Documentazione progetto Gestione Biblioteca

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

Situazione iniziale **Errore. Il segnalibro non è definito.**

1.3 Scopo 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Use case 8

2.4 Pianificazione 8

2.5 Analisi dei mezzi 9

2.5.1 Software 9

2.5.2 Hardware 9

3 Progettazione 10

3.1 Design dell’architettura del sistema 10

3.1.1 Sitemap 10

3.2 Design dei dati e database 11

3.2.1 Diagramma E/R 11

3.3 Design delle interfacce 12

3.3.1 Pagina di registrazione 12

3.3.2 Pagina libri noleggiati 12

3.3.3 Pagina ricerca libri 13

3.3.4 Pagina libro 14

3.3.5 Pagina mostra noleggi 14

3.3.6 Pagina mostra utenti 15

3.3.7 Pagina aggiungi utente 15

3.3.8 Side menu 16

4 Implementazione 17

4.1 Creazione file di database 17

4.2 Interfaccia grafica del sito 18

4.2.1 Pagina di login 18

4.2.2 Pagina libri noleggiati 18

4.2.3 Pagina ricerca libri 18

4.2.4 Pagina libro 19

4.2.5 Pagina mostra noleggi 20

4.2.6 Pagina mostra utenti 21

4.2.7 Pagina aggiungi utente 21

4.2.8 Side menu 21

5 Test 22

5.1 Protocollo di test 22

5.2 Risultati test 25

5.3 Mancanze/limitazioni conosciute 26

6 Consuntivo 27

7 Conclusioni 28

7.1 Sviluppi futuri 28

7.2 Considerazioni personali 28

8 Bibliografia 29

8.1 Sitografia 29

9 Allegati 30

# Introduzione

## Informazioni sul progetto

Allievo: Luca Fumasoli

Classe: Informatica I3AC presso la Scuola d’Arti e Mestieri di Trevano

Docente responsabile: Guido Montalbetti

Data inizio: 23/01/2021

Data fine: 23/12/2021

## Abstract

## Situazione iniziale

Lo scopo di questo progetto è di catalogare ogni libro in possesso della biblioteca della scuola SAMT e renderli visibili su un sito web alla quale ogni studente avrà accesso con un account. Questo aiuterà gli studenti a trovare libri che gli interessano grazie a filtri e recensioni con lo scopo di incentivare gli studenti a leggere.

## Attuazione

Per questo progetto sarà necessario avere un account amministratore per creare utenti e gestire i noleggi, e ci sarà bisogno di account per gli studenti che possono visualizzare tutti i libri e cercare tra loro quale gli interessa.  
Per realizzare il sito ho usato i principali linguaggi web, ossia HTML, CSS, PHP e Javascript.

## Risultati

Sono riuscito a complementare la maggior parte degli obbiettivi di questo progetto, quindi il risultato è stato un sito funzionante per la ricerca ed il noleggio dei libri da parte degli studenti con un account amministratore che può visualizzare tutti gli utenti, visualizzare e gestire tutti i noleggi e può creare nuovi utenti.

## Scopo

Lo scopo del progetto è di creare un sito facile da navigare in cui gli studenti possono trovare libri che vogliono leggere.

# Analisi

## Analisi del dominio

Il sito verrà usato dagli studenti e dalla bibliotecaria. Dagli studenti per cercare libri che gli interessano e per riservarli ad una data scelta da loro per poi andare a ritirarli in biblioteca. Dalla bibliotecaria per aggiungere utenti e per marcare i noleggi come ritornati aggiungendo anche una valutazione.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| **ID: REQ-01** | |
| **Nome** | Registrazione alunni nel sistema |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Ci sarà una pagina di login per l’autenticazione |

|  |  |
| --- | --- |
| **ID: REQ-02** | |
| **Nome** | Meccanismo di conferma con invio di e-Mail |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |

|  |  |
| --- | --- |
| **ID: REQ-03** | |
| **Nome** | Accesso al sito da parte degli studenti |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Questi utenti potranno noleggiare libri |

|  |  |
| --- | --- |
| **ID: REQ-04** | |
| **Nome** | Accesso al sito da parte della bibliotecaria |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Questo utente potrà visualizzare tutti i noleggi |
| **002** | Potrà marcare i libri noleggiati come ritornati |
| **003** | Potrà aggiungere utenti |

|  |  |
| --- | --- |
| **ID: REQ-05** | |
| **Nome** | Lista libri contenuti in biblioteca |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Il sito dovrà leggere un file contenente tutti i libri disponibili che potranno essere visibili dagli studenti |

|  |  |
| --- | --- |
| **ID: REQ-06** | |
| **Nome** | Ricerca libri |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | Pagina per cercare libri |
| **Sotto requisiti** | |
| **001** | Nella pagina di ricerca si potrà cercare libri per autore o titolo del libro |

|  |  |
| --- | --- |
| **ID: REQ-07** | |
| **Nome** | Memorizzazione libri noleggiati |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | L’applicativo memorizzerà i libri noleggiati |
| **Sotto requisiti** | |
| **001** | La data di noleggio e consegna sarà salvata e la bibliotecaria potrà visualizzare i noleggi |
| **002** | L’id dello studente che ha noleggiato il libro sarà salvato |

|  |  |
| --- | --- |
| **ID: REQ-08** | |
| **Nome** | E-mail scadenza consegna libro |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | 1 settimana prima della consegna del libro dovrà essere inviata una e-mail per ricordare all’utente di riportare il libro |
| **Sotto requisiti** | |
| **001** | La mail conterrà la data in cui doveva essere restituito il libro ed il nome del libro |
| **002** | La mail sarà inviata da: [Trevano.biblioteca@-----.ch](mailto:Trevano.biblioteca@-----.ch) |

|  |  |
| --- | --- |
| **ID: REQ-09** | |
| **Nome** | Valutazione libro |
| **Priorità** | 2 |
| **Versione** | 1.0 |
| **Note** | Quando il libro viene consegnato si potrà valutare |
| **Sotto requisiti** | |
| **001** | Il libro potrà essere valutato da 1 a 6 stelle |
| **002** | Sulla pagina del libro apparirà la media delle valutazioni degli studenti |

|  |  |
| --- | --- |
| **ID: REQ-10** | |
| **Nome** | Top 10 libri viene salvata in un file |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | - |
| **Sotto requisiti** | |
| **001** | Il file potrà essere scaricato dagli utenti |

**Spiegazione elementi tabella dei requisiti:**

**ID**: identificativo univoco del requisito

**Nome**: breve descrizione del requisito

**Priorità**: indica l’importanza di un requisito nell’insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

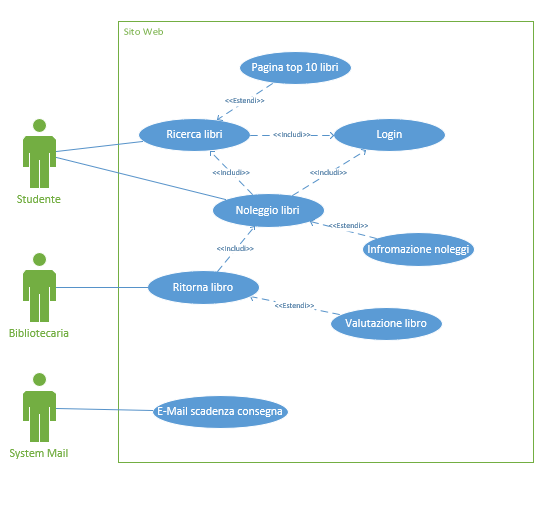
**Versione**: indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l’ultima versione, mentre le vecchie dovranno essere inserite nei diari.

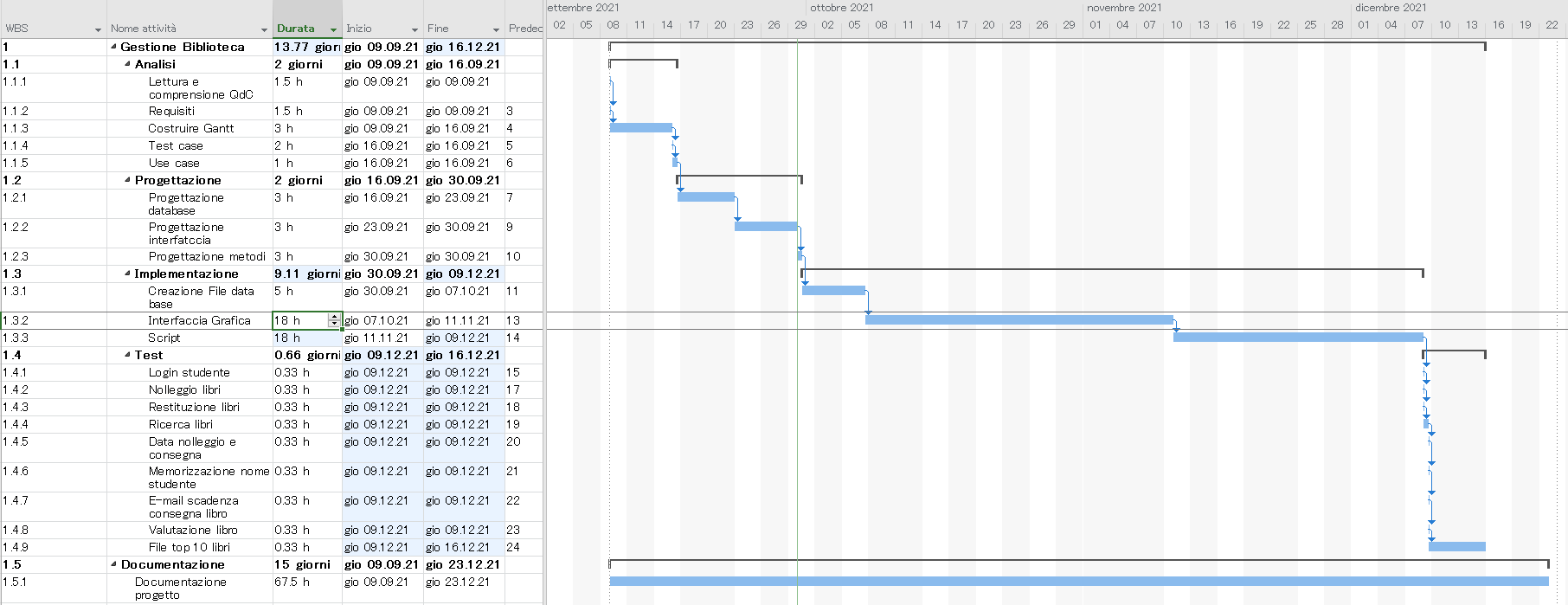
**Note**: eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti**: elementi che compongono il requisito.

## Use case



## Pianificazione



## Analisi dei mezzi

### Software

I software che ho usato sono:

* Microsoft office professional plus 2019:
  + Microsoft word: per fare la documentazione
  + Microsoft visio: per fare lo schema del sito e dei database
  + Microsoft Project: per fare il diagramma di Gantt
* Visual Studio Code versione 1.58.2
* XAMPP versione 3.3.0
* Google Chrome versione 96.0.4664.110

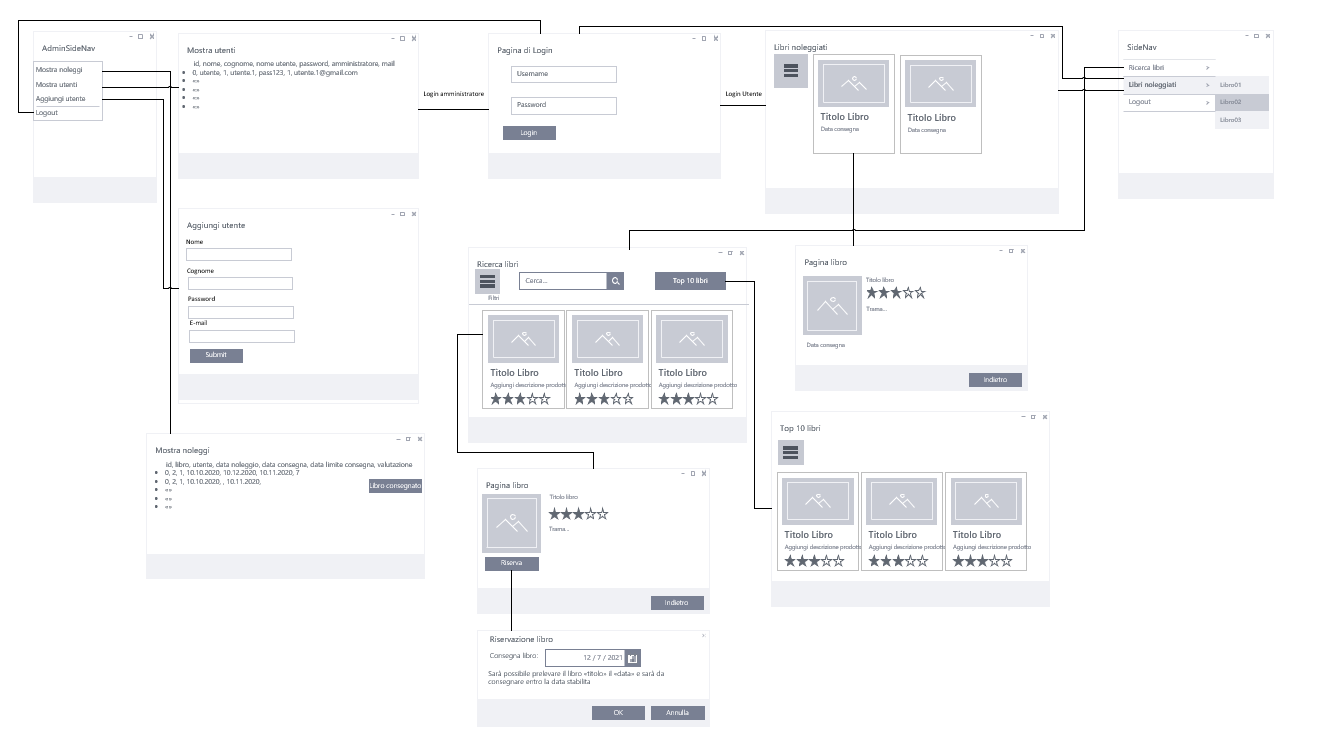
### Hardware

Come hardware ho usato solo i computer scolastici.

# Progettazione

## Design dell’architettura del sistema

### Sitemap



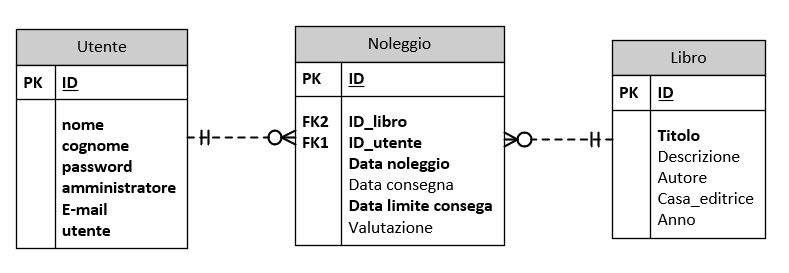
Per il sitemap ho usato visio per fare una rappresentazione minimalistica di come sembreranno le pagine.

Quando si accede al sito la prima pagina che si vede è la pagina di login, s

e si entra con un account di un utente ti manda all’interfaccia degli utenti e se entri come amministratore ti manda all’interfaccia degli amministratori. La prima pagina dopo il login dell’utente è la pagina con sui libri noleggiati, si possono vedere a che date devono essere ritornati e premendoci sopra si può andare alla pagina con su varie informazioni sul libro. Ogni pagina ha un side menu che si può aprire premendo l’icona su a sinistra, con questo menu si può cambiare pagina, andando alla pagina Ricerca libri si possono cercare libri per titolo od autore e premendo sopra al libro si può andare alla pagina del libro per noleggiarlo. Premendo su logout esce dall’account e ti riporta alla pagina di login. Accedendo come amministratore invece la prima pagina che si vede è la pagina mostra utenti, una pagina contenente tutti gli utenti e le loro informazioni. Usando il side menu degli amministratori si può andare a pagine diverse. Si può andare alla pagina aggiungi utenti che serve per aggiungere un utente immettendo nome, cognome, password e mail, o si può andare alla pagina mostra noleggi che serve per vedere tutti i noleggi fatti e segnare quando i libri vengono ritornati con una valutazione.

## Design dei dati e database

### Diagramma E/R



Questo diagramma E/R mostra le colonne dei file di database.

#### Utente

L‘entità contiene solo campi obbligatori, la primary key è un id, gli altri campi sono il nome, cognome,

password, un booleano per vedere se l’utente è un amministratore, l’E-mail ed il nome utente.

#### Noleggio

L‘entità noleggio contiene come campo obbligatorio il proprio id che è anche la primary key, l’id dell’utente che ha noleggiato il libro, l’id del libro noleggiato, la data del noleggio e la data del limite della consegna. I campi data consegna e valutazioni invece vengono compilati quando il libro viene ritornato quindi non sono obbligatori.

#### Libro

L‘entità libro ha come campi obbligatori l’id che fa da primary key ed il titolo del libro, gli altri campi sono la descrizione, l’autore, la casa editrice e l’anno di pubblicazione.

## Design delle interfacce

### Pagina di registrazione



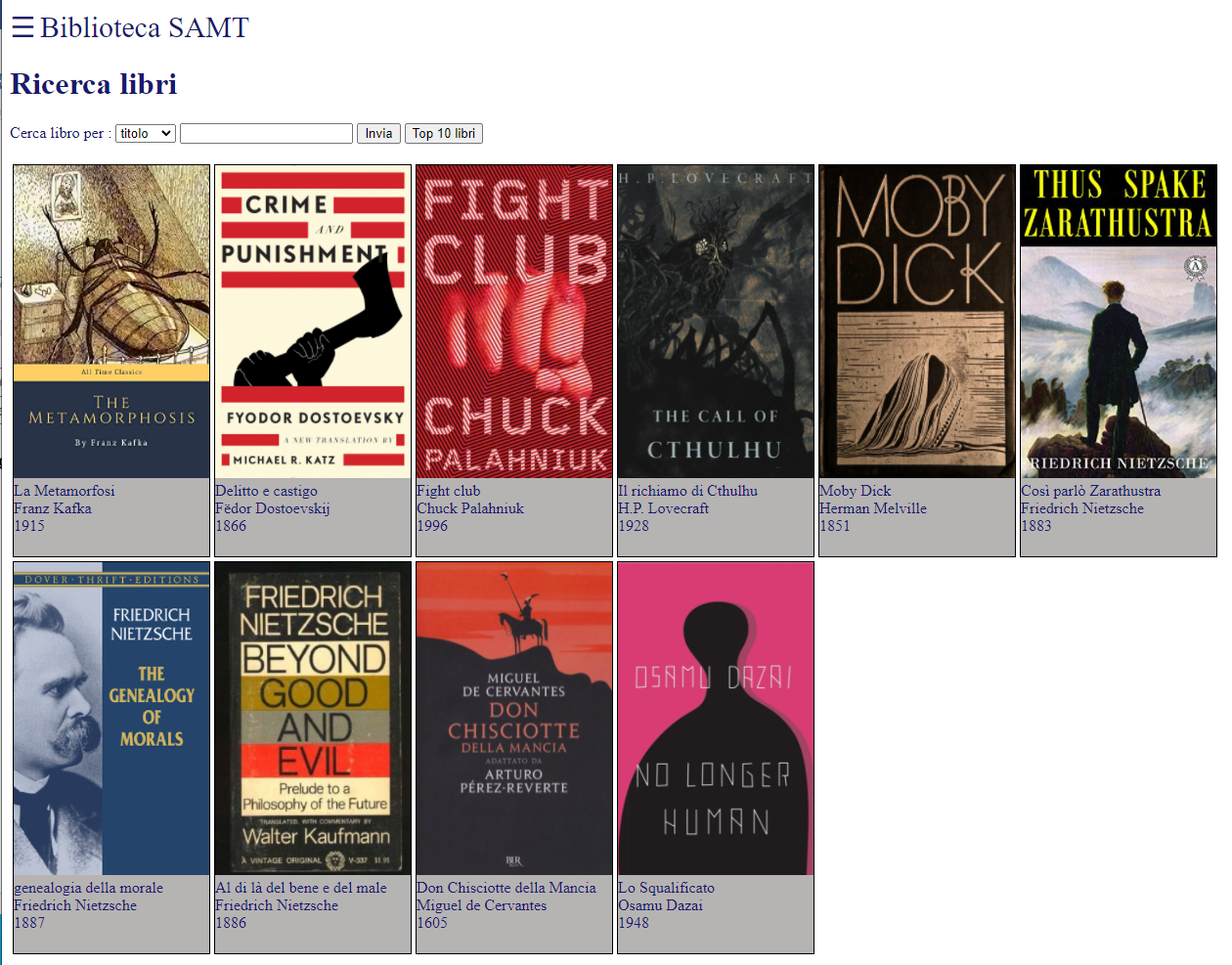
Pagina per entrare in un account per entrare nel sito.

### Pagina libri noleggiati



Pagina dei libri noleggiati dell’utente loggato.

### Pagina ricerca libri



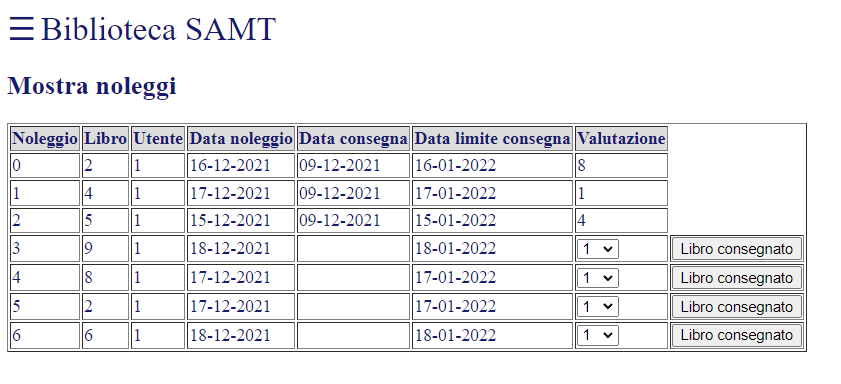
Pagina per ricercare libri.

### Pagina libro



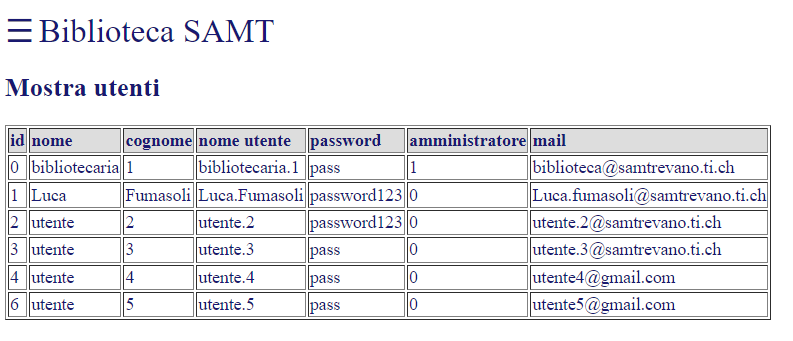
Pagina con su le informazioni del libro e per noleggiarlo.

### Pagina mostra noleggi



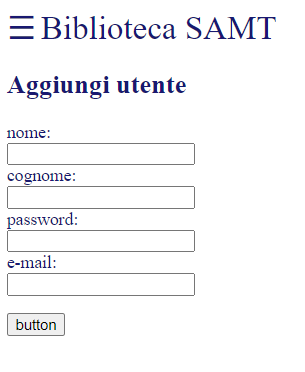
Pagina con su tutti i noleggi passati e quelli attuali con la possibilità di marcarli come ritornati.

### Pagina mostra utenti



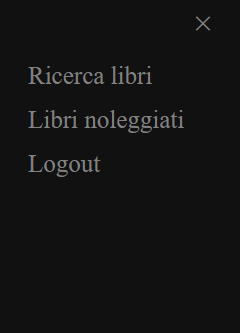
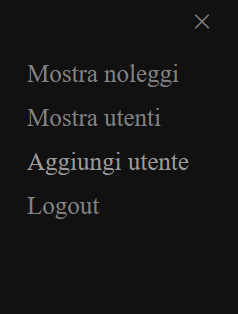
Pagina che mostra tutti gli utenti esistenti.

### Pagina aggiungi utente



Pagina per aggiungere utenti.

### Side menu

A dipendenza di se si entra come amministratore o no appaiono 2 side menu diversi per navigare le diverse pagine dei 2 tipi di profili.

# Implementazione

## Creazione file di database

Ho creato un file .csv per ogni tabella del diagramma E/R ed ho scritto i nomi delle colonne nella prima riga separati da una virgola, poi ho immesso i dati nelle righe sotto, separando ancora ogni colonna con una virgola.

Il file Libro.csv contiene una colonna per l'id, l'autore, l'anno, la casa editrice e la decsrizione.

Il file Noleggio.csv contiene una colonna per l'id, l'id dell'utente, l'id del libro, la data del noleggio, la data della consegna e la data del limite della consegna.

Il file Utente.csv contiene una colonna per l'id, iul nome, il cognome, la password, se è amministratore e l'E-mail.

## Interfaccia grafica del sito

### Pagina di login

Pagina nella quale si mette il proprio nome utente e password per entrare nel sito col proprio account.

Tutto il contenuto è in un form con dentro un div nella quale ci sono due input, uno per il nome utente e l’altro per la password, ed il bottone di Login che fa eseguire i controlli e la verifica dell’utente.  
Se si sbaglia il nome utente o la password te lo segnala mentre se sono giuste fa partire il metodo loginUtente che prende il nome utente e la password dal form e poi legge il contenuto del file Utente.csv e lo salva in una matrice. Controlla che il nome e la password non siano nulli, poi salva il nome utente in un cookie così che se si sbaglia password non si deve riscriverlo. Poi usa il metodo verfiyUser per controllare se il nome utente è presente nella matrice e se la password corrisponde a quella dell’utente. Infine usa il metodo isAdmin per controllare usando la matrice se l’account è un amministratore, e a dipendenza di sé lo è o no lo manda all’interfaccia degli utenti o quella degli amministratori.

### Pagina libri noleggiati

Questa pagina va a leggere il file Utente.csv per vedere cos’è l’id dell’utente loggato, poi legge il file Noleggi.csv per prendere tutti gli id dei libri noleggiati che non sono stati ancora consegnati sotto l’id dell’utente loggato e gli salva in un array. Va a leggere il file Libri.csv per prendere tutte le informazioni dei libri che hanno un id all’interno dell’array dei libri noleggiati. Se l’array risulta vuoto stampa un messaggio per dire che non ci sono libri noleggiati da questo utente altrimenti fa partire un metodo per stampare i libri noleggiati. Il metodo controlla quanti libri ci sono e gli stampa in righe di 6 finché può e l’ultima riga ne avrà quanti ne avanzano. Tutti i libri sono stampati in un div contenente la copertina, che viene presa da una cartella con dentro tutte le copertine dei libri con id del libro come nome dell’immagine, e sotto stampa il titolo, l’autore e l’anno di pubblicazione del libro.

### Pagina ricerca libri

Questa pagina va a leggere il file Libri.csv e gli salva in una matrice, controlla quanti libri ci sono e gli stampa in righe di 6 finché può e l’ultima riga ne avrà quanti libri avanzano dall’ultimo multiplo di 6. Tutti i libri sono stampati in un div contenente la copertina, che viene presa da una cartella con dentro tutte le copertine dei libri con id del libro come nome dell’immagine, e sotto stampa il titolo, l’autore e l’anno di pubblicazione del libro. In cima alla pagina ci sono due input per cercare tra i libri. Usando il select si può scegliere se cercare per titolo od autore del libro. Poi premendo invia viene svolta una ricerca nella matrice con dentro tutti i libri per vedere se la stringa cercata è contenuta nel titolo o nel nome dell’autore, i libri trovati vengono salvati in una matrice e poi stampati nello stesso modo in cui vengono stampati i libri al caricamento della pagina. Se non vengono trovati libri allora non stampa tutti i libri.

### Pagina libro

Questa pagina viene aperta quando si schiaccia su un libro dalla pagina libri noleggiati o la pagina di ricerca dei libri. L’id del libro viene passato dentro l’URL e viene usato per prendere la riga della matrice che corrisponde alla riga con le informazioni del libro e poi le stampa nei 2 div sotto.

Per prendere l’id ho creato il metodo getBookID che salva l’URL in una variabile e poi prende solo la parte finale della stringa che contiene l’id.

function getBookID() {

        $url = "http://$\_SERVER[HTTP\_HOST]$\_SERVER[REQUEST\_URI]";

        return substr($url, strpos($url, "=") + 1

}

Il div a sinistra contiene il titolo del libro con la copertina sotto. Mentre il div a destra contiene l’autore, l’anno di pubblicazione, la casa editrice e la trama.   
Viene letto il file dei noleggi per vedere se il libro è già stato noleggiato dall’utente loggato e se non è stato noleggiato appare il bottone Noleggia libro nel div a destra. Se invece è stato noleggiato non appare.  
Premendo sul bottone Noleggia libro la pagina viene coperta ed appare un div per noleggiare il libro.



Il div contiene un form nella quale si può immettere la data in cui si vuole ritirare il libro. Premendo su Noleggia il libro viene noleggiato e viene aggiunta una linea per il noleggio nel file Noleggio.csv.  
Premendo su cancella invece il div sparisce e la pagina torna come prima.

### Pagina mostra noleggi

Questa pagina contiene tutti i noleggi fatti dagli utenti. Viene letto il file noleggi.csv e salvato in una matrice.  
Poi questa matrice viene stampata in una tabella ed ha dipendenza di sé la riga contiene un valore nella colonna data di consegna stabilisce se il libro è già stato ritornato.

<?php if ($nol[$row][4] == "") { ?>

     <select id="rating" name="rating">

        <option value="1">1</option>

…

        <option value="10">10</option>

     </select>

<input type="submit” …>

<?php }else { ?>

     <p> <?php echo $nol[$row][6] ?></p>

<?php } ?>

Se il libro non è stato ritornato fa apparire un select per la valutazione e di fianco un bottone per marcare il libro come ritornato.  
Quando viene premuto il bottone la valutazione selezionata nel select viene aggiunta al file dei noleggi assieme alla data attuale come data di consegna. Quando questo viene fatto il libro sarà marcato come ritornato ed l’utente non visualizzerà più i libro nella pagina dei libri noleggiati.

### Pagina mostra utenti

Questa pagina legge il file degli utenti e lo salva in una matrice, poi stampa tutte le righe in una tabella.

### Pagina aggiungi utente

Pagina contenente un form con dentro un div con i campi per inserire il nome, il cognome, la password e la E-mail. Quando si preme sul bottone se sono stati compilati tutti i campi viene dichiarato un array per inserirci dentro i valori. Per l’id viene letto il file degli utenti e ricavato il numero di righe per sapere quanti id ci sono e quale dare al nuovo utente. Il nome, cognome, password ed E-mail vengono presi dal form, per l’email viene fatto un controllo. Poi per il nome utente viene messo il nome de il cognome concatenati divisi da un punto. E come amministratore viene messo 0 di default siccome non si possono creare account amministratori.

$array = array();

            $array[0] = count($csv)-1;

            $array[1] = $\_POST['nome'];

            $array[2] = $\_POST['cognome'];

            $array[3] = $\_POST['nome'] . "." .$\_POST['cognome'];

            $array[4] = $\_POST['password'];

            $array[5] = 0;

            $array[6] = $\_POST['email'];

### Side menu

A dipendenza di sé l’account loggato è un amministratore o no si apre il side menu degli amministratori o quello degli utenti.

<div id="mySidenav" class="sidenav">

    <?php include "./AdminSideNav.php" ?>

</div>

Per mostrare il side menu su ogni pagina ho usato php per fare un include della pagina e metterla dentro un div che si può aprire o chiudere nella pagina.

Dentro il div c’è una lista delle pagine alle quali l’account può andare con un riferimento alla pagina per mandare l’utente quando ci preme sopra.

### Implementazione javascript e CSS

Per avere i metodi javascript e il CSS centralizzati in un proprio file ho usato le seguenti linee di codice su ogni pagina:

<script src="myscripts.js"></script>

<link rel="stylesheet" type="text/css" href="./style.css">

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Login studente |
| **Descrizione:** | Funzionalità pagina login utente con autenticazione password | | |
| **Prerequisiti:** | È necessario che nel database ci sia un utente con la quale registrarsi e che ci sia già una pagina di login. | | |
| **Procedura:** | 1. Aprire un browser ed andare all’URL della pagina del sito 2. Immettere nome utente e password dell’account negli input 3. Premere “Login” | | |
| **Risultati attesi:** | Acceso alla pagina home del sito | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-003 | **Nome:** | Noleggio libri |
| **Descrizione:** | Gli studenti dovranno poter noleggiare i libri sul sito | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Entrare in un account 2. Andare sulla pagina di ricercar dei libri 3. Selezionare il libro che si vuole noleggiare 4. Premere il bottone Noleggia libro sulla pagina 5. Inserire la data del ritiro 6. Premere il bottone Noleggio | | |
| **Risultati attesi:** | Il libro apparirà nella pagina libri noleggiati dell’utente e la bibliotecaria vedrà il noleggio nella pagina mostra noleggi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-003 | **Nome:** | Restituzione libri |
| **Descrizione:** | La bibliotecaria dovrà poter marcare i libri come restituiti e quando succede l’utente non dovrà più poterli visualizzare nella pagina libri noleggiati. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Entrare nell’account della bibliotecaria 2. Andare sulla pagina mostra noleggi 3. Inserire una valutazione sul libro da ritornare 4. Premere il bottone Libro consegnato | | |
| **Risultati attesi:** | Il libro non sarà più presente nella pagina dei libri noleggiati dell’utente e la bibliotecaria vedrà il libro come ritornato nella tabella dei noleggi. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-005 | **Nome:** | Ricerca libri per titolo |
| **Descrizione:** | Ci sarà una pagina per cercare tra i libri contenuti in biblioteca | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Entrare nell’account di uno studente 2. Andare sulla pagina di ricerca dei libri 3. Selezionare titolo nel select della ricerca 4. Inserire un titolo nell’input 5. Premere invia | | |
| **Risultati attesi:** | Conferma che sulla pagina siano presenti solo libri con dentro il titolo la stringa inserita nell’input. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-005 | **Nome:** | Ricerca libri per autore |
| **Descrizione:** | Ci sarà una pagina per cercare tra i libri contenuti in biblioteca | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Entrare nell’account di uno studente 2. Andare sulla pagina di ricerca dei libri 3. Selezionare autore nel select della ricerca 4. Inserire il nome dell’autore nell’input 5. Premere invia | | |
| **Risultati attesi:** | Conferma che sulla pagina siano presenti solo libri scritti dall’autore cercato. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-006 | **Nome:** | Memorizzazione id studente |
| **Descrizione:** | Quando un libro sarà noleggiato e restituito verrà salvato l’id dello studente. | | |
| **Prerequisiti:** |  | | |
| **Procedura:** | 1. Entrare con l’account di uno studente 2. Selezionare un libro non noleggiato dall’account che si sta usando 3. Premere su noleggia ed inserire una data non ancora trascorsa 4. Andare sulla pagina dei noleggi della bibliotecaria | | |
| **Risultati attesi:** | Sulla pagina dovrà esserci l’id dello studente che ha noleggiato il libro | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-007 | **Nome:** | E-mail scadenza consegna libro |
| **Descrizione:** | 1 settimana prima della consegna del libro dovrà essere inviata una e-mail per ricordare all’utente di riportare il libro | | |
| **Prerequisiti:** | La mail della biblioteca dovrà essere già creata | | |
| **Procedura:** | 1. Noleggiare un libro 2. Modificare la data del limite della consegna del database a poco dopo 7 giorni. 3. Guardare nella propria mail se si ha ricevuto un’e-mail di scadenza | | |
| **Risultati attesi:** | La mail dovrà essere mandata da biblioteca@samtrevano.ti.ch e dovrà contenere la data prevista di consegna del libro con un messaggio di default. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-008 | **Nome:** | Valutazione libro |
| **Descrizione:** | Quando si consegna un libro si potrà dare una valutazione da 1 a 6 | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | 1. Entrare nell’account della bibliotecaria 2. Andare alla pagina dei libri noleggiati 3. Selezionare il numero di stelle che lo studente vuole valutare il libro 4. Premere su restituito | | |
| **Risultati attesi:** | La valutazione dovrà essere aggiunta al file delle valutazioni e dovrà contare nella media delle valutazioni sulla pagina del libro | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-009  REQ-009 | **Nome:** | File top 10 libri |
| **Descrizione:** | Quando si consegna un libro si potrà dare una valutazione da 1 a 6 | | |
| **Prerequisiti:** | - | | |
| **Procedura:** | 1. Entrare nell’account di uno studente 2. Andare nella ricerca dei libri 3. Premere sulla pagina per i top 10 libri | | |
| **Risultati attesi:** | Si dovranno vedere i 10 libri con la valutazione media più alta | | |

## Risultati test

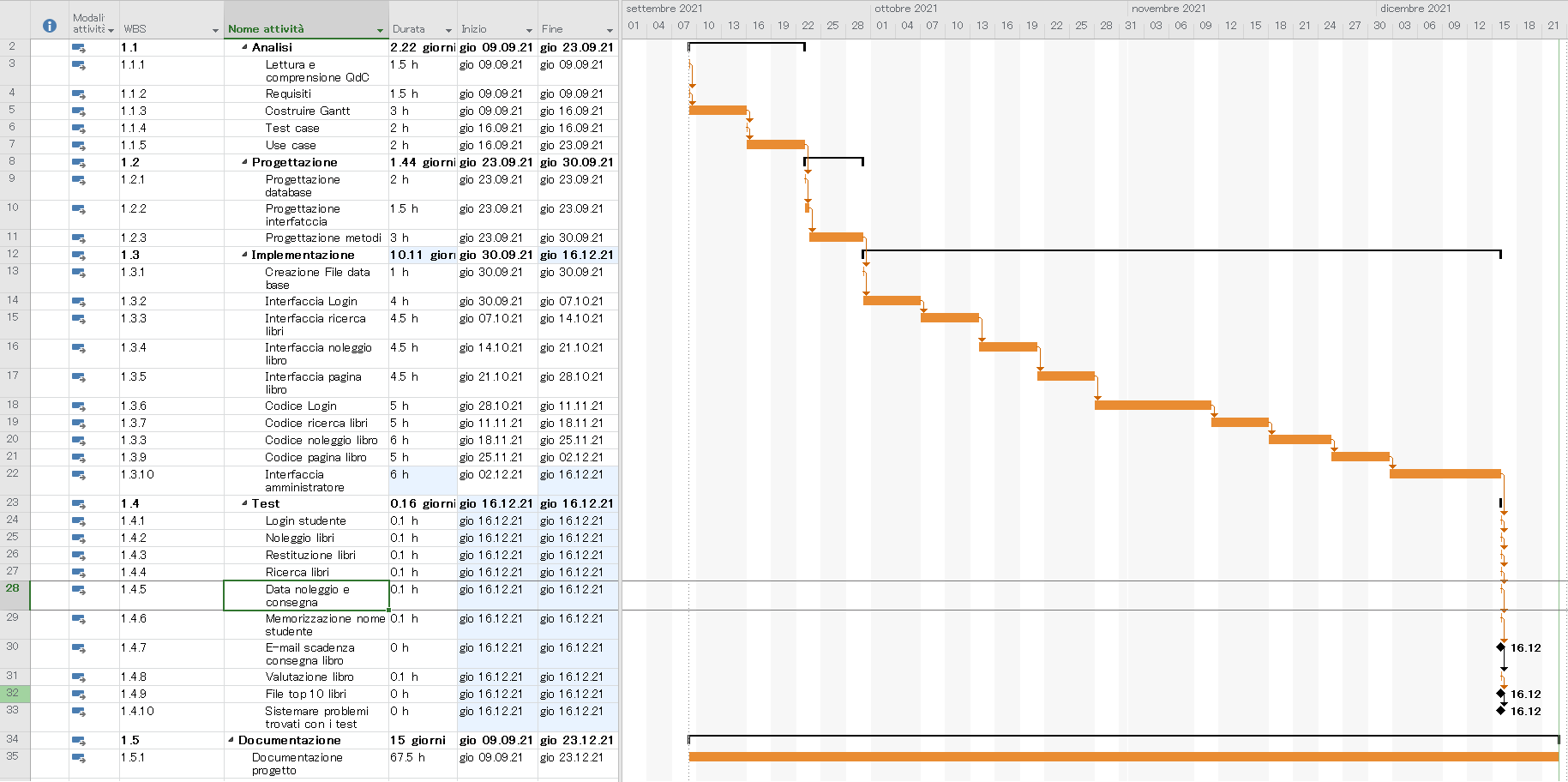
|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Funzionamento | Commento | Data |
| TC-001 | OK | Ho seguito la procedura ed ho ottenuto il risultato atteso. Ho immesso il nome utente e la password corretta e quando ho premuto su login mi ha portato alla pagina home del sito. | 23.12.2021 |
| TC-002 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso. Sono entrato in un account ed ho selezionato un libro non noleggiato dall’utente. Ho premuto sul bottone noleggia libro, ho inserito una data non ancora trascorsa nell’input della data di ritiro e poi ho premuto su noleggia. Poi il libro è apparso sulla pagina dei libri noleggiati.  Nella pagina mostra noleggi della bibliotecaria è apparsa una nuova riga sulla tabella dei noleggi con tutte le informazioni corrette sul noleggio. | 23.12.2021 |
| TC-003 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso. Sulla pagina mostra noleggi della bibliotecaria ho inserito una valutazione nel select delle valutazioni e poi ho premuto sul bottone Libro consegnato. Dopo di questo il libro è sparito dalla pagina dei libri noleggiati dell’utente che aveva noleggiato il libro e sulla pagina dei noleggi della bibliotecaria il bottone per consegnare il libro è sparito, la valutazione selezionata nel select è stata marcata come la valutazione del libro ed è stata inserita la data corrente come la data della consegna del libro. | 23.12.2021 |
| TC-004 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso. Sulla pagina della ricerca dopo aver premuto invia sono apparsi solo libri con dentro il titolo la stringa cercata. | 23.12.2021 |
| TC-005 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso. Sulla pagina della ricerca dopo aver premuto invia sono apparsi solo libri scritti da un autore con il nome contenente la stringa cercata. | 23.12.2021 |
| TC-006 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso. Quando un account noleggia un libro il suo id viene aggiunto assieme alle altre informazioni sul noleggio al file dei noleggi e quando l’account dell’amministratore va sulla pagina dei noleggi vede l’id corretto sul noleggio. | 23.12.2021 |
| TC-007 | NOK | Non ho potuto testare l’invio della mail perché il proxy blocca l’invio e non ho avuto il tempo di farlo a casa. | 23.12.2021 |
| TC-008 | OK | Ho eseguito la procedura ed ho ottenuto il risultato atteso.  Sulla pagina dei noleggi della bibliotecaria i libri noleggiati non ancora ritornati hanno nella colonna della valutazione un select per scegliere un voto da 1 a 6, quando un valore viene inserito ed il bottone del ritorno del libro viene premuto la valutazione viene marcata come il voto scelto. | 23.12.2021 |
| TC-009 | NOK | Non ho avuto tempo di fare la pagina top 10 libri. | 23.12.2021 |

## Mancanze/limitazioni conosciute

Le mancanze/limitazioni conosciute sono:

* Non ho avuto tempo di fare la pagina dei top 10 libri.
* Per via del proxy non ho potuto testare ed implementare l’invio della mail nelle funzioni del sito.

# Consuntivo



# Conclusioni

## Sviluppi futuri

Come sviluppi futuri si potrebbero implementare una pagina per aggiungere libri e le 2 cose che non sono riuscito a fare, ossia la pagina con i top 10 libri e l’invio della mail di avviso per la consegna del libro.

## Considerazioni personali

Trovo di aver lavorato abbastanza bene nello svolgimento del progetto e un po’ meno nella documentazione.

Avrei dovuto fare la documentazione in parallelo con lo svolgimento del progetto in modo da non dover svolgerne la maggior parte negli ultimi giorni. Un’altra cosa che potrei migliorare sono i diari e ricordarmi di farli durante la giornata di lavoro per scrivere dettagliatamente tutti i passaggi e tutte le problematiche riscontrate.

# Bibliografia

## Sitografia

* <https://moodle.edu.ti.ch/cpt/>
* [https://stackoverflow.com/](https://stackoverflow.com/¨)

# Allegati

I file allegati sono:

* Abstract Gestione Biblioteca.docx
* Use case Gestione Biblioteca.vsdx
* Progettazione interfaccia sito web.vsdx
* Progettazione database.vsdx
* GestioneBiblioteca.mpp