

Relazione progetto di tecnologie web

Cucinafelice

Miki Violetto matr: 1029140
Giuseppe Merlino matr: 1026694
Viviana Alessio matr: 1029720

dati di accesso al sito

indirizzo web: tecnologie-web.studenti.math.unipd.it/tecweb/~mviolett
credenziali amministratore:
username: Admin *password:* 00000000
credenziali utente semplice:
username: utente *password:* utente001

referente:

Miki Violetto
e-mail: miki.violetto@studenti.unipd.it

3 luglio 2014

Indice

1	Abstract	2
2	Analisi dell'utenza	2
3	Operazioni tipiche all'interno del sito	2
4	Divisione dei ruoli	2
5	Progettazione	3
5.1	XML schema	3
5.1.1	Perchè XML schema?	3
5.1.2	Modello adottato	3
5.1.3	Descrizione parti più significative di schemaXml.xsd	3
5.2	XML	4
5.3	HTML	6
5.4	struttura pagine - in generale	6
5.4.1	il div ricetta e il div commenti	6
5.4.2	Scelte nella stesura dell'HTML e accessibilità	8
5.5	test	9
5.6	CSS	9
5.6.1	descrizione file	9
5.7	Javascript	11
5.7.1	I controlli per la registrazione	11
5.7.2	Recupero dati	11
5.7.3	Zoom di pagina	11
5.8	Perl	11

5.8.1	librerie usate	11
5.8.2	descrizione file	12
6	Fonti	14
7	Conclusione	14

1 Abstract

Per il nostro progetto abbiamo deciso di realizzare un sito di cucina facile e veloce da consultare dove gli utenti potranno visualizzare le ricette e commentarle una volta che si saranno registrati. L'obiettivo era quello di avere pagine dinamiche con un layout responsive dall'aspetto accattivante, avendo un'occhio di riguardo per persone diversamente abili facilitandole ad un accesso più rapido alle informazioni importanti.

2 Analisi dell'utenza

Il sito si rivolge ad un pubblico ampio, quindi il linguaggio usato mira alla semplicità e alla chiarezza. Possibili categorie di visitatori:

1. La prima categoria a cui abbiamo pensato sono le **donne**, in particolare le **casalinghe**. Per loro è importante avere un sito semplice da navigare e che abbia tutte le informazioni che servono per preparare una ricetta.
2. Gli **studenti** di cucina ed eventualmente **cuochi** professionisti che cercano l'ispirazione per nuove ricette.
3. Gli **appassionati** di cucina, i quali vogliono solo vedere ricette per il semplice gusto di farlo, non necessariamente per poi realizzarle. Pensando a questi utenti abbiamo deciso di curare il CSS in modo che le pagine fossero interessanti a prima occhiata con foto e dettagli precisi.
4. Utenti casuali, provenienti ad esempio da un link esterno.
5. Gli utenti con diversi problemi come ad esempio cecità, difficoltà motorie, daltonismo.

3 Operazioni tipiche all'interno del sito

Vi sono tre tipologie di utenti: i non registrati, i registrati con privilegi di amministratore, i registrati senza privilegi.

I **non registrati** possono vedere tutte le ricette e i commenti ma non possono svolgere alcuna azione significativa, possono accedere alla pagina di registrazione.

Gli utenti che si sono **registrati** possono commentare le ricette, cancellare i propri commenti, segnalare i commenti di altri utenti, cambiare le proprie informazioni personali, recuperare le proprie credenziali.

Gli **amministratori** possono svolgere tutte le operazioni di un utente registrato, e in più possono rimuovere qualsiasi commento; inoltre vengono inseriti nella loro area personale la lista dei commenti segnalati e direttamente da lì potranno decidere se eliminarli definitivamente o se toglierli dalla lista dei segnalati.

4 Divisione dei ruoli

In generale tutti i membri del gruppo si sono aiutati a vicenda con consigli e/o correzioni del codice, collaborando sempre insieme per la buona riuscita del progetto. Inoltre tutti hanno collaborato alla stesura di questa relazione.

- **HTML:** Viviana si è occupata della struttura delle pagine, mentre Giuseppe ha curato le parti HTML contenenti le form, vista l'interazione con javascript necessaria per alcune di esse.
- **CSS:** Giuseppe ha creato i 3 fogli di stile prefissatoci.
- **JAVASCRIPT:** Se n'è occupato Giuseppe.
- **PERL:** Se n'è occupato Miki.
- **XML Schema/XML:** lo schema è stato realizzato da Miki e Viviana. I file XML sono stati popolati da Viviana.
- **ACCESSIBILITÀ:** Se ne sono occupati Viviana e Giuseppe.

5 Progettazione

5.1 XML schema

5.1.1 Perché XML schema?

Inizialmente abbiamo deciso di usare XML schema perchè volevamo che gli utenti registrati al nostro sito potessero inserire le proprie ricette. Quest'operazione necessita di controlli accurati perchè altrimenti gli utenti avrebbero potuto inserire caratteri non consentiti nei vari campi della form di inserimento.

Successivamente quindi abbiamo deciso che non ce l'avremmo fatta a curare bene come ci eravamo prefissati anche questa parte del sito e quindi abbiamo abbandonato l'idea. A quel punto però l'XML schema era già stato realizzato così l'abbiamo tenuto. Tutto lo schema è contenuto nel file schemaXml.xsd

5.1.2 Modello adottato

Abbiamo deciso di seguire il modello “tende alla veneziana” perchè ci avrebbe permesso di riutilizzare tipi. Poi effettivamente non abbiamo sfruttato molto questo aspetto, ma il compromesso tra semplicità d'uso e modularità ci ha soddisfatti.

5.1.3 Descrizione parti più significative di schemaXml.xsd

Il tipo **Tparole** l'abbiamo utilizzato in tutti quegli elementi che devono contenere una o più parole separate da uno spazio, come ad esempio il nome e il cognome dell'utente o il nome delle ricette.

```
<xs:simpleType name="Tparole">
  <xs:restriction base="xs:string">
    <xs:pattern value="\w+(\ \w+)*"/>
  </xs:restriction>
</xs:simpleType>
```

L'elemento **utenti** è l'elemento base del file utenti.xml e contiene uno o più nodi utente. È obbligatorio almeno un nodo utente in quanto il nodo dell'amministratore è sempre presente.

L'elemento **utente** contiene tutte le informazioni di un singolo utente ed è definito dal tipo **Tutente** il quale contiene i campi obbligatori per registrarsi (username, password, e-mail) e il nodo infoaggiuntive che contiene i campi opzionali. Inoltre contiene l'attributo admin che di default è settato false.

L'elemento **username** ha tipo **Tusername** il quale è una restrizione del tipo string dato dallo standard che accetta due o più caratteri scelti tra lettere maiuscole, minuscole e numeri. L'elemento **password** ha tipo **Tpassword** il quale è una restrizione del tipo string dato dallo standard che accetta da 8 a 20 caratteri scelti tra lettere maiuscole, minuscole e numeri.

L'elemento **email** ha tipo **Temail** il quale è una restrizione del tipo string dato dallo standard. In base alle nostre informazini un pattern che accetti solo e soltanto le e-mail esistenti purtroppo non esiste ancora, quindi abbiamo optato per un controllo non molto severo. Controlla infatti solo che vengano inserite e-mail accettate da questo pattern:

[a-zA-Z0-9_.-]+@[a-zA-Z0-9_.-]+\.[a-zA-Z0-9_.-]+

L'elemento **ricette** contiene zero o più nodi ricetta. L'elemento **ricetta** è definito dal tipo **Tricetta** e contiene tutte le informazioni di una ricetta (nome, tipo, descrizione, ingredienti, procedimento, ecc...).

L'elemento **commenti** contiene la lista di ricette che hanno ricevuto almeno un commento, l'attributo n.commenti viene utilizzato come contatore per l'id dei commenti.

5.2 XML

Abbiamo avuto bisogno di creare tre file XML che contenessero le ricette del sito, i commenti alle ricette, i dati degli utenti registrati.

ricette.xml

Abbiamo deciso di inserire in questo file tutte le ricette del nostro e poi di utilizzarlo come database per reperire i dati che volevamo stampare o modificare.

Il file ricette.xml come già abbiamo detto descrivendo il file schemaXml.xsd in sostanza contiene tanti tag <ricetta> quante sono le ricette e hanno questa struttura

```
<ricetta nome="...." >
  <tipo></tipo>
  <img>
    <url></url>
    <alt></alt>
  </img>
  <descrizione></descrizione>
  <ingredienti>
    <ingrediente></ingrediente>
  </ingredienti>
  <procedimento>
    <img>
      <url></url>
      <alt></alt>
    </img>
  </procedimento>
  <difficolta>bassa|media|alta</difficolta>
  <tempo_preparazione></tempo_preparazione>
  <dosi_per></dosi_per>
</ricetta>
```

utenti.xml

In questo file vengono inseriti gli utenti una volta che si sono registrati. Vi sono tanti tag <utente> quanti utenti sono registrati al sito e hanno questa struttura

```
<utente admin="true|false">
  <username></username>
  <password></password>
  <email></email>
  <infoaggiuntive sesso="M|F" >
    <nome></nome>
    <cognome></cognome>
    <abilita>principiante|intermedio|professionista</abilita>
  </infoaggiuntive>
</utente>
```

commenti.xml

In questo file vengono inseriti tutti i commenti alle ricette. Vi sono tanti tag <commento> che hanno questa struttura

```
<ricetta nome="" segnalato="">
  <commento id="..." >
    <utente></utente>
    <abilita>principiante | intermedia | professionista </abilita>
    <voto>1|2|3|4|5<voto>
    <testo></testo>
    <data></data>
  </commento> </ricetta>
```

[Tutte le ricette](#)
[Aperitivi](#)
Antipasti
[Primi](#)
[Secondi](#)
[Contorni](#)
[Dolci](#)

Cerca per difficoltà :



Polpette aromatiche al vapore

Le polpette aromatiche al vapore sono un antipasto leggero ma gustoso preparato con carne di vitello macinata e insaporita con erbe aromatiche e cipollotto. La preparazione prevede la realizzazione delle tradizionali polpette che vengono poi cotte in vaporiera con acqua aromatizzata. Le polpette aromatiche al vapore rappresentano un ottimo compromesso tra leggerezza e gusto; la cottura al vapore senza grassi aggiunti, oltre ad essere salutare, esalta i profumi delle erbe aromatiche presenti nell'impasto: erba cipollina, prezzemolo, timo, maggiorana e per finire un tocco di freschezza con il finocchietto. Ideali per un antipasto light, le polpette aromatiche al vapore stupiranno i vostri commensali con un gusto inaspettato.

Figura 1: Ricettario degli antipasti

5.3 HTML

Tutte le pagine sono dinamiche e quindi generate tramite gli script perl, comunque nella prima fase di progettazione sono state create alcune pagine statiche che ci hanno poi fatto da guida per stampare il codice XHTML 1.0 strict tramite il linguaggio perl. Tutto il sito risulta corretto rispetto allo standard W3C.

5.4 struttura pagine - in generale

In Figura 1 vi è uno screenshot preso da PC (precisamente il ricettario degli antipasti) inserito nella relazione per poter spiegare meglio la struttura base di tutte le pagine.

Abbiamo scelto un layout classico, semplice e familiare a tutti. La parte alta della pagina è occupata dal box di login e dal logo del sito che funge anche da link alla homepage. Scendendo vi è il path che contiene il percorso dalla home alla posizione corrente per far sì che l'utente sappia sempre dove si trova.

A sinistra troviamo il menu di navigazione il quale permette di accedere facilmente alle varie tipologie di ricette (aperitivi, antipasti, primi, secondi, dolci) e all'elenco di tutte le ricette.

A destra del menu troviamo il cuore del nostro sito, le ricette. A seconda della tipologia di pagina (ricetta o ricettario) vi saranno un'unica ricetta con tutte le relative informazioni oppure una lista di ricette con foto e descrizione.

5.4.1 il div ricetta e il div commenti

Per spiegare le scelte di progettazione delle pagine HTML riteniamo opportuno riportare anche qui un frammento di codice, ovvero il div ricetta e il div commenti, che sono i più significativi nelle nostre pagine dato che contengono la maggior parte dei tag HTML che abbiamo usato nel sito, ovvero:

```
<img> <ul> <form> <input>
```

Ecco il codice del div ricetta:

```

<div id="ricetta">
  <h1>(nome ricetta)</h1>
  

```

```

<div id="info">
  <p>Questa ricetta ha una <em>difficoltà</em> bassa</p>
  <p>Tempo di <em>preparazione</em> : 10 minuti</p>
</div>
<div class="ingredienti"><span class="hide"><a href="#descrizione">Salta la lista di
<h3>Ingredienti :</h3>
  <ul>
    <li>(primo ingrediente)</li>
    <li>.....</li>
  </ul>
</div>
<div id="descrizione">
  <h3>Descrizione</h3>
  <p>(testo descrizione)</p>
</div>
<div id="procedimento">
  <h3>Procedimento</h3>
  <p class="procedimento">(testo procedimento)</p>
  
  <p class="procedimento">(altro testo procedimento ...) </p>
</div>
</div>
<div id="commenti">
  <h2>Commenti</h2>
  <form method="post" id="formcommento" action="ricetta.cgi?ricetta=">
    <p id="voto">Voto</p>
    <p id="voti">
      <label for="1">1</label>
      <input type="radio" name="voto" id="1" value="1">
      <label for="2">2</label>
      <input type="radio" name="voto" id="2" value="2"></p>

    <p id="text"><label for="commenta">commenta</label>
      <textarea name="testo" id="commenta" cols="60"></textarea>
    </p>
    <p id="bottoni_form">
      <input type="submit" name="action" class="button" value="commenta">
    </p>
  </form>
  <div id="listacommenti">
    <div class="commento">
      <div class="int_commento">
        <p class="autore"> Autore: <span>miki</span>
          Abilitato;:
          <span>professionista</span>
        </p>
        <p class="data"> 15:29:27 30-06-2014</p>
      </div>
      <div class="testo_commento">(testo del commento....)</div>
      <p>Voto: <span>(voto)</span></p>
      <div class="bottoni">
        <p><input type="button" class="button"
          onclick="location.href='ricetta.cgi?ricetta=(nome)'" value="segna">
        </p>
        <p>
          <input type="button" class="button" onclick="location.href='ricetta.cgi?

```

```

        ricetta=(nome)'" value="rimuovi">
    </p>
</div>
</div></div></div>
<a id="su" href="#top">Torna a inizio pagina</a></div>

```

5.4.2 Scelte nella stesura dell'HTML e accessibilità

Le varie parti delle pagine sono racchiuse dentro a div per rendere più facile l'applicazione delle regole CSS nei fogli di stile e per dare una struttura semplice e comprensibile al codice HTML, con un'attenzione a metterle nel giusto ordine così che anche gli screen reader riescano ad accedere ai contenuti nel modo corretto, come gli utenti che hanno supporto visivo. Le regole CSS non hanno stravolto l'ordine dei contenuti.

Abbiamo poi pensato di usare dei tag **** quando il contenuto era importante ma non tanto da metterlo in un titolo. Ad esempio nel tag **** è stato inserito il tempo di preparazione e la difficoltà di una ricetta.

Altra cosa importante, è dare la possibilità a chi naviga mediante uno screen reader oppure un browser testuale di poter **saltare alcune parti della pagina** che forse non interessano all'utente o che magari sono già state lette in altre pagine e quindi la rilettura creerebbe noia. Queste parti sono il box di login (presente in ogni pagina), il menu di navigazione (presente in ogni pagina) e gli ingredienti delle ricette.

Nel codice allegato precedentemente vi è un esempio di questi salti, un link per saltare la lista degli ingredienti. Le shortcut sono state inserite dentro a tag span con classe hide che con il CSS abbiamo reso invisibile per il layout base ma visibile a dispositivi aural.

Vista la semplicità di navigazione e l'ordine dei contenuti non è stato usato il tag **tabindex** poichè non sembrava necessario. I link presenti non sono molti e i fogli di stile non modificano l'aspetto presentazionale dato dall'HTML. E' stato inserito un unico **accesskey** per la home, attivabile con alt+h oppure alt+shift+h, dipende dal browser; ritenuto utile nel caso in cui l'utente che sta navigando con uno screen reader si perdesse nel sito. La presenza dell'accesskey è comunicato tramite l'alt dell'immagine del logo, presente all'inizio di ogni pagina.

Un altro elemento utile per l'accessibilità e l'usabilità è stato quello di estendere il link alla ricetta anche alle immagini oltre che al solo titolo.

Per quanto riguarda le **form** ogni tag input è stato associato ad una label così da permettere anche a utenti con scarse capacità motorie di posizionarsi facilmente sul campo da inserire/selezionare.

★ Summary 🔍 Issues http://tecnologie-web.studenti.math.unipd.it/tecweb/~mvioletti/cgi-bin/home.cgi	
Dashboard	What was Tested
This tab shows a summary of the guidelines used to generate this report.	
Description	Checked
Accessibility	
Electronic and Information Technology Accessibility Standards (Section 508)	✓ Checked
Web Content Accessibility Guidelines 2.0 (WCAG 2.0)	✓ Checked to level AA
Browser Compatibility	
Android	✓ Checked
Blackberry	✓ Checked
Chrome	✓ Checked
Internet Explorer	✓ Checked from version 7
Firefox	✓ Checked
Netscape	✓ Checked
Opera	✓ Checked
Safari	✓ Checked
iPhone/iPad	✓ Checked

Figura 2: Test accessibilità

5.5 test

Innanzitutto abbiamo validato l'XHTML 1.0 strict e il CSS di livello 3 grazie ai validatori forniti da W3C.

Per validare i file XML e l'XMLSchema ci siamo serviti di questo validatore

<http://www.utilities-online.info/xsdvalidation>

Per testare l'accessibilità invece abbiamo usato alcuni tool presenti in rete come ad esempio

<http://www.powermapper.com/products/sortsite/checks/accessibility-checks.htm>

che ha riportato i risultati presenti in figura 2. Il sito ha livello di conformità AA ed è compatibile con tutti i browser presenti nell'immagine (alcuni li abbiamo anche testati manualmente).

5.6 CSS

Sono stati creati tre file css:

- base.css
- print.css
- aural.css

5.6.1 descrizione file

base.css

Il layout responsive principale, stilato basandosi su valori percentuali o in em per renderlo il più flessibile possibile. Come si può notare dalla figura 3, le pagine vengono adattate perfettamente a qualsiasi situazione, da pc desktop a dispositivi mobili come smartphone o tablet l'esperienza utente rimane la medesima, senza perdite o stravolgimenti dei vari contenuti. Attraverso diverse



(a) Nexus 7 (7 pollici, portrait)



(b) iPhone 5 (4 pollici, portrait)



(c) iPad mini retina (7.9 pollici, landscape)

Figura 3: Screenshot da dispositivi mobile

mediaquery si alloca al meglio il contenuto delle pagine per le diverse risoluzioni, ottimizzando così gli spazi per una migliore fruizione delle informazioni. Durante la fase di testing del foglio di stile ci si è resi conto che i diversi browser mobili tendono a ridefinire alcuni elementi delle pagine come ad esempio la forma dei bottoni, quindi per avere un design univoco su tutti si è optato per disabilitare l'aspetto dato da webkit.

print.css

Durante la progettazione del sito il team si è posto l'interrogativo su quali fossero gli elementi essenziali che l'utente vorrebbe vedere se dovesse stampare una ricetta o una pagina qualsiasi. La risposta è ricaduta su logo, path, corpo della pagina e footer finale. La rimozione della loginbox in alto a destra e del menu di navigazione non sono state fatte solo per risparmiare spazio, quindi carta e area di stampa, ma poichè la prima contiene dati che risultano irrilevanti sullo stampato ed il secondo è un aiuto visuale esclusivo alla navigazione visto che le stesse informazioni possono essere ritrovate in simile forma nel path. Sono stati rimossi i colori di sfondo del body e del corpo centrale, puntando al risparmio di cartuccia e alla chiarezza del contenuto che si va a stampare.

aural.css

Infine, come piccolo aiuto alla navigazione per browser aural e testuali in genere (lynx), sono state create alcune shortcut, che vengono attivate proprio da questo foglio di stile, per saltare la form di login, il menu di navigazione e la lista ingredienti di ogni pagina in modo che l'utente finale possa giungere al contenuto importante il prima possibile anche se il testo dovesse essere letto da uno screen reader.

5.7 Javascript

Sono state create alcune funzioni di controllo nella form di registrazione, recupero e zoom di pagina.

5.7.1 I controlli per la registrazione

Anzichè sovraccaricare di richieste inutili il server, si è pensato di gestire l'invio della form di registrazione anche dal lato client in via preventiva (ovviamente solo se il browser è abilitato per gestire il linguaggio). Gli elementi da controllare sono quei campi obbligatori contrassegnati da un asterisco, quindi l'username, la password, la conferma password e l'indirizzo e-mail. Ognuno di essi è corrisposto da una funzione che sfrutta l'evento onBlur di javascript per attivarsi ed esprimere un risultato booleano. Per procedere infine all'invio della form c'è un controllore di stati (controllatuz) che richiama controllauser, controllapass, controllapsw2 e controllamail nel caso non fossero già state eseguite da eventi onBlur, verificando i valori ritornati da ognuna di esse per valutare se procedere o meno con il submit.

5.7.2 Recupero dati

Una funzione di controllo che richiama controllamail per procedere o meno all'invio della form di recupero dati d'accesso.

5.7.3 Zoom di pagina

Durante il caricamento del body viene richiamato zoomdisabler che controlla se l'ampiezza dello schermo su cui si sta renderizzando la pagina sia minore di 600px in modo da disabilitare lo zoom alla viewport e dare un effetto webapp su smartphone o phablet.

5.8 Perl

5.8.1 librerie usate

- *CGI::Carp* per la gestione degli errori
- *CGI::Cookie* per creare e controllare i cookies
- *CGI::Session* per utilizzare le sessioni
- *XML::LibXML* per operare sui file xml più abilmente

- *Time::Piece* per ottenere la data corrente nei commenti
- *Fcntl* per bloccare i file

5.8.2 descrizione file

funzioni.pm

Per comodità abbiamo creato questo file il quale contiene delle funzioni che vengono usate in diversi file perl così da evitare di riscrivere le stesse righe di codice.

Le funzioni sono divise in due gruppi: quelle che si occupano di stampare il codice HTML e quelle che non lo fanno. Andiamo a elencarle e commentarle:

printHead si occupa di stampare l'inizio delle nostre pagine HTML, tutta la parte del tag <head> e una parte del div header (fino all'immagine del logo). Richiede il titolo della pagina e può ricevere anche keyword aggiuntive

printModuloLogin stampa il box di login presente quasi in tutte le pagine (ad esempio nella pagina di registrazione non è presente)

printPathNav produce il path e il nav facendo attenzione a non creare link circolari. Ad esempio se siamo nella pagina degli antipasti nel nav verrà omissso il link agli antipasti

printAncora stampa l'ancora che rimanda all'inizio della pagina, viene inserita solo quando serve

printFoot si occupa di stampare il footer della pagina e chiudere tutti i tag HTML necessari. Richiede l'indirizzo da inserire nel link al validatore HTML del W3C.

printCommento si preoccupa di stampare l'HTML del commento ricevuto tramite nodo <commento> con i dovuti pulsanti

previewImmagine stampa l'HTML dell'immagine ricevuta tramite nodo xml

Le funzioni che non si occupano direttamente di stampare l'HTML sono:

checkSession si occupa di eseguire tutto il carico di lavoro delle sessioni: controlla i cookie, esegue il logout, carica una sessione già aperta, crea una nuova sessione tramite i valori ricevuti dal ModuloLogin col metodo post e avverte sempre il browser che invierà in formato HTML(content-type).

cambiaData prende una data e ruota le sue componenti mostrandole nel formato che siamo abituati a vedere

cambiaGet riceve un indirizzo e decide se inserire "&" o "?" per costruire il giusto link col metodo get

operazioneCommento esegue le operazioni sul commento diverse dall'inserimento: segnalazione, rimozione della segnalazione ed eliminazione del commento

home.cgi

Questo file gestisce la pagina iniziale del sito. Si possono vedere le ultime tre ricette inserite (sfruttando il fatto che le nuove ricette verranno inserite alla fine del file xml) e gli ultimi tre commenti inseriti (ricavati tramite il loro id), sarà inoltre possibile effettuare la login o accedere alla pagina di registrazione, se si è registrati si possono segnalare i commenti visualizzati e se si possiedono privilegi di amministratore si possono anche eliminare o togliere la segnalazione (un utente può sempre eliminare un proprio commento).

ricettario.cgi

Questo file si occupa di gestire tutte quelle pagine che contengono l'anteprima di più ricette, ovvero "Aperitivi", "Antipasti", "Primi", "Secondi", "Dolci", "Tutte le ricette".

In tutte queste pagine è possibile effettuare la login, accedere alla pagina di registrazione e di recupero dati. C'è la possibilità di vedere le ricette selezionando una sola difficoltà per volta (come si può intuire da figura 1). Nelle pagine dove sarebbero comparse molte ricette abbiamo fatto in modo che se ne visualizzassero solamente quattro appena si accede al ricettario e poi grazie ad un link "pagina successiva" si può proseguire nella navigazione del ricettario. Abbiamo fatto questa scelta perchè il nostro sito potrebbe contenere anche centinaia di ricette e quindi sarebbe stato molto noioso per gli utenti scrollare pagine lunghissime. Lo script ignora i tipi non permessi, semplicemente togliendoli, e si preoccupa di tenere aggiornati i link: ad esempio se si sceglie una

difficoltà si rimane nello stesso tipo di ricette, mentre se si cambia la difficoltà si ritorna alla prima pagina di quel tipo di ricette. Inoltre se non si specifica il tipo viene mostrata la lista di tutte le ricette con anche il loro tipo, altrimenti viene mostrata solo l'immagine, il nome della ricetta e la descrizione

ricetta.cgi

Si occupa della pagina della ricetta. All'interno di essa è possibile effettuare la login, come in ricettario ed home, andare alla pagina di registrazione e recuperare i dati. La parte dinamica prevede la possibilità di inserire commenti agli utenti che hanno effettuato l'accesso. Questi potranno anche segnalare i commenti che ritengono poco opportuni, rimuovere i propri, rimuovere quelli altrui se sono amministratori. Lo script gestisce sia la mancanza del nome della ricetta sia la richiesta di una ricetta non presente nel ricettario.

area-personale.cgi Gestisce l'area personale di tutti gli utenti registrati al sito, amministratori e non. Come già detto in precedenza qui tutti gli utenti potranno modificare le proprie informazioni personali, tranne l'username che è non modificabile, e gli amministratori avranno accesso alla lista dei commenti segnalati.

recupera.cgi

In questa pagina sarà possibile recuperare i dati d'accesso al sito qualora fossero stati persi o dimenticati. La nostra idea sarebbe stata quella di inviare un'email alla casella di posta con cui l'utente si era iscritto al nostro sito, ma dato che non abbiamo i privilegi per inviare e-mail abbiamo semplicemente fatto in modo che le credenziali di accesso venissero mostrate direttamente nel sito una volta inserito il proprio indirizzo e-mail nell'apposita form.

registrazione.cgi

In questa pagina è possibile registrarsi al nostro sito. Ci si può registrare solo come utenti semplici, gli amministratori per sicurezza sono inseribili solamente tramite il file xml che contiene tutti gli utenti. Vi sono alcuni campi obbligatori (username, password ed email) ed altri opzionali (nome, cognome, abilità, sesso): su ciascuno di essi si svolgono controlli sia tramite javascript che dal perl, per essere sicuri che non vengano inserite informazioni non permesse. In questa pagina abbiamo scelto di non inserire la box di login presente in quasi tutte le altre pagine poichè un utente che si trova in questa pagina vorrà sicuramente registrarsi, azione che esclude il voler effettuare la login.

404.cgi

Questo script viene chiamato da apache quando avviene un errore 404.

redirect.cgi

questo script viene eseguito da apache in occasione di una ricerca ad un file index e provoca un redirect a home.cgi. Non era nostra intenzione risolvere la questione con un redirect, ma alla fine si è rivelata l'unica soluzione soddisfacente.

Problemi nel modificare un file xml acceduto tramite handle

In occasione dell'aggiunta dei controlli sull'accesso dei file XML, ci siamo imbattuti in alcuni problemi con il salvataggio dei parser XML dopo che sono stati modificati. Per questo in presenza di modifiche ai file XML (registrazione utente, modifica dati personali, commenti) si è mantenuto il metodo che avevamo scelto nella prima fase di progettazione (parsificazione di un file) cioè viene creato un handlefile solo per utilizzare la funzione flock (in questi casi esclusiva).

.htaccess

Vi sono due file di questo tipo, uno nella cartella public.html e uno nella cgi-bin (con la dovuta modifica ai path) e si occupano di due operazioni: chiamare 404.cgi in presenza di un errore 404 e indicare home.cgi come pagina predefinita della cartella.

6 Fonti

Quasi tutte le ricette inserite sono state prese dal sito *www.giallozafferano.it*. I contenuti sono usati a puro scopo informativo e dato che il sito resterà solamente all'interno del server universitario, non avendo quindi una ripubblicazione, il copyright non dovrebbe essere infranto.

7 Conclusione

Speriamo di aver realizzato un progetto che dimostri le competenze apprese durante il corso di tecnologie web. Il lavoro di gruppo è stato importante per capire le meccaniche in un futuro ambiente lavorativo, dove bisogna relazionarsi con altre persone che possono avere idee diverse su determinati aspetti del progetto e dove quindi è necessario trovare dei compromessi. Inoltre abbiamo imparato che l'operato di un componente dipende da ciò che produce un altro, perciò rispettare le tempistiche è fondamentale per una buona riuscita.