

Università degli Studi di Padova
Corso di laurea in Informatica
A.A. 2014/2015

Tecnologie Web

Login: admin

Password: pwd

Sito:

<http://tecnologie-web.studenti.math.unipd.it/tecweb/~lsantaca>

Email:

rocketteam.unipd@gmail.com

Studenti:

Cusinato Giacomo	1054137
Santacatterina Luca	619555



SOMMARIO

1. DESCRIZIONE GENERALE.....	3
2. SUDDIVISIONE DEI LAVORI	3
3. ARCHITETTURA	4
3.1. Progettazione layout	4
3.2. Sviluppo layout	5
3.3. Layout per dispositivi mobili.....	6
3.4. Layout di stampa	6
4. STRUTTURA.....	7
5. PRESENTAZIONE	8
6. COMPORTAMENTO	9
7. ACCESSIBILITÀ	10
7.1. Implementazione.....	10
7.2. Combinazione dei colori	10
8. LINGUAGGI UTILIZZATI	11
9. GESTIONE DEI DATI.....	12
9.1. Introduzione	12
9.2. XML.....	12
9.3. XML Schema	12
9.4. XSLT.....	12
9.5. DTD	Errore. Il segnalibro non è definito.
10. PERL	13
10.1. Introduzione	13
10.2. Descrizione moduli	13
10.2.1. Pagine visibili agli utenti	13
10.2.2. Pagine visibili agli admin.....	14
10.2.3. Moduli di supporto	14
10.2.4. Controlli e sicurezza.....	15
10.2.5. Gestione delle sessioni	15
11. VERIFICA E TEST	16
11.1. Strumenti per la scrittura del codice	16
11.2. Ambiente di lavoro	16
11.3. Criticità librerie	17
11.4. Test interazione utenti	18
11.5. Dispositivi utilizzati	19
11.6. Verifica del codice prodotto	20
11.7. Conclusioni	20

1. DESCRIZIONE GENERALE

Il progetto si propone di implementare un sito web che raggruppi le maggiori tariffe telefoniche proposte dai più noti operatori mobile. Il sito ha lo scopo di raccogliere tali offerte in modo che queste siano facilmente consultabili in un unico portale dagli utenti che lo desiderano.

Qualsiasi utilizzatore del sito web, infatti, può consultare la lista delle tariffe divisi per operatori (Wind, Vodafone, Tim, Tre) e, per ogni piano, una pagina di dettaglio in cui vengono elencate le principali caratteristiche dell'offerta selezionata. Un utente con i privilegi di amministratore, inoltre, oltre ad avere l'accesso ad un'area riservata, ha la possibilità di aggiungere o eliminare nuovi piani telefonici ed inserire nuovi amministratori nel sistema.

Il sito è stato sviluppato cercando ottimizzare criteri come usabilità, separazione tra struttura e presentazione, accessibilità e conformità agli standard W3C vista una sua possibile pubblicazione online.

2. SUDDIVISIONE DEI LAVORI

Per la realizzazione del progetto si è deciso di suddividere il carico di lavoro nel modo più uniforme possibile in modo che entrambi i componenti possano dare il giusto contributo al progetto. Sebbene il codice è stato soggetto a versionamento tramite un repository, si è cercato di distribuire gli incarichi in modo tale che ogni componente possa lavorare su sezioni del progetto indipendenti da quelle attribuite ad un secondo componente in modo da evitare conflitti.

Queste è la suddivisione dei compiti scelta:

Santacatterina Luca:

- Struttura del sito (HTML)
- Presentazione del sito (CSS di default, mobile, stampa)
- Accessibilità
- Creazione XML e XML Schema relativi ai piani
- Stesura sezione Test e Validazione
- Stesura sezione Gestione dei Dati
- Stesura sezione Contenuto e Presentazione

Cusinato Giacomo

- Comportamento del sito (Perl)
- Template zona admin
- Creazione XML e XML Schema relativi agli admin
- Testing del sito
- Stesura sezione Abstract, Suddivisione Lavori, Accessibilità
- Stesura sezione Perl

3. ARCHITETTURA

3.1. Progettazione layout

Prima di iniziare la stesura del codice abbiamo cercato di organizzare i dati che volevamo rappresentare. Questo ci ha permesso di chiarire alcuni aspetti fondamentali che il sito doveva avere. Innanzitutto la semplicità di navigazione al suo interno. Se un sito è semplice è visitabile e capibile da più persone. Il target d'utenza è l'utente in grado di utilizzare strumenti per la navigazione sul web.

Deciso questo punto siamo passati all'organizzazione del layout, proponendo su lavagna i vari schemi.

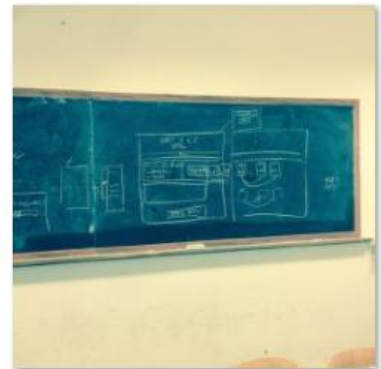
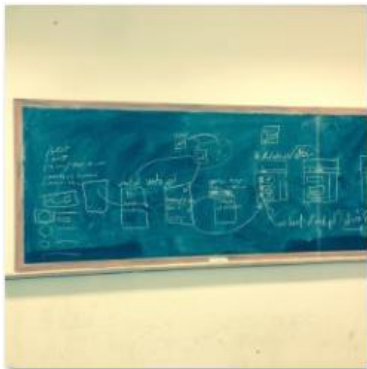


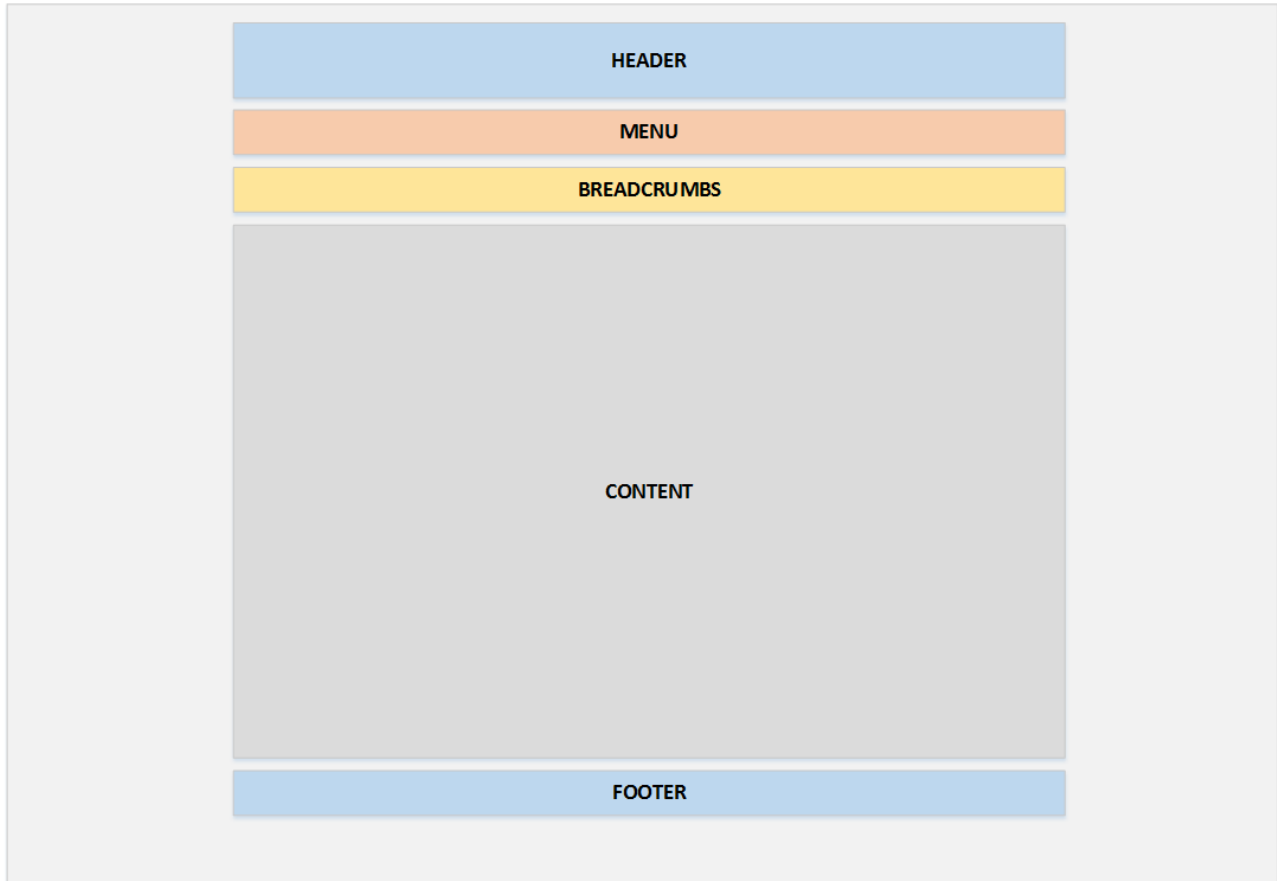
Foto bozze su lavagne

Fin da subito abbiamo optato per un layout di tipo fluido. Ha richiesto più lavoro del previsto ma ha ricompensato lo sforzo in quanto, abbiamo dovuto adattarlo ai vari tipi di monitor cambiando di molto l'aspetto, ma questo ha reso possibile un alto indice di accessibilità.

3.2. Sviluppo layout

Nell'immagine sottostante è possibile visualizzare la struttura principale dei div, i quali raccolgono per contenuto tematico le informazioni.

Il layout adottato ha come larghezza massima 1024px, mentre l'altezza non è stata in nessun modo alterata rispetto l'originale. Questo permette una evoluzione verso il basso senza problemi.



Come riportato nei capitoli sottostanti è stato applicato un foglio di stile per quanto riguarda la visualizzazione normale, nel momento in cui si va a restringere la visualizzazione fino a 640px di larghezza il contenuto rimane inalterato, dopodiché sotto i 640px si applica il foglio di stile per i dispositivi mobile.

Tutto il layout è stato progettato a pannelli adattabili, in questo modo si ha una più espandibilità.

- Partendo con l'analisi dall'alto si evidenzia subito il div "header", esso ha il compito di informare l'utente su ciò che sta visitando. Comprende il titolo del sito ed una brevissima presentazione.
- Sempre nell'area di maggior visibilità è presente il menu utente. Anch'esso è stato identificato tramite l'utilizzo del div "menu". Presenta 4 campi rappresentati da Home, Tim, Vodafone, Tre e Wind. Evidenzia il menù attivo in ogni pagina del sito. Questa area rappresenta l'asse informativo di dove l'utente può andare.
- Sottostante il menu principale è presente il div "breadcrumbs". Esso ha il compito di aiutare l'utente ad identificare in che posizione del sito si trova rispetto l'homepage del sito. Questa area rappresenta l'asse informativo di dove l'utente sia in quel determinato istante.
- Il blocco centrale è rappresentato dal div "content". Ha il compito di raccogliere la sezione principale, tutte le informazioni raccolte dagli utenti verranno rappresentate all'intero di esso. Questa area rappresenta l'asse informativo di cosa si sta trattando.
- Nella parte inferiore del sito è stato creato il div "footer", al suo interno è presente il link per poter eseguire l'autenticazione e a fianco contiene il nome del sito web.

3.3. Layout per dispositivi mobili

Il layout per i dispositivi mobili è stato sviluppato per risoluzioni inferiori ai 640px, l'aspetto del sito non varia moltissimo, ma si è cercato di eliminare alcuni "margin & padding" in modo da utilizzare quanta più area possibile per la visualizzazione.

3.4. Layout di stampa

Nel layout di stampa si è deciso di eliminare tutte le informazioni superficiali e di mantenere il contenuto semplice e privo di colori. Anche in questa parte i font non sono stati modificati rispetto a quelli di default, in modo da non alterare la relativa qualità di stampa o portabilità nel caso di stampa in PDF.

4. STRUTTURA

La struttura del sito è stata suddivisa per contenuti, in modo da aiutare l'utente nella ricerca delle informazioni. Questo porta ad una maggiore chiarezza delle sezioni presenti.

Tutto il sito è stato sviluppato in XHTML 1.0 Strict. Le pagine sono generate da PERL tramite sistemi di templating.

Di seguito vengono fornite le pagine sviluppate:

public_html/index.html tramite l'utilizzo della funzione *http-equiv="refresh"* abbiamo reso possibile il reindirizzamento alla homepage che si trova all'interno della cartella "cgi-bin". Si potrebbe con una configurazione del server APACHE effettuare questa conversione automaticamente, ma ciò non è possibile in quanto richiede l'intervento dei tecnici di laboratorio;

/cgi-bin/home.cgi la pagina viene generata da un contenuto sia dinamico che statico. I menu, gli header, il messaggio di benvenuto ed il footer sono generati automaticamente, mentre nella parte inferiore, dinamicamente, vengono prelevate le ultime 5 promozioni e visualizzate in ordine di data decrescente, per dare più evidenza alle novità;

/cgi-bin/operator.cgi anche questa pagina come la precedente possiede le stesse funzioni statiche, mentre tramite il passaggio dei parametri, quali il nome dell'operatore, la pagina provvede a caricare le informazioni di tutte le tariffe di esso. Tutto il contenuto centrale è sviluppato dinamicamente;

/cgi-bin/plan.cgi come per le pagine precedenti anche la seguente ha la parte statica di contenuto. La parte dinamica si differenzia in quanto, tramite il passaggio dell'id della promozione, va a ricercare nella base di dati la promozione e ne fa visualizzare l'offerta completa. Un ulteriore nota aggiuntiva sta nell'autenticazione, quando l'utente amministratore è loggato all'interno del sito trova la possibilità di eliminazione della promozione;

/cgi-bin/login.cgi la seguente pagina permette l'autenticazione degli amministratori per poter accedere all'area riservata. La pagina offre degli aiuti nel caso si inseriscano dei dati errati;

/cgi-bin/admin.cgi il ruolo fondamentale della pagina è l'inserimento dei dati, essa presenta alcune form per l'acquisizione dei dati e degli aiuti nel caso in cui vengano a mancare alcuni dati. Presenta inoltre controlli di sicurezza nel caso in cui si cerchi di inserire del codice malevole.

5. PRESENTAZIONE

Per la presentazione delle informazioni abbiamo cercato di essere quanto più precisi possibili ed accessibili. Abbiamo deciso di sviluppare l'interfaccia grafica tramite l'utilizzo di codice CSS. Volendo ottenere un ottimo grado di accessibilità abbiamo deciso di utilizzare CSS in versione 2, in quanto compatibile con la maggioranza dei browser in circolazione.

La pagina di amministrazione ha richiesto però una grafica leggermente diversa per essere meno prolissa e far visualizzare all'utente i campi delle form meglio ordinati. Solamente la pagina di amministrazione ha richiesto l'introduzione di poche righe di CSS3. Questo ha portato al fallimento della validazione CSS2 ma nonostante ciò ha reso possibile la visualizzazione ottimale in ugual modo. Riteniamo che anche se la pagina dovesse essere visualizzata da browser con compatibilità inferiore il contenuto e l'accessibilità non vengono alterati.

Rimane valida la compilazione con CSS3 per tutto il sito.

Ai fini del miglioramento della presentazione e delle performance abbiamo deciso di suddividere i file CSS in base alle funzioni a loro riservate. Esistono 3 tipi di CSS:

- Il file "default.css" utilizzato per la maggior parte dei dispositivi con risoluzione maggiore. Solitamente utilizzato nei computer portatili e fissi;
- Il file "mobile.css" utilizzato per la maggior parte dei dispositivi mobili quali cellulari e tablet, dove la dimensione dello schermo rimane compatta. Esso viene attivato ad una risoluzione inferiore ai 641px;
- Il file "print.css" utilizzato per la stampa delle pagine visualizzate. È stato posto particolare accorgimento a questo tipo di funzione, omettendo le parti del sito meno importanti. È stato altresì preso in considerazione la rimozione dei contenuti quali immagini e sfondi decorativi per ottenere una stampa più pulita possibile. I caratteri di default non sono stati toccati in quanto si è cercato di migliorare il risparmio di eventuale inchiostro o toner.

Il font utilizzato non è di tipo standard, bensì viene importato da una libreria esterna fornita da Google, questo comunque anche nel caso non venisse caricato utilizzerà i font di sistema disponibili e scalerà senza problemi. Tutti i caratteri utilizzati hanno una dimensione espressa in "em", questo ne permette una maggiore adattabilità alle preferenze dell'utente, senza influire negativamente nell'aspetto del sito.

6. COMPORTAMENTO

L'interazione con l'utente è molto importante. Un sito con questa "mission" punta moltissimo al continuo aggiornamento delle informazioni e richiede una costante attenzione da parte degli amministratori. Per questo abbiamo deciso di approfondire l'interazione tra sito web ed utenti.

Molti siti e molti libri consigliano l'utilizzo di JavaScript per il controllo dei dati nelle form. Noi abbiamo deciso di effettuare dei controlli più approfonditi, nel "core" del sito. Con l'utilizzo di PERL abbiamo ottenuto una più capillare omogeneità dei dati. JavaScript permette l'interazione con l'utente in tempo reale senza dover ricaricare la pagina, ma può anche essere disabilitata manualmente. Questo porterebbe a gravi disagi se non si rielaborano i dati dopo averli acquisiti. Tramite gli script che abbiamo deciso di implementare siamo riusciti ad ottenere ottimi risultati sulla formattazione dei caratteri che vengono introdotti nelle form. Inoltre abbiamo implementato una serie di funzioni che con un semplice aggiornamento della pagina si riesce a mantenere salvata l'informazione nei campi delle form. Solamente quando i dati risultano essere compilati correttamente ne sarà permesso l'inserimento.

Abbiamo aiutato l'utente nella compilazione introducendo messaggi d'errore ben formati e con una spiegazione di come risolvere l'errore.

In ogni caso tali controlli non possono essere disabilitati come nel caso di JavaScript.

7. ACCESSIBILITÀ

7.1. Implementazione

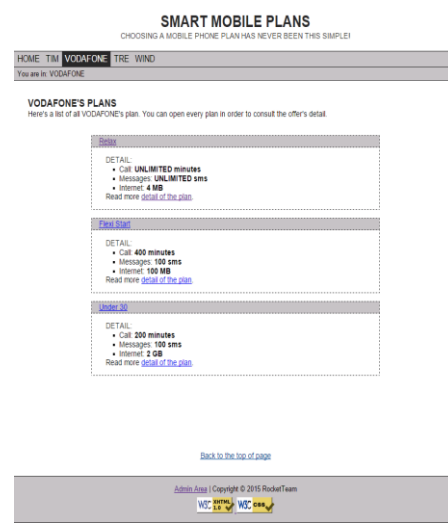
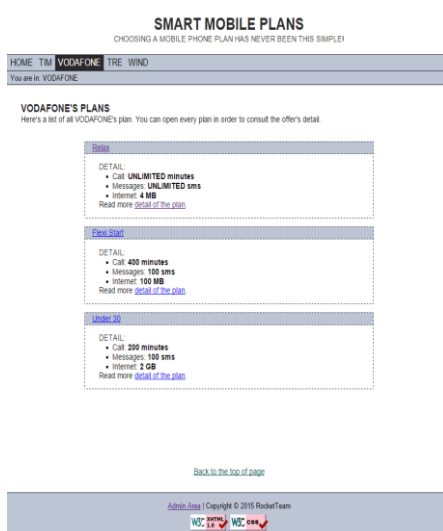
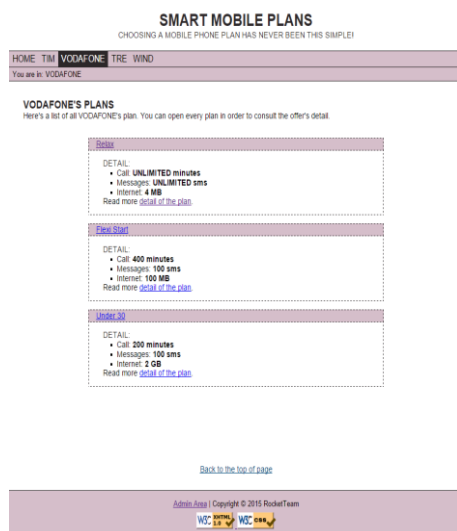
Per facilitare l'utilizzo del sito ad utenti con determinate disabilità, si è cercato di rispettare le raccomandazioni proposte dal W3C grazie ai relativi validatori per XHTML 1.0 e CSS. Il sito è stato inoltre progettato mantenendo una netta separazione tra struttura e presentazione cercando di mantenere una logica di navigazione intuitiva e semplificata.

A vantaggio dello screen reader, ogni immagine è stata corredata degli attributi alt e title la che la descrivono interamente. In ogni form, inoltre, ad ogni tag input è stata associata una label che si riferisce al tag input tramite l'attributo for in modo tale che sia ben definito il ruolo di ogni capo della form anche per lo screen reader.

Per quanto riguarda l'usabilità, invece, si è cercato di mantenere la navigazione tra le varie pagine semplice e intuitiva grazie alla creazione di un menu di navigazione (breadcrumbs), l'utilizzo dello stesso stile per tutti i link del sito (esclusi quelli del menù, facilmente individuabili) e l'utilizzo di link di ritorno ad inizio pagina. Sono stati inoltre ridefiniti i tabindex per per la navigazione tra i vari link: è stato scelto di modificare il normale flusso in quanto l'accesso all'area admin si trova a piè pagina ed è stato ritenuto importante posizionare il relativo indice a seguito del menu principale.

7.2. Combinazione dei colori

Per garantire l'usabilità del sito anche ad utenti affetti da problemi visivi è stato scelto di utilizzare uno schema a colori che esaltasse il contrasto tra sfondo e testo. Per testare tale scelta è stato utilizzato il servizio offerto <http://colorfilter.wickline.org/> che mostra come il sito può venire visualizzato da utenti con determinati problemi. Di seguito vengono riportati i risultati ottenuti.



8. LINGUAGGI UTILIZZATI

Per la struttura del sito è stato utilizzato il linguaggio XHTML 1.0 Strict validato correttamente secondo gli standard del W3C.

La presentazione è stata implementata in CSS cercando di rimanere per lo più compatibili con CSS2, che non valida correttamente per poche proprietà. Il risulta invece valido rispetto alla versione CSS3 secondo gli standard W3C.

Per la gestione dei dati è stato utilizzato XML, validato correttamente rispetto a diversi XML Schema forniti.

Per il comportamento è stato utilizzato il linguaggio Perl che consente di generare facilmente pagine dinamiche in XHTML. Per tale linguaggio, le principali librerie incluse sono: "XML::Lib" per la gestione dei file XML, "CGI::Session" per la gestione delle sessioni, "Template::Toolkit" per l'utilizzo dei template.

9. GESTIONE DEI DATI

9.1. Introduzione

Come richiesto dalle specifiche del progetto, la base di dati è formata da file di tipo XML. Essi hanno lo scopo di immagazzinare i dati raccolti in modo ben ordinato e ben formattato. Per essere così accessibili a modifiche, inserimenti e cancellazioni.

9.2. XML

Nel progetto abbiamo utilizzato due basi di dati: la prima per la raccolta delle promozioni e la seconda per il salvataggio degli utenti amministratori.

Abbiamo deciso di dividere le due basi di dati in quanto risulta molto più rapido la manipolazione dei dati ed è inoltre più corretto ai fini della validazione e della divisione dei contenuti.

9.3. XML Schema

Per ogni base di dati è stato sviluppato lo schema associato. Questo permette la validazione dei dati inseriti.

Per quanto riguarda il file “plans.xml” ha associato lo schema “plans.xsd”. Lo schema è stato sviluppato rispettando il modello “Tende alla Veneziana”, in quanto permette un ottimo riuso dei componenti e uno sviluppo futuro in caso si volessero aggiungere ulteriori informazioni da acquisire. All’interno vengono raccolti tutti i dati per poter presentare le offerte promozionali.

Mentre per il file “admins.xml” è stato associato lo schema “admin.xsd”. Lo schema è stato sviluppato rispettando il modello “Bambole Russe”, in quanto non abbiamo ritenuto necessario lo sviluppo più complesso di altri modelli. All’interno del file vengono raccolti solamente id, username e password degli amministratori.

9.4. XSLT

Non è stato adottato l’uso di XSLT (eXtensible Stylesheet Language Transformations) in quanto abbiamo deciso di utilizzare PERL perché più adattabile alle nostre esigenze di rielaborazione dei dati.

10. PERL

10.1. Introduzione

I moduli Perl contenuti nel progetto si occupano di svolgere tre funzioni essenziali:

- gestire e rendere visualizzabili i contenuti dinamici del sito;
- lettura e scrittura dei database XML relativi ad admin e piani telefonici;
- gestire le sessioni degli admin.

A supporto degli script Perl, inoltre, è stato scelto di utilizzare “Template::Toolkit”, un sistema di templating che consente di mantenere una netta separazione tra logica di comportamento e logica di presentazione rispetto a tutte le zone del sito. Tale processo è stato implementato con la creazione di un file di template ed un script Perl associato che andrà ad inizializzare tale file.

10.2. Descrizione moduli

10.2.1. Pagine visibili agli utenti

home.cgi e **home.tt**: è la home page del sito e rappresenta la pagina di root. Lo script home.cgi, oltre a inizializzare la creazione della pagina, si occupa di ottenere le ultime 5 tariffe inserite tramite una query XPath nell’XML in modo da poterle visualizzare nella sezione “latest plan” tramite l’invocazione del file di templating home.tt

operator.cgi e **operator.tt**: è la pagina che visualizza le offerte disponibili per un determinato operatore accessibile tramite il menu dell’header. Lo script operator.cgi si occupa di ricavare tutti i piani associati ad un determinato operatore per inizializzare successivamente il template operator.tt con tali dati.

plan.cgi e **plan.tt**: è la pagina relativa alla visualizzazione di un piano telefonico nel dettaglio la quale viene caricata dopo aver selezionato un’offerta qualsiasi dalla pagina del relativo operatore. Lo script plan.cgi si occupa di ricavare tutti i dettagli dell’offerta tramite il relativo id fornito nella query string dell’indirizzo e di caricare il template della tariffa inserendo tali dati. Nel caso un admin abbia effettuato l’accesso, risulta possibile effettuare la rimozione dell’offerta da questa pagina.

login.cgi e **login.tt**: è la pagina che consente di effettuare l’accesso all’area admin tramite l’apposita form visualizzata. Oltre a controllare la presenza di una sessione già corso (processo che verrà descritto in dettaglio nella sezione “Gestione delle sessioni”), si occupa di trasferire i dati inseriti dall’utente ai moduli relativi al controllo dei dati oltre a visualizzare eventuali errori effettuati in fase di login.

10.2.2. Pagine visibili agli admin

admin.cgi e **admin.tt**: è la zona relativa agli admin del sito in cui è possibile aggiungere un nuovo piano o un nuovo admin ai relativi XML tramite le form messe a disposizione. Lo script **admin.cgi**, oltre a effettuare specifici controlli sui dati in arrivo dalla pagina di login, si occupa inviare i dati inseriti ai moduli per la scrittura su database e di visualizzare eventuali errori sui dati inseriti nelle form. Il file **admin.cgi** sfrutta alcuni costrutti condizionali messi a disposizione della libreria **Template::Toolkit** per la generazione delle voci in alcuni tag select delle form e per mantenere i dati nelle form dopo un errato inserimento.

10.2.3. Moduli di supporto

adminReq.cgi: è lo script che si occupa di aggiungere un nuovo admin al relativo XML. Tale modulo è invocato dalla form presente nella pagina **admin.cgi** tramite una chiamata di tipo POST. Dopo aver effettuato determinati controlli (che verranno descritti in dettaglio nella sezione “Controlli e sicurezza”), lo script sfrutta la libreria Perl “**Lib::XML**” per creare un nuovo nodo admin ed inserirlo nel relativo documento. Nel caso alcuni dati ricevuti non siano conformi, lo script reindirizza l’utente nella zona admin con un errore che sarà visualizzato in tale pagina.

planReq.cgi: è lo script che si occupa di aggiungere un nuovo piano telefonico al relativo XML. Tale modulo è invocato dalla form presente nella pagina **admin.cgi** tramite una chiamata di tipo POST. Dopo aver effettuato determinati controlli (che verranno descritti in dettaglio nella sezione “Controlli e sicurezza”), lo script sfrutta la libreria Perl “**Lib::XML**” per creare un nuovo nodo piano ed inserirlo nel relativo documento. Nel caso alcuni dati ricevuti non siano conformi, lo script reindirizza l’utente nella zona admin con un errore che sarà visualizzato in tale pagina.

deleteReq.cgi: è lo script che si occupa di rimuovere un determinato piano telefonico dal relativo XML. Come descritto in precedenza, tale eliminazione può essere effettuata dalla pagina **plan.cgi**.

logout.cgi: è lo script che si occupa interrompere la sessione in corso. Tale modulo viene invocato alla pressione del pulsante “Logout” presente nella zona admin.

SUB.pl: è un modulo contenente tutte le subroutines Perl comuni tra più pagine o utilizzare di frequente. Oltre a contenere semplici funzioni di utilità (validazione XML, generazione ID, ecc.), espone dei metodi per la generazione di codice HTML relativo alle sezioni comuni del sito, come l’header, il menu, il breadcrumbs ed il footer.

10.2.4. Controlli e sicurezza

Per assicurare la correttezza e l'uniformità dei dati inseriti nel database XML si è cercato di codificare dei controlli per quanto più efficienti in fase di scrittura dell'XML. Tali check sono presenti nei seguenti moduli:

planReq.cgi: controllo che tutti i campi siano stati compilati, controllo di esistenza piano coincidente, controllo di presenza caratteri speciali non ammessi e conformità per determinati valori tramite espressioni regolari

adminReq.cgi: controllo che tutti i campi siano stati compilati, controllo esistenza admin, controllo password di conferma

Nel caso uno o più controlli fallissero il test, lo script si occupa di reindirizzare l'utente alla zona admin passando l'errore rilevato tramite query string in modo che questo possa essere visualizzato nella pagina.

Per mantenere sicure le credenziali degli utenti, inoltre, è stato scelto di utilizzare una cifratura di tipo "SHA-256hex" per il salvataggio delle password degli admin nel relativo file XML.

10.2.5. Gestione delle sessioni

Nel caso l'utente abbia inserito dei dati corretti nella form di login, il modulo admin.cgi si occupa di creare una nuova sessione relativa all'utente tramite la libreria Perl "CGI::Session". In questo modo l'utente non dovrà più inserire le proprie credenziali ogni ad ogni accesso all'area admin. Lo script login.cgi, infatti, prima di caricare la form di accesso effettua prima un controllo sulla presenza di una sessione già aperta il quale, se risulta positivo, consente l'accesso diretto alla zona admin. Tale controllo viene inoltre utilizzato nella pagina plan.cgi dove, in caso l'admin sia loggato, viene visualizzato il pulsante per l'eliminazione della tariffa telefonica come descritto in precedenza.

Per quanto riguarda il logout, invece, questo può essere effettuato tramite l'apposito pulsante presente nella zona admin o, in caso di una prolungata inattività dell'utente, in automatico.

11. VERIFICA E TEST

11.1. Strumenti per la scrittura del codice

Durante lo sviluppo del progetto si è fatto uso dell'editor "Sublime Text 3". Questo ci ha permesso di incrementare la qualità dei controlli, in quanto riesce a visualizzare in tempo reale errori di battitura e di sintassi del codice che si sta scrivendo. Sono stati integrati all'interno dell'editor dei plug-in che permettono la scrittura del codice più agevole ed un aiuto al programmatore. Per la scrittura di tutti i codici quali HTML, CSS e PERL, questi plug-in hanno aiutato moltissimo in quanto fornivano un ottimo supporto di auto completamento.

11.2. Ambiente di lavoro

Fin da subito ci siamo accorti che l'utilizzo del tunnel SSH per l'accesso al server di Tecnologie WEB non era la migliore soluzione per consentirci di lavorare da remoto. Abbiamo così ricreato un ambiente di lavoro che potesse essere più flessibile alle nostre esigenze.

Ci siamo affidati al software di virtualizzazione VirtualBox che ci ha permesso di ricreare un server virtuale proprio come quello di laboratorio. All'interno della macchina virtuale è stato installato il sistema operativo "Ubuntu Server 14.04 64bit". A corredo è stato poi integrato il server HTTP Apache 2, con supporto alle librerie cgi-bin, xml e loro dipendenze.

Questo non solo ci permetteva di poter lavorare in locale e quindi con latenze inferiori al microsecondo, ma bensì di poter validare l'intero codice online, in quanto tramite una configurazione del NAT presente nel modem e l'apertura della porta 80 del firewall era possibile raggiungere il sito web tramite l'indirizzo IP pubblico. Proprio grazie a questa implementazione si è riusciti a testare il sito con svariati dispositivi elencati successivamente.

E' stato inoltre utilizzato il servizio di hosting GitHub. Questo ci ha permesso di lavorare con più utenti contemporaneamente, senza interferenze ed avere una sicurezza di poter regredire in caso le modifiche non fossero corrette.

11.3. Criticità librerie

Un altro aspetto critico, riguardava le librerie PERL utilizzabili nel progetto. Per questo abbiamo sviluppato un software ad-hoc che permettesse di caricare rapidamente tutte le modifiche fatte al codice per poterlo testare sul server di “Tecnologie WEB”. Di seguito viene riportato il codice:

```
#!/bin/bash

echo -e "\n***** Inizio Script *****"
echo -ne "Posizionamento su cartella ~/tecweb/ \t"
cd ~/tecweb/
echo -e "\tOK"

echo -ne "Rimozione completa vecchi sorgenti \t"
rm -rf cgi-bin/*
rm -rf data/*
rm -rf public_html/*
echo -e "\tOK"

echo -ne "Download update repository from git \t"
git clone -q https://github.com/RocketTeam-UniPD/SmartMobilePlans.git
echo -e "\tOK"

echo -ne "Copia dei nuovi file sulle cartelle \t"
cp -R SmartMobilePlans/cgi-bin/* cgi-bin/
cp -R SmartMobilePlans/data/* data/
cp -R SmartMobilePlans/public_html/* public_html/
echo -e "\t OK"

echo -ne "Correzione permessi per l'esecuzione \t"
chmod -R 750 ~/tecweb/cgi-bin/
chmod -R 750 ~/tecweb/data/
chmod -R 750 ~/tecweb/public_html/
echo -e "\t OK"

echo -ne "Eliminazione dei files temporanei \t"
rm -rf SmartMobilePlans
echo -e "\t OK"

echo -e "***** Fine Script *****\n"
```

Script bash update codice sul server “Tecnologie WEB”

Grazie a questo script potevamo provare le ultime modifiche in modo estremamente rapido.

11.4. Test interazione utenti

Un punto fondamentale richiesto dal progetto riguardava l'interazione con l'utente. Il sito web mette a disposizione delle form per l'inserimento dei dati. Questi a loro volta vengono rielaborati e salvati nella base di dati. A volte però l'utente potrebbe commettere degli errori volontariamente o involontariamente, a volte potrebbe voler danneggiare il sito per scopi poco chiari. Per questo abbiamo voluto testare efficacemente che l'inserimento di codice malevolo non portasse a gravi problemi di stabilità del sito o peggio ancora ad un disservizio del server.

Dall'analisi svolta abbiamo ricavato tre tipi di dati che potevano portare a disguidi se inseriti nelle form. Vengono di seguito riportati i codici:

```
<script language="javascript" type="text/javascript">
    alert('This is a bug! :)');
</script>

<form name="format" action="send.php" method="post">
    <input type="hidden" name="mail" value="user@example.org">
    <input type="text" name="name">
    <input type="submit" value="submit">
</form>

<iframe src="http://www.libero.it/" width="1" height="1" frameborder="0"></iframe>
```

Codice di test per malfunzionamenti

Abbiamo concluso con successo che ogni tipo di dato inserito non portava alla visualizzazione di anomalie.

11.5. Dispositivi utilizzati

Grazie all'implementazione del server virtuale riportato in precedenza c'è stato possibile testare con svariati dispositivi il sito. Di seguito vengono riportati riassuntivamente i dispositivi provati ed i browser utilizzati:

Notebook	DELL Inspiron 3721 SO Windows 10 Monitor 17" Risoluzione 1600x900px Browser Google Chrome v46 Internet Explorer 11 Microsoft Edge 20	Desktop	Desktop assemblato SO Ubuntu Desktop 15.04 Monitor 23" Risoluzione 1920x1080px Browser Google Chrome v45 Mozilla Firefox v40
Smartphone	LG G2 SO Android Lollipop Monitor 5.2" Risoluzione 1920x1080px Browser Andorid stock Browser Google Chrome v45	Notebook	Mac Book Pro 13 Retina SO OS-X Yosemite Monitor 13" Risoluzione 2560X1600 Browser Safari v8 Browser Google Chrome v45
Tablet	LG G-PAD 8.3 SO Android KitKat Monitor 8.3" Risoluzione 1920x1200px Browser Android stock	Smartphone	Nokia Lumia 1020 SO Windows Phone Monitor 4.5" Risoluzione 768x1280px Browser default

Tutti i test effettuati con i vari browser dei dispositivi sopra riportati hanno superato l'esito correttamente.

Dato il continuo aggiornamento dei dispositivi questo non permetteva di testare il sito con browser meno recenti. Per questo ci siamo affidati ad un paio di siti che riescono ad emulare browser di varie versioni. Questi siti sono:

<http://browsershots.org/>

<https://www.browserstack.com/>

<https://www.browserling.com/>

Tramite l'utilizzo di questi siti per il controllo di compatibilità siamo riusciti ad ottimizzare la visualizzazione su gran parte dei browser.

11.6. Verifica del codice prodotto

Per la verifica del codice HTML e CSS prodotto ci siamo affidati ai validatori che il sito del W3C mette a disposizione.

Per quanto riguarda il codice HTML, è stato validato correttamente con la versione XHTML 1.0 Strict senza riportare errori o warning.

Mentre per il codice CSS lo si è validato in versione 2, 2.1 e 3.

La versione CSS3 ha validato correttamente tutti gli stili prodotti, senza riportare errori o warning. Mentre la versione CSS2 e CSS2.1 riportava un paio di errori dettati dal fatto che un paio di istruzioni non sono pienamente compatibili, ma questo comunque porta ad ottenere ugualmente risultati ottimali e non provoca problemi alla visualizzazione o all'accessibilità del sito.

Per la verifica di accessibilità abbiamo deciso di utilizzare due validatori, in quanto pensiamo sia più utile produrre un sito web accessibile a tutti i tipi di utente che di aspetto grafico accattivante. I validatori eseguono un controllo a runtime del sito. Di seguito vengono riportati i relativi siti internet:

<http://achecker.ca/checker/index.php>

<http://wave.webaim.org/>

Sono state validate tutte le pagine sviluppate del sito.

Siamo riusciti ad ottenere la certificazione più alta possibile, la WCAG 2.0 AAA. Al momento della validazione questa produce dei warning ma sono esclusivamente di contenuto in quanto non può essere il validatore, una macchina, ad eseguire determinati controlli per capire il contenuto dei testi.

Per i file XML e XSD sono stati utilizzati due validatori anch'essi online. Di seguito vengono riportati i relativi siti internet:

<http://www.freeformatter.com/xml-validator-xsd.html>

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

Entrambi permettono la validazione dei singoli file XML e XSD riguardante la corretta scrittura e permettono anche la validazione incrociata per verificare che non ci siano errori di contenuto o di forma.

11.7. Conclusioni

Il risultato di tutte le verifiche e validazioni ha portato il sito ad essere pienamente accessibile e compatibile con i vari browser/dispositivi