

## **Coordinamento eventi**

**Titolo del progetto:** Coordinamento eventi  
**Alunno/a:** Nemanja Stojanovic  
**Classe:** I4AA  
**Anno scolastico:** 2019/20  
**Docente responsabile:** Massimo Sartori

1	Introduzione .....	3
1.1	Informazioni sul progetto .....	3
1.2	Abstract .....	3
1.3	Scopo .....	3
	Analisi .....	4
1.4	Analisi del dominio .....	4
1.5	Analisi e specifica dei requisiti .....	4
1.6	Use case .....	7
1.7	Pianificazione .....	1
1.8	Analisi dei mezzi .....	1
1.8.1	Software .....	1
1.8.2	Hardware .....	1
2	Progettazione .....	2
2.1	Design dei dati e database .....	2
2.2	Design delle interfacce .....	3
2.3	Tipo di dati .....	7
2.4	Design procedurale .....	1
3	Implementazione .....	1
3.1	Grafica .....	1
3.2	Database .....	1
3.3	PHP .....	3
4	Test .....	8
4.1	Protocollo di test .....	8
4.2	Risultati test .....	10
4.3	Mancanze/limitazioni conosciute .....	11
5	Consuntivo .....	1
6	Conclusioni .....	1
6.1	Sviluppi futuri .....	1
6.2	Considerazioni personali .....	1
7	Bibliografia .....	1
7.1	Sitografia .....	1
8	Allegati .....	2

## 1 Introduzione

---

### 1.1 Informazioni sul progetto

**Titolo del progetto:** Coordinamento eventi  
**Alunno/a:** Nemanja Stojanovic  
**Classe:** I4AA  
**Anno scolastico:** 2019/20  
**Docente responsabile:** Massimo Sartori  
**Durata del progetto:** Dal 03.09.2019 al 20.12.2019

### 1.2 Abstract

*The site will handle events. The events will be public and will be visible both on a calendar and on a text list. The various associations will have the possibility to insert their own events and to visualize those already inserted in a specific place. This project lets you know who and when you set a event/meeting/event.*

*It is possible to search for events for any field. You can also print events and any research on them. For this project I used PHP and MySQL to manage the database.*

### 1.3 Scopo

Lo scopo del progetto è quello di creare un sito web e di prepararsi all'esame finale. Il sito si occuperà di gestire degli eventi. Gli eventi saranno pubblici e saranno visibili sia su calendario che su una lista testuale. Le varie associazioni avranno la possibilità di inserire i propri eventi e di visualizzare quelli già inseriti di un determinato luogo.

## Analisi

### 1.4 Analisi del dominio

Questo applicativo online ha l'obiettivo di rendere più facile e veloce la gestione degli eventi di un determinato luogo. Il sito sarà user-friendly poiché verrà utilizzato da persone che non fanno parte del mondo dell'informatica e quindi non dovranno esserci impostazioni speciali che potranno complicare loro l'utilizzo. Ogni ente che vorrà inserire gli eventi dovrà registrarsi al sito e l'amministratore dovrà accettare o rifiutare la richiesta di registrazione. Comunque esistono degli applicativi simili a questo.

### 1.5 Analisi e specifica dei requisiti

<b>ID</b>	<b>REQ-001</b>
<b>Nome</b>	<b>Pagina iniziale</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Si tratta di una applicazione WEB.
<b>002</b>	Ci dovrà essere una pagina iniziale.
<b>003</b>	Deve esserci un "tasto" "Login" nel menu che porta al form di login.

<b>ID</b>	<b>REQ-002</b>
<b>Nome</b>	<b>Amministratore</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Dovrà poter accedere al sito.
<b>002</b>	Potrà gestire le tipologie di evento.
<b>003</b>	Potrà gestire gli enti (dovrà anche poterli confermare).
<b>004</b>	Potrà gestire i luoghi.
<b>005</b>	Dovrà poter disporre della lista di eventi in formato testuale.
<b>006</b>	Dovrà poter disporre della lista di eventi sotto forma di calendario.
<b>007</b>	Dovrà poter eseguire delle ricerche su tutti i campi utili degli eventi.
<b>008</b>	Dovrà poter eseguire delle ricerche su tutti i campi utili dei luoghi/enti.
<b>009</b>	Dovrà poter stampare il risultato delle ricerche.
<b>010</b>	Dovrà disporre della statistica del numero degli eventi totali.
<b>011</b>	Dovrà disporre della statistica degli eventi totali per tipologia.
<b>012</b>	Dovrà disporre della statistica del numero degli eventi per ente.
<b>013</b>	Dovrà esserci un grafico che mostri il numero di eventi per mese/anno.
<b>014</b>	Dovrà essere filtrabile in base al luogo.

<b>ID</b>	<b>REQ-003</b>
<b>Nome</b>	<b>Enti</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Dovrà potersi registrare nel sito.
<b>002</b>	Per poter gestire gli eventi dovranno ricevere l'approvazione da parte dell'amministratore.
<b>003</b>	La registrazione sarà da fare in un determinato luogo.
<b>004</b>	Se il luogo non fosse ancora presente sarà necessario fare una richiesta all'amministratore.
<b>005</b>	Dovrà poter visualizzare un elenco degli eventi/riunioni/manifestazioni già inseriti.
<b>006</b>	Ci dovrà essere l'elenco testuale.
<b>007</b>	Ci dovrà essere l'elenco su calendario.
<b>008</b>	Ogni ente potrà inserire/cancellare/modificare i propri eventi.
<b>009</b>	Per un evento verrà inserito il titolo, il tipo di evento, l'ora di inizio e fine previsti ed eventuali osservazioni.
<b>010</b>	Dovrà poter stampare tutti gli eventi per il luogo che è registrato.

<b>ID</b>	<b>REQ-004</b>
<b>Nome</b>	<b>Visitatori</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Non dovranno registrarsi.
<b>002</b>	Devono poter scegliere un luogo per visualizzare gli eventi.
<b>003</b>	Potranno visualizzare gli eventi in modo testuale.
<b>004</b>	Potranno visualizzare gli eventi tramite calendario.
<b>005</b>	Dovranno poter ricercare per qualsiasi campo utile gli eventi (anche filtrando per ente).

<b>ID</b>	<b>REQ-005</b>
<b>Nome</b>	<b>Gestione registrazione</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Deve esserci un form per la registrazione di un nuovo ente.
<b>002</b>	Deve esserci un campo per il nome dell'ente di tipo testo.
<b>003</b>	Deve esserci un campo per l'email dell'ente di tipo testo.
<b>004</b>	Deve esserci un campo per la password di tipo testo codificato.
<b>005</b>	Deve esserci un campo per il luogo di tipo Select.

<b>ID</b>	<b>REQ-006</b>
<b>Nome</b>	<b>Gestione login e logout</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

<b>001</b>	Deve essere un form per il login di un nuovo ente.
<b>002</b>	Deve esserci un campo per l'email dell'ente/amministratore
<b>003</b>	Deve esserci un campo per la password di tipo testo codificato.

<b>ID</b>	<b>REQ-007</b>
<b>Nome</b>	<b>Inserimento di un nuovo evento</b>
<b>Priorità</b>	<b>1</b>
<b>Versione</b>	<b>1.0</b>
<b>Note</b>	<b>-</b>
<b>Sub-ID</b>	<b>Requisito</b>

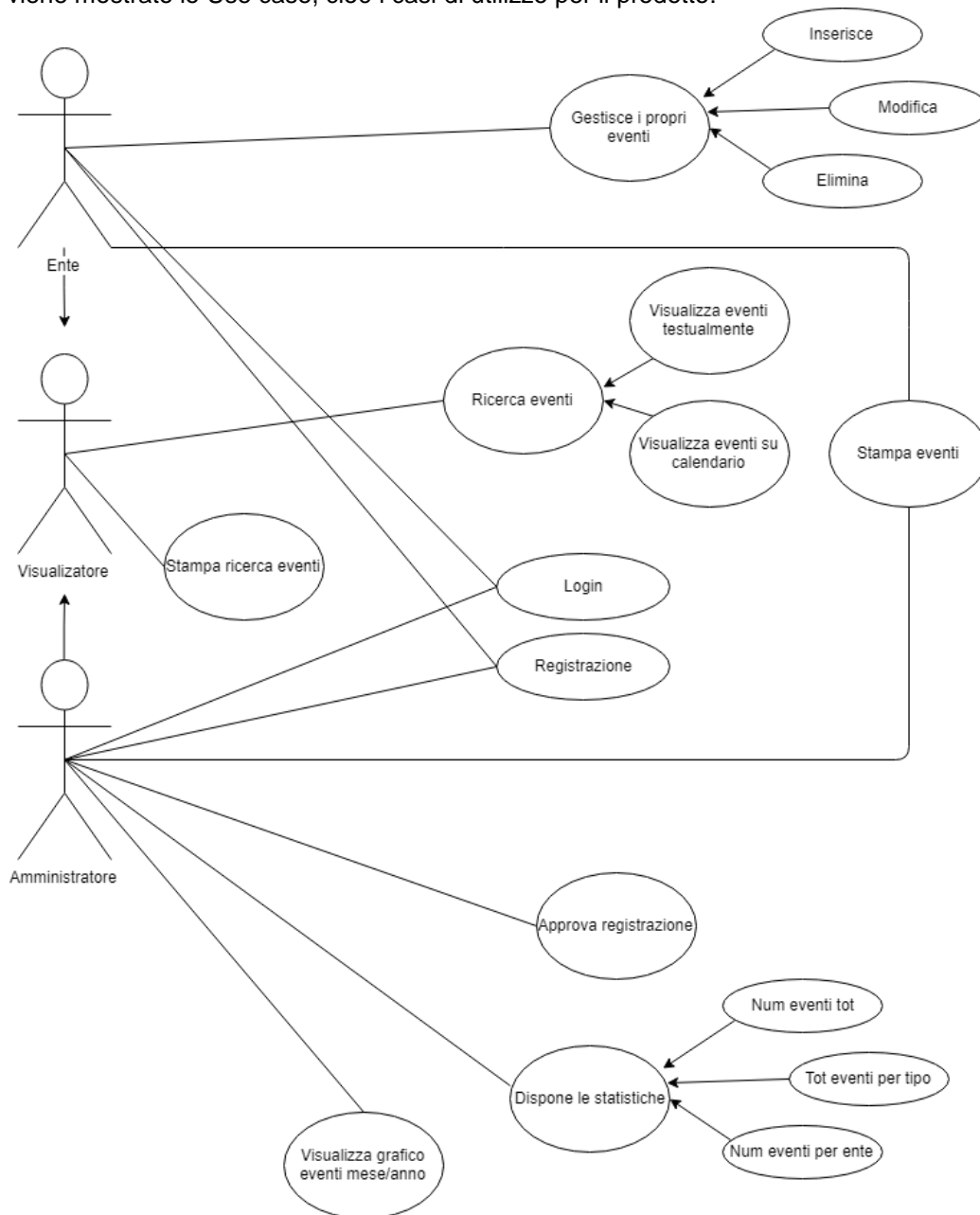
<b>001</b>	Deve essere un form per la registrazione di nuovi eventi.
<b>002</b>	Deve esserci un campo per il titolo di tipo testo.
<b>003</b>	Deve esserci un campo per il tipo di evento di tipo testo.
<b>004</b>	Deve esserci un campo per l'ora di inizio.
<b>005</b>	Deve esserci un campo per l'ora di fine.
<b>006</b>	Deve esserci un campo per le eventuali osservazioni.

## Costi

Categoria	Dettaglio	Costo
Impiegato	62 CHF/h (174 ore)	10'788

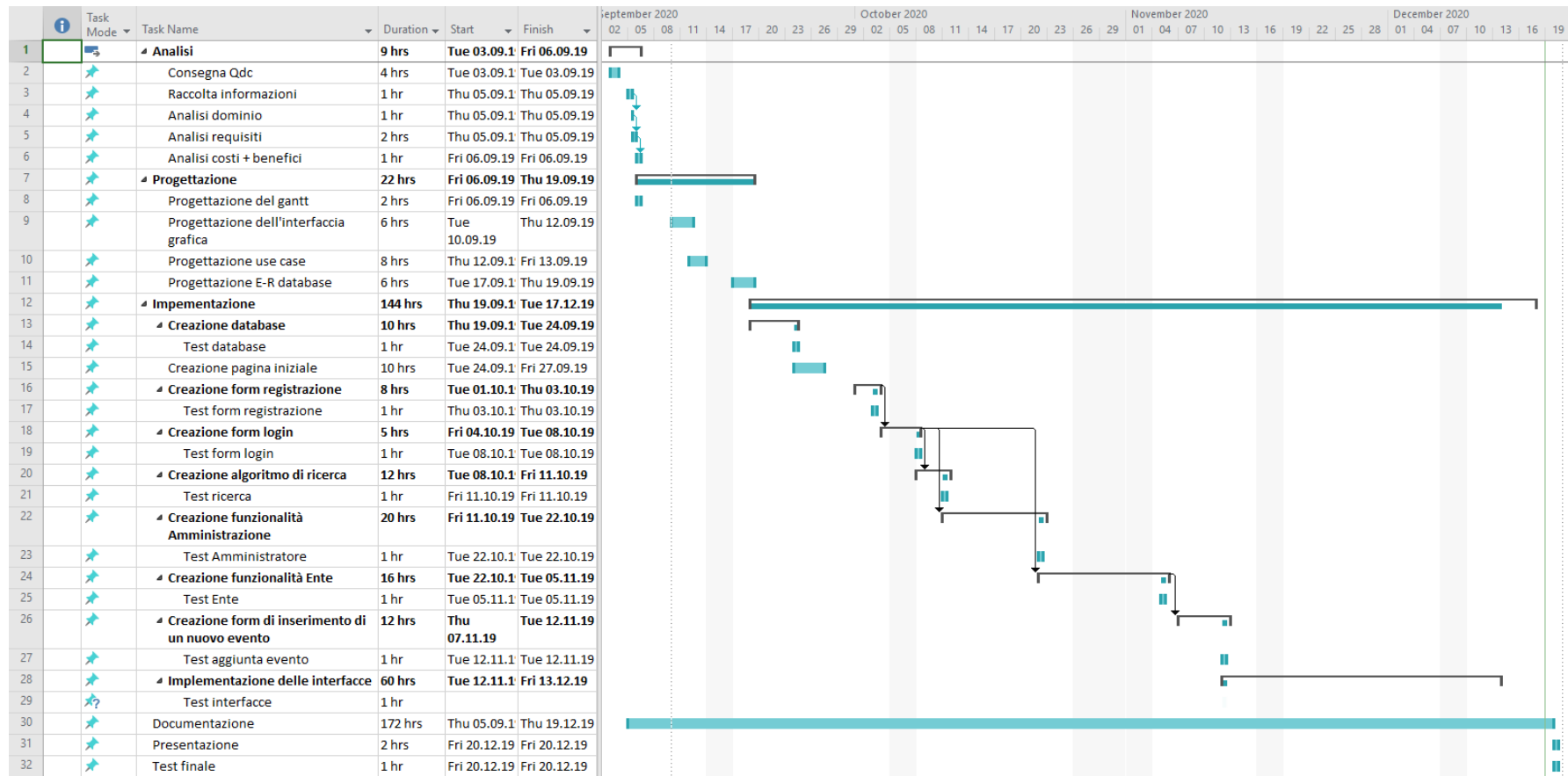
## 1.6 Use case

In seguito viene mostrato lo Use case, cioè i casi di utilizzo per il prodotto:



Lo schema contiene tre attori: il visualizzatore, l'amministratore e l'ente. Gli ultimi due fanno parte dei visualizzatori. Per quanto riguarda gli amministratori e gli enti è necessario che si registrino. L'amministratore può scegliere se accettare oppure no le richieste di registrazione. L'amministratore potrà svolgere molte attività di gestione, tra cui modificare, inserire e eliminare enti/luoghi/eventi. Avrà anche la capacità di visualizzare dei grafici e delle statistiche. L'ente oltreché a fare la registrazione e il login, potrà gestire i propri eventi oppure inserirne dei nuovi (legati sempre alla stessa città). Il visualizzatore, colui che può fare di meno tra i tre e non avrà bisogno di una registrazione. Potrà visualizzatore tutti gli eventi e farne una ricerca. Tutti e tre gli utenti possono stampare gli eventi e l'eventuale ricerca.

## 1.7 Pianificazione



Il gantt è diviso in 3 fasi (Analisi, Progettazione, Implementazione dove sono presenti anche i test). Altri due attività principali sono la documentazione, la presentazione e il Test finale, che hanno tanta importanza per il progetto.



## **1.8 Analisi dei mezzi**

Per la realizzazione di questo progetto si ha bisogno di un PC con il sistema operativo Windows 10 e il tool XAMPP installati.

### **1.8.1 Software**

I programmi utilizzati per svolgere questo progetto sono:

- Draw.io, per i disegni degli schemi e progettazioni
- Sublime Text, per lo sviluppo web
- XAMPP, per il webserver
- Word, per la documentazione
- PowerPoint, per la presentazione

### **1.8.2 Hardware**

Il PC su cui è stato sviluppato il progetto è un portatile con il sistema operativo Windows 10.

## 2 Progettazione

### 2.1 Design dei dati e database

Il database è relativamente semplice, composto da 3 tabelle (evento, utente, luogo).

Se l'utente è amministratore oppure ente viene specificato dall'attributo "granted" che fa parte della tabella utente. Se "granted" è 'ente' allora l'utente può fare accesso alla pagina dell'ente e alle varie attività a lui concesse, mentre se "granted" è 'amministratore' allora l'utente potrà fare accesso alla pagina dell'amministratore e alle varie attività a lui concesse da svolgere.

La tabella evento contiene anche gli id dell'utente e del luogo per poter comunicare tra le tabelle e per sapere quale evento è associato ad un determinato luogo e utente.

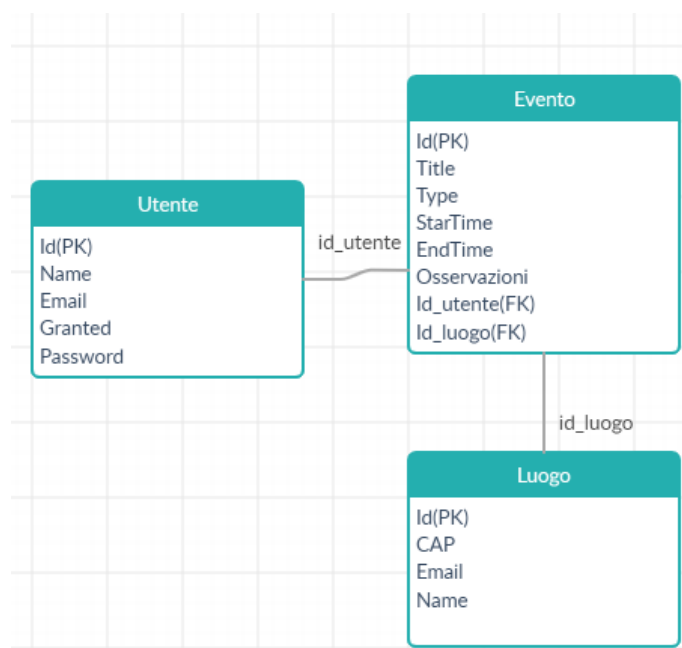


Figura 1: Design database Coordinamento eventi

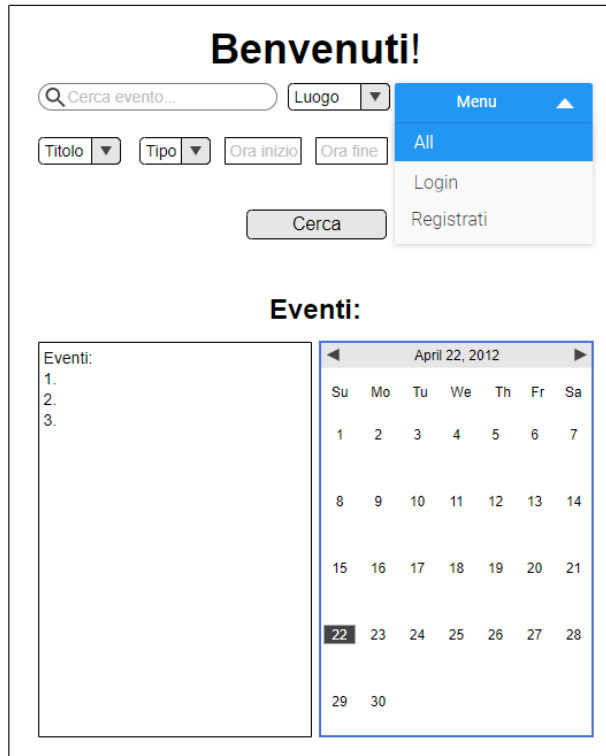
La tabella **utente** contiene le informazioni relative agli utenti registrati al sito quali id, nome, email, grado di privilegi e password.

La tabella **luogo** contiene le informazioni relative ai luoghi registrati al sito quali id, nome, email dell'utente (che fa la richiesta), CAP.

La tabella **evento** contiene le informazioni relative agli eventi inseriti nel sito quali id, titolo, tipo di evento, ora di inizio, ora di fine, le eventuali osservazioni, id dell'ente che l'ha aggiunto e l'd del luogo dell'evento.

## 2.2 Design delle interfacce

In questo capitolo sono mostrate delle anteprime delle varie interfacce delle pagine web.



The screenshot shows a web interface for event management. At the top, it says "Benvenuti!". Below this is a search bar with the placeholder "Cerca evento...". To the right of the search bar is a dropdown menu labeled "Luogo". Further right is a blue button labeled "Menu" with an upward arrow. Below the search bar are four filters: "Titolo" (dropdown), "Tipo" (dropdown), "Ora inizio" (text input), and "Ora fine" (text input). Below these filters is a "Cerca" button. To the right of the search bar is a dropdown menu with options "All", "Login", and "Registrati". Below the search bar is a section titled "Eventi:". To the left of the calendar is a list of events, numbered 1, 2, and 3. To the right of the calendar is a calendar for April 22, 2012, showing the days of the week (Su, Mo, Tu, We, Th, Fr, Sa) and the dates (1-30). The date 22 is highlighted.

Figura 2: Pagina principale (Visualizzatore)

## Benvenuto!

Luogo ▼

Menu ▲

All

Logout

**Eventi:**

Eventi:

- 1.
- 2.
- 3.

April 22, 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figura 3: Pagina dell'ente

## Benvenuto!

Luogo ▼

Menu ▲

All

Statistiche

Logout

**Eventi:**

Eventi:

- 1.
- 2.
- 3.

April 22, 2012

Su	Mo	Tu	We	Th	Fr	Sa
1	2	3	4	5	6	7
8	9	10	11	12	13	14
15	16	17	18	19	20	21
22	23	24	25	26	27	28
29	30					

Figura 4: Pagina principale dell'amministratore

### Statistiche e grafici

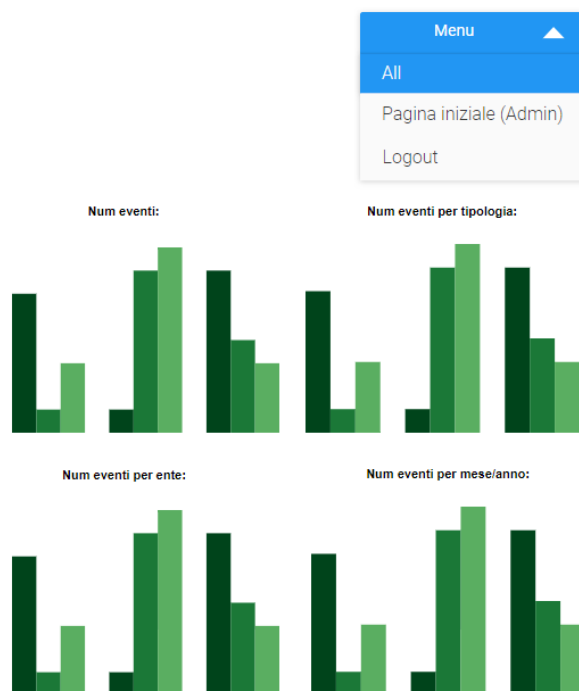


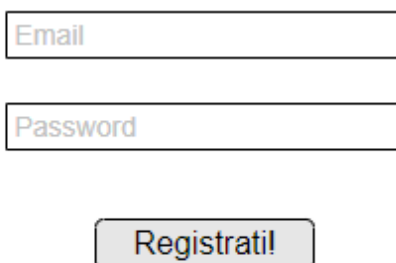
Figura 5: Pagina delle statistiche per l'amministratore

## Form di inserimento eventi

<input type="text" value="Titolo"/>	<input type="text" value="Tipo di evento"/>
<input type="text" value="Ora di inizio"/>	<input type="text" value="Ora di fine"/>
<input type="text" value="Osservazioni"/>	
<input type="button" value="Registrali!"/>	

Figura 6: Form di inserimento eventi

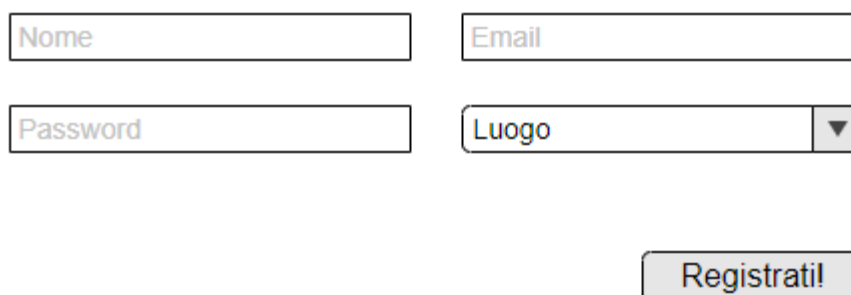
## Form di Login



The login form consists of two text input fields stacked vertically. The top field is labeled 'Email' and the bottom field is labeled 'Password'. Below these fields is a button labeled 'Registrati!'.

Figura 7: Form di Login

## Form di registrazione



The registration form consists of four input fields arranged in a 2x2 grid. The top-left field is labeled 'Nome', the top-right field is labeled 'Email', the bottom-left field is labeled 'Password', and the bottom-right field is labeled 'Luogo' with a dropdown arrow on its right side. Below these fields is a button labeled 'Registrati!'.

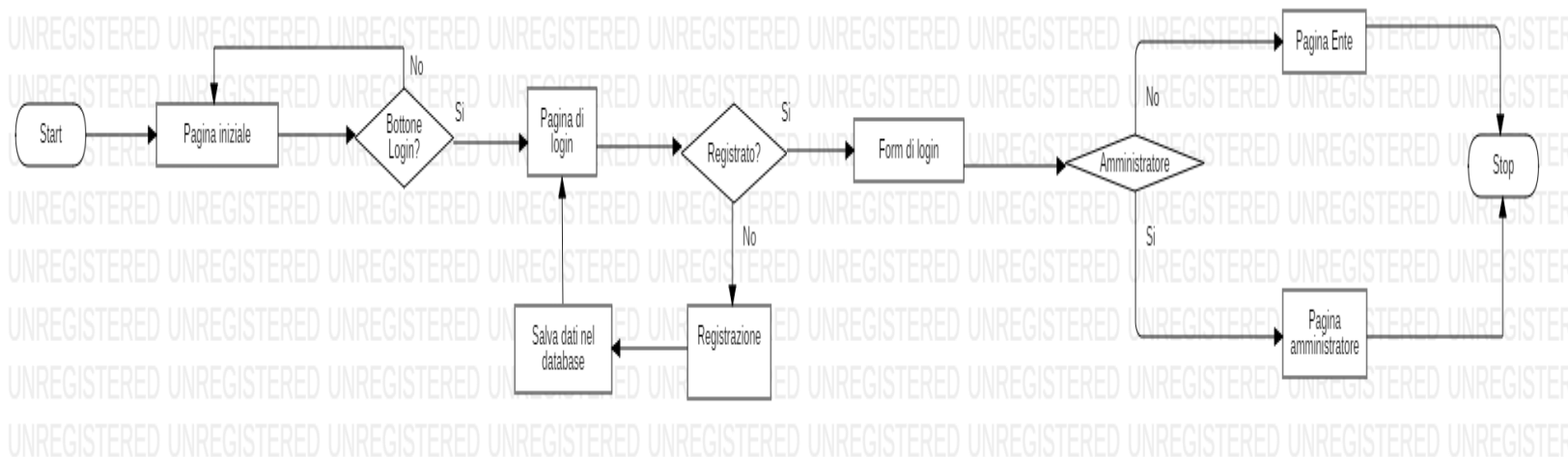
Figura 8: Form di registrazione enti

## 2.3 Tipo di dati

Nome campo	Tipo di dato
<b>Ente o amministratore:</b>	
Nome dell'ente o amministratore	Text
E-mail dell'ente o amministratore	Text
Password dell'ente o amministratore	Password
Luogo dell'ente	Text
<b>Evento:</b>	
Titolo evento	Text
Tipo di evento	Text
Ora di inizio evento	Datetime
Ora di fine evento	Datetime
Osservazioni evento	Text
<b>Luogo:</b>	
Nome luogo	Text
E-mail ente o amministratore	Text
CAP	Text

Al momento dell'inserimento dei dati per la modifica o per l'inserimento degli enti/eventi/luoghi vengono usati i soprastanti tipi di dati per quando riguarda i form di inserimento.

## 2.4 Design procedurale



Ho creato un diagramma di flussi in modo da potermi basare su di esso per non dimenticare dei passaggi che devono essere svolti. Inizialmente viene avviato la pagina iniziale, quella del visualizzatore. In seguito, se viene fatta la richiesta di login, viene controllato che l'utente esista. In caso contrario bisogna fare la registrazione, la quale salva le credenziali di accesso su un database. Se l'utente è già registrato viene fatto un controllo se esso è un ente o un amministratore e così viene indirizzato alla pagina corrispondente.



### 3 Implementazione

#### 3.1 Grafica

Ho iniziato l'implementazione con la creazione delle interfacce grafiche, cioè la parte legata all'html e al css. Tutta la parte di css si trova all'interno del file style.css e definisce lo stile delle pagine web. Ho utilizzato una grafica abbastanza basilare, perché mi volevo focalizzare più all'efficacia del prodotto che alla grafica.

#### 3.2 Database

Finite le interfacce grafiche mi sono spostato sulla parte pratica, iniziando a creare il database sempre dallo schema che avevo fatto precedentemente, l'ho dovuto ristrutturare un paio di volte.

Il file *database.sql* se eseguito creerà il database usato in locale.

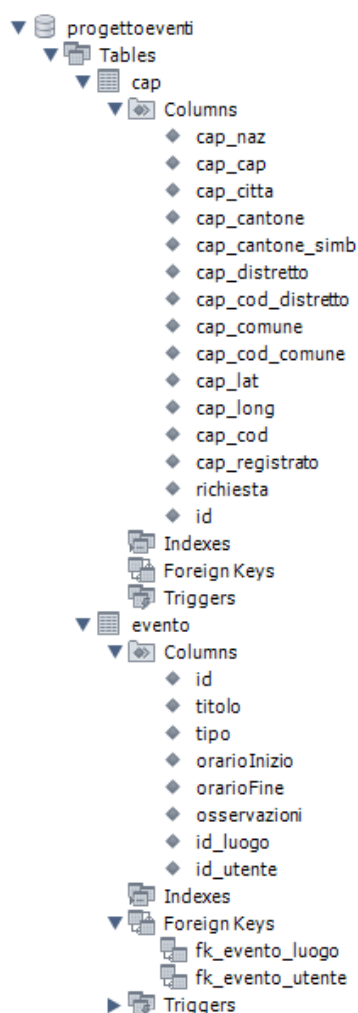


Figura 9: tabelle "CAP", "evento"

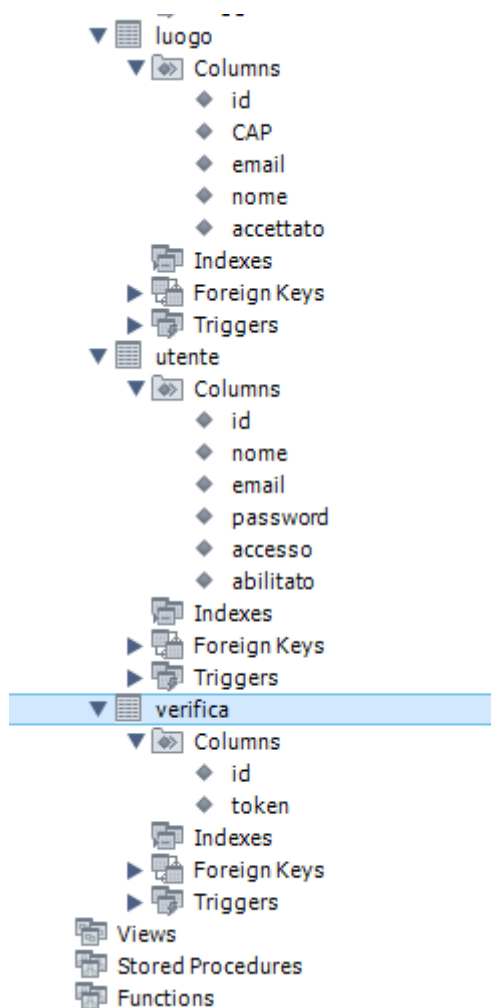


Figura 10: Tabelle "luogo", "utente" e "verifica"

Come già detto, il database è stato modificato più volte. Sono state aggiunte due tabelle, tra cui: "verifica" e "cap". La tabella "verifica" contiene i codici di attivazione per confermare la registrazione al sito, vengono conservati finché l'utente non ne fa uso. La tabella "cap" contiene tutte le città in Svizzera. Verrà utilizzata per fare i controlli se la città richiesta dall'ente fa parte dello stato.

Oltre alle tabelle sono stati aggiunti anche degli attributi nuovi nelle tabelle:

- "abilitato" è stato aggiunto alla tabella "utente" e serve per assicurarsi che l'amministratore abbia accettato la registrazione di un ente.
- "accettato" è stato aggiunto alla tabella "luogo" per assicurarsi che la richiesta del luogo sia stata accettata dall'amministratore.
- "cap\_registrato" è stato aggiunto alla tabella "cap" per sapere che quella città è già stata aggiunta alla tabella "luogo" e non c'è bisogno della richiesta.
- "richiesta" è stato aggiunto alla tabella "cap" per sapere che è stata fatta una richiesta per quella città.
- "id" è stato aggiunto alla tabella "cap" per identificare la città.

### 3.3 PHP

La prossima fase è stata creare i file PHP che si occupassero di gestire le interrogazioni al database.

La prima cosa da fare è stato creare un file di connessione al database in modo da non avere lo stesso identico codice ridondante in ogni modulo, viene definito come segue:

```
define('DBHOST', 'localhost:3306');
define('DBUSER', 'root');
define('DBPASS', '');
define('DBNAME', 'progettoeventi');

$conn = new mysqli(DBHOST, DBUSER, DBPASS, DBNAME);

function __construct() {
    print "In BaseClass constructor\n";
}

if (!$conn) {
    die("Connection failed : " . $conn->connect_error);
}
```

In questo script vengono definite quattro variabili globali ognuna delle quali contiene una parte dell'autenticazione per l'accesso al database locale, viene poi creato un oggetto di tipo *mysqli* che stabilisce la connessione ed infine controllato se quest'ultima è valida.

Ho creato lo script di registrazione, incaricato di interrogare il database per l'inserimento di un nuovo utente ed inviare un email di conferma all'email passatagli.  
I punti importanti del file sono definiti da:

```
function getToken($length){
    $token = "";
    $codeAlphabet = "ABCDEFGHIJKLMNOPQRSTUVWXYZ";
    $codeAlphabet.= "abcdefghijklmnopqrstuvwxyz";
    $codeAlphabet.= "0123456789";
    $max = strlen($codeAlphabet);

    for ($i=0; $i < $length; $i++) {
        $token .= $codeAlphabet[random_int(0, $max-1)];
    }

    return $token;
}
```

La funzione soprastante genera un token casuale di una lunghezza x.

```
function sendMail($id, $email, $token){
    $subject = "Conferma registrazione per Coordinamento Eventi";
    $from = 'nema.sto2000@gmail.com';
    $to = $email;
    $headers = "Da: ".$from."\r\n";
    $headers .= "Content-Type: text/html; charset=UTF-8\r\n";
    $body="Grazie per essersi registrato/a! Perfavore vada al seguente <a
href='http://localhost/ProgettoI4AA/Sito/Verifica.php?id=".$id."&token=".$token."'>link
</a> per completare l'attivazione del suo account Carthafind.";
    mail($to, $subject, $body, $headers);
}
```

La Funzione sendMail() che invia l'email di conferma con un link generato in HTML tramite il tag <a> il quale punta al file verifica.php.

La password non deve essere in chiaro, perciò ho fatto la dodifica della stringa password in un hash di tipo SHA256:

```
$password = hash('sha256', $psw);
```

Fatto lo script di registrazione ho scritto quello di login in modo da testare se le informazioni passate funzionassero come previsto. Questo modulo si occupa di convalidare i dati inseriti, viene fatto tramite un controllo sull'email esistente o meno e, tramite una comparazione tra l'hash sul database e quello locale, dato che la codifica è one-way.

Inoltre, ha il compito di inizializzare una sessione con l'id dell'utente che ha effettuato il login e reindirizzarlo sul file *Amministratore.php* o *Ente.php*, in base se l'accesso è di tipo "Ente" o "Amministratore".

```
if($count == 1 && $row['password'] == $loginPassword) {
    $_SESSION['user'] = $row['id'];

    if($row['accesso'] == "Ente" && $row['abilitato'] == true){
        header("Location: Ente.php");
    }elseif ($row['accesso'] == "Amministratore") {
        header("Location: Amministratore.php");
    }
} else {
    $outMessage = "Credenziali non valide, riprovare.";
}
}
```

Insieme al login è stato necessario utilizzare il logout che d'altra parte si occupa di resettare la sessione, riportando l'utente alla pagina di login:

```
session_start();
if (!isset($_SESSION['user'])) {
    header("Location: PaginaPrincipale.php");
} else if(isset($_SESSION['user']) != "") {
    header("Location: PaginaPrincipale.php");
}

if(isset($_GET['logout'])) {
    unset($_SESSION['user']);
    session_unset();
    session_destroy();
    header("Location: PaginaPrincipale.php");
    exit;
}
```

```
if (empty($nome)) {
    $error = true;
    $nomeError = "Inserisci il tuo nome.";
} else if (strlen($nome) < 3) {
    $error = true;
    $nomeError = "Il nome deve contenere almeno tre lettere.";
} else if (!preg_match("/^[a-zA-Z ]+$/", $nome)) {
    $error = true;
    $nomeError = "Il nome può solo contenere lettere dell'alfabeto.";
}
}
```

Al momento dell'inserimento dei dati negli input, bisogna fare il controllo di ciò che è stato inserito nel campo e in base al dato (nome, email, ecc) bisogna fare dei controlli. Nel caso ci siano errori non viene permesso di continuare e viene stampato un messaggio d'errore, come per esempio nello <span> sottostante:

```
<span class="text-danger"><?php echo $nomeError; ?></span>
```

Per quanto riguarda la gestione richieste dell'inserimento del luogo da parte dell'ente, se ne occupa l'amministratore:

```
$queryRichiesta = "SELECT * FROM `cap`";
$resultRichiesta = $conn->query($queryRichiesta);
$fetchRichiesta = $resultRichiesta->fetch_assoc();

while($row1 = $resultRichiesta->fetch_array(MYSQLI_ASSOC)){
    if($row1["richiesta"]==true){
        //Codice di update
    }
}
```

Nella tabella “cap”, quando un ente fa la richiesta di inserimento di un Luogo, viene modificato il valore del campo “richiesta” a true. Viene fatta scorrere tutta la tabella “cap” e se il valore della richiesta è true, allora la richiesta apparirà e l'amministratore potrà accettarla o meno. Se l'amministratore sceglie di accettare la richiesta, esso viene modificato a “false” perché altrimenti la richiesta continuerebbe ad apparire. Il campo “cap\_registrato” viene modificato a “true” così da non fare più la richiesta di aggiunta per quanto riguarda quella città. Nella tabella luogo viene modificato il valore dell'attributo “accettato” a “true” così che ci si assicura che il luogo sia accettato.

Anche se il filtro non funziona, ho fatto in modo che la selezione del luogo per filtrare in base ad esso mostri i luoghi presenti nella tabella “luogo”.

```
<?php
$querySelectLuogo = "SELECT * from luogo";
$resultSelectLuogo = $conn->query($querySelectLuogo);
while($row2 = $resultSelectLuogo->fetch_array(MYSQLI_ASSOC)){
    $querySelectLuogo1 = "SELECT * from luogo where nome='".$row2['nome']."' limit 1";
    $resultSelectLuogo1 = $conn->query($querySelectLuogo1);
    $fetchtSe = $resultSelectLuogo1->fetch_assoc();
}>
    <option value="<?php echo $fetchtSe["nome"]; ?>">
        <?php echo $fetchtSe["nome"]; ?>
    </option>
<?php } ?>
```

Vengono fatti scorrere tutti i dati della tabella “luogo” e per ogni città presente nella tabella viene stampato il nome.

Per quanto riguarda la lista degli eventi ho utilizzato il seguente metodo per visualizzarli:

```
<?php
    $query1 = "SELECT * from evento";
    $result1 = $conn->query($query1);
?>

<thead>
    <tr>
        <td>Id</td>
        <td>Titolo</td>
        <td>Tipo</td>
        <td>Orario di Inizio</td>
        <td>Orario di Fine</td>
        <td>Osservazione</td>
        <td>Id del Luogo</td>
        <td>Id dell'utente</td>
    </tr>
</thead>
<tbody>
    <?php while($row2 = $result1->fetch_array(MYSQLI_ASSOC)){?>
        <tr>
            <td><?php echo $row2['id'] ?></td>
            <td><?php echo $row2['titolo']?></td>
            <td><?php echo $row2['tipo']?></td>
            <td><?php echo $row2['orarioInizio']?></td>
            <td><?php echo $row2['orarioFine']?></td>
            <td><?php echo $row2['osservazioni']?></td>
            <td><?php echo $row2['id_luogo']?></td>
            <td><?php echo $row2['id_utente']?></td>
        </tr>
    <?php } ?>
</tbody>
```

Viene creata una tabella con tutte le informazioni di ogni evento.

È possibile modificare i dati riguardanti un ente/evento/luogo, nel codice di seguito viene fatta la modifica di un evento.

```
$eventoID = $_REQUEST['eventoID'];

if(!$error){
    $updateLuogo = "UPDATE `evento` SET `tipo`='".$tipo."' WHERE `id`='".$eventoID."'";
    $resultUpdateLuogo= $conn->query($updateLuogo);

    header("Location: ../gestione_eventi.php");
}
```

Se i dati vengono inseriti correttamente e non vengono presentati degli errori, questa query aggiorna la tabella "evento" con i nuovi valori.

È possibile anche inserire dei nuovi eventi/luoghi/enti, nel codice seguente viene visto il codice che lo permette:

```
if (!$error) {
    $queryEmailUtente = "SELECT `email` FROM `utente` WHERE `id` = '". $userID. "'";
    $resultEmailUtente = $conn->query($queryEmailUtente);
    $fetchEmailUtente = $resultEmailUtente->fetch_assoc();

    $queryIdLuogo = "SELECT `id` FROM `luogo` WHERE `email` = '". $fetchEmailUtente['email']. "'";
    $resultIdLuogo = $conn->query($queryIdLuogo);
    $fetchIdLuogo = $resultIdLuogo->fetch_assoc();

    $queryIns = "INSERT INTO `evento` (`titolo`, `tipo`, `orarioInizio`, `orarioFine`, `osservazioni`, `id_luogo`, `id_utente`) VALUES
    ('. $titolo. '",
    '". $tipo. '", '". $orarioInizio. '", '". $orarioFine. '", '". $osservazione. '", '". $fetchIdLuogo[
    "id"]."', '". $userID. "')";

    if ($conn->query($queryIns)=== TRUE ) {
        unset($titolo);
        unset($tipo);
        unset($orarioInizio);
        unset($orarioFine);
        unset($osservazione);
        $titolo = "";
        $tipo = "";
        $orarioInizio = "";
        $orarioFine = "";
        $osservazione = "";
        header("Location: ../../Ente.php");
    }
}
```

Se non vengono riscontrati errori, vengono ritornati: l'email dell'utente e l'id del luogo, i quali poi insieme a tutti gli altri dati verranno inseriti nella tabella "evento".

Oltre alla possibilità di modifica e di inserimento, è presente anche quella di cancellazione di un ente/evento/luogo, nel codice di seguito viene fatta l'eliminazione di un evento:

```
<?php
include_once ' ../../Database/dbconnect.php';

$eventoID = $_REQUEST['eventoID'];

$querys1 = "DELETE FROM `evento` WHERE `id_utente` =' $userID'";

if ($conn->query($querys1) === TRUE ) {
    echo "Record deleted successfully";
} else {
    echo "Error deleting record: " . $conn->error;
}
header("Location: ../../Ente.php");
?>
```

Alla pagina viene passato l'id dell'evento da eliminare. Se va a buon fine viene stampato un messaggio di successo altrimenti un messaggio di errore. Infine viene riportato alla pagina specificata.

## 4 Test

### 4.1 Protocollo di test

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Controllo dell'esistenza del sito
<b>Riferimento:</b>	REQ-001		
<b>Descrizione:</b>	Controllo che il sito esista inserendo l'URL.		
<b>Prerequisiti:</b>	XAMPP		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Avviare l'applicazione XAMPP.</li> <li>2. Cliccare su "start" per i servizi "Apache" e "MySQL".</li> <li>3. Aprire un browser.</li> <li>4. Inserire il percorso del progetto nella cartella "xampp\htdocs\", in questo caso <a href="http://localhost/ProgettoI4AA/Sito/PaginaPrincipale.php">http://localhost/ProgettoI4AA/Sito/PaginaPrincipale.php</a>.</li> </ol>		
<b>Risultati attesi:</b>	La pagina viene caricata correttamente		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Controllo funzionamento del ruolo di amministratore
<b>Riferimento:</b>	REQ-002		
<b>Descrizione:</b>	Controllo che l'amministratore possa fare l'accesso e che riesce a svolgere tutte le attività.		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Cliccare sull'opzione "Login".</li> <li>2. Inserire le credenziali dell'amministratore.</li> <li>3. Controllare che non ci siano richieste di inserimento ente o luogo.</li> <li>4. Provare a gestire le tipologie di evento, gli enti e i luoghi.</li> <li>5. Provare a visualizzare la lista di eventi in format testuale e calendario.</li> <li>6. Provare a fare la ricerca su tutti i campi utili dei luoghi/enti.</li> <li>7. Provare a stampare il risultato delle ricerche.</li> <li>8. Cliccare sull'opzione del menu "Statistiche".</li> <li>9. Visualizzare le statistiche e i grafici che mostrano: <ol style="list-style-type: none"> <li>a. Il numero degli eventi totali</li> <li>b. Gli eventi totali per tipologia</li> <li>c. Il numero degli eventi per ente</li> <li>d. Il numero di eventi per mese/anno (grafico)</li> </ol> </li> <li>10. Deve essere filtrabile in base al luogo.</li> </ol>		
<b>Risultati attesi:</b>	L'amministratore riesce a fare l'accesso e a svolgere tutta la procedura		



<b>Test Case:</b>	TC-003	<b>Nome:</b>	Controllo funzionamento del ruolo di Ente
<b>Riferimento:</b>	REQ-003		
<b>Descrizione:</b>	Controllo che l'ente possa fare l'accesso e che riesce a svolgere tutte le attività.		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Cliccare sull'opzione "Login".</li> <li>2. Inserire le credenziali ente.</li> <li>3. Visualizzare l'elenco degli eventi/riunioni/manifestazioni sià inseriti.</li> <li>4. Provare a inserire/cancellare/modificare i propri eventi</li> <li>5. Provare a visualizzare la lista di eventi in format testuale e calendario.</li> <li>6. Provare a stampare tutti gli eventi per il rispettivo luogo</li> </ol>		
<b>Risultati attesi:</b>	L'ente riesce a fare l'accesso e a svolgere tutta la procedura		

<b>Test Case:</b>	TC-004	<b>Nome:</b>	Controllo funzionamento dei visitatori
<b>Riferimento:</b>	REQ-004		
<b>Descrizione:</b>	Controllo che il visitatore riesca a svolgere tutte le attività a lui concesse senza fare il login		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Scegliere un luogo e visualizzare gli eventi</li> <li>2. Provare a visualizzare la lista di eventi in formato testuale e calendario.</li> <li>3. Provare a ricercare per qualsiasi campo utile gli eventi (anche filtrando per ente).</li> </ol>		
<b>Risultati attesi:</b>	Riesce a svolgere tutta la procedura senza la necessità dell'accesso.		

<b>Test Case:</b>	TC-005	<b>Nome:</b>	Controllo funzionamento della registrzione
<b>Riferimento:</b>	REQ-005		
<b>Descrizione:</b>	Controllo che la registrazione venga fatta		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Cliccare sull'opzione "Registrazione".</li> <li>2. Inserire nei campi i dati (nome, email, password e luogo).</li> <li>3. Controllare il profilo dell'email e nell'email inviata dall'amministratore cliccare sul link di verifica dell'utente</li> <li>4. L'amministratore deve accettare la richiesta di registrazione</li> </ol>		
<b>Risultati attesi:</b>	L'ente riesce a fare la registrazione e riceve l'email per confermare la stessa.		

<b>Test Case:</b>	TC-006	<b>Nome:</b>	Controllo funzionamento della gestione
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Controllo che l'utente riesca a fare il login e il logout.		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Cliccare sull'opzione "Login".</li> <li>2. Inserire le credenziali dell'ente o dell'amministratore.</li> <li>3. Controllare l'indirizzamento alla pagina sia quella dell'ente (nel caso l'utente fosse un amministratore, allora la pagina deve essere quella dell'amministratore)</li> <li>4. Cliccare su "Logout"</li> </ol>		
<b>Risultati attesi:</b>	L'utente riesce a fare il login e il logout.		

<b>Test Case:</b>	TC-007	<b>Nome:</b>	Controllo funzionamento inserimento di un nuovo evento
<b>Riferimento:</b>	REQ-007		
<b>Descrizione:</b>	Controllo che l'ente riesca ad aggiungere un nuovo evento		
<b>Prerequisiti:</b>	TC-001		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Cliccare su "Inserisci", nella barra infondo alla pagina dell'ente</li> <li>2. Nel form inserire i dati necessari (Titolo, tipo, ora di inizio, ora di fine, e le eventuali osservazioni).</li> <li>3. Provare ad inserire dei numeri nel "tipo" e nel "titolo" e assicurarsi che non appaiano errori.</li> <li>4. Cliccare su "Aggiungi"</li> </ol>		
<b>Risultati attesi:</b>	L'utente riesce ad aggiungere un nuovo evento senza la presenza di errori.		

## 4.2 Risultati test

Test	Risultato
TC-001	Riuscito
TC-002	Riuscito (fino al punto 4, gli altri punti no perché non sono stati implementati)
TC-003	Riuscito (fino al punto 4, gli altri punti no perché non sono stati implementati)
TC-004	Non riuscito (il visualizzatore vede soltanto la lista di eventi)
TC-005	Riuscito
TC-006	Riuscito
TC-007	Riuscito (tranne che per la gestione degli apostrofi nei campi di testo)

#### **4.3 Mancanze/limitazioni conosciute**

Ci sono dei requisiti che non sono stati implementati, i quali:

- Lista degli eventi in formato calendario
- La ricerca su tutti i campi utili degli eventi e dei luoghi/enti
- La stampa degli eventi o delle ricerche
- Le tre statistiche per quanto riguarda l'amministratore (numero eventi totali, eventi totali per tipologia, numero eventi per ente.)
- Grafico che mostra il numero di eventi per mese/anno per quanto riguarda l'amministratore
- Filtrazione in base al luogo

## 5 Consuntivo

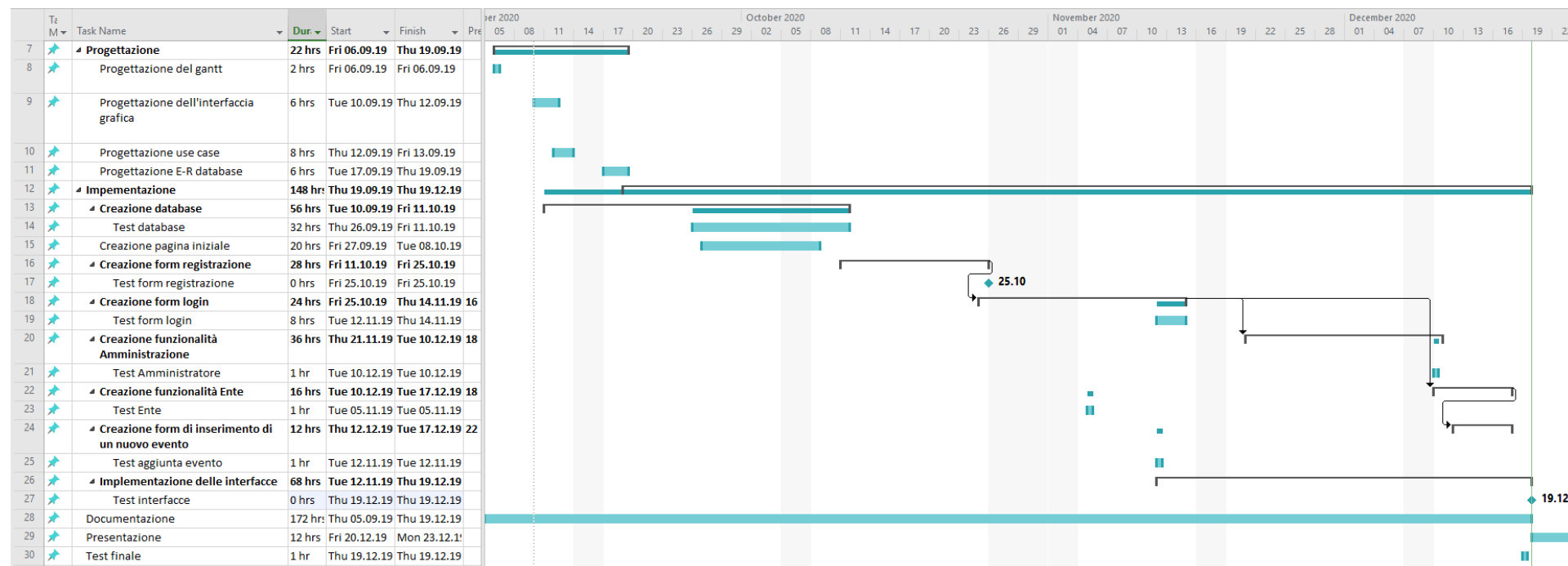


Figura 11: Consuntivo

Come si può notare dal consuntivo, le ore impiegate effettivamente per realizzare il progetto sono molto diverse rispetto a quelle pianificate (Per la parte di implementazione). Si nota anche l'assenza dell'attività "Creazione algoritmo di ricerca". È stata rimossa perché l'attività non è stata svolta. Per continuare il progetto ho impiegato anche delle ore a casa per portarmi avanti il più possibile. Le ore stimate del lavoro a casa sono circa 30.

## 6 Conclusioni

L'idea del progetto potrà essere utile ma non cambierà perché probabilmente esisteranno già siti simili al mio anche più efficaci e performanti.

### 6.1 Sviluppi futuri

Migliorare lo stile del sito e perfezionare i controlli dei campi, visto che se per esempio viene messo un apostrofo nel titolo dell'evento non esegue la query. Potrà essere migliorato così che saranno utilizzati metodi diversi più efficaci o meno pesanti.

Potrà essere implementata la struttura web MVC (Model View Controller), che permette di separare chiaramente il codice e le funzioni. Inoltre bisognerà implementare tutta la parte che non sono riuscito a svolgere, come: la ricerca degli eventi, la stampa degli eventi e delle ricerche su di essi, le statistiche, i grafici ecc.

### 6.2 Considerazioni personali

Il progetto nel suo insieme è interessante da creare perché un prodotto del genere potrebbe essere utile ai cantoni, se non anche a livello nazionale.

Ho trovato personalmente il progetto difficile da realizzare poiché io non sia molto bravo in programmazione. Ho riscontrato tanti problemi, i quali mi hanno fatto perdere tempo e non mi hanno permesso di finire il prodotto, anche se ogni tanto erano molto banali.

Penso che sia stato utile per farmi capire che magari dovrei tendere a realizzare un progetto legato alla sistemistica nel secondo semestre.

Pur essendoci stati diversi problemi legati alla programmazione ci sono anche state delle nuove conoscenze riguardo ai linguaggi di programmazione utilizzati.

Mi è stato molto utile per capire quanto dei problemi riscontrati possano rallentare la procedura dell'implementazione.

## 7 Bibliografia

### 7.1 Sitografia

- <https://app.moqups.com/unsaved/82924190/edit/page/ad64222d5>, Creazione Mockup, 20-09-2019
- <https://www.w3schools.com/howto/default.asp>, W3School How to, utilizzato diverse volte durante il progetto
- <https://www.php.net/>, Documentazione ufficiale PHP, utilizzato più volte
- <https://1bestcsharp.blogspot.com/2015/10/php-populate-html-select-option-values-from-mysql-database-code.html>, PHP Select, 06-12-2019
- <http://forum.wininizio.it/index.php?/topic/27502-calendario-cn-la-possibilit%C3%A0-di-inserire-eventi/>, Calendario
- <https://stackoverflow.com/questions/12234545/refresh-page-without-losing-the-post-value>, Aggiornare la pagina senza perdere i valori
- <https://forum.html.it/forum/showthread/t-1411781.html>, Conservare i dati, 17-10-2019
- <https://stackoverflow.com/questions/20622300/delete-row-button-for-mysql-in-php>, Delete row, 29-11-2019

## **8 Allegati**

---

Allegati:

- Diari di lavoro
- Codici sorgente
- Abstract

Tutti i file riguardanti il progetto sono presenti anche su GitHub:

- <https://github.com/nemastojanovic/CoordinamentoEventi>