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 1

1.7.1 Analisi 1

1.7.2 Progettazione 1

1.7.3 Implementazione 2

1.7.4 Test 2

1.8 Analisi dei mezzi 3

1.8.1 Software 3

1.8.2 Hardware 3

2 Progettazione 3

2.1 Design dell’architettura del sistema 3

2.2 Design dei dati e database 3

2.3 Design delle interfacce 3

2.4 Design procedurale 3

3 Implementazione 5

4 Test 5

4.1 Protocollo di test 5

4.2 Risultati test 5

4.3 Mancanze/limitazioni conosciute 6

5 Consuntivo 6

6 Conclusioni 6

6.1 Sviluppi futuri 6

6.2 Considerazioni personali 6

7 Bibliografia 6

7.1 Bibliografia per articoli di riviste: 6

7.2 Bibliografia per libri 6

7.3 Sitografia 6

8 Allegati 6

# 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

### Hardware

Su quale piattaforma dovrà essere eseguito il prodotto? Che hardware particolare è coinvolto nel progetto? Che particolarità e limitazioni presenta? Che hw sarà disponibile durante lo sviluppo?

# Progettazione

## Design dell’architettura del sistema



Figure 8 Design di 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

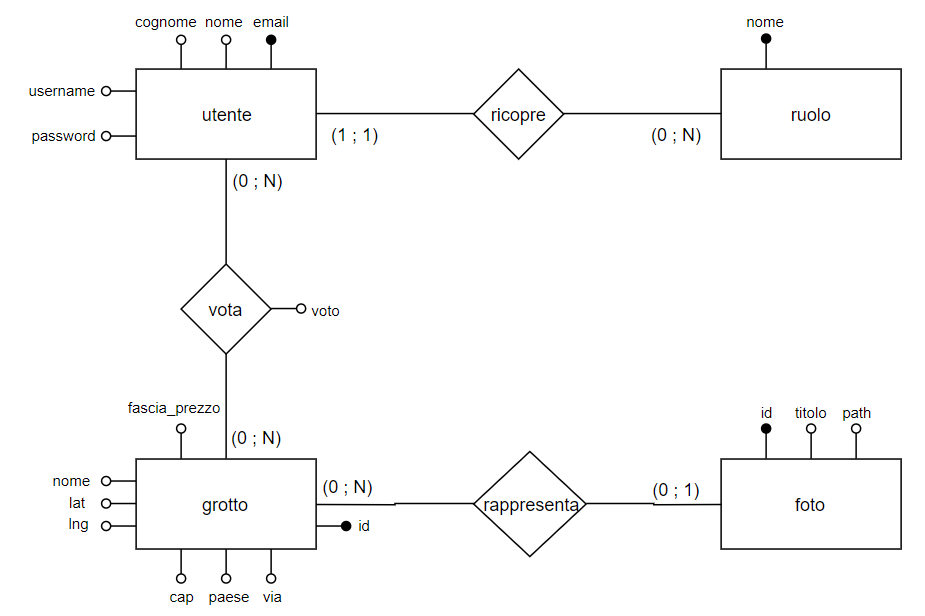


Figure 9 Diagramma ER del database

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

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.

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.

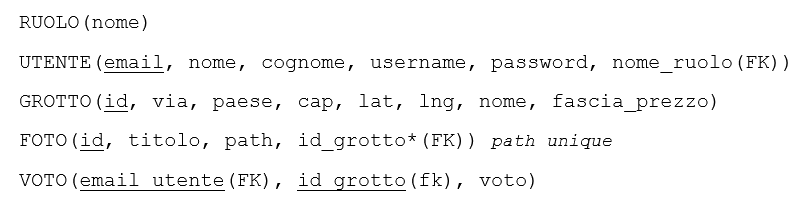


Figure 10 Schema logico del database

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(25) | Rappresenta il nome di battesimo dell’utente. |
| cognome | varchar(25) | Rappresenta il cognome dell’utente. |
| password | varchar(64) | Rappresenta la password dell’utente, verrà salvato l’hash del valore. |

## Design delle interfacce

Descrizione delle interfacce interne ed esterne del sistema e dell’interfaccia utente. La progettazione delle interfacce è basata sulle informazioni ricavate durante la fase di analisi e realizzata tramite mockups.

## Design procedurale

Descrive i concetti dettagliati dell’architettura/sviluppo utilizzando ad esempio:

* Diagrammi di flusso e Nassi.
* Tabelle.
* Classi e metodi.
* Tabelle di routing
* Diritti di accesso a condivisioni …

Questi documenti permetteranno di rappresentare i dettagli procedurali per la realizzazione del prodotto.

# 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.

# 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

1. URL del sito (se troppo lungo solo dominio, evt completo nel diario),
2. Eventuale titolo della pagina (in italico),
3. Data di consultazione (GG-MM-AAAA).

**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
* …