Piattaforma per la scoperta dei grotti in Ticino

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 3

Analisi 4

1.4 Analisi del dominio 4

1.5 Analisi e specifica dei requisiti 4

1.6 Use case 6

1.7 Pianificazione 7

1.7.1 Analisi 8

1.7.2 Progettazione 8

1.7.3 Implementazione 9

1.7.4 Test 10

1.8 Analisi dei mezzi 10

1.8.1 Software 10

1.8.2 Hardware 10

2 Progettazione 11

2.1 Design dell’architettura del sistema 11

2.2 Design dei dati e database 12

2.3 Design delle interfacce 14

2.3.1 Pagina iniziale 14

2.3.2 Pagina di login 14

2.3.3 Pagina di registrazione 15

2.3.4 Pagina di creazione dei grotti 15

2.3.5 Pagina di amministrazione 16

3 Implementazione 17

3.1 Struttura del progetto 17

3.2 Database 17

3.2.1 Tabella utente 17

3.2.2 Tabella grotto 18

3.2.3 Tabella foto 18

3.2.4 Tabella voto 18

3.3 Pagina Home 18

4 Test 18

4.1 Protocollo di test 18

4.2 Risultati test 19

4.3 Mancanze/limitazioni conosciute 19

5 Consuntivo 19

6 Conclusioni 19

6.1 Sviluppi futuri 19

6.2 Considerazioni personali 19

7 Bibliografia 20

7.1 Bibliografia per articoli di riviste: 20

7.2 Bibliografia per libri 20

7.3 Sitografia 20

8 Allegati 20

# Introduzione

## Informazioni sul progetto

* Allievo: Matteo Forni
* Docente responsabile: Luca Peduzzi
* Scuola Arti e Mestieri di Trevano, sezione informatica, classe I4AA
* Data di inizio: 03.09.2019
* Termine di consegna: 20.12.2019

## Abstract

E’ una breve e accurata rappresentazione dei contenuti di un documento, senza notazioni critiche o valutazioni. Lo scopo di un abstract efficace dovrebbe essere quello di far conoscere all’utente il contenuto di base di un documento e metterlo nella condizione di decidere se risponde ai suoi interessi e se è opportuno il ricorso al documento originale.

Può contenere alcuni o tutti gli elementi seguenti:

* **Background/Situazione iniziale**
* **Descrizione del problema e motivazione**: Che problema ho cercato di risolvere? Questa sezione dovrebbe includere l'importanza del vostro lavoro, la difficoltà dell'area e l'effetto che potrebbe avere se portato a termine con successo.
* **Approccio/Metodi**: Come ho ottenuto dei progressi? Come ho risolto il problema (tecniche…)? Quale è stata l’entità del mio lavoro? Che fattori importanti controllo, ignoro o misuro?
* **Risultati**: Quale è la risposta? Quali sono i risultati? Quanto è più veloce, più sicuro, più economico o in qualche altro aspetto migliore di altri prodotti/soluzioni?

Esempio di abstract:

*As the size and complexity of today’s most modern computer chips increase, new techniques must be developed to effectively design and create Very Large Scale Integration chips quickly. For this project, a new type of hardware compiler is created. This hardware compiler will read a C++ program, and physically design a suitable microprocessor intended for running that specific program. With this new and powerful compiler, it is possible to design anything from a small adder, to a microprocessor with millions of transistors. Designing new computer chips, such as the Pentium 4, can require dozens of engineers and months of time. With the help of this compiler, a single person could design such a large-scale microprocessor in just weeks.*

## Scopo

Lo scopo di questo progetto è quello di creare un sito web che consenta di esplorare in maniera semplice e intuitiva i vari grotti del Canton Ticino. Grazie a questo prodotto si potranno visualizzare le posizioni dei punti di ristorazione con le informazioni su di essi. È disponibile inoltre una ricerca dei grotti a cui si possono applicare dei filtri come il nome, la località e la fascia di prezzo.

Eseguendo l’accesso al sito si avrà la possibilità di inserire dei nuovi ristoranti con una valutazione, la fascia di prezzo e le informazioni su di esso.

## Analisi

## Analisi del dominio

Il progetto dovrà essere una piattaforma per la scoperta dei grotti del Ticino, momentaneamente esiste un sito che fa già questo e si tratta di ticino.ch. Esso si comporta in maniera molto simile a come dovrà fare questo progetto, consente infatti di visualizzare tramite una mappa i vari grotti e di filtrarli tramite dei filtri simili a quelli richiesti in questa piattaforma. Il sito già esistente utilizza però delle mappe che non consentono di inserire i marker tramite un indirizzo dato che in esso un utente non può aggiungere una località.

Il contesto del prodotto è quindi in parte organizzato ma in maniera molto semplice e che necessita di un aggiornamento continuo da parte degli amministratori dato che solo loro possono aggiungere i grotti alla mappa.

Gli utenti che utilizzeranno il sito saranno principalmente senza alcuna conoscenza tecnica ed è per questo motivo che le interfacce dovranno essere molto semplici e facili da utilizzare. Il sito mira ad essere utilizzato dal maggior numero di persone nel cantone e fuori.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Creazione pagina riservata agli amministratori |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Si necessitano i permessi di amministratore per accedervi |
| **Sotto requisiti** | |
| **001** | Si dovrà avere la possibilità di creare/modificare/eliminare gli utenti |
| **002** | Si dovrà avere la possibilità di creare/modificare/eliminare dei grotti |
| **003** | Si dovrà avere la possibilità di rendere un utente normale amministratore |
| **004** | Si dovrà avere la possibilità di approvare o rifiutare la creazione di un grotto da parte di un utente |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Creazione pagina principale |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci dovrà essere una mappa interattiva con la posizione dei grotti |
| **002** | Ci dovrà essere una sezione riservata alla ricerca dei grotti |
| **003** | Si dovrà potere applicare alla ricerca dei filtri in base al zona/fascia di prezzo/nome |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Creazione pagina di login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Si dovrà avere la possibilità di aggiornare la password |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Creazione pagina di registrazione |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci dovrà essere un controllo istantaneo dei dati inseriti |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Creazione pagina di creazione di un grotto |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Bisognerà avere effettuato il login per accedervi |
| **Sotto requisiti** | |
| **001** | Si dovrà avere la possibilità di inserire testo e immagini |
| **002** | Si dovrà avere la possibilità di assegnare una valutazione (stelle) |
| **003** | Si dovrà avere la possibilità di assegnare una fascia di prezzo |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Condivisione sui social di una località |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |

## Use case



Figure 1 Use Case

La foto precedente rappresenta li use case del prodotto, come si può vedere dallo schema l’utente di base ha la possibilità, dopo aver eseguito il login, di creare un nuovo grotto o di condividerne uno sui social. Senza effettuare il login esso può solamente visualizzare i grotti sulla mappa interattiva o tramite la sezione di ricerca, essendo questo scontato non è stato inserito nello schema.

L’amministratore può invece creare, modificare o eliminare un utente oppure un grotto ed inoltre è incaricato di verificare e accettare o rifiutare le creazioni degli utenti. Per essere un admin bisogna per forza eseguire il login e quindi questa parte non è stata inserita nell’immagine per evitare di ripetere inutilmente una cosa scontata.

## Pianificazione



Figure 2 Diagramma di Gantt

Nella figura superiore si può vedere il diagramma di Gantt completo, esso rappresenta in maniera indicativa come dovrebbe andare il progetto. Esso è stato diviso in quattro grandi categorie che sono Analisi, Progettazione, Implementazione e Test, oltre ad esse vi è la documentazione che procede lungo tutto il progetto dato che verrà completata con l’avanzare delle altre attività.

### Analisi



Figure 3 Gantt capitolo Analisi

L’analisi si suddivide in quattro attività, esse sono la scrittura del diagramma di Gantt, l’analisi del dominio, l’analisi dei requisiti e l’analisi dei costi. La prima operazione è quella che a parer mio richiede più tempo e quindi le ho assegnato mezza giornata di lavoro composta da quattro ore scolastiche.

La seconda attività è piuttosto facile e veloce da fare ed ho quindi previsto di impiegarci circa due ore e un quarto considerando la revisione finale dello scritto.

L’analisi dei requisiti è anche piuttosto veloce da completare e quindi le ho assegnato il resto della giornata rimanente dopo aver fatto la seconda operazione.

La lezione successiva, il 10.09.2019, ho previsto di iniziarla con l’ultima attività compresa nell’analisi che è l’analisi dei costi che comprende l’analisi dei mezzi necessari per lo sviluppo del progetto, i software utilizzati e l’hardware necessario.

### Progettazione



Figure 4 Gantt capitolo Progettazione

La progettazione è stata suddivisa in quattro attività ma l’ultima rappresenta soltanto una milestone che indica la fine della progettazione e l’inizio dell’implementazione.

Il 10.09.2019, dopo aver finito l’ultimo incarico di analisi, ho previsto di creare il design dell’architettura del sistema. Lo schema non dovrebbe prendere troppo tempo dato che sono abbastanza in chiaro sul flusso di lavoro del sito. Fatto ciò ho occupato la giornata seguente con il design della banca dati, questo perché può prendere abbastanza tanto tempo per arrivare ad una forma ottimizzata.

Infine, come ultima attività della progettazione, vi è il design delle interfacce che potrebbe prendere abbastanza tanto tempo a dipendenza di quanto in fretta riesco a sviluppare la base di tutte le pagine in modo da renderle semplici e veloci da comprendere.

### Implementazione



Figure 5 Gantt capitolo implementazione prima parte

L’implementazione, essendo la parte più lunga e complessa del progetto, è stata suddivisa in due parti così da semplificarne la spiegazione. La prima parte comprende otto attività ed è previsto che si prolunghi fino al 05.11.19.

Inizialmente si crea il database che era stato progettato in precedenza e si sviluppa il codice che andrà a collegare esso con le pagine web.

Quando il database è terminato si può iniziare a sviluppare le pagine vere e proprie iniziando da quella principale che, secondo le previsioni, è quella che occupa più tempo. Essa comprende infatti tutta la mappa interattiva e la sezione di ricerca.

Finita la prima pagine si passa a quella di registrazione e una volta terminata si scriverà il codice che verifica l’inserimento dell’utente sia front end che back end. La stessa procedura avviene in seguito con la pagina di login che utilizza un codice molto simile a quello utilizzato per la registrazione.

Come ultima attività della prima parte di implementazione vi è la scrittura del codice che verifica se un utente è admin o meno così da consentirgli, se possiede tutti i permessi, l’accesso alla sezione dedicata agli amministratori.



Figure 6 Gantt capitolo implementazione seconda parte

La seconda parte d’implementazione comprende quattro attività ed una milestone che rappresenta il termine dello sviluppo.

La prima delle attività consiste nel creare la pagina che gestisce l’inserimento di un grotto da parte degli utenti, con essa bisogna sviluppare il codice che verifica ciò che l’utente scrive e il codice che gestisce la trasformazione da indirizzo a coordinate fatto grazie alle API di Google Maps.

L’ultima pagina da creare è quella dedicata agli admin che prevedo sia piuttosto complicata e per questo l’attività dura parecchio tempo. Essa deve gestire le creazioni, eliminazioni e modifiche di utenti e grotti ed inoltre deve contenere una sezione dedicata alla verifica degli inserimenti da parte degli utenti.

### Test



Figure 7 Gantt capitolo Test

L’ultima parte del progetto sono i test, essi saranno test di funzionamento generale e mirano a verificare che l’integrazione delle pagine funzioni anche se si fanno volutamente cose sbagliate.

I test procedono in maniera piuttosto lineare iniziando a controllare la prima pagina per poi procedere in ordine di creazione. Infine vi sarà un test generale per controllare la sicurezza del progetto. Alla fine dei test vi sono due lezioni che vengono utilizzate per creare la presentazione ed infine vi è una milestone che rappresenta il termine del progetto.

## Analisi dei mezzi

### Software

I software che sono stati utilizzati per questo progetto sono i seguenti:

* Jetbrains PhpStorm 2019 2.3
* MySQL Workbench 6.3.8
* Google Chrome 77
* Mozilla Firefox 69
* WPS Writer 11.1
* WPS Presentation 11.1
* Microsoft Project 2010

Mentre le librerie utilizzate sono le seguenti:

* Material Design Bootstrap 4.8.7
* API Google Maps Javascript 3.38
* API Google Geocoding 3.

### Hardware

L’hardware su cui è stato svolto il progetto è un laptop Dell XPS 15 9570 con installato Ubuntu 18.04 come sistema operativo. Il prodotto può essere fatto funzionare su una qualsiasi macchina che abbia un webserver funzionante che funzioni con il pattern MVC di PHP, quindi con il modulo di rewrite attivato e l’override delle cartelle consentito.

# Progettazione

## Design dell’architettura del sistema



Figure 8 Design del sistema

Il design del sistema di questo progetto è piuttosto semplice, l’utente inizia collegandosi alla pagina principale che sarà quella contenente la mappa interattiva e la sezione di ricerca dei grotti con possibilità di filtrare i risultati. Da questa pagina l’utilizzatore del sito potrà decidere se restare dove si trova, in caso necessiti solo di eseguire una ricerca, oppure di spostarsi verso la pagina di login. Se esso non possiede un account avrà la possibilità di crearne uno, grazie ad un pulsante che lo porterà alla sezione di registrazione, altrimenti potrà accedere al suo account. Se verrà per caso dimenticata la password ci sarà un bottone che consentirà di ripristinarla.

Dopo aver eseguito il login, se l’utente possiederà i privilegi di amministratore, si troverà nella pagina riservata agli admin che consente di creare, modificare e eliminare grotti ed utenti e anche di approvare o rifiutare i grotti inseriti dagli utenti. Se vorrà potrà poi spostarsi alla pagina di creazione dei grotti.

Se l’utente non è un amministratore potrà solo visualizzare la pagina di aggiunta di una località.

In tutte le parti e sezioni del sito vi sarà la possibilità di navigare verso tutte le pagine grazie ad un menu.

## Design dei dati e database



Figure 9 Diagramma ER del database

Il diagramma ER del database è piuttosto semplice come si può vedere dall’immagine soprastante. Esso comprende cinque tabelle che rappresentano rispettivamente il **ruolo** che uno user possiede quindi admin o normale, l’**utente** con tutti i suoi attributi, il **grotto**, le **foto** che possono essere collegate ad una località e la **fascia di prezzo** di un ristorante.

La tabella ruolo poteva essere sostituita da due attributi all’interno di utente ma così facendo si avrebbero problemi se un giorno si volesse inserire un nuovo incarico come ad esempio il gerente di un grotto. Un utente può avere solo un ruolo mentre un ruolo può essere ricoperto da zero a molti utenti.

La tabella utente comprende le informazioni di base di una persona e utilizza l’email come chiave dato che essa è sicuramente univoca.

La tabella grotto contiene un id che identifica la località e poi le informazioni sulla sua posizione sia in coordinate che come indirizzo, questo perché contattare tutte le volte le API di Google renderebbe il sito lento. Inoltre comprende un attributo rappresentante la fascia di prezzo in cui si trova il grotto ed uno con la valutazione di esso.

L’ultima tabella contiene le foto che vengono identificate da un id e possiedono un titolo e un percorso locale. Un grotto può possedere molte immagini e un’immagine può essere associata ad un solo grotto.



Figure 10 Schema logico del database

Lo schema logico rende chiara un’ultima particolarità del database, l’associazione vota dello schema ER viene tradotta in una tabella che contiene come chiavi primarie le chiavi di utente e grotto ed ha inoltre un attributo voto.

|  |  |  |
| --- | --- | --- |
| **Tabella ruolo** | | |
| Attributo | Tipo | Descrizione |
| nome | varchar(25) | Chiave primaria, rappresenta il nome del ruolo. |

|  |  |  |
| --- | --- | --- |
| **Tabella utente** | | |
| Attributo | Tipo | Descrizione |
| email | varchar(50) | Chiave primaria, rappresenta il l’email dell’utente. |
| username | varchar(50) | Rappresenta il nome utente. |
| nome | varchar(50) | Rappresenta il nome di battesimo dell’utente. |
| cognome | varchar(50) | Rappresenta il cognome dell’utente. |
| password | varchar(64) | Rappresenta la password dell’utente, verrà salvato l’hash del valore. |

|  |  |  |
| --- | --- | --- |
| **Tabella grotto** | | |
| Attributo | Tipo | Descrizione |
| id | int | Chiave primaria, l’identificatore del grotto |
| nome | varchar(50) | Il nome del grotto |
| lon | double | La coordinata longitudine del grotto |
| lat | double | La coordinata latitudine del grotto |
| via | varchar(50) | La via in cui si situa il grotto |
| paese | varchar(50) | Il paese (comune) in cui si situa il grotto |
| cap | int | Il cap del paese |
| no\_civico | varchar(10) | Il numero civico |
| fascia\_prezzo | enum | La fascia di prezzo (buon mercato/nella norma/caro) |
| valutazione | int | La valutazione ricevuta |

|  |  |  |
| --- | --- | --- |
| **Tabella foto** | | |
| Attributo | Tipo | Descrizione |
| id | int | Chiave primaria, l’identificatore della foto |
| titolo | varchar(50) | Il titolo dell’immagine |
| path | varchar(50) | Il percorso assoluto dell’immagine sul server |

|  |  |  |
| --- | --- | --- |
| **Tabella voto** | | |
| Attributo | Tipo | Descrizione |
| email\_utente | varchar(50) | Chiave primaria, chiave esterna, l’email dell’utente |
| id\_grotto | int | Chiave primaria, chiave esterna, l’identificatore del grotto |
| voto | int | Il voto assegnato |

## Design delle interfacce

### Pagina iniziale



Figure 11 Mockup pagina Index

La pagina principale dovrà essere pulita e semplice da utilizzare, la mia idea era quella di suddividerla in due parti: la prima contenente la mappa interattiva che mostra le posizioni dei grotti mentre la seconda con la sezione di ricerca a cui potranno venire applicati dei filtri.

### Pagina di login



Figure 12 Mockup pagina di login

La pagina di login sarà molto semplice dato che conterrà solamente quattro elementi. Il primo sarà il campo per inserire l’email, il secondo sarà l’input per la password e poi vi saranno due bottoni. Il primo dei pulsanti servirà a cambiare la password del proprio account mentre il secondo consentirà di raggiungere la pagina di registrazione.

### Pagina di registrazione



Figure 13 Mockup pagina registrazione

La pagina di registrazione conterrà tutti gli input necessari a inserire i campi necessari per creare un account che sono:

* Nome
* Cognome
* Username
* Email
* Password
* Ripetizione della password

### Pagina di creazione dei grotti



Figure 14 Mockup pagina creazione grotti

La pagina di creazione dei grotti conterrà tutti i campi necessari per la creazione di un grotto quindi:

* Nome
* Fascia di prezzo
* Valutazione
* CAP
* Paese
* Via
* Numero Civico

Alla fine dell’inserimento si potrà controllare tramite un bottone apposito se la posizione sulla mappa viene marcata nel punto corretto. Se tutto sarà stato inserito nella maniera giusta si potrà salvare l’inserimento che verrà passato agli admin da verificare.

### Pagina di amministrazione



Figure 15 Mockup pagina admin

La pagina degli admin sarà di base molto vuota e semplice, conterrà tre bottoni che consentiranno di modificare gli utenti, i grotti o di verificare gli inserimenti. Una volta premuto su uno di questi bottoni si aprirà una finestra modale che mostrerà la lista dei campi che si vorranno gestire.

# Implementazione

*In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.*

*Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.*

*In questa parte è richiesto l’inserimento di codice sorgente/print screen di maschere solamente per quei passaggi particolarmente significativi e/o critici.*

*Inoltre dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.*

*Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.*

*Per eventuali dettagli si possono inserire riferimenti ai diari.*

## Struttura del progetto

Il progetto è basato su un pattern MVC e quindi tutto il codice è suddiviso nelle seguenti cartelle principali:

* views
* models
* controllers
* assets

Nella cartella ***views*** si trovano tutti i files contenenti la parte frontend del sito, essi sono suddivisi in ulteriori cartelle relative alla pagina che vanno a caricare. Questi files contengono principalmente tag di HTML e codice Javascript oltre ad alcuni controlli e inserimenti di dati in PHP.

La cartella ***models*** contiene codice PHP che si occupa di un compito preciso come gestire il database, verificare la validità dei dati o inviare email.

Nella sezione ***controllers*** vi sono dei files PHP che si occupano di mettere in comunicazione le views (front end) con i models (back end). Essi servono anche a caricare le views quando la pagina relativa viene richiamata.

L’ultima cartella contiene i fogli di stile della pagina così come le librerie utilizzate, i codici in Javascript e le immagini dei grotti che vengono mostrate quando si seleziona una località.

## Database

Il database è stato sviluppato sulla base della progettazione ma nel corso del progetto ho subito alcune modifiche. Esso è suddiviso in sei tabelle che rappresentano rispettivamente i ruoli degli utenti, gli utenti, le fasce di prezzo dei grotti, i grotti, le immagini dei locali e le valutazioni di essi.

Le tabelle rappresentanti i ruoli e le fasce di prezzo sono state create così da semplificare una futura espansione del progetto dato che consentono di aggiungere un campo senza dover modificare nient’altro.

### Tabella utente

La tabella utente è una delle più importanti di tutto il database, essa contiene otto campi di cui i primi cinque (nome, cognome, username, email e password) saranno inseriti dall’utente alla registrazione mentre i seguenti tre (reset\_token, first\_login, nome\_ruolo) verranno usati per verificare gli stati dell’utente e valutare cosa mostrargli.

L’email rappresenta anche l’identificatore di un account dato che non ve ne possono essere due uguali. La password è salvata codificata utilizzando l’algoritmo sha256 per evitare problemi di sicurezza. Il campo reset\_token verrà compilato e inviato per email al momento che un utente vorrà cambiare la propria password per accertarsi che non sia un estraneo con intenzioni dannose. Il campo first\_login contiene un booleano che indicherà se l’utente è stato creato da un admin o meno, se questo è vero l’utente dovrà modificare la password al primo login.

### Tabella grotto

Nello schema rappresentante i grotti vi sono dodici campi, i primi sei (nome, via, cap, paese, fascia di prezzo e telefono) saranno valori inseriti da un utente alla creazione dello stesso. I campi lon e lat indicano le coordinate geografiche della località e serviranno a mostrare il luogo sulla mappa, verranno inserite automaticamente calcolandole dall’indirizzo passato dall’utente. Il campo verificato indica se un grotto è stato creato da un admin o da un utente, se lo ha creato quest’ultimo allora la località dovrà essere valutata da qualcuno con privilegi più elevati.

### Tabella foto

La tabella foto contiene i dati relativi alle immagini associate ai grotti, la path di essa indica il percorso relativo al pattern MVC dell’immagine. Il nome della foto è un hash del suo contenuto così che non vi siano problemi di caratteri speciali in esso e per evitare doppioni della stessa immagine.

### Tabella voto

La tabella voto contiene la valutazioni da uno a cinque fatte dagli utenti ai grotti. Essa contiene la chiave dell’utente, quella del grotto e il voto, ad ogni inserimento su di essa un trigger di MySQL farà la media di tutte le valutazioni del grotto che ne ha appena ricevuta una nuova e aggiornerà il campo valutazione nella tabella del grotto. A seguito il codice che esegue l’operazione appena spiegata.

delimiter $$$

create trigger avg\_after\_review after insert on voto for each row

begin

    set @avg\_review = (select avg(voto) from voto where id\_grotto=NEW.id\_grotto);

    update grotto set valutazione = @avg\_review where id=NEW.id\_grotto;

end;

$$$

delimiter ;

## Pagina Home

La pagina home è suddivisa in due sezioni e comprende due dei requisiti principali che sono una mappa interattiva per visualizzare i grotti e una lista di essi in cui poterli ordinare per i vari campi.

Nella sezione superiore vi è la mappa generata grazie alle credenziali di Google Maps, essa al caricamento della pagina andrà a prendere tutti i grotti caricati dal database e genererà per ogni uno di essi un puntatore alle coordinate corrispondenti. Se si premerà sul puntatore verrà mostrata, dopo una brave animazione di zoom, una finestra modale contenente le informazioni di base relative alla località scelta. Il codice per creare la mappa e tutti i puntatori è in Javascript.



Figure 16 Homepage, mappa con puntatori Figure 17 Homepage, mappa con grotto selezionato

Nella sezione di pagina inferiore si trova la tabella che consente di visualizzare tutti i grotti e filtrarli per uno qualsiasi dei loro campi. La tabella è stata creata utilizzando le DataTables di Material Design Bootstrap ed è impostata per mostrare 15 località per pagina. È possibile filtrare i campi semplicemente premendo sui nomi delle colonne evidenziati in grassetto.



Figure 18 Homepage, tabella dei grotti

## Pagina Grotto

La pagina grotto serve a mostrare tutti i dettagli di una località specifica e di dare la possibilità agli utenti che hanno eseguito il login di votare un ristorante e di assegnargli delle immagini. Essa è strutturata in maniera dinamica, se vi sono immagini all’inizio genera un carosello che le mostrerà cambiando foto a intervalli di cinque secondi mentre se non ve ne saranno assegnate a quel grotto non lo mostrerà. Sotto il carosello vi sarà una lista contenente tutte le informazioni della località ed in seguito, se sarà stato eseguito il login, vi saranno le sezioni per votare e aggiungere foto.

 

Figure 19 Pagina grotto, informazioni Figure 20 Pagina grotto, valutazione e aggiunta immagini

## Pagina di login

La pagina di login è strutturata in maniera semplice e contiene un form che consentirà di collegarsi al proprio account inserendo l’email e la password, di raggiungere la pagina per la registrazione oppure di cambiare la password del proprio account.

Per verificare se un utente può collegarsi o meno vengono caricati tutti gli utenti dal database e si verifica se ne esiste uno con l’email inserita nel campo apposito, questo viene fatto richiamando il model DBConnection che gestisce la connessione con il database, se l’hash della password inserita equivale a quello salvato nel database e che tipo di permessi ha l’utente. Se vi è stato un errore la pagina verrà ricaricata mostrando l’errore in basso al form e ricompilando automaticamente i campi riempiti in precedenza.



Figure 21 Pagina di login, form di connessione

## Pagina di registrazione

La pagina di registrazione consentirà ad un utente di creare un nuovo account, essa contiene un form con i campi per inserire il nome, il cognome, lo username, l’email e la password ripetuta due volte per sicurezza.

Per generare un nuovo account si verifica che le password inserite siano uguali e che non ne esisti già uno con quell’email. Se vi è stato un errore la pagina verrà ricaricata mostrando l’errore in basso al form e ricompilando automaticamente i campi riempiti in precedenza. Una volta eseguita la registrazione con successo si verrà mandati alla pagina di login per accedere con l’utente appena generato.



Figure 22 Pagina di registrazione, form d'inserimento

## Pagina di ripristino della password

La pagina di ripristino della password è accessibile dalla pagina di login e consente di inserire l’email dell’account e la nuova password che si vuole impostare all’account. Una volta premuto il tasto di invio verrà impostato un token nel campo reset\_token dell’utente composto da 15 bytes di caratteri casuali che verrà anche inviato per email all’indirizzo inserito in precedenza. Quando si riceverà il messaggio di posta elettronica basterà premere sul link contenente il token e se il token

# Test

## Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l’adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-012 | **Nome:** | Import a card with KIC, KID and KIK keys, but not shown with the GUI |
| **Descrizione:** | Import a card with KIC, KID and KIK keys with no obfuscation, but not shown with the GUI | | |
| **Prerequisiti:** | Store on local PC: Profile\_1.2.001.xml (appendix n\_n) and Cards\_1.2.001.txt (appendix n\_n).  PIN (OTA\_VIEW\_PIN\_PUK\_KEY) and ADM (OTA\_VIEW\_ADM\_KEY) user right not set. | | |
| **Procedura:** | 1. Go to “Cards manager” menu,  in main page click “Import Profiles” link, Select the “1.2.001.xml” file, Import the Profile 2. Go to “Cards manager” menu,  in main page click “Import Cards” link, Select the “1.2.001.txt” file, Delete the cards,  Select the “1.2.001.txt” file, Import the cards 3. Research the “41795924770” Card, Click the imsi card link Check the card details 4. Execute the SQL: SELECT imsi, dir, keyset, cntr, rawtohex(kickey), rawtohex(kidkey), rawtohex(kikkey), rawtohex(chv), rawtohex(dap)FROM otacardkey a where imsi='340041795924770' ORDER BY keyset; | | |
| **Risultati attesi:** | Keys visible in the DB (OtaCardKey) but not visible in the GUI (Card details) | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

## Mancanze/limitazioni conosciute

Descrizione con motivazione di eventuali elementi mancanti o non completamente implementati, al di fuori dei test case. Non devono essere riportati gli errori e i problemi riscontrati e poi risolti durante il progetto.

# Consuntivo

Consuntivo del tempo di lavoro effettivo e considerazioni riguardo le differenze rispetto alla pianificazione (cap 1.7) (ad esempio Gannt consuntivo).

# Conclusioni

Quali sono le implicazioni della mia soluzione? Che impatto avrà? Cambierà il mondo? È un successo importante? È solo un’aggiunta marginale o è semplicemente servita per scoprire che questo percorso è stato una perdita di tempo? I risultati ottenuti sono generali, facilmente generalizzabili o sono specifici di un caso particolare? ecc

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc

# Bibliografia

## Bibliografia per articoli di riviste:

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo dell’articolo (tra virgolette),
3. Titolo della rivista (in italico),
4. Anno e numero
5. Pagina iniziale dell’articolo,

## Bibliografia per libri

1. Cognome e nome (o iniziali) dell’autore o degli autori, o nome dell’organizzazione,
2. Titolo del libro (in italico),
3. ev. Numero di edizione,
4. Nome dell’editore,
5. Anno di pubblicazione,
6. ISBN.

## Sitografia

* <https://www.gr.ch/DE/institutionen/verwaltung/dvs/alg/dokumentation/Vermessung/Dokumentliste%20AV/Brosch_LV95_it.pdf>, *Nuove coordinate per la Svizzera*, 20-09-2019
* <https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/tutorial>, *Google Maps API Documentation,* 20-09-2019
* <https://developers-dot-devsite-v2-prod.appspot.com/maps/documentation/javascript/examples/geocoding-reverse>, *Reverse Geocoding*, 20.09.2019
* <https://mdbootstrap.com/docs/jquery/>, *Material Design Bootstrap Documentation,* 20.09.2019

**Esempio:**

* http://standards.ieee.org/guides/style/section7.html, *IEEE Standards Style Manual*, 07-06-2008.

# Allegati

Elenco degli allegati, esempio:

* Diari di lavoro
* Codici sorgente/documentazione macchine virtuali
* Istruzioni di installazione del prodotto (con credenziali di accesso) e/o di eventuali prodotti terzi
* Documentazione di prodotti di terzi
* Eventuali guide utente / Manuali di utilizzo
* Mandato e/o Qdc
* Prodotto
* …