

# MY EVENTI

Sito di annunci di eventi.



<http://tecweb2016.studenti.math.unipd.it/ebile>

**Numero di componenti del gruppo: 1**

**Cognome, nome e matricola dei membri del gruppo:** Bile Ezanin Christian Prince Carlos 1096677

**email referente:** ezaninchristianprincecarlos.bile@studenti.unipd.it

## CREDENZIALI DI ACCESSO

### 1) Amministratore

**nome utente:** admin

**password:** admin

### 2) Utente premium

**nome utente:** chrissbill

**password:** chrissbill

### 3) Utente non premium

**nome utente:** ombretta

**password:** ombretta

## NOTA IMPORTANTE:

Per installare il sito su un altro server, è necessario caricare i database DB.sql e istat.sql presenti nella cartella Modello, e infine configurare l'accesso al database modificando il nome dell'host, il nome utente, la password e il nome del database presenti nel file config.php nella cartella Modello.

# INDEX

I.PRESENTAZIONE DEL SITO.....	3
i.Obbiettivo.....	3
ii.User target.....	3
iii.Compatibilità browser.....	3
II.PROGETTAZIONE E REALIZZAZIONE DEL SITO.....	4
i.Architettura dell'informazione.....	4
1.Schemi organizzativi usati.....	4
2.Strutture organizzative usate.....	4
ii.Struttura HTML.....	5
1.Scelta dei colori.....	6
2.Contenuto delle pagine.....	7
a)Home page.....	7
b)Pagina tipi di eventi, categorie di eventi, eventi nella tua città.....	7
c)Pagina dei risultati.....	7
d)Pagina dettagli dell'annuncio.....	8
e)Pagine utente.....	8
f)Pagina profilo utente.....	9
g)Pagina login, registrazione e impostazioni account.....	9
h)Pagina Informazioni e FAQ.....	9
3.Layout.....	10
4.Metafora della pesca.....	10
iii.Javascript.....	10
iv.PHP.....	12
1.Il database.....	12
2.Classi e interfacce php.....	13
v.Sicurezza.....	13
III.ACCESSIBILITÀ E USABILITÀ.....	15
1.Percepibile.....	15
2.Utilizzabile.....	15
3.Comprensibile.....	16
4.Robusto.....	17
IV.TEST ESEGUITI.....	17

## I. PRESENTAZIONE DEL SITO

### i. Obbiettivo

Per il progetto di Tecnologie Web, ho deciso di implementare un sito web di annunci di eventi dal nome My Eventi. L'obiettivo del sito è quello di pubblicare annunci di eventi accessibili a tutte le categorie di utenti. Per eventi intendo conferenze, seminari, concerti ecc... Su My Eventi gli annunci vengono pubblicati dagli utenti iscritti che sono in possesso di un privilegio che chiamo *privilegio premium*. Ogni utente iscritto al sito è dotato di un cognome, un nome, un nome utente e una password. La suddivisione degli utenti iscritti è la seguente:

1. **Gli utenti non premium:** Sono utenti che godono dei servizi di prenotazione ad eventi, ed hanno inoltre una cosiddetta rete sociale che è un mini social, stile Twitter, in cui possono seguire un' altro utente iscritto non amministratore e cercare un utente iscritto. Seguire un utente significa ricevere delle notifiche su alcune attività dell'utente. Per ora l'utente seguace riceve notifiche sui nuovi annunci pubblicati e sulle partecipazioni ad eventi degli utenti che segue, sullo stato dei suoi privilegi premium, cioè se la sua domanda è stata accettata. Gli utenti non premium hanno inoltre un profilo utente visitabile da tutti gli utenti del sito, che siano iscritti o non iscritti, perciò sul profilo utente non vengono mostrati delle informazioni sensibili, ma solo il nome utente, il cognome, il nome e il tipo di utente. Gli utenti non premium non possono pubblicare annunci, ma possono richiedere di essere utenti premium per pubblicare annunci, e possono anche modificare le proprie informazioni personali, tra cui la loro password.
2. **Gli utenti premium:** Sono utenti che , oltre a godere di tutti i servizi degli utenti non premium, possono pubblicare e gestire i loro annunci. Sui loro profili utente vengono visualizzate altre informazioni aggiuntive, che sono il totale degli annunci pubblicati e una lista parziale di tali. L'utente premium riceve inoltre notifiche sulle nuove partecipazioni ai suoi annunci di eventi, sull'eventuale blocco o sblocco di un suo annuncio, e sulla revoca dei suoi privilegi premium.
3. **L'utente amministratore:** L'utente amministratore non ha i servizi offerti agli altri tipi di utente, ma ha dei compiti. L'amministratore ha il compito di mettere in quarantena, ossia bloccare la visualizzazione di un annuncio qualora violasse le regole di pubblicazione stabilite, può anche sbloccarlo se il suo pubblicatore lo modifica entro 15 giorni di tempo, passato questo lasso di tempo, egli deve provvedere all' eliminazione per liberare lo spazio occupato nel database. L'amministratore può mettere in primo piano un annuncio, e può bloccare un utente se viola delle regole, ma non può eliminarlo, l'eliminazione avverrà in php admin quando l'utente non risponde al motivo del blocco account dopo 1 mese di tempo. Inoltre l'amministratore riceve le richieste di passaggio ad utente premium e deve provvedere a contattare l'utente via mail per un colloquio via skype per assicurarsi che sia una persona affidabile al fine di assegnargli i privilegi. Per ridurre il carico di lavoro, ho tolto molti compiti all'amministratore, ad esempio l'aggiunta di una nuova categoria di eventi. Questo perché è un progetto che ho svolto da solo e il lavoro da fare era tanto.

### ii. User target

Gli utenti mirati dal sito web sono persone fisiche di tutte le categorie, come richiesto dalla consegna, senza limite di età che sono residenti in Italia o all'estero, con la condizione che abbiano una conoscenza della lingua italiana al livello buono.

### iii. Compatibilità browser

Per la versione desktop il sito prevede di essere compatibile con browser di versioni anche datate con supporto di Javascript di Firefox, Google Chrome, Opera , Safari e dalla versione 8 in su di Internet Explorer. Mi sono limitato alla versione 8 di Internet Explorer in quanto alcune proprietà CSS non sono applicabili alle versioni più vecchie, e trovare un'alternativa non è banale. Inoltre scrivere codice JavaScript per le versioni vecchie di IE è difficile e più corposo con gravi conseguenze sul caricamento delle pagine per i dispositivi che hanno risorse limitate. Percui ogni volta che un

utente proverà ad entrare sul sito con tale browser, verrà ridiretto ad un'altra pagina che gli dice di accedere con un browser più aggiornato. Non è stato usato HTML5 per rimanere compatibile con IE 8 senza code forking. Per quanto riguarda la versione mobile, il sito è previsto di essere compatibile con i dispositivi android dalla versione Honeycomb in su, e dall'iphone 3 in su.

## II. PROGETTAZIONE E REALIZZAZIONE DEL SITO

### i. Architettura dell'informazione

Un annuncio di evento pubblicato sul sito è caratterizzato principalmente da : un titolo, il tipo di evento, La categoria dell'evento, l' indirizzo (composto dalla via, la città e la provincia ), la data di svolgimento, l' ora in cui inizia, una breve introduzione dell'annuncio, e la descrizione chiara dell'evento. Gli annunci presentano anche altri campi secondari, facoltativi, che sono: le etichette e le foto. Le etichette si possono considerare come sottocategorie create con parole chiavi dagli utenti che pubblicano, mentre le foto sono immagini che caricano per rendere il loro annuncio più attraente.

#### 1. Schemi organizzativi usati

Per organizzare gli oggetti informativi del sito che sono gli annunci pubblicati, ho usato entrambi i tipi di schemi discussi in aula, e cioè lo schema organizzativo esatto e lo schema organizzativo ambiguo. L'uso dello schema organizzativo esatto è dovuto al fatto che gli eventi si distinguono univocamente per tipo, *es: una conferenza non può essere un concerto*, e quindi era buona norma raggrupparli per tipo . Per quanto riguarda l'uso dello schema ambiguo, il motivo sta nelle categorie di eventi e nella città in cui si svolgerà l'evento. Infatti una categoria di evento rinchiede un insieme di eventi che con alta probabilità sono tutti eterogenei, e una città può ospitare molti eventi di vari tipi in diverse categorie. Inoltre le etichette inserite dagli utenti generano delle sottocategorie di annunci che aumentano la complessità a suddividere gli oggetti informativi in uno schema solo esatto. Il tipo di schema esatto usato è lo schema alfabetico mentre per lo schema ambiguo ho usato uno schema per argomento.

Sul sito gli schemi sono presentati separatamente, con 3 pagina distinte accedibili con collegamenti sul menù di navigazione principale, uno per lo schema esatto e due per lo schema ambiguo ( categorie e città ). Non viene dedicata nessuna pagina alle sottocategorie degli annunci(le etichette scritte dai pubblicatori durante l'inserimento dell'annuncio) che possono essere in numero molto elevato, perciò ho deciso di usarli come parole chiavi nella ricerca di annunci. Qui sotto è riportato uno screenshot dei link alle pagine degli oggetti informativi organizzati secondo lo schema esatto e secondo lo schema ambiguo, presenti nel menù di navigazione:




il link cerchiato in verde porta alla pagina dei tipi di eventi, mentre i link cerchiati in marroncino portano rispettivamente alle categorie di eventi e alle città ospitanti.

#### 2. Strutture organizzative usate

Come struttura organizzativa , ho usato la struttura a database. Questa scelta è dovuta al fatto che ogni oggetto informativo (gli annunci di eventi pubblicati ) sono caratterizzati da campi dati raggruppati in tabelle relazionali senza gerarchie. Un oggetto di tipo Conferenza scienze può portare ad un gruppo di oggetti di tipo Conferenza sport tramite la relazione Conferenza. Può sembrare che ci sia gerarchia, ma un annuncio non è una conferenza. Per evidenziare la relazione tra i vari annunci, e cioè la struttura organizzativa , con lo scopo di aiutare l'utente ad avere un'idea chiara di come navigare nel sito, senza disorientarsi, tutti gli elenchi di annunci ricercati dall'utente in base al tipo, la categoria, al pubblicatore, ecc.. vengono mostrati come risultati di una ricerca avanzata eseguita indirizzando l'utente alla pagina

Sono stati trovati 7 annunci di eventi.


[Informazioni sugli annunci che sono stati trovati.](#)

Pagina 1 su 2

Annuncio 1 di evento trovato pubblicato da chrissbill che si terrà il giorno 21/02/2018 : Congresso - sdvdsdvd sdvdsdvs svdsv  
 jxxb jkabdsabj jkabjcdwylb jcwbbjwbk wykwubukhw bkwbsbjwvjwb  
[Leggi tutto l'annuncio 1 trovato](#)

Annuncio 2 di evento trovato pubblicato da chrissbill che si terrà il giorno 21/02/2017 : Conferenza - P000000dd  
 jksb jbdjdbbbs bybwleljwn wjwewjebk dbjkewekjdb  
[Leggi tutto l'annuncio 2 trovato](#)

Annuncio 3 di evento trovato pubblicato da chrissbill che si terrà il giorno 21/02/1995 : Congresso - sdvdsdvd sdvdsdvs svdsv  
 nxbykblhgbqkbbqkqkjb bskjgbjkb  
[Leggi tutto l'annuncio 3 trovato](#)

Annuncio 4 di evento trovato pubblicato da chrissbill che si terrà il giorno 21/12/1995 : Concerto - sdvdsdvd sdvdsdvs svdsv  
 phecgegvrvhgvgvghvgvhgh  
[Leggi tutto l'annuncio 4 trovato](#)

Annuncio 5 di evento trovato pubblicato da chrissbill che si terrà il giorno 29/12/2017 : Concerto - DJ me in concerto  
 jlnfjsknjksfjks sdjsjkbjsk djs  
[Leggi tutto l'annuncio 5 trovato](#)

Annuncio 6 di evento trovato pubblicato da chrissbill che si terrà il giorno 29/12/2017 : Conferenza - Legge dei grandi numeri.  
 Dobbiamo crederci mhbuhyghfntvjvhyb jhhjb?  
 jkbkbbbbbhjmn  
[Leggi tutto l'annuncio 6 trovato](#)

Pagina 1 su 2

Vai a pagina dei risultati:

Seleziona una pagina  
 dei risultati  
 2 su 2


Vai

[pagina dopo](#)

**Informazioni sulla ricerca eseguita:**

Hai cercato :

- Annunci di eventi pubblicati in qualsiasi intervallo di tempo
- In tutte le categorie
- Eventi di Tutti i tipi
- Eventi in qualsiasi città
- Eventi che si svolgeranno, oppure che si sono svolti, in qualsiasi intervallo di tempo


[Torna agli annunci trovati](#)

Ricerca avanzata di annunci

Ottieni informazioni e aiuto sulla ricerca avanzata  
 (verrai indirizzato alla pagina di aiuto)  
 Tutti i campi sono facoltativi

Inserisci una parola chiave caratteristica dell'evento ( al massimo 30 caratteri), ad esempio: Informatica

Inserisci una città dove si svolgerà l'evento, ( al massimo 30 caratteri), ad esempio: Treviso

Inserisci il nome utente, nome o cognome del pubblicatore dell'annuncio ( al massimo 30 caratteri)

Seleziona il tipo di evento ( Modifica la voce corrente scegliendo una delle opzioni )

Tutte le tipologie di eventi

Seleziona una categoria ( Modifica la voce corrente scegliendo una delle opzioni )

Tutte le categorie

Data dell' evento

Scegli un intervallo di tempo in cui si svolgerà l'evento. Inserisciti 2 cifre per il giorno e il mese, e 4 cifre per l'anno. Esempio: 23 per il giorno, 09 per il mese e 2017 per l'anno.

Dal:

Giorno	Mese	Anno
<input type="text"/>	<input type="text"/>	<input type="text"/>

Al:

Giorno	Mese	Anno ad esempio:
<input type="text"/>	<input type="text"/>	2017
<input type="text"/>	<input type="text"/>	<input type="text"/>

Data di pubblicazione dell'annuncio

Scegli un intervallo di tempo in cui l' annuncio di evento è stato pubblicato. Inserisciti 2 cifre per il giorno e il mese, e 4 cifre per l'anno. Esempio: 23 per il giorno, 09 per il mese e 2017 per l'anno.

Dal:

Giorno	Mese	Anno
<input type="text"/>	<input type="text"/>	<input type="text"/>

Al:

Giorno	Mese	Anno
<input type="text"/>	<input type="text"/>	<input type="text"/>

Cerca

## ii. Struttura HTML

Una pagina visualizzata è divisa in 5 parti principali dentro al tag body: Header, Navigazione, Breadcrumb, Contenuto e Footer. queste parti sono rappresentate da div, tag paragrafo per il Breadcrumb, con relativo id: La parte Header contiene le informazioni sul sito: nome sito e breve descrizione del suo scopo; La parte Navigazione contiene il menù di navigazione principale per rispondere alla domanda dove posso andare dell'utente; La parte Breadcrumb è un tag paragrafo che contiene il percorso relativo dalla Home page fino alla pagina dove si trova l'utente, così tale da rispondere alla domanda dove sono? ; La parte Contenuto contiene tutto il contenuto della pagina; La parte Footer

contiene solo un link per tornare al contenuto, il nome del sito, e il copyright con anno di pubblicazione. Le pagine che sono solo di compilazione di un modulo o di avviso, non contengono l'Header, la Navigazione e il Breadcrumb ma uno o più link per andare indietro, in una pagina specifica a seconda di quali potrebbero essere le richieste dell'utente. Dentro al div contenuto della pagina, tutti gli elementi che sono in relazione rispetto al contesto, sono raggruppati in div con eventuale intestazione, questo mi permette di avere una sequenza logica degli elementi e un maggior controllo delle priorità e proprietà CSS. Alcuni elementi subiscono una modifica della loro posizione nella sequenza applicando il CSS, ma è solo una modifica visiva e non al livello strutturale (esempio: gli oggetti flotanti). Questo perché ho cercato di immagazzinare il più possibile di informazioni importanti dentro l'area visibile per evitare che gli utenti usino continuamente lo scrollbar.

## 1. Scelta dei colori

I colori sono stati scelti in base a 4 criteri: contesto del sito, rapporto contrasto luminosità, convenzioni, daltonismo o altre malattie simili. I colori contesto principale del sito è il nero, il grigio chiaro, e il bianco, in quanto prima di partecipare ad un evento, siamo allo scuro di quello che succederà veramente in quell'evento, questo spiega il colore nero, poi quando iniziamo a scoprire cose in più su quell'evento, l'oscurità svanisce poco a poco, ecco perché il colore grigio chiaro, poi quando finisce l'evento, abbiamo un'idea più che chiara di cosa si trattasse, ecco perché è stato scelto il colore bianco. Inoltre come in uno spettacolo gli spettatori sono al buio e gli artisti alla luce, ho cercato di riprodurre la scena con i colori del sito. Per convenzioni ho lasciato i link visitati al colore standard ed ho messo ai link non visitati il colore standard un po' scurito. Per il focus e hover dei link, scelto di evidenziare la scritta in un verde scuro, che in esadecimale è #003300, con sfondo bianco. Gli input di tipo submit sono stati colorati in bianco con sfondo di colore blu vicino al colore standard dei link, questo per aiutare l'utente vedente a ricordare che tutto quello che è di colore blu o viola (link visitati) è cliccabile. La sovrapposizione tra colore del testo e colore di sfondo è stata fatta rispettando un rapporto tra contrasto e luminosità di almeno 7,0:1, sia per l'utente non affetto di daltonismo o malattie simili, sia anche per l'utente che è affetto da queste malattie. Qui sotto viene mostrata la pagina home vista da persone che non distinguono bene alcuni colori o affette da altre malattie visive, L'elemento cerchiato in rosso è un link visitato visto da questi utenti:





## 2. Contenuto delle pagine

### a) Home page

La Home page viene generata dinamicamente dall'oggetto Controller in php, e contiene principalmente il primo piano, che è un insieme di annunci scelti dall'amministratore, per i quali pensa che possano essere di interesse dell'utente, o che siano di grande importanza. Vengono elencati con un'intestazione (con titolo, tipo ecc.), breve introduzione dell'annuncio e una foto descritta nell'attributo alt, se tale foto dell'annuncio esiste. Per motivi di accessibilità per gli utenti non vedenti e per gli utenti che hanno difficoltà a muovere il mouse, il link ai dettagli dell'evento ha come testo l'intestazione dell'annuncio, ed inoltre lo stesso collegamento è stato messo con contenuto la foto dell'annuncio, in modo tale che se l'utente passa accidentalmente il titolo con il tab subito dopo trova la foto cliccabile. In più la foto è più grande, quindi c'è più spazio cliccabile.

### b) Pagina tipi di eventi, categorie di eventi, eventi nella tua città

La pagina tipi di eventi viene generata dinamicamente dall'oggetto Controller in php, ed è accessibile con un collegamento presente nel menù di navigazione. Il suo contenuto si divide in due parti, la prima da un glossario dei tipi di eventi disponibili, che è una lista di collegamenti ad id interni alla pagina, un link dentro il glossario ha, per esempio, come testo: "Tipi di eventi che iniziano con la lettera C". Se l'utente clicca su uno di questi link verrà portato alla lista di nomi di tipi utenti che iniziano con la C. Se non ci sono eventi di tipo con nome che inizia con la A, non ci sarà nessun link nel glossario che porta a questa lista che non esiste. Tutto questo evita all'utente che sa quello che vuole, di scorrere tutti i tipi di eventi prima di avere quello che vuole, o di scorrerla inutilmente. La seconda parte è un elenco di collegamenti agli annunci pubblicati per ogni tipo di evento. Per adesso i tipi di eventi sono: Concerto, Conferenza, Congresso, Convegno, Manifestazione e Seminario, e vengono elencati per ordine alfabetico crescente.

La pagina categorie di eventi viene anche lei, generata dinamicamente dal oggetto Controller in php, ed è presentata esattamente come la pagina dei tipi di eventi (indice alle categorie disponibili e l'elenco delle categorie).

La pagina eventi nella tua città viene generata dinamicamente dal oggetto Controller in php. Diversamente dalle pagine tipi di eventi e categorie di eventi, la pagina eventi nella tua città ha una form di ricerca che costringe l'utente ad inserire una città in cui cercare gli annunci. Questo perché i comuni e città italiani che ho trovato sono tantissimi, circa 8000, ed elencarli tutti nella pagina sarebbe proibitivo.

### c) Pagina dei risultati

La pagina dei risultati viene generata dinamicamente dall'oggetto Controller in php come risposta ad una ricerca eseguita. Essa è composta da una lista di x annunci nella pagina y, dove x è una variabile  $\geq 0$  e  $\leq$  di una costante  $> 0$ , presente nella classe views nella parte PHP, il cui scopo è quello di stabilire il massimo di record trovati per ogni pagina y, con y variabile intera positiva tale che  $1 \leq y \leq (\text{tot record}/x) + 1*b$ , dove  $b=1$  se  $\text{tot record} \% x > 0$ , 0 altrimenti. Oltre agli annunci trovati, c'è un div che contiene informazioni sulla ricerca eseguita, e un modulo per eseguire un'altra ricerca avanzata. Il fatto di aver spostato le informazioni sulla ricerca eseguita in un div in fondo alla pagina, è per evitare che l'utente legga ogni volta quello che ha cercato prima di avere i risultati, perché molto probabilmente sa già

quello che sta cercando. Come anticipato prima nella struttura organizzativa, la ricerca non viene fatta soltanto quando viene compilata una form, ma anche cliccando su un link, ad esempio cliccando sul link con scritto “Concerto: 50 annunci”, è come eseguire una ricerca scegliendo l’opzione Concerto nella select per il tipo di evento lasciando tutti gli altri campi vuoti, o con valore di default. L’utente può visualizzare gli annunci trovati alla pagina y+1 o y-1 se esistono, andando nella mini form sotto alla lista, che gli permette di andare direttamente alla pagina successiva o precedente con un link, oppure di scegliere una pagina specifica grazie ad una select.

#### d) *Pagina dettagli dell’annuncio*

La pagina dettagli dell’annuncio, è la pagina che mostra l’annuncio nei minimi dettagli. Viene generata dinamicamente dall’oggetto Controller in php in base all’id dell’evento fornito dal collegamento AJAX. Il contenuto si divide in 5 parti: la prima parte contiene il titolo, la breve introduzione e descrizione dell’evento. La seconda parte contiene le informazioni sulla data di pubblicazione, l’autore dell’annuncio, il tipo di evento, l’indirizzo, la data e l’ora in cui l’evento si svolgerà. La terza parte si compone del numero di partecipanti attuali, di un link verso la lista dei partecipanti e di un link per apportare la sua partecipazione, o ritirarla se l’utente è connesso e partecipa già all’evento. La quarta parte è composta dalle foto caricate del pubblicatore per l’annuncio, con un’opzione non consigliata agli utenti non vedenti, di visualizzare foto non ridimensionate. E infine la quinta parte è una lista di opzioni che viene data all’utente: se l’utente non è connesso, oppure se è connesso e l’annuncio non è suo, avrà come unica opzione quella di poter segnalare l’annuncio per vari motivi che dovrà scegliere o scrivere prima di inviare la segnalazione; se l’utente è connesso e proprietario dell’annuncio, avrà le opzioni di modifica e di eliminazione dell’annuncio. Se l’utente è l’amministratore, avrà le opzioni di blocco dell’annuncio e quelle per eliminarlo.

#### e) *Pagine utente*

Un utente iscritto è dotato di una home page utente, detta spazio personale per gli utenti non amministratori, e spazio amministratore per l’utente amministratore, che viene generata dall’oggetto Controller in php. Oltre a contenere la navigazione principale, l’home page utente ha un contenuto diviso in 3 parti principali: Il menù di navigazione dello spazio personale o amministratore, il sottocontenuto ottenuto cliccando su una delle voci del menù spazio personale o amministratore, e la parte di ricerche particolari nel sottocontenuto. La seconda parte e la terza dipendono fortemente dalla voce cliccata nel menù spazio personale, che a sua volta cambia in base ai privilegi dell’utente:

- *Per l’utente non premium*, il menù spazio personale è una lista di link: alle ultime notizie, cioè notizie che riguardano le attività degli utenti dentro la sua rete sociale; link alle sue partecipazioni ad eventi futuri, cioè eventi, per cui si è prenotato che devono ancora svolgersi; link alle sue partecipazioni ad eventi passati, cioè eventi, per cui si è prenotato che si sono già svolti; link alla lista degli utenti che segue; Link alla lista degli utenti che lo seguono; link al suo profilo, che lo indirizza al proprio profilo; e un link alle impostazioni del suo account.
- *Per l’utente premium*, il menù spazio personale è uguale a quello dell’utente non premium con due voci in più, e sono la voce “Pubblica un annuncio di evento” che indirizza al modulo di inserimento di un nuovo annuncio, e la voce “Gestisci i tuoi annunci” che gli permette di avere i suoi annunci pubblicati con diverse opzioni di modifica.
- *Per l’utente amministratore* il menù spazio utente cambia completamente e diventa un menù spazio amministratore con le voci: Annunci segnalati dagli utenti, che mostra tutti gli annunci segnalati; Annunci messi in quarantena, che mostra gli annunci bloccati da l’amministratore o altri amministratori; lista in ordine alfabetico crescente degli utenti iscritti e lista in ordine alfabetico decrescente degli utenti iscritti che hanno entrambi lo scopo di mostrare gli utenti iscritti non amministratori, secondo un certo ordine, con varie opzioni di controllo; e la voce Impostazioni account.

Tutti i link dentro il menù utente che non indirizzano ad una nuova pagina, producono un contenuto messo nel sottocontenuto, questo per evitare la creazione di molte pagine inutili che potrebbero disorientare l’utente.



Come detto prima, l'utente premium ha la possibilità di inserire o modificare un annuncio, perciò egli ha a disposizione due pagine:

- **Pagina inserimento e modifica annuncio:** Serve a pubblicare un nuovo annuncio o modificare uno vecchio. Questa è una pagina statica, in quanto contiene solo una form da compilare, tuttavia i campi della form sono generati dinamicamente dall'oggetto Controller in php per poter dare i valori di default agli input, textarea e select, questo è utile quando l'utente vuole modificare i dati di un annuncio, oppure quando si è verificato un errore e l'utente non deve scrivere tutto da capo.
- **Pagina di gestione foto:** questa pagina permette all'utente di gestire le foto di un suo annuncio. Può eliminare vecchie foto, oppure aggiungere nuove foto rispettando il limite massimo permesso. Questa pagina è statica perché senza header e nav, ma il suo contenuto viene generato dinamicamente dall'oggetto Controller in php per poter elencare le foto caricate dell'annuncio e controllare quante ne può ancora aggiungere.

#### *f) Pagina profilo utente*

La pagina profilo utente viene generata dinamicamente dall'oggetto Controller in php, e il suo scopo è quello di dare brevi informazioni su un utente iscritto. E' stata implementata per aumentare il potere informativo del sito, infatti non avrebbe avuto senso vedere un annuncio pubblicato da qualcuno che nessuno sa chi è. Il contenuto della pagina si compone di alcune informazioni sull'utente (nome, cognome, nome utente e tipo di utente), del suo social, cioè da link verso la lista degli utenti che segue e che lo seguono, se l'utente visitato è premium vengono mostrati i suoi primi x annunci, dove x è la variabile spiegata prima per la pagina dei risultati, e un link verso tutti i suoi annunci pubblicati, che in realtà è una ricerca avanzata per nome di pubblicatore. E infine a seconda che il visitatore del profilo sia l'utente stesso o l'utente amministratore, c'è l'opzione di modifica profilo o di blocco profilo.

#### *g) Pagina login, registrazione e impostazioni account*

La pagina di login è quella che permette l'accesso allo spazio personale utente ed è accessibile tramite un link nel menù di navigazione. Viene chiesto all'utente di inserire il suo nome utente o la sua mail con cui si è iscritto, e la sua password. Contiene anche: un link verso la pagina di iscrizione per gli utenti non iscritti, un link ad informazioni e aiuti per gli utenti in difficoltà, e un link per gli utenti che hanno dimenticato la password, che consiste nel simulare l'invio di una mail di istruzioni per il ripristino della password, in quanto il server della scuola non permette l'invio di mail.

La pagina di registrazione serve ad registrare un nuovo utente, ed è accessibile da un link nella pagina di login, questo perché la probabilità che un visitatore voglia iscriversi subito quando entra nel sito, è bassa. Nel form della pagina viene chiesto all'utente di inserire il suo nome, il suo cognome, un indirizzo di posta elettronica, un nome utente che desidera e una password. Per gli utenti in difficoltà, c'è un link verso informazioni e aiuti.

La pagina di impostazioni account permette all'utente di modificare le sue informazioni inserite durante la sua iscrizione, tranne il nome utente perché cambiare nome utente implica andare a modificare molti file XML e tabelle del database, ad esempio la lista degli utenti che segue o dei suoi seguaci, o ancora gli annunci pubblicati, se sono tantissimi, ci potranno essere gravi conseguenze sull'efficienza dell'operazione, ho quindi tralasciato questa funzionalità. Nella pagina c'è anche un link per cancellarsi dal sito. Questo non viene permesso agli amministratori, in quanto la loro cancellazione avverrà in php admin.

#### *h) Pagina Informazioni e FAQ*

Questa pagina è la pagina che contiene tutte le informazioni, tutorial, guide, e domande frequenti degli utenti. E' una pagina statica che dovrà purtroppo essere modificata anche lei se alcuni tag dell'header del sito cambiano sintatticamente. Questa poca manutenibilità del codice è dovuto al fatto che se avessi voluto generarla dinamicamente attaccando l'header al contenuto e facendo echo alla fine, questo non mi permetteva di fare l'echo andando direttamente su un elemento di un certo id, e questo era un problema per linkare tante pagine esterne e dinamiche verso informazioni ed aiuti precisi, l'utente si sarebbe dovuto scorrere tutta la pagina prima di arrivare all'informazione di proprio interesse. Per quelli che visitano direttamente la pagina dal link presente nel menù di navigazione, senza passare per link esterni, si troverà prima del lungo testo un indice ai contenuti.

### 3. Layout

L'interfaccia principale del sito è la classica interfaccia a 3 pannelli per rispondere alle tre domande dell'utente : Dove sono? Di cosa si tratta? Dove posso andare?. Alla prima domanda rispondono il tag title della pagina, il breadcrumb e la brevissima presentazione del sito nell'header. Alla seconda domanda risponde il contenuto della pagina. Infine alla terza domanda rispondono i menù di navigazione(principale ed eventualmente quello utente). Vengono inoltre forniti strumenti di interazione come la ricerca, sempre presente nell'area visibile.

Il layout utilizzato, è un layout a variabilità controllata, usando misure relative in percentuali per gli elementi che si devono adattare alle dimensioni del browser, e misure relative in em per fissare le proprietà dimensionali di elementi che non si devono adattare alle dimensioni dello schermo, ad esempio alcuni bottoni. Il primo punto di rottura scelto, è una misura in pixel della larghezza media in modalità landscape, dei device mobili target ad oggi, e cioè 1064px. Da quel punto in giù vengono definiti certi intervalli dove vengono applicati certi stili CSS. In generale, gli stili applicati sono sull'eliminazione dei margini e il riadattamento visivo della sequenza logica degli elementi che avevano dimensioni massime minori di 90% al fine che si possano vedere meglio. Nella versione desktop dai 1065px in su, per avere più cose nell'area visibile, sono stati applicati tanti stili grafici float, che in un device dallo schermo ridotto non devono esserci, per cui vengono tolti questi stili. Per adattare meglio le misure relative alle dimensioni della finestra, è stato inserito il meta tag viewport con width=device-width. Viene inoltre fornito un layout per la stampa delle pagine.

### 4. Metafora della pesca

Per non deludere l'utente che naviga nel sito, a seconda che si tratti di una situazione di tiro perfetto, trappole per aragoste, pesca con rete o di boa di segnalazione, ho fornito navigazione, ricerca, informazioni e aiuti. Gli utenti possono navigare tra i tipi e le categorie di eventi grazie alle pagine dedicate accedibili con i link presenti nel menù di navigazione sempre visibile tranne nelle pagine dove bisogna compilare una form. Possono cercare un annuncio preciso grazie alla form di ricerca presente anche lei nel menù di navigazione principale del sito. Inoltre esiste la ricerca avanzata per soddisfare le richieste di tutti. Per informazioni, aiuti e domande, ho fornito loro una pagina che raccoglie quelli che potrebbero essere domandati con alta probabilità, e questa pagina è sempre presente nel menù di navigazione principale per consultazioni se necessario. **Sul sito non è permesso la ricerca di utenti iscritti agli utenti che non sono connessi al loro spazio personale, questa decisione è dovuta al fatto che non è l'obiettivo prefissato del sito e voglio evitare ambiguità tra sito di annunci e social network.** La ricerca di un utente è permessa agli utenti che sono in possesso di una rete sociale dentro al sito, e per avere una rete sociale bisogna essere iscritti.

**La ricerca di annunci per parola chiave** potrebbe avere insuccesso qualora l'annuncio non possedesse etichette, tipo o categoria che contenga la parola chiave. Il motivo per cui non ho implementato la ricerca per parola chiave sul testo della descrizione dell'evento o su quello dell'introduzione dell'annuncio, è perché non garantiscono che l'annuncio trovato sia in contesto con quello che cerca veramente l'utente. Infatti l'utente potrebbe scrivere una sigla del tipo "ATO", con l'intenzione di trovare annunci di un argomento specifico di questo nome, e potrebbe trovarsi tanti annunci che non c'entrano niente con tale, ma sono stati mostrati perché contengono la parola ATO come sillaba di numerosi verbi al passato, ad esempio: trovato, mangiato ecc... L'utente si ritroverebbe a leggerli tutti per trovare quello che cerca veramente, magari anche senza trovarlo, e rimarrebbe deluso da questa esperienza. Potrebbe servire all'utente che ha una vaga idea di quello che vuole, ma per accontentare tutti, non l'ho implementata lasciando la navigazione per questi utenti. Tuttavia tale ricerca è fornita al pubblicatore di annunci per aiutarlo a cercare meglio tra i suoi annunci pubblicati.

### iii. Javascript

Nel sito viene usato Javascript per due funzionalità: Il controllo input e il menù a tendina. Ho cercato di scrivere un codice più efficiente ed efficace possibile per le risorse limitate dei telefoni. Gli input per leggere dati dal database, ad esempio la ricerca di annunci, non vengono controllati al lato client, ma al lato server. Tuttavia i controlli fatti su questi input al lato server si limitano all'escaping delle stringhe per evitare attacchi di tipo Sql injection. Gli input che vengono controllati sia al lato client che al server, sono input che vanno a modificare il database, che nel sito si trovano nella pagina di inserimento/modifica di un annuncio, nella pagina di gestione foto di un annuncio, nella pagina di

iscrizione e nella pagina di impostazioni account. Il eseguito da Javascript mira a controllare che i campi obbligatori e facoltativi siano stati compilati con valori solo accettabili, cioè valori che potrebbero avere un senso, questo perché è difficile dire che quello inserito dall'utente è comprensibile e non sono solo lettere casuali. In certi campi come l'ora e la data è facile implementare il controllo completo, ma in campi come il titolo e la descrizione dell'evento, il controllo si limita a verificare che il campo non sia vuoto. Non controlla gli input relativi alle etichette di un annuncio nuovo o modificato, in quanto gli inserimenti di valori strani avranno gravi conseguenze sull'apparizione dell'annuncio nei risultati della ricerca, viene quindi fatto solo un escaping delle stringhe al lato server prima di inserirle nel database. Il codice Javascript scritto per il controllo degli input, è simile a quello visto in laboratorio: Ho creato degli array multidimensionali globali che hanno come chiave gli id degli input da controllare, con relativa espressione regolare. Gli array multidimensionali sono 2, uno per gli input che necessitano un tipo di controllo comune, e uno per gli input della data e dell'ora che richiedono ulteriori controllo. La funzione di controllo viene chiamata onsubmit sulla form con un parametro attuale di tipo booleano che se true dice che la form in questione contiene degli input di data e ora, perciò devono essere fatti dei controlli su questi, altrimenti salta il controllo su questi. La funzione di controllo svolge il suo compito chiamando ulteriori funzioni di controllo in base all'array di input da controllare, Queste funzioni ritornano degli array di stringhe di errori che vengono concatenati in un array unico, se questo array ha length > 0, allora c'è stato almeno un errore e viene chiamata una funzione che inserisce quella lista di errori in un div con id, e viene piazzato in alto alla pagina portando il focus sulla lista, e colorando in rosso gli input che hanno generato l'errore tramite un'assegnazione di una classe CSS. Si nota che per il controllo delle foto non ho dichiarato nessun array globale, mi sono limitato a dichiarare una variabile globale che contiene il numero massimo di foto caricabili. Questa scelta si basa sul fatto che il controllo sulle foto è sul formato dei file e delle descrizioni date. La funzione incaricata del controllo delle foto, controlla prima se il file caricato è teoricamente un'immagine, se lo è allora controlla se è stata data una descrizione della foto. Purtroppo è difficile scrivere un'espressione regolare per la descrizione foto per sapere se quello inserito ha senso, perciò viene solo controllato se la stringa è vuota, mentre viene lasciata la possibilità all'utente non vedente di segnalare l'annuncio in caso di contenuto insensato. La manutenibilità della funzione non è un problema in quanto basterà modificarla leggermente in caso di cambio di id delle foto, un costo che trova vantaggio nella sua complessità asintotica che è lineare nel numero di foto caricabili.

La seconda parte di JavaScript riguarda il menù a tendina. L'implementazione di questa funzionalità è dovuta ai menù di navigazione, sia quello principale che quello utente, che sono lunghi ed occupano molto spazio nell'area visibile di un telefono, obbligando l'utente a scrollare sempre per avere il contenuto, la soluzione era quindi quella di nascondarli quando non servono e di mostrarli a richiesta, quindi creare i menù a tendina. Se Javascript è disattivato, i menù a tendina non appariranno e si avrà il classico menù fisso allungato che copre tutta l'area visibile. Se invece i menù a tendina erano presenti prima della disattivazione di Javascript, non spariranno finché non venga ricaricata la pagina. Al caricamento del body viene chiamata una funzione che controlla la larghezza dello schermo o del browser, se questa supera la soglia definita da una variabile globale, che rappresenta il breakpoint medio in modalità landscape di alcuni dispositivi android e iphone, che ho fissato a 1064px, vengono prima presi gli elementi dal codice HTML, identificati da una classe, che diventeranno quelli cliccabili con onclick uguale alla chiamata della stessa funzione. Per motivi di accessibilità questi elementi non sono link ma intestazioni alla lista delle voci, quindi "Menù", "Menù spazio personale" e "Menù spazio amministratore". A questi elementi vengono assegnati una classe css che modifica la loro presentazione sotto una media query, aggiungendo un colore blu uguale al colore convenzionato per gli input di tipo submit, e una freccetta che indica che qualcosa apparirà sotto se si clicca. Poi viene preso la lista del menù e le viene applicata una classe che la nasconde dall'utente vedente finché l'utente non clicca su menù. Quando l'utente clicca su menù, verrà chiamata la stessa funzione che verificherà di nuovo la dimensione dello schermo o del browser, e se si verifica il breakpoint, guarda se la lista del menù ha la classe che nasconde assegnata, se ce l'ha allora la rimuove, altrimenti la assegna. Questo comportamento consiste nel mostrare e nascondere le voci del menù. Quando invece non si verifica il breakpoint, la funzione riporta tutto come prima, cioè come se fosse in modalità laptop con dimensioni sufficienti per l'area visibile. Per svolgere al meglio il suo compito, la funzione chiama funzioni ausiliare in modo da avere un costo lineare nel numero di menù di navigazione presenti nel sito, che in questo caso son al più 2, quindi un costo medio costante. Si fa notare che quando il menù viene nascosto, viene nascosto solo agli utenti vedenti, ma non a quelli non vedenti.

Il sito funziona perfettamente senza Javascript, in quanto è stato applicato il principio di trasformazione elegante. Il controllo degli input è ripetuto al lato server e i menù a tendina sono solo un' accessorio per avere più contenuti nell'area visibile del telefono. Inoltre vengono offerti strumenti di navigazione in più come il classico torna su e i link per saltare elementi.

#### **iv. PHP**

Il pattern architetturale su cui riposa il sito è il Model-View-Controller. Le view sono i file html che interagisco con l'utente, il Model è un insieme di classi php i cui oggetti ricevono certe richieste di operazioni e ne restituiscono il risultato, infine il Controller funge da intermediario tra View e Model.

##### **1. Il database**

Per il database ho fatto uso di tabelle relazionali Mysql e di file XML. Le tabelle Mysql che ho creato, sono principalmente 6:

- ➔ la tabella "utente" è una raccolta di dati degli utenti iscritti sul cognome, nome, nome utente, email, password, percorso al filexml, data ultimo accesso, booleano del possesso dei privilegi premium, booleano del blocco dell'account ed eventuale giustificazione del blocco.
- ➔ la tabella "evento" è una raccolta di dati degli annunci di eventi sul codice identificativo, la data dell'evento, l'ora dell'evento, la data di pubblicazione, il pubblicatore, il titolo, la breve introduzione, il tipo di evento, la categoria di evento, la città che ospita l'evento, le etichette(sottocategorie generate dal pubblicatore), il percorso al file xml dell'evento, booleano sul blocco ed eventuale motivo del blocco.
- ➔ la tabella "tipoevento" è una tabella che raccoglie i tipi di eventi disponibili.
- ➔ la tabella "categoria" è una tabella che raccoglie le categorie di eventi disponibili.
- ➔ la tabella "primopiano" è una tabella che contiene il codice identificativo e il pubblicatore degli annunci di eventi messi in primo piano.
- ➔ la tabella "segnalazione" è una tabella che contiene le segnalazioni ad annunci, da parte degli utenti.

Il motivo per cui ho creato ed ho deciso di memorizzare queste informazioni, si basa su una stima della probabilità che un'informazione venga richiesta frequentemente. Ad esempio il tipo dell'evento e il titolo dell'annuncio ha una frequenza di apparizione nel sito elevata, per cui immagazzinarli in un database Mysql, migliora la velocità di reperimento dell'informazione con delle queries efficienti. Nel database ci sono anche tabelle non scritte da me, ma prese da un sito, che rappresentano le città, i comuni e le province in Italia, le ho usate per poter controllare gli input sul campo città dei nuovi annunci pubblicati. Infatti queste non vengono messe in relazione con nessuna delle mie tabelle, in quanto contengono un numero di record elevato, e quindi mettere un vincolo di chiave esterna con queste tabelle, avrebbe avuto una conseguenza sulla velocità delle queries. Ho scelto quindi di usarle per verificare se l'input inserito dall'utente sul campo città, o sul campo provincia, è veramente una città o una provincia.

Il motivo per cui ho usato anche i file XML per l'utente e l'annuncio, si basa sulla frequenza dell'informazione e per evitare troppe tabelle oppure colonne di tabelle. Ogni utente possiede una cartella personale che memorizza i file XML relativi ai suoi annunci privati e alle sue partecipazioni ad eventi, e una cartella che contiene le foto caricate dei suoi annunci. Nel file XML dell'utente, vengono memorizzati gli id degli annunci di eventi ai quali partecipa, se avessi deciso di inserire questa informazione nel database relazionale, le soluzioni sarebbero di creare una tabella partecipazione che contiene tutti i partecipanti ad un dato evento, oppure di inserire la colonna "partecipazioni" nella tabella utente per contenere l'informazione, in entrambi i casi le queries di lettura si rivelano sempre meno efficienti al crescere della tabella, mentre con il filexml l'operazione ha un costo costante ammortizzato. Inoltre troppe tabelle e troppe informazioni sono poco mantenibili e gestibili. Lo stesso ragionamento ho fatto per il file XML dell'evento che contiene altre informazioni caratteristiche(descrizione totale, tag img per le foto) e i partecipanti all'evento. Ci sono inoltre altri due file XML per ogni utente registrato, che sono uno per memorizzare gli utenti della sua rete sociale, e

uno per le ultime notizie. Per ogni tipo di file XML, esiste un file XML modello, presente nella cartella Modello, il quale viene copiato nella cartella dell'utente quando si iscrive la prima volta.

## 2. Classi e interfacce php

Le interfacce sono dei file php che ho scritto per permettere alle views di comunicare con il Controller al fine di avere una completa separazione tra struttura e comportamento del sito. L'utente interagisce con le pagine, tramite i link e le form, richiedendo una certa operazione che viene inviata con metodo get o post all'interfaccia corrispondente, la quale chiama un metodo della classe Controller, grazie ad un suo oggetto creato e salvato nell'array SESSION. Tale metodo in generale chiede il risultato dell'operazione ad una classe del modello, ad esempio se è un'operazione utente, allora verrà chiamato un metodo dell'oggetto della classe utente. Una volta ricevuti i risultati, il Controller aggiorna la pagina per informare l'utente. I file interfacce sono trovabili nella cartella radice del progetto consegnato.

Ci sono 6 classi nel Modello e 1 classe per il Controller, e sono:

- ➔ la classe *"DB\_handler"*: è una classe del modello i cui oggetti sono degli intermediari tra il database e gli utenti del database. permettono grazie ai loro metodi di svolgere operazioni di lettura o di modifica dei dati nel database, a seconda permessi di accesso dell'utente.
- ➔ La classe *"domoperations"*: è una classe del modello i cui oggetti permettono di eseguire delle operazioni di lettura e modifica su file XML.
- ➔ La classe *"utente"*: è una classe del modello i cui oggetti rappresentano utenti iscritti connessi.
- ➔ La classe *"evento"*: è una classe i cui oggetti rappresentano annunci di eventi pubblicati.
- ➔ La classe *"formsTag"*: è una classe astratta del modello che rappresenta l'insieme dei tag interattivi discendenti del tag form. Ad esempio: input, textarea, select ecc.. Da questa classe derivano delle classi concrete, che sono input, textarea e select. Il cui compito è quello di semplificare l'aggiunta di questi tag nelle form create a runtime dal controller.
- ➔ La classe *"views"*: è stata creata al fine di fornire delle costanti e delle funzioni statiche, utili al controller per creare pagine html.
- ➔ La classe *"Controller"*: è una classe i cui oggetti rappresentano degli intermediari tra le pagine che interagiscono con l'utente, e il modello. un oggetto controller crea/aggiorna dinamicamente le pagine in base alle richieste e ai risultati delle operazioni ritornate dal modello. Un oggetto di tipo controller viene creato ogni volta che un utente si collega al sito, e viene distrutto quando l'utente chiude la finestra del browser.

Tutte queste classi sono contenute in diversi file PHP presenti nella cartella Modello, e sono stati commentati per evidenziare il ruolo di ciascun metodo, funzione statica, variabile statica e costante di classe.

## v. Sicurezza

Per quanto riguarda la sicurezza, ho provveduto ad implementare alcune funzionalità per proteggere i dati degli utenti e il database da operazioni malintenzionate. Ad ogni utente viene chiesto di inserire una password quando si iscrivono per la prima volta, quest'ultima viene cifrata con una funzione PHP disponibile con la segnatura: `string password_hash ( string $password , integer $algo [, array $options ] )`, dove si è usato un algoritmo di cifratura di tipo Blowfish. Nessuno è in grado di decifrare la password, a parte l'utente stesso, senza sapere prima il testo in chiaro. Quando l'utente esegue l'accesso, gli viene chiesto di inserire il testo in chiaro della sua password nell'input di tipo password, quindi non visibile. Questa password viene inviata al server con metodo post assieme al suo nome utente, il server poi esegue lo script dell'interfaccia php che riceve i dati (il file `spaziopersonale.php`), in cui c'è una chiamata al metodo `spazioUtente()` dell'oggetto di tipo Controller salvato dentro l'array `$_SESSION`. Questo metodo del Controller crea temporaneamente un oggetto di tipo utente chiamando il costruttore a 2 parametri (username e password), nel quale

viene inizializzato il campo dato di tipo `DB_handler`, che verifica la correttezza dei dati ricevuti tramite una chiamata al proprio metodo `esisteUtenteConPassword($idu, $password, $blocked = FALSE, $admin = false)`. In questo metodo viene verificato che il nome utente, o la mail, `$idu` esiste, in caso affermativo controlla, con la funzione PHP `password_verify`, se il testo in chiaro `$password` corrisponde alla password cifrata attualmente in database dell'utente `$idu`, in caso affermativo viene restituito il nome utente e la password cifrata dell'utente, altrimenti ritorna un array vuoto, al costruttore della classe utente il quale inizializza gli altri campi dati dell'utente se esiste l'utente, oppure li lascia indefiniti. Il Controller poi controlla se l'utente è stato inizializzato correttamente con il metodo `isConnected()`. Se ritorna true, il Controller aggiorna la finestra portando l'utente al suo spazio personale e salva dopo l'oggetto utente creato che lo rappresenta, dentro l'array `$_SESSION` per future operazioni lungo tutta la sua navigazione nel sito. Se ritorna false, il Controller aggiorna la finestra della login dando un messaggio di errore.

Per aumentare la sicurezza di alcuni dati sensibili, e proteggere il database, tutte le queries sono state incapsulate in metodi della classe `DB_handler`, alcuni di questi metodi chiedono l'username e la password, o semplicemente il nome utente dell'utente connesso, ecco perché la password cifrata dell'utente connesso viene memorizzata in un campo dato dell'oggetto di tipo utente che lo rappresenta durante la sua navigazione. Il fatto che alcuni metodi chiedano la password e altri no, è dovuto al fatto che questi metodi operano modifiche sugli annunci o dati personali dell'utente, oppure restituiscono dati sensibili dell'utente, mentre gli altri metodi sono principalmente di lettura, a volte anche di scrittura, poco influente. Ci sono anche funzioni statiche che svolgono solo operazioni di lettura su tabelle extra che non contengono dati particolari.

Gli input inseriti dagli utenti subiscono anche loro un controllo per evitare attacchi di tipo Mysql injection. Innanzitutto viene fatto l'escaping di tutti i parametri attuali di tipo string che vengono usati nelle queries sia di lettura che di scrittura, grazie al metodo PHP `real_escape_string` dell'oggetto `mysqli` che rappresenta la connessione al database. Questo per evitare gli attacchi Mysql injection. Ho scritto poi una funzione statica `escapeCharacters($stringa)`, dentro alla classe `DB_handler`, per fare l'escape di alcuni caratteri XML (& " ' < >) dentro alle stringhe da inserire nel database. Questo per evitare che l'utente inserisca dei tag HTML che potrebbero invalidare le pagine.

Nel sito ci sono 4 form che vanno ad inserire o modificare dati nel database: la form di iscrizione, la form di impostazioni account, la form per l'inserimento o modifica di un annuncio e la form di gestione foto. Tutti gli input di tipo text, password o textarea hanno un numero minimo e massimo di caratteri. Nel caso in cui un utente malintenzionato si divertisse a cambiare questi valori nel codice HTML, riceverà dei messaggi di errore prima con Javascript, e se disattivo interviene il server. I controlli effettuati dal codice Javascript sono molto limitati perché controllano soltanto se i valori degli input hanno un certo senso, quindi rifiuta campi vuoti o campi non numerici quando non richiesti e viceversa, ma non controlla mai il massimo dei caratteri. Tale controllo è effettuato dal server, oltre a ripetere gli stessi controlli effettuati da Javascript. Per certi valori inseriti, è facile controllare se hanno senso, ad esempio il giorno, il mese e l'anno, mentre per altri è complicato verificarlo, ad esempio alcuni campi di inserimento o modifica di un annuncio come la descrizione dell'evento. Una soluzione sarebbe quella di implementare un algoritmo di intelligenza artificiale, ma per i mezzi e il tempo che mi sono stati dati, non era realistico. Ho quindi optato per un'altra soluzione che è quella di stabilire delle regole di pubblicazione e di lasciare gli utenti dirmi con segnalazioni se certi contenuti li rispettano. Un annuncio viene messo in quarantena quando viola almeno una di queste regole, lasciando 15 giorni di tempo al pubblicatore per modificarlo, altrimenti viene eliminato, con eventuale blocco dell'utente se viola continuamente le regole. Il numero di segnalazioni che può ricevere un annuncio è fissato a 20, oltre questa soglia, le segnalazioni vengono ignorate, e quindi non memorizzate nel database, così da evitare di saturare il database di false segnalazioni.

Un altro problema che si è posto, è stato quello di fermare l'utente quando supera il massimo numero massimo di caratteri nei textarea. La tecnologia XHTML strict non permette di inserire l'attributo `maxlength` nel tag `textarea`, ma HTML5 sì. Purtroppo HTML5 non è compatibile con Internet Explorer 8 che mi sono prefissato di raggiungere, e fare il controllo in Javascript presentava alcuni problemi di accessibilità per avvisare l'utente del limite superato. Ho quindi scelto di notificare l'utente del numero massimo di caratteri da non superare nel label del textarea, come tutti gli altri tag di modifica testo, e di effettuare un controllo al lato server che consiste nel calcolare il numero di caratteri ricevuti e di

restituire un messaggio di errore personalizzato che dice di quanto si è superato il limite, al fine di dare un'idea all'utente del numero di caratteri da cancellare.

Per quello che riguarda i file XML privati, alcuni file XML sono accedibili tramite password o nello scoping della rete sociale dell'utente proprietario, ad esempio i file per le notizie, mentre altri sono di dominio pubblico, come i file degli annunci, per solo lettura e modificabili tramite password.

### III. ACCESSIBILITÀ E USABILITÀ

Per avere un sito accessibile e usabile, ho prima validato tutte le pagine, sia quelle dinamiche ( nei diversi modi in cui possono presentarsi, ad esempio quando contengono informazioni e quando non le contengono), che statiche. Ho anche mantenuto la completa separazione tra struttura, presentazione e comportamento del sito. E infine ho cercato di seguire le seguenti linee guida WCAG 2.0:

#### 1. Percepibile

- 1.1 *Alternative testuali*: Tutti gli elementi non testuali presenti nel sito hanno un'alternativa testuale. Viene obbligatoriamente chiesto agli utenti di fornire un'alternativa testuale alle immagini caricate per i loro annunci, e tutti i tag interattivi delle form (input e textarea) , hanno un tag label associato che indica la finalità e dà indicazioni sull'input da inserire. Tutti gli altri elementi non testuali di tipo decorativo sono applicati tramite CSS.
- 1.2 *Alternative tipi di media temporizzati*: Il sito non usa media temporizzati.
- 1.3 *Adattabile*: Per applicare questa linea guida, ho scritto i contenuti delle pagine HTML in una sequenza logica che non confonda e disorienti l'utente quando il CSS non c'è. Ho fatto uso : delle intestazioni per introdurre un sottocontesto del contenuto; dei tag strong per comunicare un'informazione importante; dei tag em per dare un suggerimento. I glossari e le caratteristiche di un elemento sono stati raggruppati in un tag lista, ul per quelli non ordinati, e ol per quelli ordinati, tutto questo per assicurarmi che il senso del contenuto percepito dall'utente vedente, sia uguale a quello che non vede e che sia corretto. Infine i colori per i link visitati sono stati lasciati ai loro colori standard mentre i link non visitati hanno ricevuto il colore standard un po' scurito. Infine tutti i tag interattivi ricevono dei colori distinguibili per i link e per gli input quando sono in focus o attivi, tutto ciò per non disorientare l'utente vedente. Non ho usato parole che fanno riferimento a posizioni nella pagina(ad esempio clicca in alto ), o a delle forme di oggetti o immagini o ancora colori(ad esempio clicca sul pesciolino rosso in mezzo bottone rotondo).
- 1.4 *Distinguibile*: Nel sito, nessuna informazione viene rappresentata basandosi solo sul colore del testo. Il testo cambia colore solo in base al colore del sottofondo in modo da avere un rapporto contrasto luminosità elevato. I colori principali usati per il testo sono nero per sottofondi chiari e bianco per sottofondi scuri. Non sono state usate immagini di testo e nemmeno sottofondi sonori. Il font-size del testo è stato settato con il CSS a 105% per i dispositivi da 1065px in su di width, e 120% per i dispositivi con dimensioni minori. Il testo può essere ridimensionato fino a più di 200 % senza perdita di contenuto e funzionalità, si passa solo in modalità mobile. Per quanto riguarda il contrasto, mi sono assicurato che il rapporto sia sempre di almeno 7:1.

#### 2. Utilizzabile

- 2.1 *Accessibile da tastiera*: Tutte le funzionalità del sito si ottengono cliccando su link ed input che sono accessibili tramite tastiera, schiacciando tab per esempio, inoltre per gli utenti vedenti, essi cambiano colore quando ricevono il focus. Non viene richiesto nessun tempo specifico per le battiture.

- 2.2 *Adeguata disponibilità di tempo*: Nel sito non viene usato nessun tipo di informazione temporizzata e le pagine non si aggiornano in modo automatico .
- 2.3 *Convulsioni*: Il contenuto del sito è testuale, con alcune foto di annunci, non vengono usati lampeggiamenti, o immagini che possono creare attacchi epilettici.
- 2.4 *Navigabile*: Per aumentare la navigabilità del sito, ho inserito dei link interni che permettono di saltare blocchi di contenuti che si ripetono su più pagine, ad esempio il menù di navigazione e l'header, ma anche di contenuti che potrebbero non interessare l'utente che conosce già abbastanza il sito. Alcuni link che hanno maggiore frequenza di utilizzo hanno l'attributo accesskey. Inoltre ogni pagina, ad eccezione delle pagine di accesso, iscrizione, inserimento annuncio, gestione foto e impostazioni account, contiene il breadcrumb che indica la posizione parziale dell'utente tra tutte le pagine partendo dalla Home page, mentre tutte le pagine senza eccezioni contengono il tag title che descrive l'argomento o la finalità di quest'ultima. Tuttavia il meta tag description appare uguale su molte pagine in quanto usano lo stesso file header, visto che sono generate dinamicamente in PHP, perciò ho cercato di essere il più breve e preciso possibile nel tag title che cambia dinamicamente a seconda delle pagine, per dire all'utente dove si trova e quale contenuto si aspetta di trovare. Le pagine molto lunghe, come la pagina faq e il modulo di inserimento annunci, vengono divise in sezioni con intestazioni in modo da aiutare l'utente non vedente a farsi una mappa del contenuto. Per quanto riguarda l'ordine del focus, il contenuto di ogni pagina è stato scritto con una sequenza logica di elementi, usando tag che rispettano il contesto(ad esempio lista, intestazione ecc...), in modo da essere comprensibile senza foglio di stile applicato. Questo mi ha permesso di avere un ordine di focus sugli elementi che possono riceverlo, link ed input, che non disturba il significato, ed evita l'uso dell'attributo tabindex che è poco mantenibile. Per fare in modo che il contenuto della pagina si capisca soltanto dagli elementi che possono ricevere il focus, link, input, select, textarea ecc..., ho fornito un label ai tag delle form con testo più esplicito e comprensibile possibile, e lo stesso per ogni link ho fornito un testo che permetta di comprendere lo scopo del collegamento. **Alcuni di link potrebbero sembrare poco significativi e alcuni label potrebbero sembrare poco espliciti all'utente non vedente, tuttavia parte del testo è stata nascosta all'utente vedente per motivi di spazio o per non essere troppo ripetitivo visto che ha una panoramica visiva di tutta la pagina, e quindi vede informazioni in più che l'utente vedente che sta navigando con il tab non ha sentito con lo screen reader, perciò questa informazione gli viene aggiunta e nascosta all'utente vedente.** Ogni volta che un elemento interattivo riceve il focus, cambia il colore dei bordi, o del testo(i link).

### 3. Comprensibile

- 3.1 *Leggibile*: La lingua delle pagine è impostata ad italiano con codifica UTF-8, in quanto il sito è destinato ad utenti italiani, o che sappiano bene la lingua. Le parti in lingua straniera sono messi in tag di linea con attributi di lingue(xml:lang), ad esempio le parole "My", "Home" e "premium". Ho cercato di essere il più chiaro e comprensibile senza usare termini troppo tecnici, e avere un livello di lettura non superiore al livello di istruzione secondaria inferiore. Tutte le abbreviazioni hanno l'attributo title per contenere il suo significato.
- 3.2 *Prevedibile*: I meccanismi di navigazione ripetuti su molte pagine come i menù di navigazione, sono sempre nello stesso ordine e identificati univocamente in quanto viene usato un file unico che cambia solo di contenuto, così l'utente sa quali blocchi saltare per avere il blocco desiderato. Gli elementi che ricevono il focus non cambiano il contesto della pagina, almeno che sia l'utente a volerlo, ad esempio inviare i dati di una form compilata, o cliccare su un link.
- 3.3 *Assistenza nell'inserimento*: Ogni form contiene un certo insieme di fieldset con legenda, che sono raggruppamenti logici di input con label. Nei label vengono forniti le istruzioni per compilare l'input corrispettivo, e per molti viene anche fornito un esempio. Inoltre, in ogni form, c'è un link verso ad una guida più approfondita presente nella pagina informazioni e domande frequenti. Nelle form di inserimento, presenti nelle pagine di iscrizione, inserimento annuncio e gestione foto annuncio, quando si verifica un errore, gli



elementi che hanno causato l'errore vengono identificati e descritti tramite testo con suggerimenti per correggere, se Javascript è attivo, l'utente vedente vedrà bordi di questi elementi, segnati con un altro colore. L'utente ha inoltre la possibilità di modificare i dati inseriti senza ricominciare da capo qualora si verificasse un errore di inserimento o che voglia aggiungere/rimuovere/sostituire contenuto, grazie ad un metodo della Controller che permette di rimostrare la form con tutti i dati inseriti dall'utente.

#### **4. Robusto**

L'intero sito riposa sulla tecnologia XHTML strict ed è stato validato sul sito [validator.w3.org](http://validator.w3.org) con input diretto in quanto molte pagine sono dinamiche. E' stata mantenuta una netta separazione tra struttura, presentazione e comportamento. Tutto ciò per garantire una massima compatibilità con i programmi utenti che mi sono prefissato di raggiungere, ma anche per quelli futuri comprese le tecnologie assistive.

Per aumentare l'usabilità, oltre ad aver cercato di applicare le linee guida del WCAG 2.0, ho cercato di migliorare l'esperienza utente con gli schemi e strutture organizzative spiegate prima, con tempi di risposta delle ricerche più brevi possibili dividendo i risultati in pagine di record limitati per ogni pagina. Ho anche cercato di fornire una grafica attraente.

### **IV. TEST ESEGUITI**

Per assicurarmi di aver raggiunto gli obbiettivi prefissati, ho testato il sito sui browser Firefox, Google Chrome, Microsoft Edge, Opera, e Internet Explorer dalla versione 8 in su, nel sistema operativo Windows 10. Non ho potuto testarlo su Safari perché non sono riuscito a venire in possesso di un computer apple. Il sito è stato anche testato su sistemi operativi linux, Ubuntu per la precisione, presenti nei laboratori dell'università.

Per quanto riguarda i dispositivi cellulari che mi prefissato di raggiungere, un vero test è stato eseguito su uno smartphone di fascia media, Huawei p8 lite 2017, mentre altri test sono stati fatti sull'opzione per sviluppatori presente in Google Chrome, che simula vari dispositivi mobile.

In più alcuni test di sicurezza sono stati fatti, solo i test di Mysql Injection e dei controlli input di valori corretti senza passare per pagine HTML, e anche disattivando JavaScript.

Il sito risponde bene a questi test fatti, tranne la stampa della pagina informazioni e domande frequenti, e la pagina di inserimento di un nuovo annuncio su Firefox, per un bug del browser.

Per quello che è dei test di accessibilità, ho prima disattivato JavaScript e tolto il CSS alle pagine, per vedere se il contenuto aveva senso e si potesse navigare da tastiera. Ho anche usato lo screenreader ChromeVox, un'estensione per sviluppatori scaricabile e installabile su Chrome, per testare la navigazione non vedente.