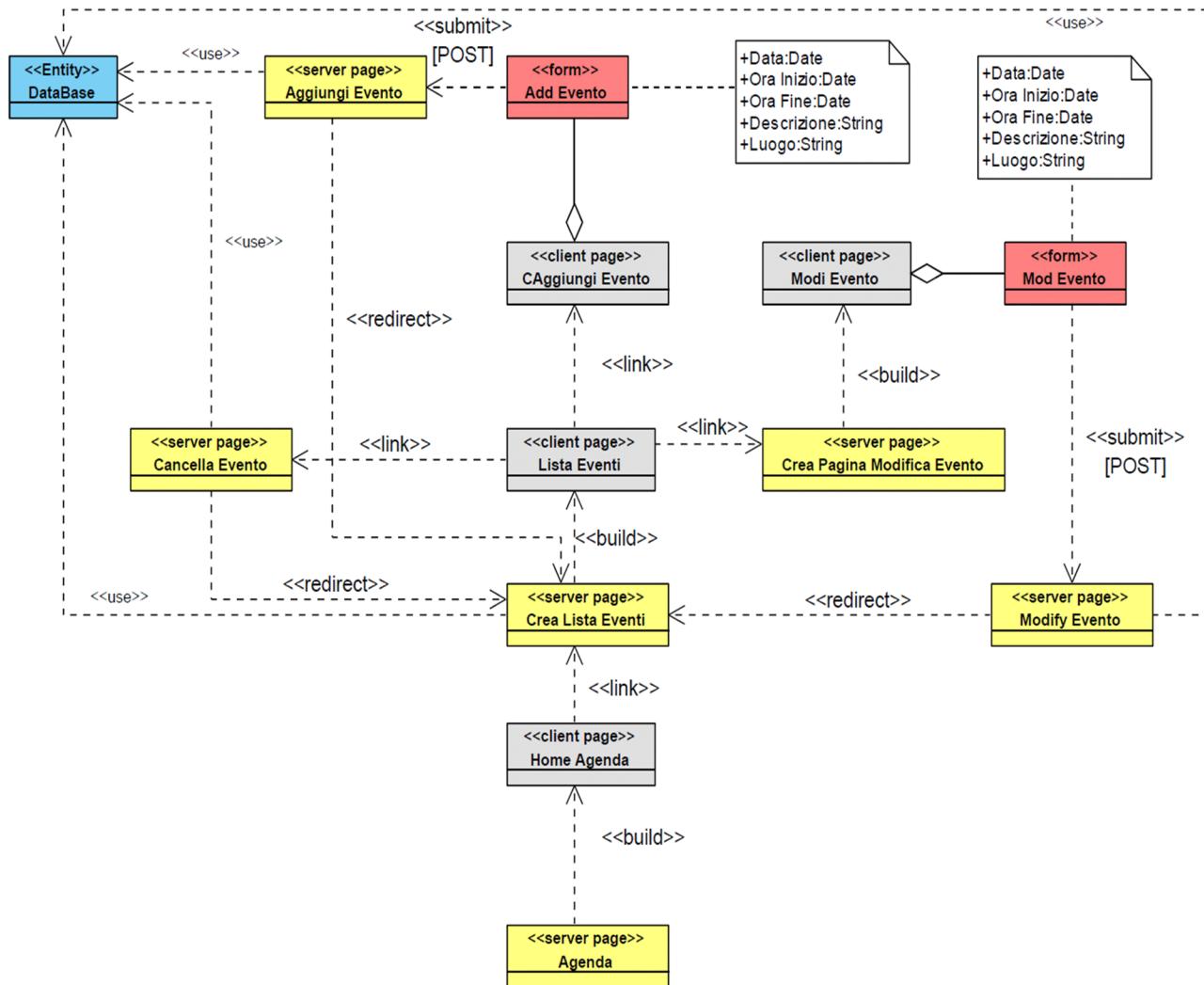


Figura 3 - Sitemap Homepage

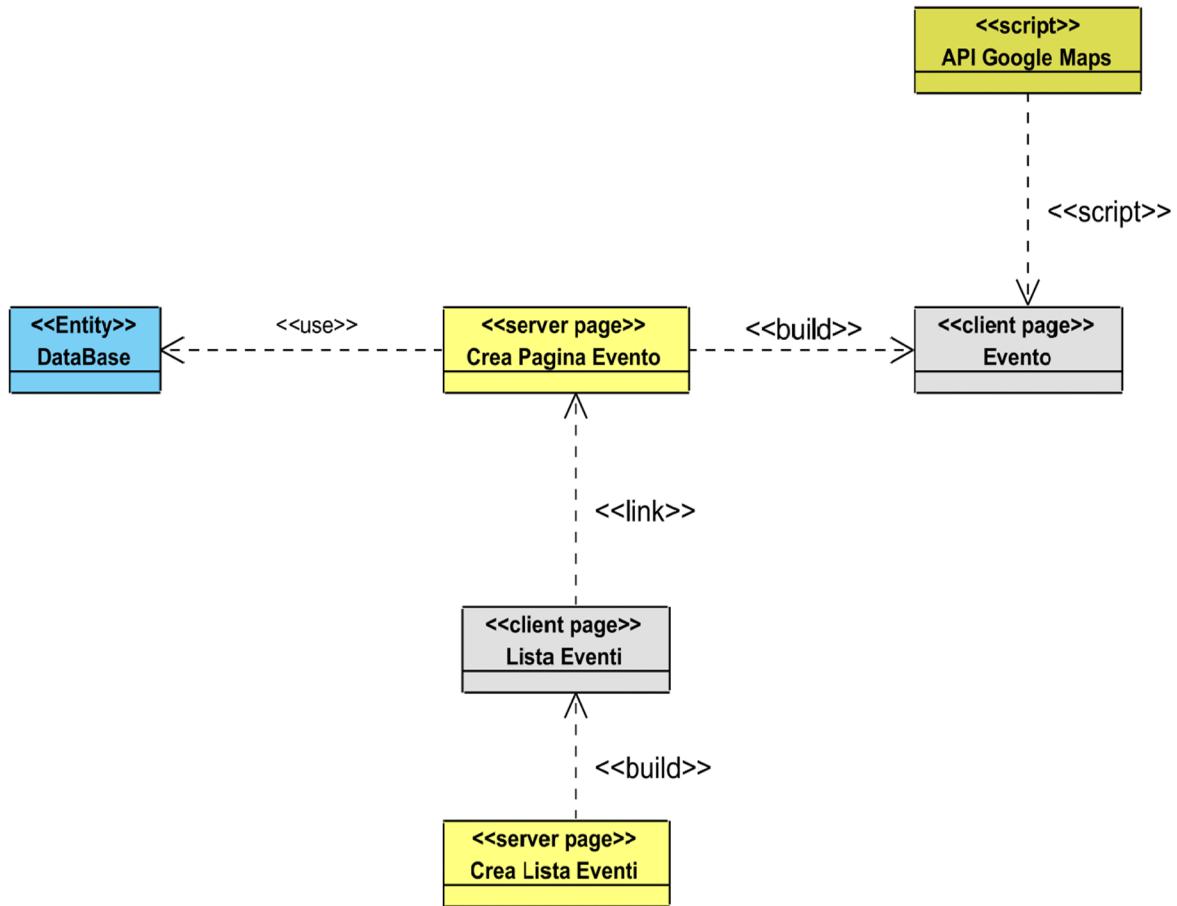


Figura 4 - Sitemap Luogo

Gestione dei Tasks

In questa sezione descriveremo le operazioni che è possibile effettuare per quanto riguarda i tasks in sospeso.

Esse si suddividono in :

- Modifica
- Cancellazione
- Aggiunta

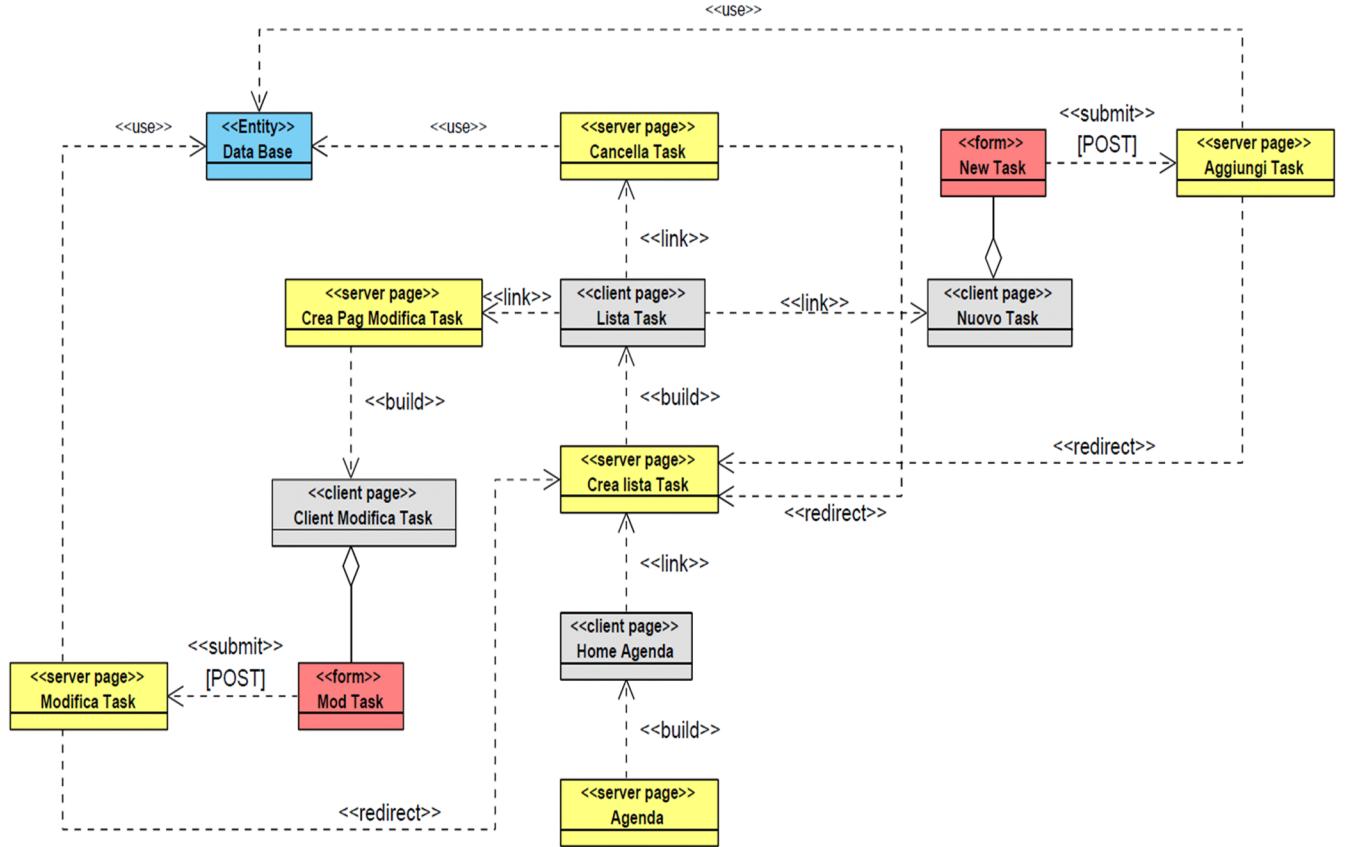


Figura 5 - Sitemap Task

Gestione della Rubrica

In questa sezione descriveremo le operazioni che è possibile sui contatti contenuti nella rubrica :

- Modifica
- Cancellazione
- Aggiunta
- Visualizzazioni

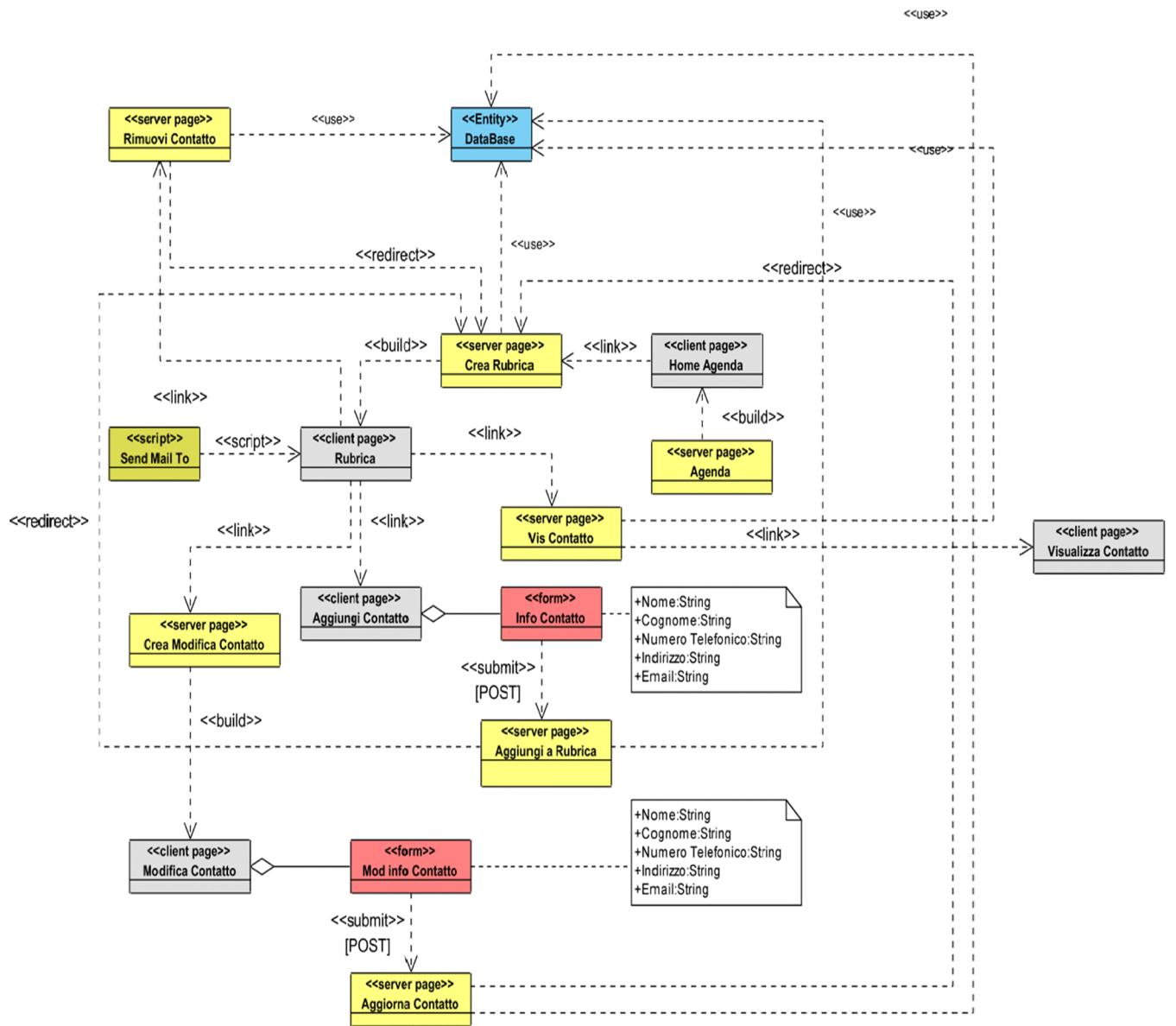


Figura 6 - Sitemap Email

4. Metodi di Session Tracking

La necessità dell'utilizzo di una o più tecniche di session tracking nasce dalla differenza tra le pagine statiche e le pagine dinamiche. Una pagina statica è un documento che è sempre lo stesso, cioè ciascun utente lo visualizza allo stesso modo. Una pagina dinamica, invece, è un documento che cambia a seconda dell'utente che lo visualizza. Un esempio può essere una semplice pagina che

visualizza il nome utente precedentemente inserito in una apposita pagina per il login. È chiaro che tale pagina deve essere necessariamente dinamica, in quanto il nome visualizzato dipenderà da quale utente si collega al sistema.

L'importanza dell'uso di tecniche di session tracking deriva direttamente dalle caratteristiche stesse del protocollo HTTP: tale protocollo è senza stato, cioè non conserva memoria delle richieste già avvenute tra un dato client ed il server. Il session tracking si occupa proprio di questo, cioè dell'implementazione di meccanismi per permettere al server di "riconoscere" il client, servendogli quindi una pagina che rispetta le sue richieste.

Per l'implementazione potranno essere usate varie metodologie:

1. Variabili di sessione
2. Riscrittura dell'URL
3. Campi nascosti nelle form

La tecnica delle variabili di sessione prevede la creazione di un insieme di variabili sul server che sono associate allo specifico utente. Tutte le tecnologie lato server potranno quindi accedere a tali variabili utilizzandone il valore oppure aggiungerne di nuove. Per accedere a tali variabili, il server produce, alla prima richiesta del client, un identificativo della sessione che viene allegato alla risposta HTTP. Una volta ricevuto tale identificativo, esso sarà memorizzato nella cache dello user agent e verrà allegato ad ogni successiva richiesta. A questo punto il server, leggendo tale identificativo potrà accedere alle variabili da esso indicate. Tale tecnica di session tracking potrà essere implementata essenzialmente in due modi: attraverso la tecnica dei cookies oppure, se essi sono disabilitati, attraverso la tecnica della riscrittura dell'URL.

La tecnica della riscrittura dell'URL prevede la modifica dell'URL corrispondente alla risorsa richiesta con l'aggiunta di alcune parti ulteriori. L'esempio tipico è la concatenazione, dopo l'URL della richiesta di parametri da inviare alla risorsa richiesta. Tale tecnica è implementata aggiungendo all'URL un punto interrogativo seguito da delle coppie del tipo "nome = valore" separate da dei simboli &. L'utilizzo di questo metodo è abbastanza rapido quando il numero di parametri da passare è molto basso, mentre diventa più macchinoso quando il numero di parametri aumenta.

La tecnica dei campi nascosti nelle form prevede l'inserimento nei form di campi aggiuntivi, non visibili all'utente, che conservano le informazioni che sono utili al server per identificare il client. Il principale svantaggio di tale tecnica è che quando l'informazione comincia a crescere di dimensione, trasportarla in ogni ciclo richiesta-risposta può diventare molto oneroso

5.Tecnologie Utilizzate

Si elencano in questa sezione delle possibili scelte implementative dei componenti presentati nelle precedenti sitemap.

5.1.Tecnologie lato Server

I meccanismi utilizzati per la programmazione lato server sono essenzialmente due : il meccanismo CGI (Common Gateway Interface) oppure le servlet.

Si è scelto di utilizzare le servlet in quanto esse non sono altro che un' ottimizzazione del meccanismo CGI. Esse ci consentono di caricare la classe in memoria solo alla prima richiesta mentre per le successive viene solo creato un thread che esegue il metodo service() della servelet , inoltre, un altro vantaggio è la possibilità di condividere delle informazioni tra varie istanze della stessa servelet.

5.2.Tecnologie lato Client

Il meccanismo usato per la programmazione lato client è JavaScript, linguaggio di scripting che può essere integrato nel codice HTML con una sintassi molto simile a quella del linguaggio Java.

Per la visualizzazione del luogo di un evento sulla mappa utilizzeremo le API(Application Programming Interface)di Google maps, che forniscono tutte le funzionalità necessarie per la completa gestione di una mappa.

5.3.Base di dati

Per l'implementazione della base di dati utilizzata dagli strumenti lato server si sceglie di utilizzare MySQL un RDBMS (Relational Database Management System) con la caratteristica di essere scaricabile liberamente

5.4.Strumenti di Supporto

Uno strumento essenziale per la realizzazione di un applicazione web è il server web. In questo caso si è scelto di utilizzare Apache per la sua notorietà e la sua elevata integrazione con MySQL. Siccome è previsto l'utilizzo delle servlet si è scelto di utilizzare Tomcat, che è uno strumento di implementazione software per le Java Servelet e le Java Server Page

6.Introduzione

In questo documento verrà esposta l'analisi dell'applicazione implementata. Innanzitutto saranno elencate e descritte adeguatamente tutte le funzionalità oggetto dell'implementazione e, per ognuna di esse, mostreremo la sitemap. Ad esse seguirà una spiegazione dettagliata che metterà in luce tutte le possibili operazioni che l'utente potrà compiere. Dove necessario si metterà in risalto la differenza tra la sitemap proposta e quella implementata. Dopo di che passeremo all'analisi della scelta architettonica effettuata per l'applicazione con successiva discussione sulle tecnologie utilizzate in fase di implementazione. In ultima analisi verranno proposte delle ipotesi per migliorare l'applicazione.

7.Funzionalità da implementare

L'applicazione implementerà le seguenti funzionalità:

- Fase di registrazione
- Fase di login
- Gestione della rubrica (con georeferenziazione)
- Gestione degli eventi (con georeferenziazione)
- Proposta di un luogo di incontro che ottimizzi le distanze percorse dai partecipanti

Per ogni funzionalità verrà mostrata la relativa sitemap mettendo in risalto le eventuali differenze con quella presentata nella prima parte.

Fase di registrazione

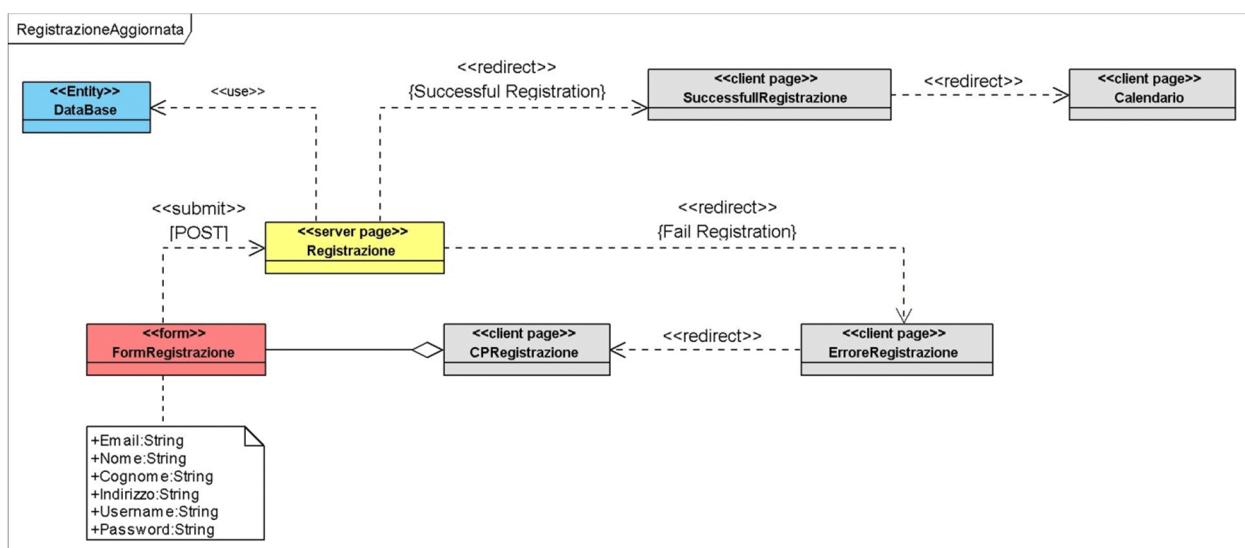


Figura 7 Fase di Registrazione

Per la fase di registrazione l'utente si troverà di fronte ad una pagina contenente il form di registrazione (CPRegistrazione). La pressione del tasto dà inizio alla fase di registrazione inviando una richiesta http alla java server (JSP) page Registrazione. Essa si occupa di prelevare i parametri dalla richiesta e di inserire i dati del nuovo utente nel database. Nella client page di registrazione sono stati inseriti dei controlli di conformità sui campi. Questo tipo di controlli sono effettuati mediante l'utilizzo degli attributi "pattern" e "required" del tag input. L'attributo "pattern" consente di verificare su un campo di tipo testo un'espressione regolare, mentre l'attributo "required" segnala l'obbligatorietà della compilazione del campo. In caso di registrazione fallita (es. username già presente nel database) verrà mostrata una pagina di errore e si verrà reindirizzati sulla client page di registrazione.

Fase di login

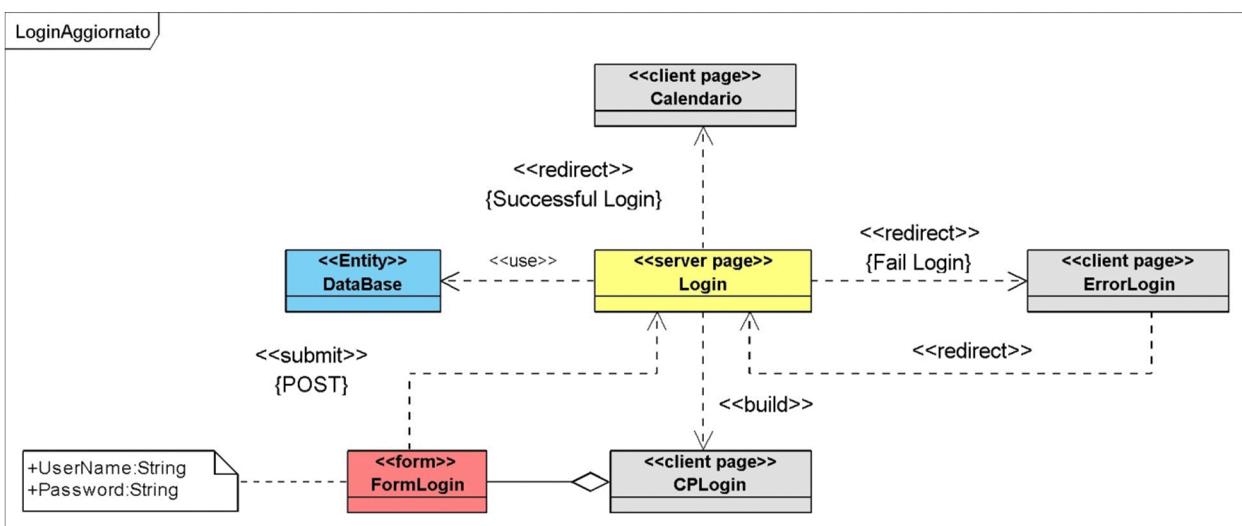


Figura 8 Fase di Login

La fase di login è gestita dalla JSP login. Essa riceve la richiesta http e assume comportamenti diversi a seconda delle situazioni. Se nella richiesta http è stata già istanziata una variabile di sessione significa che il login è già stato effettuato e si viene reindirizzati alla home page utente (calendario). Se nella richiesta non è presente nessuna sessione e mancano i parametri di login viene mostrata la client page di login con relativo form. Se, invece, i parametri sono presenti nella richiesta, viene effettuato un controllo sul database. In caso di dati coerenti viene creata la variabile di sessione e viene mostrata l'home page utente. Nel caso contrario verrà mostrata una client page di errore e si verrà reindirizzati alla client page di login. Anche in questo caso vengono utilizzati gli attributi

“pattern” e “required” del tag input per il controllo relativo ai campi del form.

Differenze con la Sitemap del proposal

Nella precedente sitemap la client page contenente il form di login era una pagina statica con relativo file html, quindi accessibile anche in caso di login già effettuato. La correzione apportata prevede che sia la JSP login a decidere se costruire la pagina contenente il form o reindirizzare, in caso di login già effettuato, sulla home page utente.

Gestione della Rubrica

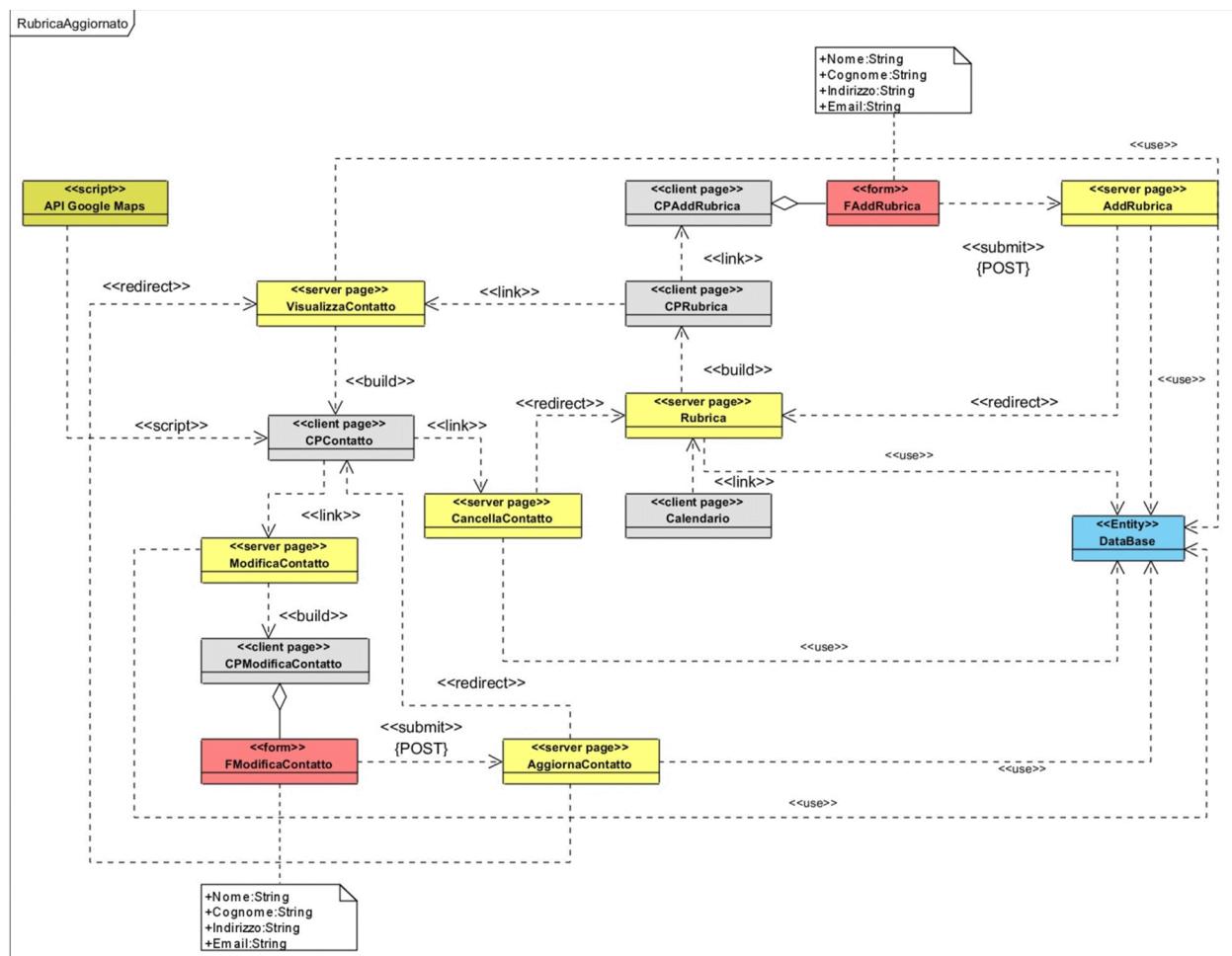


Figura 9 Gestione Rubrica

L'utente dopo aver effettuato il login, ed essere stato reindirizzato alla propria home page avrà a disposizione un link che farà una richiesta http alla JSP Rubrica. Questa si occuperà di ricevere la richiesta, appurare che il login sia

stato effettuato e, in caso affermativo, di prelevare dalla variabile di sessione l'userid settato nella fase di login, infine, dopo un'opportuna interrogazione al database, mostrerà i contatti dell'utente. Da qui sarà possibile aggiungere nuovi contatti utilizzando l'apposito tasto. La JSP che si occuperà di questa funzione è AddRubrica. Essa preleverà dalla richiesta i dati relativi al form sottoscritto in precedenza in una pagina statica, e si occuperà di aggiornare la base di dati. Sarà, inoltre, possibile visualizzare per ogni contatto una pagina che ne elencherà tutte le informazioni, mostrando, attraverso l'utilizzo delle API di Google maps, la georeferenziazione dell'indirizzo. A farsi carico del recupero dei dati e della costruzione della pagina del contatto è la JSP paginacontatto (nella sitemap VisualizzaContatto). Raggiunta la pagina relativa al contatto l'utente avrà la possibilità di cancellarlo o modificarlo. La cancellazione viene affidata alla JSP cancellacontatto (nella sitemap CancellaContatto), che, una volta recuperati i dati dalla richiesta, provvederà ad aggiornare il database. La modifica viene affidata a due JSP modificacontatto (nella sitemap ModificaContatto) e aggiornacontatto (nella sitemap AggiornaContatto). La prima costruirà la client page composta dal form di modifica precompilato con i precedenti dati, mentre alla pagina aggiornacontatto sarà affidato il compito di recuperare i dati sottoscritti nel form precedente e di apportare le modifiche alla base di dati, infine si verrà reindirizzati alla pagina di visualizzazione del contatto.

Differenze con la Sitemap del proposal

La vecchia sitemap prevedeva che la modifica e la cancellazione fossero accessibili già nella client page Rubrica e che, nella stessa pagina, fosse possibile inviare email ai contatti. Nella nuova versione si è preferito, per comodità e chiarezza, che le operazioni di modifica e cancellazione fossero accessibili esclusivamente nella pagina relativa al contatto. Per quanto riguarda la possibilità di inviare email, in seguito alla discussione sul proposal e alle funzionalità assegnato, si è scelto di tralasciare questa funzionalità in quanto l'invio di email avviene già all'interno delle fasi aggiunta modifica e cancellazione di un evento.

Gestione degli eventi

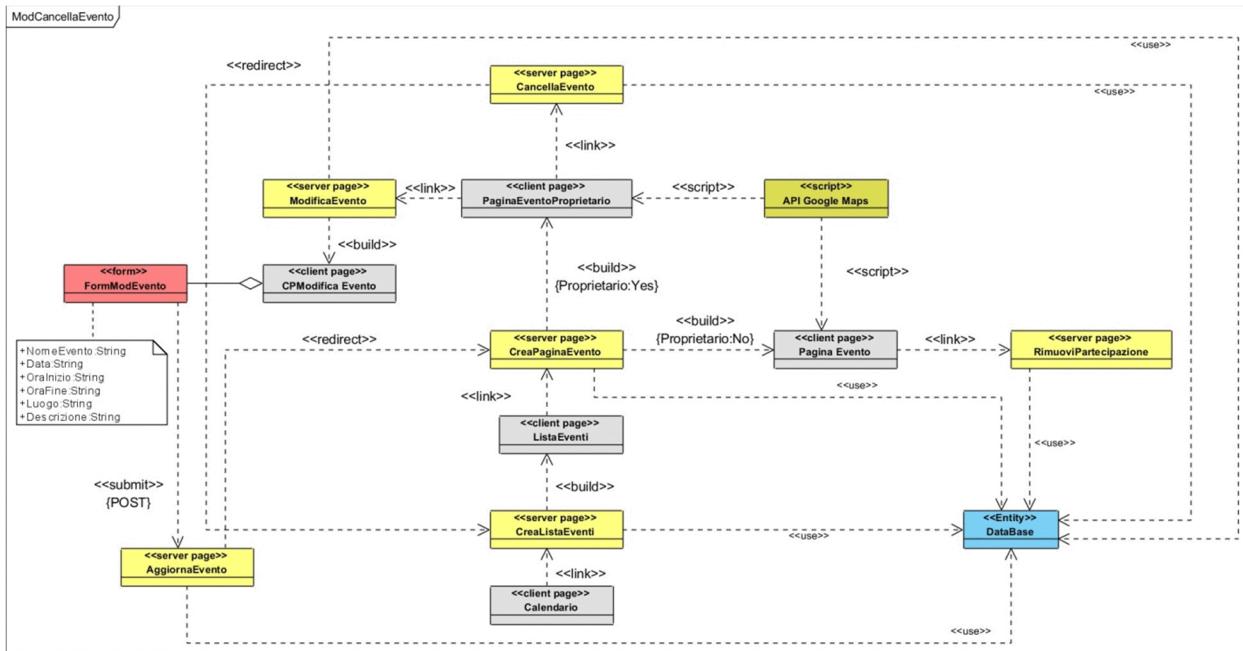


Figura 10 Gestione degli eventi

La client page Calendario è stata progettata per visualizzare tutti i giorni del mese corrente, e per ognuno di questi è possibile, cliccandoci , effettuare una richiesta http alla JSP listaeventi (nella sitemap CreaListaEventi). Essa, prelevata la data e l'userid dalla richiesta, invierà nella risposta la lista degli eventi in programma. Per ogni evento sarà possibile visualizzare una pagina che ne descriverà in dettaglio tutte le informazioni, mostrando su mappa (mediante API di Google maps) il luogo d'incontro. La creazione di questa pagina è affidata alla JSP paginaevento (nella sitemap CreaPaginaEvento), che si occuperà del recupero dei dati e di mostrare all'utente, in base al suo ruolo (creatore dell'evento o semplice partecipante), i tasti relativi alla modifica e alla cancellazione o alla rimozione della propria partecipazione. Le fasi di modifica e cancellazione non differiscono, se non per l'invio di una mail di notifica ai partecipanti, in modo particolare da quelle descritte per la gestione della rubrica, per tanto avremo una JSP che si occuperà della cancellazione ed altre due che si occuperanno della precompilazione del form e dell'aggiornamento dei dati nel database. La fase di rimozione della partecipazione avviene contattando l'apposita server page che si occuperà di rimuovere l'utente dai partecipanti all'evento. La fase di aggiunta di un evento è stata trattata

separatamente poiché includeva lo sviluppo della funzionalità “proposta di un luogo d’incontro”.

Differenze con la Sitemap del proposal

La vecchia sitemap prevedeva solo la visualizzazione della lista degli eventi (senza una pagina specifica che ne visualizzasse le informazioni in modo dettagliato), e da qui sarebbe stato possibile selezionare gli eventi per poi cancellarli o modificarli. Questa soluzione si è rivelata, però, poco pratica e difficilmente realizzabile, in più il fatto di non poter visualizzare i dettagli di un evento o di doverli visualizzare nella lista avrebbe dato luogo ad un risultato poco user friendly. Di conseguenza si è scelto di aggiungere la possibilità di visualizzare una pagina con informazioni dettagliate sull’evento e di aggiungere in quest’ultima le funzionalità di cancellazione e modifica. Si è osservato inoltre che l’utente può assumere due ruoli: creatore dell’evento o semplice partecipante; quindi si è scelto di rendere accessibili le funzionalità di modifica e di cancellazione solo al proprietario dell’evento e la rimozione della partecipazione al semplice partecipante.

Proposta Luogo d'incontro

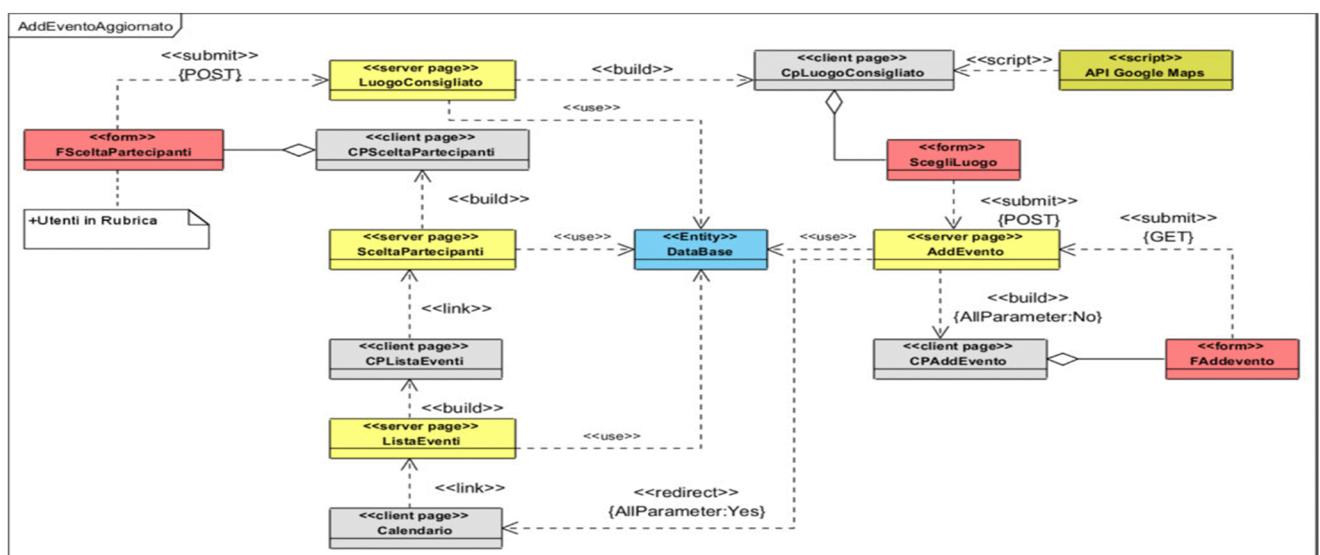


Figura 11 Proposta luogo d'incontro

A partire dalla lista degli eventi sarà data la possibilità all’utente di aggiungere un evento attraverso l’utilizzo del tasto “Aggiungi Evento” che invierà una richiesta alla server page SceltaPartecipanti (implementata nella JSP checkrubrica). Quest’ultima avrà il compito di recuperare i dati dalla rubrica dell’utente e mostrare la lista dei contatti, selezionabili attraverso checkbox.

Una volta selezionati i contatti viene inviata una richiesta alla server page LuogoConsigliato (implementata nella JSP sceltaluogo) che, mediante la formula matematica per il calcolo del baricentro, troverà e indicherà su mappa (mediante la API di Google maps) il punto consigliato come luogo d'incontro. Ovviamente l'utente avrà il libero arbitrio sulla scelta del luogo potendo inserire manualmente l'indirizzo preferito. Dopo di che verrà inviata una richiesta alla server page AddEvento. Essa ha due comportamenti: se nella richiesta sono presenti tutti i parametri di cui ha bisogno, effettua l'inserimento nel database e reindirizza l'utente alla home page (client page calendario), altrimenti risponde visualizzando una pagina contenente il form da compilare per poter aggiungere un evento. L'avvenuta aggiunta dell'evento implica l'invio di una email a tutti i partecipanti contenente una richiesta di partecipazione e un link che permette di confermare la propria presenza all'evento.

Link di partecipazione

Il link di partecipazione contenuto nelle email consente di effettuare una richiesta http alla JSP partecipaevento. La JSP controllerà che l'utente abbia effettuato il login e, in caso affermativo, effettua la registrazione dell'utente per quel determinato evento. Nel caso in cui l'utente non abbia effettuato il login, avrà la possibilità di compilare manualmente il form con i dati necessari per la partecipazione all'evento. Una volta compilato il form viene inviata una nuova richiesta alla JSP partecipaevento che aggiornerà il database confermando la partecipazione con i dati inseriti dall'utente.

Problematica "incontri improbabili"

Dato che il calcolo del luogo consigliato come punto di incontro avviene mediante una formula matematica utilizzando la latitudine e la longitudine, non si è in grado di stabilire a priori se il luogo trovato si trovi in un punto accettabile. Ad esempio abbiamo riscontrato che, nel caso in cui l'utente si trovasse a Bacoli e il partecipante si trovasse a Sorrento, la proposta d'incontro sarebbe nel mar Tirreno.

Differenze con la Sitemap del proposal

La vecchia sitemap prevedeva una pagina statica che contenesse il form da compilare per la creazione dell'evento senza preoccuparsi di permettere all'utente di recuperare le informazioni dei partecipanti dalla propria rubrica. Inoltre non prevedeva l'individuazione di un punto d'incontro ideale. La nuova sitemap, invece, elimina la pagina statica e affida il compito di costruirla ad una

server page (AddEvento), introduce delle nuove server page che permettono all’utente di selezionare dalla rubrica i partecipanti all’evento e di indicare un punto d’incontro ideale utilizzando le API di Google maps.

8. Architettura del sistema

Si è scelto di utilizzare un’architettura a tre livelli per l’implementazione del sistema in oggetto. Il primo livello è rappresentato dalla sorgente dell’informazione, che nel nostro caso sarà il database, il secondo livello che sarà anche quello intermedio, implementa il meccanismo di controllo, il quale si occupa dell’elaborazione delle richieste e di procurare i dati da presentare, il meccanismo di presentazione che elabora i dati recuperati dal meccanismo di controllo rendendoli comprensibili per il formato del client, e la business logic che si occupa del controllo della coerenza dei dati prodotti in base alle regole fornite dal problema in questione. L’ultimo livello è rappresentato dall’interfaccia utente che è user friendly e permette l’interazione con l’applicazione all’utente. L’architettura scelta e descritta implementa il design pattern MVC(Model-View-Controller) dove la parte del Model è rappresentata dalla sorgente dell’informazione, la parte del View è rappresentata dall’interfaccia utente e la parte del controller è rappresentata dagli strumenti utilizzati dal server. Il livello del Model è accessibile solo al livello del controller, mentre dal livello delle viste si può accedere solo al livello del controller.

9. Tecnologie utilizzate

Lato server

Per quanto riguarda il lato server nella proposta di soluzione si era parlato molto di servlets poiché rappresentavano un miglioramento del CGI, ma le Java Server Page (JSP) si sono rivelate la soluzione più comoda e versatile. Quando ad un web server (che supporta le JSP) è richiesta una pagina JSP, esso verifica, innanzitutto, se tale pagina è già stata compilata (viceversa produce il relativo bytecode), quindi carica ed esegue il codice Java della pagina JSP come una qualsiasi altra servlet, producendo in output la pagina HTML da inviare al browser. Il principale vantaggio delle JSP è quello di poter introdurre diversi linguaggi (come HTML e JavaScript) in un unico documento rendendo la programmazione più semplice e intuitiva. Rispetto all’invocazione di servlets pure per la generazione di pagine web dinamiche, si semplifica la stesura e la modifica dell’HTML poiché è possibile separare la sezione di produzione dei

contenuti da quella di visualizzazione vera e propria (aspetto grafico e stilistico del sito). I vantaggi di questo disaccoppiamento, si ripercuotono anche sulla modalità di produzione e manutenzione del sito, permettendo a team di lavoro differenti di lavorare in maniera indipendente ma parallela.

Lato client

Per gli strumenti lato client, come già accennato nella proposta di soluzione, è stato utilizzato strumento di JavaScript. Script di tale linguaggio sono inseriti nelle pagine in due modi differenti: caricando una libreria di funzioni da un file esterno oppure inserendo direttamente nella pagina del codice JavaScript. Questo linguaggio è stato utilizzato principalmente per interfacciarsi alle Google Maps API delle quali, oltre allo strumento della mappa, sono stati usati gli strumenti dei Marker e del Geocoder.

Database

Come già detto nella proposta di soluzione, il DBMS utilizzato è MySql. La base di dati rappresenta una parte fondamentale del sistema infatti, come è possibile notare dalle pagine precedenti, è soggetto a continue interrogazioni. La struttura logica del database è la seguente:

Evento (Id, Nome, OraInizio, OraFine, Data, Luogo, Descrizione, Proprietario)
Utente (IdUtente, Nome, Cognome, Password, Email, Indirizzo)
PartecipaEvento (IdEvento, Nome, Cognome, IdUtente, Email)
Rubrica (Proprietario, Nome, Cognome, Indirizzo, Email, Telefono)

Nella tabella Evento si terrà nota di tutti gli eventi creati con le relative informazioni.

Nella tabella Utente saranno presenti le informazioni degli utenti registrati necessarie anche per l'autenticazione.

Nella tabella PartecipaEvento si terrà nota, per ogni evento, di tutti i suoi partecipanti.

Nella tabella Rubrica saranno registrati tutti i contatti relativi ad ogni utente.

10.Tecnologie di Session Tracking

Lo strumento di session tracking principalmente utilizzato è quello delle variabili di sessione. L'oggetto **Session** ci permette di memorizzare valori per la sessione corrente. Quindi, memorizzando un determinato valore nell'oggetto, sarà rintracciabile per tutta la durata della sessione. Una volta che l'utente sarà uscito dal nostro sito, queste informazioni verranno perse (con l'oggetto **Session**), ma per tutta la navigazione all'interno del nostro sito, saranno disponibili. In pratica, l'oggetto **Session** ha la funzionalità di un cookie di memorizzare informazioni durante la navigazione del sito, ma non è possibile utilizzarlo nel lungo periodo (non è possibile impostarne una scadenza lunga, solo alla fine della sessione esso viene scaricato). Attenzione però! Questo oggetto è disponibile solo sui browser che supportano i cookies. Nel nostro caso la variabile di sessione viene istanziata al momento del login tramite il metodo `getSession()` e, subito dopo, viene memorizzato al suo interno il valore `userid` che conterrà l'identificativo dell'utente. Tale valore sarà sfruttato dalle altre JSP per verificare l'avvenuta autenticazione dell'utente nel sistema. Nel caso in cui l'utente non risulti autenticato il sistema provvederà a reindirizzarlo sulla pagina di login.

Un'altra tecnica utilizzata è la riscrittura dell'URL che prevede la modifica dell'URL corrispondente alla risorsa richiesta con l'aggiunta di alcune parti ulteriori. L'esempio tipico è la concatenazione, dopo l'URL della richiesta di parametri da inviare alla risorsa richiesta. Tale tecnica è implementata aggiungendo all'URL un punto interrogativo seguito da delle coppie del tipo “nome = valore” separate da dei simboli &. Un esempio di questa tecnica si riscontra al momento della creazione della lista degli eventi, infatti, l'URL del link che contatta la JSP `listaeventi` viene riscritto concatenando giorno mese e anno, così da poter recuperare i giusti dati.

L'ultima tecnica utilizzata è quella dei campi nascosti nelle form che prevede l'inserimento nei form di campi aggiuntivi, non visibili all'utente, per memorizzare informazioni utili al server. È possibile riscontrare un esempio di questa tecnica nella pagine di selezione dei partecipanti ad un evento, infatti il form include alcuni campi nascosti in cui sono memorizzati giorno mese anno e il numero di contatti . Grazie a questo metodo è stato quindi possibile effettuare uno scambio di dati tra due diverse pagine in completa trasparenza, contrariamente a come avviene nel metodo della riscrittura dell'URL.

11. Ipotesi aggiuntive

Un'analisi approfondita dell'applicazione mette in luce alcune problematiche. Innanzitutto, ipotizzando un utilizzo dell'applicazione su rete geografica, potrebbero sorgere problemi dovuti al sovrappiattamento di connessioni al database. Ciò avviene perché la connessione al database avviene ad ogni chiamata del metodo `service()` delle servlets (derivanti dalle JSP), quindi ad ogni richiesta su quella servlet. Una possibile soluzione potrebbe essere quella di ridefinire il metodo `init()` prevedendo la connessione al database. Così facendo ad ogni chiamata del metodo `service()` verrebbe utilizzata la medesima connessione.

Un'altra problematica riscontrata è rappresentata dal fatto che effettuando il calcolo del luogo consigliato come punto di incontro tramite una formula matematica che utilizza la latitudine e la longitudine di più luoghi non si è in grado di stabilire a priori se il luogo trovato si trovi in un punto accettabile. Una possibile soluzione potrebbe essere quella di definire dei poligoni (mediante le API di Google maps) escludendo le zone irraggiungibili e prevedendo che, nel caso in cui la formula restituisca coordinate non incluse nei poligoni, venga selezionato come punto d'incontro un punto appartenente ad un poligono con distanza minima dalla posizione calcolata.

Al momento non è prevista la gestione conflitti tra eventi, cioè la gestione di eventi che si tengono nello stesso arco temporale. Una soluzione potrebbe essere quella di avvertire l'utente, in fase di inserimento, modifica o partecipazione ad un evento, di un possibile conflitto visualizzando un messaggio in cui l'utente possa scegliere se partecipare o rinunciare all'evento.

Attualmente l'applicazione non dispone di fogli di stile o di un layout che la renderebbero più utilizzabile. In particolar modo sarebbe utile la presenza di un tasto Home e di un form di ricerca in ogni pagina. Un ulteriore miglioria potrebbe essere l'utilizzo di fogli di stile diversi a seconda del tipo di periferica che effettua le richieste http.

Un altro problema risiede nel fatto che tutta la logica di funzionamento dell'`HttpSession` usata nell'applicazione parte dal presupposto che si possa fare affidamento sulla tecnica dei cookie. Sappiamo invece che lato client questa non è una certezza perché è sempre possibile istruire, per diverse ragioni, un browser internet a disabilitare i cookie. Un'applicazione costruita con questo presupposto è quindi poco robusta perché non può funzionare in questa "sciagurata" eventualità. Fortunatamente è possibile predisporre la nostra applicazione affinché funzioni anche con un browser che ha i cookie

disabilitati. Basta utilizzare la tecnica dell'URL Rewriting o "riscrittura dell'URL".

Si tratta di aggiungere ad ogni link contenuto nelle pagine della nostra applicazione l'informazione relativa all'identificativo di sessione. Questa tecnica agisce come backup della prima; in questo modo il server può sempre risalire all'identificativo della sessione associata alla richiesta in quanto questo viene trasmesso accodato a ciascuna richiesta dell'applicazione stessa. Per implementare la tecnica del URL rewriting occorre fare quello che si definisce l'encoding del URL. Esiste il metodo

```
public String encodeURL(String url)
```

della javax.servlet.HttpServletResponse che assolve a questo compito; riceve in input un URL e lo restituisce dopo aver effettuato l'encoding. In questo modo si è garantiti del funzionamento dell'applicazione anche nel caso in cui i cookie siano disabilitati.

