

Candidato: Mattia Lazzaroni
Azienda: Scuola Arti e Mestieri / CPT Trevano-Canobbio
Periodo: 03.09.2019 – 20.12.2019
Presentazione: 07-17 gennaio 2020

Situazione iniziale

Al momento esistono già molti siti che permettono di svolgere delle riservazioni, tuttavia il mio prodotto ha dei meccanismi particolari e delle funzionalità diverse dagli altri. L'obiettivo del lavoro è quello di realizzare un sito web che dia la possibilità all'utente di vedere una lista di alloggi, vedere i dettagli di un alloggio ed eventualmente riservarlo. Dopo aver completato una riservazione, l'utente riceverà una conferma per email ed eventualmente potrà stampare il pdf che riceve. Nel mio sito è possibile registrarsi solo come utenti, mentre per il login oltre agli utenti normali, ci sono anche gli amministratori gerenti e gli amministratori. I due amministratori avranno dei menu riservati in cui potranno gestire il database, ad esempio l'amministratore gerente potrà stampare tutte le riservazioni, gestire i clienti e caricare le disponibilità, mentre l'amministratore potrà gestire i gerenti e caricare delle strutture. Deve inoltre essere presente un sistema di gestione della fatturazione del sito.

Attuazione

Per la risoluzione del problema del cliente ho iniziato dall'analisi, in cui sono andato a capire quale fosse lo scopo del progetto e in cui ho tenuto delle interviste con il cliente per chiarirmi al meglio. Finita questa fase, sono passato alla progettazione, in cui ho inizialmente designato il database e dove ho poi realizzato dei semplici mockup delle varie pagine del sito. A quel punto ho poi iniziato ad implementare il sito web, creando prima il database con MySQL e poi la struttura vera e propria partendo da un template Bootstrap. I linguaggi di programmazione utilizzati sono: PHP per la logica, HTML per la grafica e CSS per lo stile e SQL per la gestione della banca dati. Ho anche usufruito di due librerie PHP, la prima, chiamata PHPMailer, mi è servita per riuscire a mandare delle email agli utenti tramite il codice, mentre la seconda, mPDF, mi ha permesso di creare dei PDF da mostrare agli utenti o agli amministratori gerenti, contenenti degli elementi HTML (ad esempio delle tabelle). La fase finale è stata quella di testing, in cui ho testato le funzionalità principali realizzate nell'implementazione. Durante tutto il corso del progetto c'è stata una fase di documentazione delle attività svolte, anche grazie a un diario che tenevo giornalmente.

Risultati

Il progetto è stato completato quasi interamente, se non fosse che manca la gestione della fatturazione del sito. Ho deciso di lasciare questa parte per ultima in quanto non mi era tanto chiara e in quanto non la ritenevo indispensabile per la riuscita e la funzionalità del prodotto, e dal momento che mi sono accorto di non aver più abbastanza tempo a disposizione per poter implementare questa parte, ho deciso di tralasciarla. Per quanto riguarda la riservazione, ho trovato molto complessa la query che andava a entrare in ben 5 tabelle per poter trovare una camera libera in una certa data, di conseguenza la query non è precisa al 100% nel cercare una disponibilità. Per il resto, c'è la possibilità di effettuare una riservazione e di gestire tutti i vari elementi della banca dati tramite delle pagine riservate, che era l'obiettivo principale del progetto. Mi posso quindi ritenere soddisfatto del lavoro svolto.

Gestione disponibilità alloggi

Titolo del progetto: Gestione disponibilità alloggi
Alunno/a: Mattia Lazzaroni
Classe: SAM I4
Anno scolastico: 2019/20
Docente responsabile: Guido Montalbetti

1	Introduzione.....	3
1.1	Informazioni sul progetto.....	3
1.2	Abstract	3
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	7
2.4	Pianificazione	7
2.4.1	Pianificazione intermedia	9
2.5	Analisi dei mezzi.....	11
2.5.1	Software	11
2.5.2	Hardware.....	11
3	Progettazione	12
3.1	Design dei dati e database.....	12
3.2	Design delle interfacce.....	13
4	Implementazione	16
4.1	Struttura.....	16
4.2	PHPMailer	17
4.3	mPDF	18
4.4	"ob_start()" e "ob_end_clean()"	19
4.5	Database.sql	20
4.6	ClearData.sql.....	21
4.7	.htaccess	21
4.8	errore.php.....	21
4.9	server.php	22
4.10	index.php	22
4.11	signup.php	24
4.12	check-your-email.php	25
4.13	verify.php	26
4.14	login.php	27
4.15	logout.php	28
4.16	amministratore.php e amministratore_gerente.php	29
4.17	carica-disponibilita.php	30
4.18	gestisci-clienti.php	31
4.19	stampa.php	33
4.20	carica-strutture.php.....	35
4.21	gestisci-gerenti.php	36
4.22	dettagli.php	39
5	Test.....	43
5.1	Protocollo di test.....	43
5.2	Risultati test.....	51
5.3	Mancanze/limitazioni conosciute.....	56
6	Consuntivo.....	57
7	Conclusioni	59
7.1	Sviluppi futuri.....	59
7.2	Considerazioni personali.....	59
8	Bibliografia	59
8.1	Sitografia	59
9	Allegati.....	60

1 Introduzione

1.1 Informazioni sul progetto

- Allievo: Mattia Lazzaroni.
Superiore professionale: Guido Montalbetti.
- Scuola d'Arti e Mestieri di Trevano, sezione Informatica, classe 4, progetti individuali.
- Data inizio: 03.09.2019
Data fine: 20.12.2019

1.2 Abstract

Do you want to organize a last-minute trip for a weekend at the sea with your girlfriend, but you don't have time to reserve an accommodation with a complex site? Then my site for accommodation availability management is what you need. This is an incredibly simple, intuitive and user-friendly website that will allow you to find the accommodation that's right for you in less than 5 minutes. With the simple registration mechanism, you can create your account and you can immediately make one or more reservations. The filter system will help you search for what you want with minimum effort. The detail page of an accommodation gives you the opportunity to see a map with the place of accommodation, to facilitate the travel to your hotel/camping site/bed & breakfast. Surely there are many other websites that offer the same services, but mine is unique thanks to its simplicity and its speed in booking an accommodation.

1.3 Scopo

Lo scopo di questo progetto è quello di creare un sito Web che permetta agli utenti di visualizzare e riservare degli alloggi. Innanzitutto, l'utente deve poter trovare degli alloggi disponibili in base ai filtri che andrà ad inserire. Una volta scelto il proprio alloggio, se volesse riservarlo, l'utente dovrà registrarsi andando a riempire dei campi minimi. Per confermare la registrazione, verrà inviato un link di conferma per email. Inoltre, l'utente dovrà confermare la riservazione tramite email. Ci dovrà anche essere una sezione riservata unicamente agli amministratori e una riservata agli amministratori gerenti. Dopo aver riservato un alloggio l'utente potrà stampare la propria riservazione con tutti i dettagli. Gli amministratori gerenti avranno la possibilità di stampare le riservazioni di una certa persona o di un certo periodo. Ci dovrà inoltre essere una parte per la gestione delle fatturazioni.

2 Analisi

2.1 Analisi del dominio

Spesso per poter trovare un alloggio in cui passare la notte occorre girare molti siti perché quelli presenti sono incompleti e non forniscono tutte le informazioni oppure sono troppo complicati e risultano difficili da usare. Il mio prodotto servirà a quegli utenti che cercano un sito semplice ed intuitivo con cui riservare un alloggio in pochi minuti. Per poter accedere al mio sito sarà necessario come minimo un dispositivo con accesso ad Internet, un'email, un recapito telefonico e una conoscenza basilare per poter usare un sito Web munito di registrazione e conferma tramite email.

2.2 Analisi e specifica dei requisiti

In questo capitolo sono andato a suddividere i vari requisiti che ho trovato nel QdC, con i rispettivi sotto requisiti. Come primo requisito ho messo la verifica dell'hosting, senza la quale il sito non può esistere. In seguito, ho aggiunto i requisiti della realizzazione dei menu per gli amministratori e gli amministratori gerenti. Come quarto requisito c'è la stampa degli alloggi, con le varie possibilità di filtrare i risultati e di vedere le informazioni di ogni alloggio. Il successivo è il requisito su cui si basa il sito, ovvero la possibilità di svolgere una riservazione. Questo sarà possibile solo dopo essersi registrati o loggati. Gli ultimi due requisiti concernono la gestione delle stampe e delle fatturazioni.

ID: REQ-001	
Nome	Verifica dell'hosting
Priorità	1
Versione	1.0
Note	È necessario verificare che l'hosting fornito dalla scuola funzioni correttamente

ID: REQ-002	
Nome	Creazione menu amministratori
Priorità	1
Versione	1.0
Note	È necessario creare una sezione riservata agli amministratori del sito
Sotto requisiti	
001	L'amministratore potrà gestire gli amministratori gerenti
002	L'amministratore potrà caricare le strutture legate al sito
003	L'amministratore potrà stampare la fatturazione per un determinato periodo e per una specifica struttura

ID: REQ-003	
Nome	Creazione menu amministratori gerenti

Priorità	1
Versione	1.0
Note	È necessario creare una sezione riservata agli amministratori gerenti
Sotto requisiti	
001	L'amministratore gerente potrà gestire i clienti
002	L'amministratore gerente potrà caricare la disponibilità delle strutture
003	L'amministratore gerente potrà stampare le riservazioni per un determinato periodo e/o persona

ID: REQ-004	
Nome	Stampa degli alloggi
Priorità	1
Versione	1.0
Note	Dovrà esserci una pagina che mostra tutti gli alloggi disponibili
Sotto requisiti	
001	Sarà possibile filtrare i risultati
002	Sarà possibile visualizzare i dettagli di ogni singolo alloggi
003	Sarà possibile fare una riservazione di un alloggio

ID: REQ-005	
Nome	Possibilità di svolgere una riservazione
Priorità	1
Versione	1.0
Note	L'utente avrà la possibilità di svolgere delle riservazioni
Sotto requisiti	
001	Sarà obbligatorio registrarsi per poter completare una riservazione
002	Bisognerà mandare per email un link per confermare la registrazione
003	Sarà necessario anche di una pagina per svolgere il login al secondo accesso al sito

ID: REQ-006	
Nome	Gestire il sistema di stampa

Priorità	1
Versione	1.0
Note	Coloro che riservano un alloggio possono stampare la riservazione con tutti i dettagli
Sotto requisiti	
001	Gli amministratori del sito potranno stampare le fatturazioni di un certo periodo e per una struttura
002	Gli amministratori gerenti potranno stampare le riservazioni di un certo periodo o di una certa persona

ID: REQ-007	
Nome	Gestire le fatturazioni
Priorità	2
Versione	1.0
Note	Il sito deve prevedere di poter gestire la fatturazione delle strutture
Sotto requisiti	
001	Il sito guadagna tramite le riservazioni
002	Vi deve essere la possibilità di modificare le percentuali percepite

2.3 Use case



Use Case del sito web

Nel seguente Use Case Diagram troviamo tre attori: utente, amministratore gerente e amministratore. Per quanto riguarda l'amministratore e l'amministratore gerente non mi soffermerò a spiegare, in quanto hanno entrambi solo tre Use Case facilmente comprensibili tramite l'immagine qua sopra. L'utente invece è l'attore che richiede spiegazioni in quanto complicato. Egli può riservare un alloggio, tuttavia per compiere questa azione deve prima cercarlo obbligatoriamente. Durante la ricerca, volendo, può filtrare i risultati. Questa ricerca produrrà degli alloggi che se cliccati mostreranno una pagina con i dettagli. È proprio da questa pagina dei dettagli che è possibile svolgere la riservazione. Per poter riservare, l'utente deve registrarsi al sito web e deve confermare la registrazione tramite e-mail. In seguito, se l'utente dovesse tornare al sito non dovrà più registrarsi, bensì loggarsi.

2.4 Pianificazione

La pianificazione del progetto è stata effettuata mediante la realizzazione di un diagramma di Gantt. Inizialmente ho pianificato un'analisi di 16 ore, in cui vado a leggere il QdC e a porre le domande al mio responsabile. Successivamente vado ad analizzare il dominio, i requisiti e i mezzi a disposizione. Infine, realizzo il Gantt e l'analisi termina qui, con una milestone. Dopo di questo, inizio la progettazione, che dura 8 ore. Durante questa fase vado a progettare il database e le interfacce. Finita la progettazione ho inserito un'altra milestone. Successivamente inizia l'implementazione, che è la fase che dura di più in cui ho diviso tutte le varie attività per poter realizzare il vero e proprio sito web. Anche la fine di questa fase è segnata da una milestone. Più sotto è possibile notare tutti i vari test che vengono svolti sull'applicazione e la milestone di fine fase. Infine, come ultima attività possiamo vedere la documentazione, che dura tutto il progetto e contiene i diari giornalieri e le 4 fasi principali. All'ultima riga è presente la milestone che indica la fine del progetto. Di seguito si può notare il Gantt che ho appena descritto:

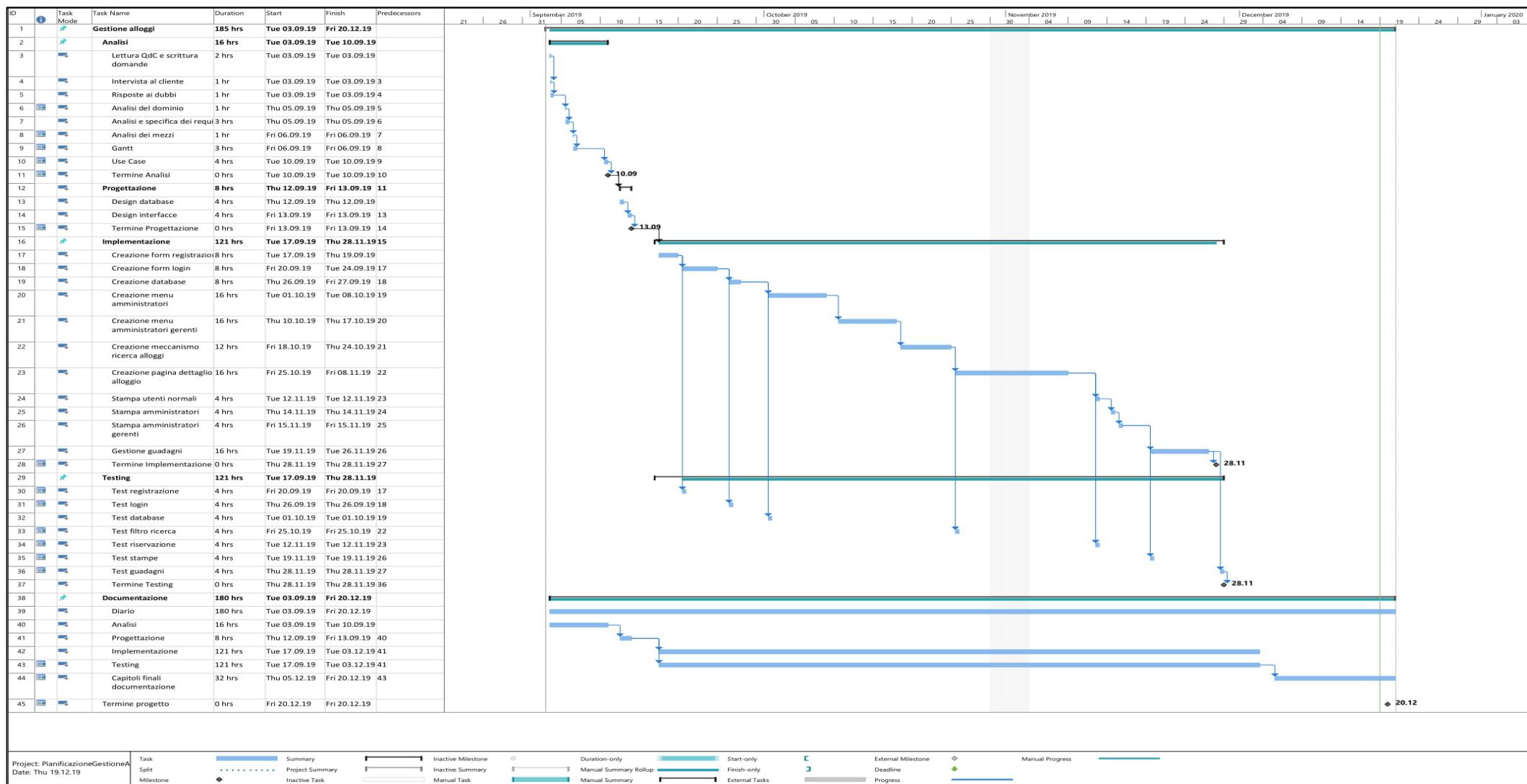


Diagramma di Gantt rappresentante la pianificazione del progetto

2.4.1 Pianificazione intermedia

All'incirca un mese prima della consegna del progetto mi sono reso conto di essere completamente sfasato con i tempi della pianificazione, di conseguenza sotto consiglio del mio responsabile, ho realizzato una Gantt intermedio. Questo Gantt mi ha aiutato un attimo a fare il punto della situazione e a pensare quali elementi dell'implementazione fossero più importanti rispetto ad altri. Questa pianificazione intermedia mi ha permesso di capire che ero troppo tirato coi tempi, di conseguenza avrei dovuto tralasciare la parte della gestione delle fatturazioni.

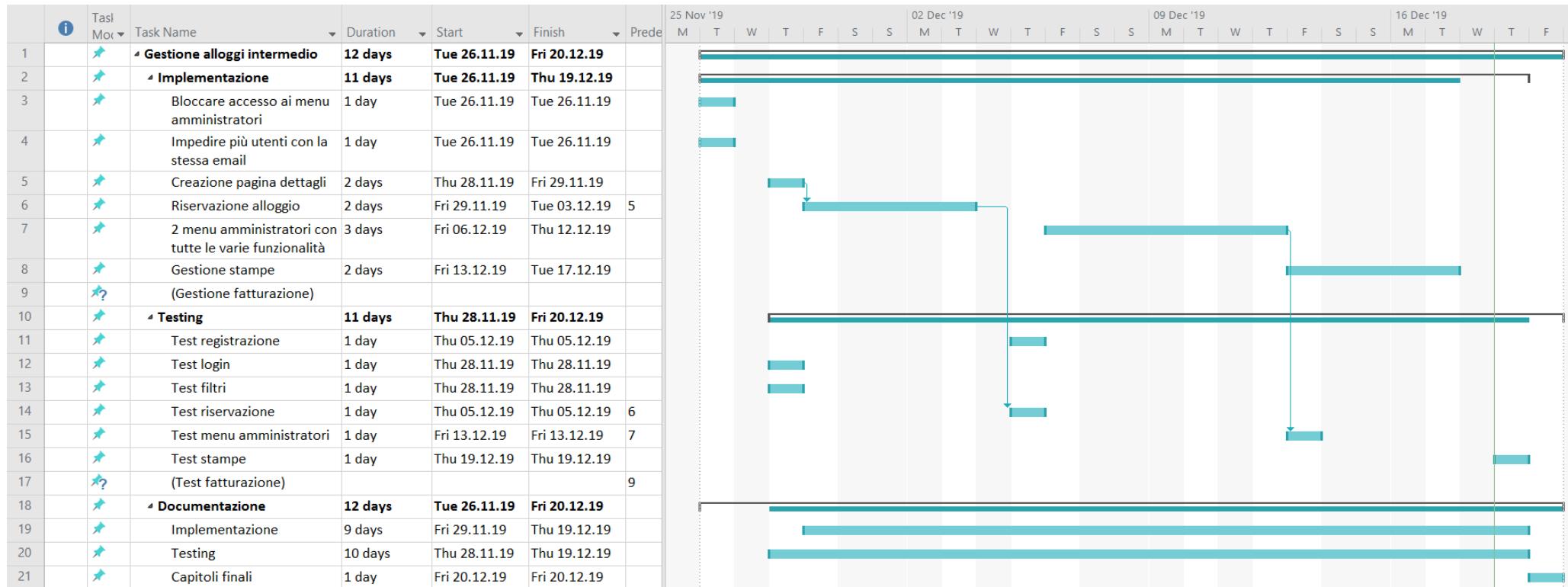
Gestione disponibilità alloggi


Diagramma di Gantt rappresentante la pianificazione intermedia del progetto

2.5 Analisi dei mezzi

2.5.1 Software

I software utilizzati per la realizzazione di questo progetto sono:

- Microsoft Word 2016
- Microsoft Power Point 2016
- Microsoft Project Professional 2016
- Google Chrome 71.0.3578.98
- GitHubDesktop 2.1.3
- Visual Studio Code 1.34.0
- MySQL Workbench 8.0.17
- FileZilla 3.8.0
- phpMyAdmin 4.9.2 (<https://h2-phpmyadmin.infomaniak.com/>)
- Draw.io 12.4.2 (<https://draw.io>)
- MockFlow 2019 (<http://wireframepro.mockflow.com/>)

2.5.2 Hardware

- Computer portatile:
 - Modello: Acer Aspire E 15
 - Processore: Intel Core i7 7500u
 - RAM: 16GB LPDDR4 2133 MHz
 - GPU: NVIDIA 940MX
 - SSD: 256 GB

3 Progettazione

3.1 Design dei dati e database

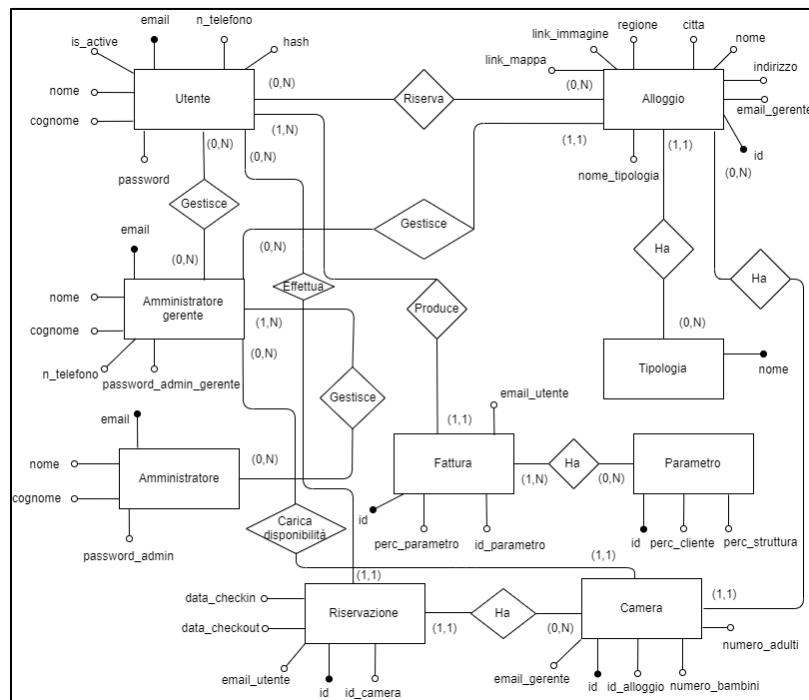
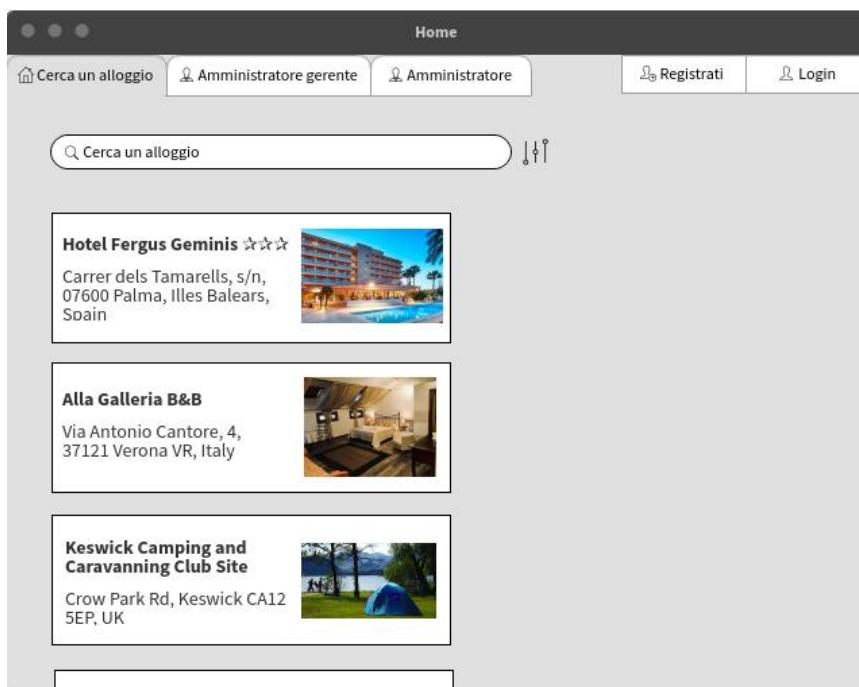


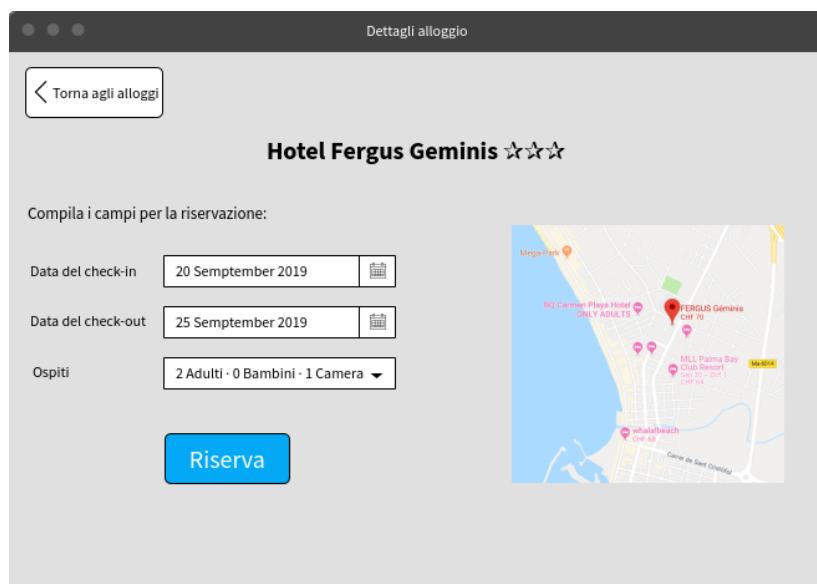
Diagramma E-R rappresentante la progettazione del Database

Il mio schema E-R del database è formato da 9 tabelle. La tabella "Fattura" e la tabella "Parametro" non sono state utilizzate per mancanza di tempo. Sulla sinistra è possibile tre tabelle: "Utente", "Amministratore gerente" e "Amministratore". Queste tabelle rappresentano essenzialmente i 3 tipi di utenti disponibili nel mio progetto. Tutti quanti sono identificati dalla propria email e dispongono di altre informazioni tipo il nome, il cognome e la password. Le ultime 4 tabelle rimaste sono legate tra di loro. "Alloggio", ha sicuramente una "Tipologia", viene identificato da un id e come attributi ha delle informazioni che sono quelle che verranno mostrate all'utente per la scelta dell'alloggio, come un'immagine, una regione, una città, ecc. C'è poi la tabella "Camera", che ha sicuramente un alloggio, viene identificata da un id e ha degli attributi tipo il numero di bambini o il numero di adulti. L'ultima tabella, "Riservazione", si lega ad una camera, anche questa è identificata da un id e ha degli attributi tipo la data del check-in e quella del check-out.

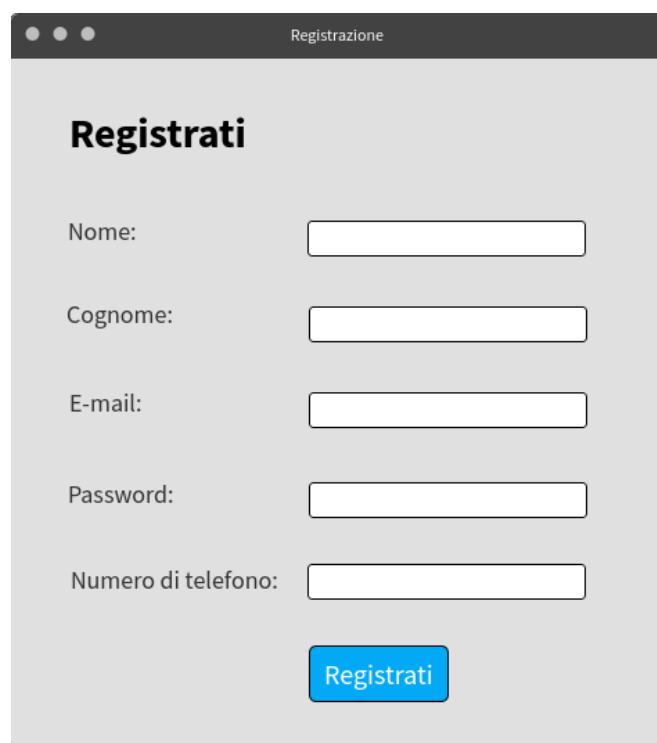
3.2 Design delle interfacce



Mockup della pagina principale



Mockup della pagina di riservazione



Registrazione

Registrati

Nome:

Cognome:

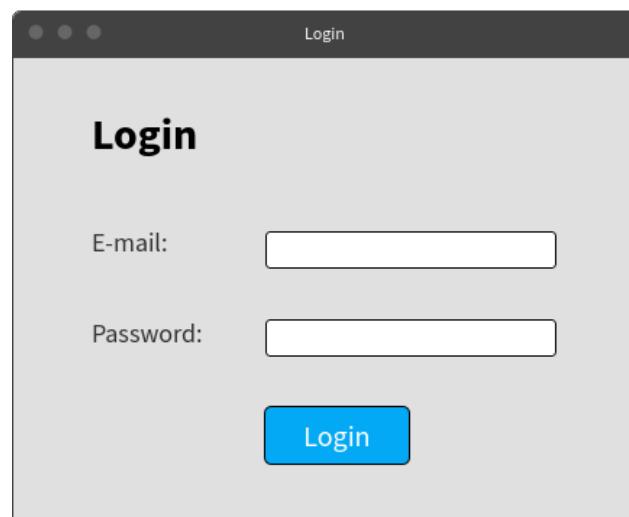
E-mail:

Password:

Numero di telefono:

Registrati

Mockup della pagina di registrazione



Login

E-mail:

Password:

Login

Mockup della pagina di login



Mockup della pagina riservata agli amministratori



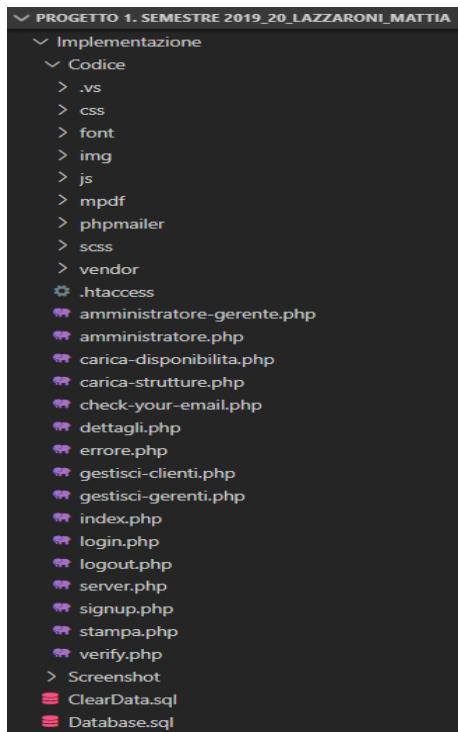
Mockup della pagina riservata agli amministratori gerenti

Quelli appena visti sono dei mockup realizzati in maniera molto basilare e non definitiva, solo per farsi un'idea iniziale di come dovranno poi venir realizzate tutte le pagine del sito. Per poter realizzare queste illustrazioni ho utilizzato MockFlow, che è un tool che permette appunto di pianificare delle UI (User Interfaces).

4 Implementazione

4.1 Struttura

Per cominciare fornirò una breve spiegazione della struttura, che si presenta nel seguente modo:



Struttura della parte di implementazione

- **Implementazione** → è la cartella che contiene il codice, i vari screenshot dell'implementazione e i file .sql per la gestione del database.
 - Codice → è la directory che contiene il template bootstrap 4, che ho scelto in quanto la sua disposizione one-column (ovvero ogni riga della homepage possiede solamente una colonna, quindi ogni riga contiene solo un alloggio) favoriva quello che avevo pensato di realizzare.
 - css, scss, img, js... → sono tutte cartelle che contengono elementi aggiuntivi per il sito, come lo stile, delle immagini o degli script.
 - vendor → è la cartella aggiunta tramite composer che contiene le librerie di terze parti. Nel mio caso ho aggiunto PHPMailer e mPDF.
 - mpdf → è la cartella che contiene la libreria di php chiamata "mpdf". Questa libreria è basata su fpdf, con l'aggiunta della possibilità di creare file PDF da codice HTML codificato in UTF-8.
 - phpmailer → è la cartella che contiene la libreria PHPMailer. Questa libreria, come dice il nome, semplicemente permette di mandare delle email tramite PHP.
 - .htaccess → è il file in cui ho aggiunto la regola che serve a reindirizzare l'utente ad una pagina di errore nel caso in cui egli provi ad accedere ad un file che non esiste (es: "samtinfo.ch/i16lazmat/gestionealloggi2019/fileinesistente.php").
 - amministratore-gerente.php → file per la pagina di gestione per l'amministratore gerente, accessibile solo se viene effettuato il login come amministratore gerente.
 - amministratore.php → file per la pagina di gestione per l'amministratore, accessibile solo se viene effettuato il login come amministratore.
 - carica-disponibilita.php → file accessibile solo agli amministratori gerenti in cui è possibile aggiungere delle camere per gli alloggi.

- carica-strutture.php → file accessibile solo agli amministratori in cui è possibile aggiungere degli alloggi.
 - check-your-email.php → file che viene mostrato all'utente nel momento in cui effettua una registrazione. Indica all'utente di controllare l'email.
 - dettagli.php → file per la pagina di dettagli di un alloggio, che viene mostrata all'utente quando egli sceglie un alloggio dalla lista. In questa pagina è possibile svolgere la riservazione.
 - errore.php → pagina generale che viene mostrata quando l'utente inserisce un file non esistente.
 - gestisci-clienti.php → file accessibile solo agli amministratori gerenti in cui il gerente può gestire le riservazioni dei clienti che hanno riservato un alloggio di sua proprietà.
 - gestisci-gerenti → file accessibile solo agli amministratori in cui è possibile gestire tutti gli amministratori gerenti presenti nel database.
 - index.php → pagina principale del sito, contenente la lista di tutti gli alloggi, la barra di ricerca con i filtri, i link alle pagine di login e di registrazione e altre piccole informazioni.
 - login.php → pagina contenente il form di login in cui appunto loggarsi.
 - logout.php → file per eseguire il logout dopo aver realizzato un login.
 - server.php → file che esegue la connessione al database.
 - signup.php → pagina contenente il form di registrazione in cui appunto registrarsi.
 - stampa.php → pagina accessibile solo agli amministratori gerenti che permette di stampare le riservazioni effettuate in un certo periodo e/o per un certo utente.
 - verify.php → pagina in cui viene portato l'utente quando clicca il link ricevuto per email. È la pagina che si occupa di verificare che l'utente abbia confermato la sua email.
- Screenshot → cartella contenente tutte le immagini realizzate durante l'implementazione, che possono essere sia pezzi di codice, sia risultati grafici che il codice produce.
 - Database.sql → file che contiene le query che generano il database con già dei dati dentro.
 - ClearData.sql → file che contiene le query che puliscono il database svuotando tutte le tabelle e reinserendo i dati.

4.2 PHPMailer

Per installare questa libreria, mi è bastato digitare il seguente comando:

```
C:\Users\matti\OneDrive\Desktop\Progetti individuali\Progetto 1. semestre  
2019_20_Lazzaroni_Mattia\Implementazione\Codice\phpmailer> composer require  
phpmailer/phpmailer
```

A quel punto, composer si occuperà di creare la cartella “vendor” e il file “vendor/autoload.php”. Per poter utilizzare poi la libreria, ho aggiunto le seguenti righe nei file in cui dovevo mandare delle email:

```
//Importo le classi PHPMailer nel namespace globale.  
use PHPMailer\PHPMailer\PHPMailer;  
use PHPMailer\PHPMailer\SMTP;  
use PHPMailer\PHPMailer\Exception;  
  
//Includo il file autoload di composer per PHPMailer.  
require './phpmailer/vendor/autoload.php';
```

Dopo aver importato le varie classi e incluso l’autoload, sono passato alla istanziazione dell’oggetto per poter mandare un email.

```
//Istanzia un nuovo oggetto PHPMailer.  
$mail = new PHPMailer(true);
```

```
$mail->isSMTP();  
$mail->SMTPDebug = 2;  
$mail->Host      = 'smtp.live.com';  
$mail->SMTPAuth   = true;  
$mail->Username   = 'gestionealloggi@hotmail.com';  
$mail->Password   = 'LaTuaPassword';  
$mail->SMTPSecure = 'tsl';  
$mail->Port       = 587;  
$mail->setFrom('gestionealloggi@hotmail.com', 'Gestione alloggi');  
$mail->addAddress('destinazione@gmail.com', 'Nome Destinazione');  
$mail->addReplyTo('gestionealloggi@hotmail.com', 'Gestione alloggi');  
$mail->CharSet = "UTF-8";  
$mail->Subject = 'Titolo dell'email di prova';  
//Body di prova.  
$body = "Corpo dell'email di prova";  
//Assegno il body all'email.  
$mail->Body = $body;  
$mail->isHTML(true);  
//Mando l'email.  
if (!$mail->send()) {  
    echo 'Mailer Error: ' . $mail->ErrorInfo;  
}
```

4.3 mPDF

Per l'installazione di mPDF, come per PHPMailer, mi è bastato lanciare un comando.

```
C:\Users\matti\OneDrive\Desktop\Progetti individuali\Progetto 1. semestre  
2019_20_Lazzaroni_Mattia\Implementazione\Codice\mpdf> composer require mpdf/mpdf
```

Fatto ciò basterà includere l'autoload e il gioco è fatto.

```
//Includo il file autoload di composer per mPDF.  
require './mpdf/vendor/autoload.php';
```

A questo punto si potrà istanziare un oggetto mPDF e passare alla creazione del PDF.

```
//Istanzia un nuovo oggetto mPDF.  
$mpdf = new \Mpdf\Mpdf();  
$mpdf->WriteHTML('<h1>Hello world!</h1>');  
//Salvo l'output in una variabile.  
$pdf_prova = $mpdf->Output('doc.pdf', 'S');  
//Rendo il pdf un allegato dell'email.  
$mail->addStringAttachment($pdf_prova, 'doc.pdf');  
$mail->isHTML(true);
```

Nell'esempio appena riportato qua sopra vado a creare un file chiamato "doc.pdf" che allego poi in un email.

4.4 “ob_start()” e “ob_end_clean()”

Molto spesso all'interno del mio progetto mi è capitato di usare queste funzioni.

```
//Metodo che impedisce a tutto quello che segue di essere stampato a schermo.  
ob_start();  
//Includo il file che esegue il login.  
include('login.php');  
//Metodo che torna a permette di stampare tutto quello che segue.  
ob_end_clean();
```

In questo esempio vado ad includere il file “login.php” nel mio “index.php”. Se non avessi usato le due funzioni, tutta la parte html del login sarebbe uscita nel mio index. Grazie alle due funzioni, l'output del login viene catturato e gettato via, invece di venir mandato al browser.

4.5 Database.sql

```

create database if not exists efof_i16lazmat;

use efof_i16lazmat;

create table utente(
    email varchar(30) primary key,
    nome varchar(30) not null,
    cognome varchar(30) not null,
    password_utente varchar(255) not null,
    n_telefono varchar(20) not null,
    hash varchar(255) not null,
    is_active int(1) default 0
);

create table amministratore_gerente(
    email varchar(30) primary key,
    nome varchar(30) not null unique,
    cognome varchar(30) not null,
    password_admin_gerente varchar(255) not null,
    n_telefono varchar(20) not null
);

create table amministratore(
    email varchar(30) primary key,
    password_admin varchar(255) not null,
    nome varchar(30) not null,
    cognome varchar(30) not null
);

```

Prima parte di "Database.sql"

```

create table parametro(
    id int primary key,
    perc_cliente decimal(5,2) not null,
    check (perc_cliente >= 0.00 and perc_cliente <= 100.00),
    perc_struttura decimal(5,2) not null,
    check (perc_struttura >= 0.00 and perc_struttura <= 100.00)
);
create table fattura(
    id int primary key,
    perc_parametro decimal(5,2) not null,
    check (perc_parametro >= 0.00 and perc_parametro <= 100.00),
    id_parametro int not null,
    email_utente varchar(30) not null,
    foreign key(id_parametro) references parametro(id),
    foreign key(email_utente) references utente(email)
);

create table tipologia(
    nome varchar(30) primary key
);

```

Seconda parte di "Database.sql"

```

create table alloggio(
    id int primary key auto_increment,
    nome varchar(50) not null,
    indirizzo varchar(255) not null,
    link_immagine text not null,
    link_mappa text not null,
    regione varchar(50) not null,
    citta varchar(50) not null,
    email_gerente varchar(50) not null,
    nome_tipologia varchar(50) not null,
    foreign key(email_gerente) references amministratore_gerente(email),
    foreign key(nome_tipologia) references tipologia(nome)
);

create table camera(
    id int primary key auto_increment,
    numero_bambini int not null,
    numero_adulti int not null,
    id_alloggio int not null,
    email_gerente varchar(50) not null,
    foreign key(id_alloggio) references alloggio(id),
    foreign key(email_gerente) references alloggio(email_gerente)
);

create table riservazione(
    id int primary key auto_increment,
    data_checkin date not null,
    data_checkout date not null,
    id_camera int not null,
    email_utente varchar(50) not null,
    foreign key(id_camera) references camera(id),
    foreign key(email_utente) references utente(email)
);

```

Terza parte di "Database.sql"

Queste sono le parti più importanti del mio file “Database.php” in cui vado a creare il database, le tabelle e tutti i vari attributi. La tabella “fattura” deve avere per forza l’id di un parametro e l’email di un utente. La tabella “alloggio” deve avere per forza l’email del gerente che lo gestisce e il nome della sua tipologia. Anche la tabella “Camera” ha dei campi obbligatori, ovvero l’ID dell’alloggio in cui si trova e l’email del gerente che la gestisce. Infine, anche la tabella riservazione deve avere obbligatoriamente l’email dell’utente che ha effettuato la riservazione e l’ID della camera che è stata riservata.

Nel file ci sono delle query (che negli screenshot qua sopra non è possibile notare) che vanno semplicemente a popolare le tabelle con dati inventati da me.

4.6 ClearData.sql

```
use efof_i16lazmat;

delete from riservazione;
delete from camera;
delete from alloggio;
delete from tipologia;
delete from amministratore_gerente;
delete from amministratore;
```

Parte del file in cui vado a svuotare tutte le tabelle.

Questo file invece ha lo scopo di svuotare tutte le tabelle e reinserire i dati di default. Questo durante l’implementazione mi ha permesso di resettare le mie tabelle quando si trovavano piene di nuove righe a causa dei test.

4.7 .htaccess

```
RewriteEngine on
ErrorDocument 404 http://samtinfo.ch/gestionealloggi2019/errore.php
```

Questo file contiene una regola che si occupa semplicemente di reindirizzare l’utente alla pagina d’errore nel caso che egli cerchi di accedere ad un file che non esiste, causando un errore 404.

4.8 errore.php

Questa è la pagina di cui abbiamo appena parlato nel punto precedente. L’utente viene portato in questa pagina quanto inserisce un URL non valido, ad esempio [“www.samtinfo.ch/gestionealloggi2019/blabla.php”](http://www.samtinfo.ch/gestionealloggi2019/blabla.php). Questo è il semplice codice che mostra l’errore all’utente e gli propone il link per la homepage:

```
<article class="card-body mx-auto" style="max-width: 450px;">
  <h4 class="card-title mt-3 text-center">Errore!</h4>
  <p class="text-
  center">Questa pagina non esiste oppure non hai il permesso per accederci.</p>
  <form action="index.php">
    <div class="form-group">
```

```
<button type="submit" class="btn btn-primary btn-block"><a href="index.php"></a> Torna alla home </button>
</div> <!-- form-group// -->
</form>
</article>
```

4.9 server.php

Server.php è un file in cui l'utente non verrà mai portato. Infatti, lo scopo di questa pagina non è quello di mostrare qualcosa al cliente, bensì quello di effettuare la connessione con il database.

```
$servername = "efof.myd.infomaniak.com";
$username = "efof_i16lazmat";
$password = "Alloggi_Admin2019";
$database = "efof_i16lazmat";
```

Inizialmente vado a salvare in delle variabili le credenziali per accedere al database.

```
try {
    $db = new PDO("mysql:host=$servername;dbname=$database", $username, $password);
    $db->setAttribute(PDO::ATTR_ERRMODE, PDO::ERRMODE_EXCEPTION);
    $db->setAttribute(PDO::ATTR_STRINGIFY_FETCHES, false);
} catch (PDOException $e) {
    echo "Connessione fallita: " . $e->getMessage();
}
```

Poi provo a connettermi al database con le credenziali appena citate. Nel caso di un fallimento, all'utente viene mostrato un messaggio di errore.

4.10 index.php

Questa è la homepage del sito, contiene la lista di tutti gli alloggi, la barra di ricerca con i filtri, i link alle pagine di login e di registrazione, e a dipendenza del tipo di login fatto, una navbar con eventualmente l'accesso ai menu riservati.

Per evitare che gli utenti possano accedere ai menu riservati agli amministratori, vado a fare i seguenti controlli:

```
<?php
//Se si è loggati come amministratore gerente o amministratore mostro il rispettivo bottone per il menu dedicato.
if (isset($_SESSION['amministratore_gerente'])) :
?>
    <li class="nav-item">
        <a class="nav-link waves-effect" href="amministratore-
gerente.php">Amministratore gerente</a>
    </li>
<?php endif; ?>
<?php
if (isset($_SESSION['amministratore'])) :
?>
```

```
<li class="nav-item">
    <a class="nav-link waves-
    effect" href="amministratore.php">Amministratore</a>
</li>
<?php endif; ?>
```

Vado quindi a controllare se è stata impostata la sessione “amministratore” o rispettivamente “amministratore_gerente” e in quel caso mostro il link all’utente. Queste sessioni vengono impostate nel file “login.php” al momento del login, a dipendenza appunto del tipo di utente che si logga.

Per sapere se l’utente è loggato o meno e quindi per sapere se dargli la possibilità di fare il login/registrarsi oppure fare il logout, uso anche qui le sessioni:

```
<?php
//Se non è stato effettuato il login, dò all'utente la possibilità di farlo o di creare
//un account.
if (!isset($_SESSION['loggedin'])) :
?>
<li class="nav-item">
    <a href="signup.php" class="nav-link border border-light rounded waves-effect">
        <i class="fas fa-user-plus"></i>Registrati
    </a>
</li>&nbsp;
<li class="nav-item">
    <a href="login.php" class="nav-link border border-light rounded waves-effect">
        <i class="fas fa-sign-in-alt"></i>Login
    </a>
</li>&nbsp;
<?php endif;
//Stampo il bottone per il logout se l'utente ha effettuato il login.
if (isset($_SESSION['loggedin'])) :
?>
<li class="nav-item">
    <a href="logout.php" class="nav-link border border-light rounded waves-effect">
        <i class="fas fa-sign-out-alt"></i>Logout
    </a>
</li>
<?php endif; ?>
```

Quello che viene fatto nelle righe di codice qua sopra è controllare se la sessione “loggedin” (che imposta a “true” al momento del login) è settata, in quel caso mostro all’utente il bottone per fare il logout, altrimenti gli mostro i bottoni per fare il login o per registrarsi.

Per mostrare gli alloggi, invece, vado a creare una query (che nel caso vengano scelti dei filtri sarà filtrata) che prende tutti gli alloggi dal database. Vado poi ad eseguirla in loop e ogni volta stampo le informazioni dell’alloggio alla posizione corrente:

```
...
$stmt = $db->prepare($accommodation_query);
//Esegue la query.
```

```
$stmt->execute();
$row = $stmt->fetch();
//Salvo alcune variabili utili.
${"nome" . $i} = $row["nome"];
${"indirizzo" . $i} = $row["indirizzo"];
${"link_immagine" . $i} = $row["link_immagine"];
${"regione" . $i} = $row["regione"];
${"città" . $i} = $row["città"];
${"nome_tipologia" . $i} = $row["nome_tipologia"];
${"link_mappa" . $i} = $row["link_mappa"];
${"email_gerente" . $i} = $row["email_gerente"];
//Stampo un alloggio.
echo '<form id="dettagli" . $i . '" action="dettagli.php" method="post">
    <div class="row wow fadeIn">
        <div class="col-lg-5 col-xl-4 mb-4">
            <div class="view overlay rounded z-depth-1">
                <img src=' . ${"link_immagine" . $i} . '" class="img-fluid" alt="">
            </div>
        </div>
        ...
    </div>
```

4.11 signup.php

È la pagina contenente il form di registrazione in cui l'utente può appunto registrarsi. Ovviamente tutti i valori inseriti dall'utente sono controllati e l'utente riceve dei messaggi di errore se i campi inseriti non dovessero essere validi.

```
if (empty($password_1)) {
    array_push($errors, "La password è richiesta");
} else {
    if (strlen(trim($password_1)) < 6) {
        array_push($errors, "La password è troppo corta");
    }
    $password_1 = trim($password_1);
}
if ($password_1 != $password_2) {
    array_push($errors, "Le due password non coincidono");
}
```

Questa è solo una parte dei controlli che vado a fare e si occupa di validare le due password obbligatorie.

```
$hash = md5(time());
//Definisco la query che inserisce l'utente.
$query = "INSERT INTO utente (email, nome, cognome, password_utente, n_telefono, hash)
VALUES(:email, '$name', '$surname', :password, '$full_number', '$hash')";
```

In questo pezzo di codice vado a creare una hash univoca in base all'orario corrente espresso in secondi.

Creo poi una query che serve ad inserire l'utente all'interno del database. All'utente, oltre ai campi oramai validati che ha inserito lui stesso nel form, ci aggiungo questa hash univoca che servirà poi in seguito al momento dell'attivazione dell'account.

```
$stmt->bindParam(":password", $param_password, PDO::PARAM_STR);
$param_password = password_hash($password_1, PASSWORD_DEFAULT);
```

Per quanto riguarda la password, visto che ovviamente non posso salvarla in chiaro nel database, vado a creare una hash della password dell'utente tramite il metodo “password_hash()” e la inserisco nel database.

**\$body = "Grazie per esserti registrato!
**

**Il tuo account è stato creato, puoi accedere con le seguenti credenziali dopo aver attivato l'account.
**

```
----- <br>
Email: $email <br>
Password: $pointed_password <br>
----- <br>
```

**Fai clic su questo link per attivare il tuo account:
**

http://samtinfo.ch/gestionealloggi2019/verify.php?email=\$email&hash=\$hash;

Questo, invece, è il corpo dell'email che viene mandata all'utente al momento della registrazione, in cui gli si dice di cliccare un link che lo porta alla pagina per l'attivazione dell'account. All'interno di questo link viene aggiunta l'email inserita dall'utente nel form e l'hash univoca creata in precedenza. In questo modo, si può essere certi che il link sarà univoco, nessuno oltre a lui potrà attivare l'account e dunque non ci sarà nessun modo di attivare l'account se l'email inserita è inesistente o fasulla.

4.12 check-your-email.php

In questa pagina viene semplicemente mostrata un'immagine all'utente quando completa la registrazione, spiegandogli di andare a vedere le sue email per attivare l'account.

```
<article class="card-body mx-auto" style="max-width: 450px;">
  <h2 class="card-title mt-3 text-center">Il tuo account è stato creato!</h2>
  <p class="text-center">Sei pregato di verificarlo cliccando il link di attivazione che ti è stato mandato per email.</p>
  
</article>
```



Immagine che fa capire all'utente che sta facendo le cose in modo corretto.

4.13 verify.php

È la pagina in cui viene indirizzato l'utente quando clicca il link ricevuto per email.

```
//Controllo di ricevere un email e una hash tramite url.  
if (isset($_GET['email']) && !empty($_GET['email']) and isset($_GET['hash']) && !empty($_GET['hash'])) {  
    $email = $_GET['email']; // Imposto l'email  
    $hash = $_GET['hash']; // Imposto l'hash  
    //Preparo la query  
    $search = $db->  
        >prepare("SELECT email, hash, is_active FROM utente WHERE email=:email AND hash=:hash A  
ND is_active='0'");  
    $search->bindParam(":email", $email, PDO::PARAM_STR);  
    $search->bindParam(":hash", $hash, PDO::PARAM_STR);  
    $search->execute();  
    $match = $search->rowCount();  
    ...
```

In questo pezzo di codice, controllo innanzitutto che l'accesso alla pagina sia stato effettuato correttamente, ovvero tramite link inviato per email e quindi con una richiesta di tipo "GET" (parametri richiesti vengono aggiunti al URL). In quel caso, vado a fare una select nel database, cercando almeno un utente che abbia l'email e la hash che sono passate come parametro della richiesta. Vediamo ora cosa succede nel momento in cui viene trovato almeno un utente dalla select appena eseguita:

```
//Nel caso che la query produce un risultato, modifico l'active dell'utente nel db e stampo un messaggio di riuscita all'utente.  
if ($match > 0) {  
    $update = $db->  
        >prepare("UPDATE utente SET is_active='1' WHERE email='".$email."' AND hash='".$hash.'  
        "' AND is_active='0'");  
    $update->execute();  
}
```

Se viene trovato un utente, vado essenzialmente a modificarne l'attributo "is_active" da 0 a 1. Da questo momento l'account è attivo e pronto a fare il login.

Se invece la query non ritorna nessun risultato, indico all'utente che probabilmente l'account è già stato attivato e che quindi l'URL non è valido.

```
else {
    echo '<div class="form-group text-center card-body mx-auto" style="max-width:450px;">
        <br><div>L'URL non è valido o hai già attivato il tuo account.</div><br>
        <a href="index.php" class="btn btn-primary btn-block"> Torna alla home </a>
    </div>';
}
```

Da notare “'”, che serve per stampare l'apostrofo.

4.14 login.php

Come dice anche il nome, questa è la pagina in cui l'utente può loggarsi dopo aver creato l'account e dove anche gli amministratori e gli amministratori gerenti possono accedere. Ovviamente anche in questa pagina vengono effettuati i controlli sulla validità dei campi inseriti dall'utente.

```
//Query per controllare che un utente esista con quella email.
$query_utente = "SELECT * FROM utente WHERE email = :email";
//Query per controllare che un amministratore esista con quella email.
$query_amministratore = "SELECT * FROM amministratore WHERE email = :email";
//Query per controllare che un amministratore gerente esista con quella email.
$query_amministratore_gerente = "SELECT * FROM amministratore_gerente WHERE email = :email";

//Preparo le query.
$stmt1 = $db->prepare($query_utente);
$stmt2 = $db->prepare($query_amministratore);
$stmt3 = $db->prepare($query_amministratore_gerente);

//Associo i parametri alla variabile contenente l'email.
$stmt1->bindParam(":email", $param_email, PDO::PARAM_STR);
$stmt2->bindParam(":email", $param_email, PDO::PARAM_STR);
$stmt3->bindParam(":email", $param_email, PDO::PARAM_STR);
$param_email = trim($email);

//Eseguo le query.
$stmt1->execute();
$stmt2->execute();
$stmt3->execute();
```

Quello che faccio dopo aver verificato gli input è creare 3 query in cui vado a cercare nel database l'email inserita nel form. Dopo aver eseguito le 3 query semplicemente tramite if/elseif, vado a controllare quale degli “stmt” ha ritornato una riga. Nel caso che sia lo “\$stmt1” vuol dire che quell'email è legata ad un utente, nel caso sia lo “\$stmt2” è un amministratore e nel caso sia lo “\$stmt3” si tratta di un amministratore gerente. Questo è quello che vado a fare ad esempio nel caso che l'email sia legata ad un utente.

```
if ($stmt1->rowCount() == 1) {
    $row = $stmt1->fetch(PDO::FETCH_ASSOC);
    $email = $row["email"];
```

```
$hashed_password = $row["password_utente"];
$active = $row["is_active"];

if ($active == 1) {
    //Se la password inserita dall'utente coincide con la password hashata presente
    nel db è legata a quell'email.
    if (password_verify($password, $hashed_password)) {
        //Imposto delle variabili session e sposto l'utente alla homepage.
        $_SESSION["name"] = $row["nome"];
        $_SESSION["email"] = $row["email"];
        $_SESSION["utente"] = true;
        $_SESSION["loggedin"] = true;
        header('location: index.php');
    }

    //Altrimenti indico all'utente che la password inserita è errata.
    else {
        array_push($errors, "La password che hai inserito non è corretta");
    }
    //Altrimenti indico all'utente che non ha ancora verificato l'account.
} else {
    array_push($errors, "Non hai verificato l'account, controlla la tua email!");
}
}
```

In queste poche righe vado a prendere dalla tabella “utente” le informazioni che mi servono dell’utente con quell’email. Una di quelle, è il valore di “is_active”, che se vale 1 permette di passare al controllo della password. Se anche la password dovesse essere giusta vengono salvate delle sessioni utili per sapere che tipo di utente ha fatto il login, per sapere il suo nome e la sua email. Dopo aver salvato queste informazioni reindirizzo il cliente alla home page, in cui vedrà un messaggio di benvenuto affiancato dal tipo di utente con cui ha effettuato l’accesso.

4.15 logout.php

Questo file si occupa semplicemente di eliminare tutti le sessioni presenti. Quando il cliente effettua il login, andando nella homepage vedrà un bottone apposta per il logout. Dopo che le sessioni sono state svuotate, l’utente viene reinidirizzato alla pagina principale.

```
<?php
//Inizializzo la sessione.
session_start();

//Deseleziono tutte le variabili di sessione.
$_SESSION = array();

//Distruggo la sessione.
session_destroy();
```

```
//Reindirizzo alla homepage.  
header("location: index.php");  
exit;  
?>
```

4.16 amministratore.php e amministratore_gerente.php

Questi due file, abbastanza simili, possono essere accessibili solo se si è eseguito l'accesso con un amministratore o rispettivamente un amministratore gerente.

Il controllo (in questo caso in amministratore.php) viene fatto in questo modo:

```
if (isset($_SESSION['amministratore'])) :  
?>  
<!--Main Navigation-->  
<header>  
...
```

Questo è quello che viene mostrato invece se si prova ad accedere alla pagina senza avere i permessi viene gestito da questo codice:

```
<?php else : ?>  
<!-- Contenuto visivo della pagina -->  
<article class="card-body mx-auto" style="max-width: 450px;">  
    <h4 class="card-title mt-3 text-center">Errore!</h4>  
    <p class="text-center">Non hai il permesso di accedere a questa pagina.</p>  
    <form action="index.php">  
        <div class="form-group">  
            <button type="submit" class="btn btn-primary btn-block"><a href="index.php"></a> Torna alla home </button>  
        </div> <!-- form-group// -->  
    </form>  
</article>  
<?php endif; ?>
```

Quindi viene mostrato un messaggio d'errore e l'utente viene esortato ad abbandonare la pagina.

Questo è invece il contenuto che viene mostrato al cliente se egli ha eseguito il login con permessi elevati (in questo caso come amministratore gerente):

```
<article class="card-body mx-auto" style="max-width: 450px; margin-top: 75px;">  
    <h4 class="card-title mt-3 text-center">Benvenuto!</h4>  
    <p class="text-  
center">In questa pagina puoi caricare le disponibilit&agrave; delle strutture, ges  
tire i clienti che hanno riservato gli alloggi e stampare le riservazioni per un d  
eterminato periodo e/o persona.</p>  
    <form method="post" action="amministratore-gerente.php">  
        <div class="form-group">  
            <button id="disponibilita" name="disponibilita" type="submit" class="bt  
n btn-primary btn-block"> Carica disponibilit&agrave; </button>
```

```
</div> <!-- form-group// -->
<div class="form-group">
    <button id="clienti" name="clienti" type="submit" class="btn btn-primary btn-block"> Gestisci clienti </button>
</div> <!-- form-group// -->
<div class="form-group">
    <button id="stampa" name="stampa" type="submit" class="btn btn-primary btn-block"> Stampa riservazioni </button>
</div> <!-- form-group// -->
</form>
</article>
```

Si riesce quindi a capire che quello che l'utente vede è un form contenente dei buttoni, ognuno dei quali lo reindirizza verso una pagina di gestione. L'amministratore gerente, come si vede da quest'immagine ha la possibilità di caricare le disponibilità, gestire i clienti e stampare le riservazioni. L'amministratore, invece, può caricare le strutture e gestire i gerenti.

A dipendenza di quale bottone viene cliccato, eseguo il redirect alla pagina richiesta.

```
<?php
if (isset($_POST['disponibilita'])) {
    header('Location: carica-disponibilita.php');
} else if (isset($_POST['clienti'])) {
    header('Location: gestisci-clienti.php');
} else if (isset($_POST['stampa'])) {
    header('Location: stampa.php');
}
?>
```

4.17 carica-disponibilita.php

File accessibile solo agli amministratori gerenti in cui è possibile aggiungere delle camere per gli alloggi. Questa pagina presenta un form in cui il gerente può inserire il numero di bambini, il numero di adulti e il nome dell'alloggio in cui si troverà la camera. Dopo aver controllato la validità degli input del gerente, vado ad aggiungere la camera nel database:

```
//Query che prende l'id più grande della tabella delle camere.
$room_query = "SELECT camera.id
FROM camera
JOIN alloggio
ON alloggio.nome = '" . $nome . "'"
WHERE alloggio.email_gerente = '" . $email . "'"
ORDER BY id DESC LIMIT 1";
//Query che prende l'id dell'alloggio che il nome uguale a quello inserito dal gerente.
$id_query = "SELECT alloggio.id
FROM alloggio
WHERE alloggio.nome = '" . $nome . "'"
AND alloggio.email_gerente = '" . $email . "'"
LIMIT 1";
```

```
else if ($stmt->rowCount() > 0) {
    $row = $stmt->fetch(PDO::FETCH_NUM);
    $row1 = $stmt1->fetch(PDO::FETCH_NUM);
    //Prendo un id disponibile.
    $nextId = $row[0] + 1;
    $idAlloggio = $row1[0];
    $insert_room_query = "INSERT INTO camera values (" . $nextId . ", " . $bambini
. ", " . $adulti . ", " . $idAlloggio . ", '" . $email . "')";
    $stmt = $db->prepare($insert_room_query);
    //Eseguo la query.
    $stmt->execute();

    ...
}
```

Questo codice verrà eseguito soltanto nel caso in cui esista l'alloggio inserito nel form dal gerente e nel caso in cui questo alloggio sia gestito proprio da lui.

Quello che fa è andare a prendere il prossimo ID libero nella tabella camera e l'ID dell'alloggio in cui verrà inserita la camera. Una volta che sono stati presi questi due valori, la query per l'inserimento è semplicissima

4.18 gestisci-clienti.php

Anche questa è una pagina dell'amministratore gerente, in cui può eliminare o modificare delle riservazioni effettuate su camere di sua proprietà

Una volta entrati in questa pagina, si presenta una tabella che stampa le riservazioni fatte sulle camere del gerente loggato.

```
echo "<br><form id='formTabella' action='gestisci-
clienti.php' method='post'><div class='table-responsive'>
<table class='text-center table table-striped'>
<tr style='background-color:#D3D3D3;'>
    <th>ID Riservazione</th>
    <th>Data check-in</th>
    <th>Data check-out</th>
    <th>Numero bambini</th>
    <th>Numero adulti</th>
    <th>Email</th>
    <th>Nome</th>
    <th>Cognome</th>
    <th>Numero di telefono</th>
    <th>Modifica</th>
    <th>Elimina</th>
</tr>
<tr>;
    //Per ogni riga ritornata dalla query, aggiungo una riga alla tabella.
    for ($i = 1; $i < $stmt->rowCount() + 1; $i++) {
        $row = $stmt->fetch(PDO::FETCH_NUM);
        echo "<tr>
            <td><input type='hidden' name='id' value='" . $i . "'>
            " . $row[5] .
            "</td>
```

```
<td><input type='hidden' name='checkin' value='".$i . "'>
    . date("d-m-Y", strtotime($row[6])) .
    "</td>
<td><input type='hidden' name='checkout' value='".$i . "'>
    . date("d-m-Y", strtotime($row[7])) .
    "</td>
<td><input type='hidden' name='bambini' value='".$i . "'>
    . $row[8] .
    "</td>
    ...
    ...
```

Tutti i vari elementi che va a stampare sono presi da una query, che come già detto, prende tutte le riservazioni fatte nelle camere gestite dal gerente che ha fatto il login.

Le ultime due colonne danno la possibilità di modificare o eliminare una tabella. Prima mi soffermerò sull'eliminazione.

```
//Query che prende le info della riservazione in base alla riga specificata da rownumber.
$email_query = "SELECT * FROM
(SELECT row_number() OVER (ORDER BY id ASC) AS rownumber, id
FROM riservazione)
as get_id
where rownumber = " . $id_da_controllare . "";

//Preparo la query.
$stmt = $db->prepare($email_query);

//Eseguo la query.
if ($stmt->execute()) {

    $row = $stmt->fetch(PDO::FETCH_NUM);
    $riservazione_da_eliminare = $row[1];

    //Query che disattiva i controlli.
    $disable_checks = "SET FOREIGN_KEY_CHECKS=0";
    //Query che elimina la riservazione.
    $delete_query = "DELETE FROM riservazione
WHERE id = '" . $riservazione_da_eliminare . "'";
    //Query che riattiva i controlli.
    $enable_checks = "SET FOREIGN_KEY_CHECKS=1";
```

Innanzitutto, vado a creare una query che mi va a prendere tutte le informazioni della riservazione cliccata. Per far questo, utilizza “\$id_da_controllare”, che è una variabile che contiene appunto l’ID dell’alloggio che andrà eliminato. A questo punto semplicemente prendiamo l’id della riservazione da eliminare e andiamo ad eseguire una semplice query che elimina la riservazione con quell’ID. Come è possibile vedere, ci sono altre due query che circondano quella dell’eliminazione. La prima disattiva i controlli sulle foreign key, così che io possa eliminare la riservazione senza problemi, mentre la seconda serve a riattivare i controlli in un secondo momento.

Per quanto riguarda la parte della modifica, invece, una volta che si schiaccia il bottone, viene mostrato un nuovo form:

```
<form id="formClienti" method="post" action="gestisci-clienti.php">
    <h4 class="card-title mt-3 text-center">Gestisci i clienti</h4><br>
    <?php if (isset($_POST['modifica']) || isset($_POST['salva_modifiche'])) : ?>
        <p class="grey-text text-center">Modifica la data di check-
in e/o la data di check-out</p>
        <div class="form-group input-group">
            <div class="input-group-prepend">
                <span class="input-group-text"> <i class="fa fa-calendar-
alt"></i> </span>
            </div>
            <input name="checkin" class="form-control" type="date" max="9999-12-
31" min="<?php echo date('Y-m-d'); ?>" title="Data di check-
in" value="<?php echo $checkin; ?>">
        </div>
        <div class="form-group input-group">
            <div class="input-group-prepend">
                <span class="input-group-text"> <i class="fa fa-calendar-
alt"></i> </span>
            </div>
            <input name="checkout" class="form-control" type="date" max="9999-12-
31" min="<?php echo date('Y-m-d'); ?>" title="Data di check-
out" value="<?php echo $checkout; ?>">
        </div>
        <button id="modifica" type="submit" name="salva_modifiche" class="btn btn-
primary btn-block">Salva modifiche</button>
    <?php endif ?>
</form>
```

Questo form mostra due input di tipo data in cui bisogna inserire le nuove date per le riservazioni. Ovviamente queste date verranno controllate validate. Se le date saranno valide, la riservazione verrà aggiornata nel database per merito del seguente codice:

```
//Query per modificare la riservazione.
$modifica_query = "UPDATE riservazione
SET data_checkin = '" . $checkin . ',
data_checkout = '" . $checkout . '
WHERE id = ".$_SESSION['id_da_modificare']."'";
```

//Preparo ed eseguo la query.

```
$stmt = $db->prepare($modifica_query);
$stmt->execute();
```

Questa è la query che modifica la riservazione. Il modo in cui viene preso l'id della riservazione da modificare dalla tabella è lo stesso medesimo di prima, non andrà dunque a rispiegarlo.

4.19 stampa.php

Questa è l'ultimo file disponibile all'amministratore gerente. La pagina gli si presenta con un form in cui può inserire data di check-in e data di check-out e/o email dell'utente. I campi non sono quindi tutti obbligatori ma

vengono comunque validati. Dopo di questo, all'invio del form vengono mostrate tutte le riservazioni fatte nel range di date inserito prima e/o fatte dall'utente di cui si è inserita l'email.

```
$SESSION['stampa_multipla'] = "La tua ricerca ha restituito le seguenti riservazioni:<br><br><table class='table table-<striped'>
<tr>
    <th>Email utente</th>
    <th>ID Camera</th>
    <th>Adulti</th>
    <th>Bambini</th>
    <th>Check-in</th>
    <th>Check-out</th>
</tr>";
while ($row = $stmt->fetch(PDO::FETCH_NUM)) {
    $SESSION['stampa_multipla'] .= "<tr>
        <td>" . $row[2] . "</td>
        <td>" . $row[3] . "</td>
        <td>" . $row[4] . "</td>
        <td>" . $row[5] . "</td>
        <td>" . date("d-m-Y" , strtotime($row[0])) . "</td>
        <td>" . date("d-m-Y" , strtotime($row[1])) . "</td>
    </tr>";
}
$SESSION['stampa_multipla'] .= "</table>";
```

Questo è il caso in cui vengono trovate più riservazioni (nel caso in cui ne viene trovata una sola, stampo ugualmente la tabella, soltanto che non è necessario un ciclo while). Una volta mostrata questa tabella, l'utente avrà anche a disposizione un bottone “Invia le riservazioni”. Cliccando questo bottone si passerà a PHPMailer e all'invio della tabella per email:

```
//Se l'utente decide di inviare la riservazione.
if (isset($_POST['invia'])) {
    //Istanzia un nuovo oggetto PHPMailer.
    $mail = new PHPMailer(true);
    //Imposto tutte le informazioni per poter inviare l'email.
```

```
$mail->...  
...  
...  
//Se il risultato ritornato conteneva una sola riservazione.  
if (isset($_SESSION['stampa_singola'])) {  
    $body = $_SESSION['stampa_singola'];  
    $mail-  
>Body = $body . "<br>In allegato è possibile trovare un pdf da stampare.";  
  
    $pdf = new \Mpdf\Mpdf();  
    $pdf->WriteHTML($_SESSION['stampa_singola']);  
    $pdf_riservazioni = $pdf->Output('riservazioni.pdf', 'S');  
//Se il risultato ritornato conteneva più riservazioni.  
} else if (isset($_SESSION['stampa_multipla'])) {  
    $body = $_SESSION['stampa_multipla'];  
    $mail-  
>Body = $body . "<br>In allegato è possibile trovare un pdf da stampare.";  
  
    $pdf = new \Mpdf\Mpdf();  
    $pdf->WriteHTML($_SESSION['stampa_multipla']);  
    $pdf_riservazioni = $pdf->Output('riservazioni.pdf', 'S');  
}  
  
//Aggiungo un pdf in allegato all'email.  
$mail->addStringAttachment($pdf_riservazioni, 'riservazioni.pdf');  
$mail->isHTML(true);
```

Ora, sia nel caso che sia stata trovata una sola riservazione, sia che ne siano state trovate di più, vado ad inserire la tabella nel body dell'email. In aggiunta è anche possibile notare mPDF, che semplicemente mi permette di creare un PDF che supporti codice HTML e lo vado ad allegare all'email tramite "\$mail->addStringAttachment(...)". Il soggetto dell'email, anche se in questa foto non si può notare, è ovviamente il gerente che ha effettuato l'accesso.

```
//Mando l'email.  
if (!$mail->send()) {  
    echo 'Mailer Error: ' . $mail->ErrorInfo;  
}
```

Quest'ultima parte di codice è quella che si occupa di mandare l'email e di stampare un errore nel caso che ci sia.

4.20 carica-strutture.php

File accessibile solo agli amministratori in cui è possibile aggiungere degli alloggi.
Questa pagina presenta un form in cui l'amministratore deve inserire tutte le informazioni minime per mostrare un alloggio:

```
$nome = "";
$indirizzo = "";
$link_immagine = "";
$link_mappa = "";
$regione = "";
$citta = "";
$email = "";
$tipologia = "";
```

Dopo aver controllato la validità degli input dell'amministratore, vado ad aggiungere l'alloggio nel database:

```
//Query che seleziona l'id dell'ultimo alloggio.
$alloggio_query = "SELECT alloggio.id
FROM alloggio
JOIN amministratore_gerente
ON amministratore_gerente.email = '" . $email . "'"
ORDER BY id DESC LIMIT 1";

//Preparo la query da eseguire.
$stmt = $db->prepare($alloggio_query);
//Eseguo la query.
$stmt->execute();
//Se eseguendo la query viene trovata una riga, preparo una nuova query.
else if ($stmt->rowCount() > 0) {
    $row = $stmt->fetch(PDO::FETCH_NUM);
    //Aggiungo 1 all'ultimo id trovato, questo sarà l'id della struttura che l'amministratore aggiunge.
    $nextId = $row[0] + 1;
    $insert_alloggio_query = "INSERT INTO alloggio values (" . $nextId . ", '" . $nome . "', '" . $indirizzo . "', '" . $link_immagine . "', '" . $link_mappa . "', '" . $regione . "', '" . $citta . "', '" . $email . "', '" . $tipologia . "')";
    $stmt = $db->prepare($insert_alloggio_query);
    //Eseguo la query.
    $stmt->execute();
    ...
}
```

Quello che vado a fare è prendere l'id dell'ultimo alloggio inserito, così che nella variabile "\$nextId" posso avere l'ID dell'alloggio che inserirò, prendendo l'ID dell'ultimo alloggio e facendoci +1.

Nel caso in cui venga trovato un amministratore gerente con l'email inserita nel form, vado ad aggiungere l'alloggio nel database.

4.21 gestisci-gerenti.php

Anche questa è una pagina dell'amministratore, in cui può aggiungere, eliminare o modificare gli amministratori gerenti.

Una volta entrati in questa pagina, si presenta una tabella che stampa semplicemente tutti gli amministratori gerenti presenti nel database.

Oltre a questa tabella si potrà vedere un form che permette di inserire un gerente. Per iniziare soffermiamoci su questo form.

```
$email = "";
$nome = "";
$cognome = "";
$password_1 = "";
$password_2 = "";
$telefono = "";

...
...
...

if (count($errors) == 0) {
    $password_hashata = password_hash($password_1, PASSWORD_DEFAULT);
    //Query che inserisce l'amministratore gerente se non ci sono errori.
    $insert_query = "INSERT INTO amministratore_gerente VALUES
        ('" . $email . "', '" . $nome . "', '" . $cognome . "', '" . $password_hashata . "'"
        , '" . $telefono . "')";

    $stmt = $db->prepare($insert_query);

    //Eseguo le query.
    if ($stmt->execute()) {
        $_SESSION['message'] = "<div class='text-success text-center'>
            <p style='color:green'>L'amministratore gerente con email '" . $email . "'"
        &egrave; stato aggiunto con successo.</p>
        </div>";

        //Ricarico la pagina così da svuotare il form.
        echo "<meta http-equiv='refresh' content='0'>";
    } else {
        array_push($errors, "Ops! Qualcosa è andato storto. Riprova");
    }
}
```

Se tutti gli input inseriti nel form non ritornano errori, vado semplicemente a creare una hash della password inserita nel form ed eseguo una query che inserisce il gerente nel database.

Tuttavia, questo form non serve soltanto all'inserimento di un gerente, bensì può anche modificare un gerente già esistente. Basterà cliccare "Modifica" dalla tabella del gerente e tutte le informazioni di quest'ultimo verranno trasferite nel form. Questo è quello che ho fatto per riuscire a spostare gli elementi dalla tabella al form al click del bottone per modificare:

```
$email_query = "SELECT * FROM
(SELECT row_number() OVER (ORDER BY email ASC) AS rownumber,
email,
nome,
cognome,
n_telefono
FROM amministratore_gerente)
as get_email
where rownumber = " . $id_da_controllare . "";
```

```
//Preparo la query
$stmt = $db->prepare($email_query);

//Eseguo la query.
if ($stmt->execute()) {
    $row = $stmt->fetch(PDO::FETCH_NUM);
    //Salvo i valori restituite dalla query.
    $email = $_SESSION['email_definitiva'] = $row[1];
    $nome = $_SESSION['nome_definitivo'] = $row[2];
    $cognome = $row[3];
    $password_1 = "";
    $password_2 = "";
    $telefono = $row[4];
```

Id_da_controllare indica l'ID dell'gerente che è stato cliccato e starà per venir modificato. Dunque, prendo dal database quel gerente e nelle variabili ci salvo tutti i vari attributi che si potrebbero voler modificare. Queste variabili sono poi stampate nel form.

```
//Mi salvo tutte le varie informazioni del gerente.
if (empty($email)) {
    $email = $row[0];
}
if (empty($nome)) {
    $nome = $row[1];
}
if (empty($email)) {
    $cognome = $row[2];
}
if (empty($password_1)) {
    $password_1 = $row[3];
} else if (!empty($password_1)) {
    $password_1 = password_hash($password_1, PASSWORD_DEFAULT);
}
if (empty($password_2)) {
    $password_2 = $row[3];
} else if (!empty($password_2)) {
    $password_2 = password_hash($password_2, PASSWORD_DEFAULT);
}
if (empty($telefono)) {
    $telefono = $row[4];
}

//Query che aggiorna il gerente.
$update_query = "UPDATE amministratore_gerente
SET email = '" . $email . "', nome = '" . $nome . "', cognome = '" . $cognome . "', pas
sword_admin_gerente = '" . $password_1 . "', n_telefono = '" . $telefono . "'"
WHERE email = '" . $_SESSION['email_definitiva'] . "'";
```

La modifica funziona nel seguente modo: tutti i campi che nel form non vengono toccati o sono lasciati vuoti, rimangono così come sono già nel database, tutti gli altri che invece vengono modificati, vanno ad aggiornare la colonna di quel gerente nel database. Quelli qua sopra sono i controlli in cui le variabili che andrò ad inserire nel database prendono semplicemente il valore dalla select degli amministratori gerenti nel caso in cui quel campo specifico non venga modificato nel form. Ad esempio, se un gerente ha l'email "mattia.lazza@gmail.com" e nel form questo valore non viene toccato oppure viene cancellato, io vado a reinserire questa email all'interno della variabile "\$email".

Per quanto riguarda la cancellazione funziona allo stesso modo della modifica, soltanto che invece di stampare i valori che prendo dalla tabella nel form, vado a creare una query che mi elimina il gerente in base all'email trovata nella tabella.

4.22 dettagli.php

Questa è la pagina forse più importante del sito, in quanto è quella che permette di svolgere una registrazione. Soltanto gli utenti hanno il permesso di svolgere una riservazione. Come prima cosa, questa pagina stampa le informazioni dell'alloggio cliccato:

```
//Se la pagina tramite invio di un form o se viene inviato un form della seguente pagina.  
if ($_SERVER['REQUEST_METHOD'] == 'POST') {  
    //Stampo l'alloggio cliccato prendendo tutte le informazioni della pagina "index.php".  
    echo '<div class="row <!--wow fadeIn--> text-center" style="margin-top:12.5vh;">  
  
        <div class="col-lg-12 col-xl-12 ml-xl-4 mb-4">  
            <h3 class="mb-3 font-weight-bold dark-grey-text">  
                <strong>' . $_SESSION['nomeDettagli'] . '</strong>  
            </h3>  
            <p>  
                <strong>' . $_SESSION['indirizzoDettagli'] . '</strong>  
            </p>  
            ...  
    </div>  
</div>
```

Se la pagina viene aperta con una richiesta di tipo POST, vado a stampare tutte le sessioni che ho salvato nell'index (vedi 4.10) all'interno di un div.

L'utente oltre al div con tutti i dettagli dell'alloggio, vedrà un form in cui potrà inserire: data check-in, data-checkout, numero di adulti e numero di bambini. Queste sono le informazioni che un utente deve inserire per poter completare la riservazione.

```
//Inizio a stampare un form.  
$stampa_camere = "<form action='dettagli.php' method='post'>";  
//Per ogni camera possibile.  
for ($i = 1; $i < $stmt->rowCount() + 1; $i++) {  
    //Salvo delle variabili utili.  
    $row = $stmt->fetch(PDO::FETCH_NUM);  
    $r_id = $_SESSION['camera_seelta'] = $row[0];  
    ${"id" . $i} = $row[0];  
    //La query che controlla le date delle riservazioni delle camere.  
    $date_query = "SELECT i.startDate, i.endDate, i.bambini, i.adulti
```

```
        FROM (SELECT '' . $_SESSION['checkin'] . '' startDate, '' . $_SESSION['checkout']
'] . '' endDate, '' . $_SESSION['bambini'] . '' bambini, '' . $_SESSION['adulti'] . ''
adulti) i
        where not exists (
    select 1
    from (select riservazione.id, riservazione.data_checkin, riservazione.data_checkout
, riservazione.id_camera
        from riservazione
        join camera on camera.id = " . ${"id" . $i} . "
        join alloggio on alloggio.email_gerente = camera.email_gerente
        where camera.id_alloggio = " . $_SESSION['id'] . "
        and camera.numero_adulti = " . $_SESSION['adulti'] . "
        and camera.numero_bambini = " . $_SESSION['bambini'] . "
        group by (riservazione.id_camera)) r
    where r.data_checkin < i.endDate and r.data_checkout > i.startDate);
//Preparo la query da eseguire.
$stmt1 = $db->prepare($date_query);
//Eseguo la query.
$stmt1->execute();
$row1 = $stmt1->fetch(PDO::FETCH_NUM);
//Preparo gli elementi della tabella se le camere trovate dovessero essere più di u
na.
$stampa_camere .= "<tr>
    <td>
        . {"id" . $i} .
    </td>
    <td>
        . $row1[3] .
    </td>
    <td>
        . $row1[2] .
    </td>
    <td>
        . date("d-m-Y", strtotime($row1[0])) .
    </td>
    <td>
        . date("d-m-Y", strtotime($row1[1])) .
    </td>
    <td>
        <input type='radio' name='camera" . $i . "' value='camera" . $i . "'>
    </td>
</tr>";
...
...
...
echo "<br>Per quella data sono disponibili le seguenti camere:<br><br>
<table class='table text-center table-striped'>
```

```
<tr style='background-color:#D3D3D3;'>
    <th>Numero</th>
    <th>Adulti</th>
    <th>Bambini</th>
    <th>Check-in</th>
    <th>Check-out</th>
    <th>Riserva</th>
</tr>";

echo $stampa_camere;
echo "</table>";
```

Quello che ho fatto qua sopra è stato stampare una tabella con tutte le camere disponibili trovate in base alla ricerca. Queste camere vengono tutte affiancate da un radio button, così che l'utente possa scegliere quale riservare tra quelle proposte.

```
//Se è stato cliccato almeno un radio button.
if ($isRadioClicked) {

    //Query per le informazioni della camera scelta.
    $camera_scelta_query = "SELECT * FROM (SELECT row_number() OVER (ORDER BY riservazione.id ASC) AS rownumber, riservazione.id_camera
        from riservazione
        join camera on camera.id = riservazione.id_camera
        join alloggio on alloggio.email_gerente = camera.email_gerente
        where camera.id_alloggio = " . $_SESSION['id'] . "
        and camera.numero_adulti = " . $_SESSION['adulti'] . "
        and camera.numero_bambini = " . $_SESSION['bambini'] . "
        group by (riservazione.id_camera)) as get_id where rownumber = " . $id_camera . "";

    //Preparo ed eseguo la query.
    $stmt = $db->prepare($camera_scelta_query);
    $stmt->execute();
    $row = $stmt->fetch(PDO::FETCH_NUM);
    //L'id della camera scelta.
    $camera_scelta = $row[1];

    //La query che salva la riservazione nel database.
    $salvataggio_query = "INSERT INTO riservazione values ('" . $nuovo_id . "', '" . $_SESSION['checkin'] . "', '" . $_SESSION['checkout'] . "', '" . $camera_scelta . "', '" . $_SESSION['email'] . "')";

    //La preparo e la eseguo.
    $stmt = $db->prepare($salvataggio_query);
    $stmt->execute();

    //Preparo il body dell'email.
    unset($body);
```

```
$body = "Hai riservato la camera " . $camera_scelta . ", dal " . date("d-m-Y", strtotime($_SESSION['checkin'])) . " al " . date("d-m-Y", strtotime($_SESSION['checkout'])) . ", per " . $_SESSION['adulti'] . " adulti/o e per " . $_SESSION['bambini'] . " bambini/o.";  
$allegato = "<br>In allegato è possibile trovare un pdf da stampare.";  
$mail = new PHPMailer(true);  
...  
...  
$mail->Body = $body . $allegato;  
  
//Creo un nuovo pdf che allegherò all'email.  
$pdf = new \Mpdf\Mpdf();  
$pdf->WriteHTML($body);  
$pdf_riservazione = $pdf->Output('riservazione.pdf', 'S');
```

Nel caso in cui venga selezionato un radio button e che quindi la camera venga scelta, vado a creare una query per prendere le informazioni della camera da riservare. Lo scopo di questa query è quello di ritornare l'ID della camera che si sta per riservare, così da poter creare l'insert nel della riservazione nel database. Dopo aver salvato la riservazione nel database, semplicemente vado a mandare un email all'utente sempre con PHPMailer e in più gli allego un pdf con le informazioni della riservazione effettuata. Per creare questo PDF ho usato mPDF, che mi ha permesso di inserire del codice HTML all'interno del PDF.

5 Test

5.1 Protocollo di test

Test Case:	TC-001	Nome:	Verifica del funzionamento di: form di registrazione, conferma tramite email e form di login.		
Riferimento:	REQ-005	Descrizione: Sarà necessario verificare che la registrazione funzioni correttamente. Bisognerà inoltre verificare che a registrazione completata arrivi l'email e cliccando il link si attivi l'account e sia poi possibile loggarsi con le credenziali inserite in precedenza.			
Prerequisiti:	-				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php Premere il bottone "Registrati" nella navbar. Riempire il form con dei dati validi (es.: "Mattia", "Lazzaroni", "lazza.yt@gmail.com", "+41761234567", "prova123", "prova123", "Utente base") Premere "Crea Account" (si verrà reindirizzati ad una pagina che indicherà che la registrazione è stata effettuata correttamente e chiederà di controllare la propria inbox). Recarsi nell'inbox dell'email inserita durante la registrazione e cliccare il link presente nella nuova email appena arrivata. Si verrà reindirizzati alla pagina che conferma che l'account è stato attivato, premere "Vai al login". Inserire l'email e la password inserite in precedenza nel form di registrazione e premere "Accedi". 				
Risultati attesi:	Si deve venir reindirizzati alla homepage e deve uscire un messaggio tipo "Ciao Mattia, hai effettuato correttamente l'accesso come utente. Cerca un alloggio."				

Test Case:	TC-002	Nome:	Verifica dei controlli e dei messaggi di errore del form di registrazione.		
Riferimento:	REQ-005	Descrizione: Sarà necessario verificare che nella pagina di registrazione i controlli funzionino correttamente, quindi che non si possano registrare più utenti con la stessa email, che non si possa registrare un utente con dei numeri nel nome, ecc. Dovranno anche uscire dei messaggi di errore.			
Prerequisiti:	È necessario che nel database ci sia almeno un utente dall'email conosciuta.				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php Premere il bottone "Registrati" nella navbar. Riempire il form con dei dati non validi o duplicati (nel caso dell'email). Premere "Crea Account". 				

	5. Ripetere i punti 3. e 4. Più volte, con dati sbagliati diversi (es. numero di telefono con lettere, email senza “@”, nome con numeri, password che non coincidono ed eventualmente campi vuoti).
Risultati attesi:	Dovranno venire mostrati degli errori in base al dato sbagliato o vuoto e la registrazione risulterà non completata.

Test Case:	TC-003	Nome:	Verifica dei messaggi di errore del form di login.
Riferimento:	REQ-005		
Descrizione:	Sarà necessario verificare che i messaggi di errore nel form di login funzionino correttamente, quindi che non ci si possa loggare con un email non presente nel database o con una password che non coincide con l'email		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che sia stata svolta almeno una registrazione in precedenza con email e password conosciute. Attenzione: l'account non deve venir attivato tramite il link.		
Procedura:	<ol style="list-style-type: none"> 1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure <code>localhost/index.php</code> 2. Premere il bottone “Login” nella navbar. 3. Riempire il form con un email che non sia quella conosciuta e una password a scelta. 4. Premere “Accedi”. 5. Riempire il form con l'email conosciuta e presente nel database e la password che coincide con l'email inserita. 6. Pemere ancora “Accedi”. 7. Attivare l'utente dal link ricevuto per email. 8. Riempire il form con l'email conosciuta e la password che non coincide con l'email. 9. Premere un'ultima volta “Accedi”. 		
Risultati attesi:	Dopo la prima volta che si preme “Accedi” dovrà venire mostrato un messaggio che indica che l'email non esiste, dopo la seconda volta il messaggio dovrà che l'utente non è stato attivato e dopo l'ultima volta il messaggio dovrà indicare che la password non coincide con l'email.		

Test Case:	TC-004	Nome:	Verifica del funzionamento del filtro tipologia.
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che scegliendo la tipologia di un alloggio nella sezione “Filtri”, rimangano soltanto gli alloggi di quella tipologia e gli altri spariscano.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi con diverse tipologie.		
Procedura:	<ol style="list-style-type: none"> 1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure <code>localhost/index.php</code>. 2. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. 		

	<ol style="list-style-type: none"> 3. Aprire il menu a tendina della tipologia schiacciando su “Qualsiasi”. 4. Scegliere una tipologia, ad esempio “Camping”. 5. Premere il bottone blu “Applica”. 6. Premere il bottone blu con la lente.
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli con tipologia scelta precedentemente.

Test Case:	TC-005	Nome:	Verifica del funzionamento del filtro regione.
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che scegliendo la regione di un alloggio nella sezione “Filtri”, rimangano soltanto gli alloggi di quella regione e gli altri spariscano.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi con diverse regioni.		
Procedura:	<ol style="list-style-type: none"> 1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure <code>localhost/index.php</code>. 2. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. 3. Inserire una regione nel campo di testo “Regione”. 4. Premere il bottone blu “Applica”. 5. Premere il bottone blu con la lente. 		
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli con la regione scelta precedentemente.		

Test Case:	TC-006	Nome:	Verifica del funzionamento del filtro città.
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che scegliendo la città di un alloggio nella sezione “Filtri”, rimangano soltanto gli alloggi di quella città e gli altri spariscano.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi con diverse città.		
Procedura:	<ol style="list-style-type: none"> 1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure <code>localhost/index.php</code>. 2. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. 3. Inserire una città nel campo di testo “Città”. 4. Premere il bottone blu “Applica”. 5. Premere il bottone blu con la lente. 		
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli con la città scelta precedentemente.		

Test Case:	TC-007	Nome:	Verifica del funzionamento del filtro nome gerente della struttura.
Riferimento:	REQ-004		

Descrizione:	Sarà necessario verificare che scegliendo il nome del gerente di un alloggio nella sezione “Filtri”, rimangano soltanto gli alloggi inseriti da quel gerente e gli altri spariscano.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi inseriti da diversi gerenti.		
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. Inserire il nome di un gerente nel campo di testo “Nome gerente della struttura”. Premere il bottone blu “Applica”. Premere il bottone blu con la lente. 		
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli inseriti dal gerente scelto precedentemente.		

Test Case:	TC-008	Nome:	Verifica del funzionamento della ricerca tramite nome		
Riferimento:	REQ-004				
Descrizione:	Sarà necessario verificare che inserendo un nome della barra di ricerca, vengano mostrati solo gli alloggi che contengono una parte del nome inserito.				
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi.				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Inserire il nome di un alloggio o una parte del nome (ad esempio “Fergus” oppure “ergu”) nella barra di ricerca. Premere il bottone blu con la lente. 				
Risultati attesi:	Dovranno scomparire tutti gli alloggi, tranne quello/i che soddisfa/no la ricerca (in questo caso l’Hotel Fergus Geminis).				

Test Case:	TC-009	Nome:	Verifica del funzionamento dei filtri combinati		
Riferimento:	REQ-004				
Descrizione:	Sarà necessario verificare che scegliendo più filtri, essi continuino a funzionare.				
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi.				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. Scegliere o inserire almeno un valore nei quattro filtri presenti (ad esempio tipologia “Albergo” e nome gerente della struttura “Mattia”). Premere il bottone blu “Applica”. 				

	<p>5. Inserire il nome di un alloggio o una parte del nome (ad esempio “Hotel”) nella barra di ricerca.</p> <p>6. Premere il bottone blu con la lente.</p>
Risultati attesi:	Dovranno scomparire tutti gli alloggi, tranne quello/i che soddisfa/no la ricerca (in questo caso l’Hotel Fergus Geminis e il Katikies Hotel).

Test Case:	TC-010	Nome:	Verifica del funzionamento della riservazione con email di conferma e PDF da stampare		
Riferimento:	REQ-005				
Descrizione:	Sarà necessario verificare che funzioni la riservazione di un alloggio e che arrivi un email con le informazioni della riservazione.				
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po’ di alloggi, camere e riservazioni. Bisognerà inoltre aver effettuato il login come utenti, se non si sa come fare, riferirsi al TC-001.				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Scegliere ad esempio l’alloggio “Hotel Fergus Geminis” dalla lista e premere “Mostra dettagli”. Inserire 2 adulti e 0 bambini nei campi sotto. Scegliere una data di check-in, ad esempio 02.02.2020. Scegliere una data di check-out, ad esempio 09.02.2020. Premere il bottone blu “Cerca disponibilità”. Usciranno 2 camere disponibili, scegliere ad esempio la camera 36 selezionando il secondo radio button sulla destra. Premere il bottone blu “Riserva la camera”. 				
Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita della riservazione e controllando le email, dovrebbe essercene una nuova con le informazioni della riservazione e con un pdf in allegato.				

Test Case:	TC-011	Nome:	Verifica del funzionamento della pagina del caricamento delle disponibilità per i gerenti.		
Riferimento:	REQ-003				
Descrizione:	Sarà necessario verificare che funzioni correttamente il caricamento di una camera.				
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po’ di alloggi, camere e riservazioni. Bisognerà inoltre aver fatto l’accesso come amministratore gerente, se non si sa come fare, riferirsi al TC-001.				
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Nella barra di navigazione sopra selezionare “Amministratore gerente”. Premere il bottone blu “Carica disponibilità”. Scegliere il numero di adulti e di bambini, ad esempio 2 e 3. 				

	<p>5. Inserire il nome dell'alloggio in cui si vuole inserire la camera, ad esempio "Alla Galleria B&B".</p> <p>6. Premere il bottone blu "Aggiungi".</p>
Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita dell'aggiunta e controllando il database, dovrebbe esserci la nuova camera appunto nella tabella "Camera".

Test Case:	TC-012	Nome:	Verifica del funzionamento della pagina della gestione dei clienti.		
Riferimento:	REQ-003	Descrizione:			
Prerequisiti:	Sarà necessario verificare che funzioni correttamente la gestione di un cliente.				
Procedura:	<p>1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php.</p> <p>2. Nella barra di navigazione sopra selezionare "Amministratore gerente".</p> <p>3. Premere il bottone blu "Gestisci clienti".</p> <p>4. Premere il bottone azzurro "Modifica", ad esempio nella riga della riservazione 22.</p> <p>5. Nel form, cambiare le date di check-in e check-out, ad esempio in 15.01.2020 e 20.11.2019</p> <p>6. Premere "Salva modifiche"</p>				
Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita della modifica e controllando il database, dovrebbe esserci le nuove date della riservazione appunto nella tabella "Riservazione".				

Test Case:	TC-013	Nome:	Verifica del funzionamento della stampa per gli amministratori gerenti		
Riferimento:	REQ-006	Descrizione:			
Prerequisiti:	Sarà necessario verificare che gli amministratori gerenti possano stampare correttamente le riservazioni fatte in un certo periodo e/o da un certo utente.				
Procedura:	<p>1. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php.</p> <p>2. Nella barra di navigazione sopra selezionare "Amministratore gerente".</p> <p>3. Premere il bottone blu "Stampa riservazioni".</p> <p>4. Nel form, inserire le date di check-in e check-out per cui si vogliono vedere le riservazioni, ad esempio dal 01.11.2019 al 25.12.2019. Inserire eventualmente anche l'email dell'utente, ad esempio "lazza.yt@gmail.com".</p> <p>5. Premere "Cerca".</p> <p>6. Verrà mostrata una lista di riservazioni, premere "Invia le riservazioni".</p>				

Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita dell'invio dell'email e controllando l'email del gerente, dovrebbe essercene una nuova con le informazioni delle riservazioni stampate precedentemente a schermo.		
--------------------------	--	--	--

Test Case: Riferimento:	TC-014 REQ-002	Nome:	Verifica del funzionamento della pagina del caricamento delle strutture per gli amministratori.
Descrizione:	Sarà necessario verificare che funzioni correttamente il caricamento di un alloggio.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi, camere e riservazioni. Bisognerà inoltre aver fatto l'accesso come amministratore, se non si sa come fare, riferirsi al TC-001.		
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Nella barra di navigazione sopra selezionare "Amministratore". Riempire il form come si preferisce, con dei dati però consoni a quanto indica l'input box. Premere il bottone blu "Aggiungi". 		
Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita dell'aggiunta e tornando nella home page e scorrendo infondo alla pagina dovrebbe esserci il nuovo alloggio.		

Test Case: Riferimento:	TC-015 REQ-003	Nome:	Verifica del funzionamento della pagina della gestione dei gerenti.
Descrizione:	Sarà necessario verificare che funzioni correttamente la gestione di un gerente.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci sia magari qualche amministratore gerente. Bisognerà inoltre aver fatto l'accesso come amministratore, se non si sa come fare, riferirsi al TC-001.		
Procedura:	<ol style="list-style-type: none"> Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php. Nella barra di navigazione sopra selezionare "Amministratore". Premere il bottone blu "Gestisci gerenti". Premere il bottone azzurro "Modifica", ad esempio nella riga del gerente "mattia.lazza@gmail.com". Nel form, cambiare dei valori come si preferisce, nel mio caso andrò a modificare il numero di telefono in "+41764650110". Premere "Salva modifiche". 		
Risultati attesi:	Dovrà venire mostrato un messaggio di riuscita della modifica e controllando il database, dovrebbe esserci le nuove informazioni del gerente appunto nella tabella "Amministratore_gerente".		

Test Case: Riferimento:	TC-016 REQ-007	Nome:	Verifica del funzionamento della gestione della fatturazione del sito
--	-------------------	--------------	---

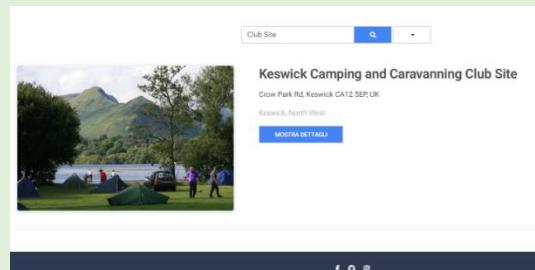
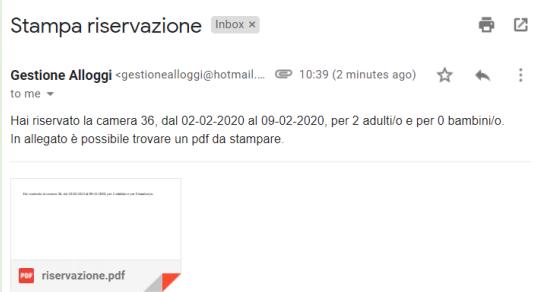
Descrizione:	Sarà necessario verificare che funzioni correttamente la gestione della fatturazione del sito web.
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che venga effettuata qualche riservazione.
Procedura:	-
Risultati attesi:	-

5.2 Risultati test

Test Case	Funzionamento	Commento	Data
TC-001	OK	Si riesce correttamente a registrarsi, l'email arriva, il link attiva l'account e si riesce a fare il login.	29.11.19
TC-002	OK	<p>Ho provato ad inserire il nome “Mattia123”, l'email “mattia.lazzaronisamtrevano.ch”, il numero “0764650110r”, la password “prova123”, la seconda password “prova12” e ho lasciato il cognome vuoto. Il risultato è stato che sono stati mostrati i seguenti errori:</p> <p style="color: red;">Il nome non può contenere valori numerici</p> <p style="color: red;">Il cognome è richiesto</p> <p style="color: red;">Il formato dell'email non è valido</p> <p style="color: red;">Il formato per il numero di telefono non è valido</p> <p style="color: red;">Le due password non coincidono</p>	29.11.19
TC-003	OK	<p>Credenziali presenti nel database: “lazza.yt@gmail.com:prova123”. Inizialmente nel form di login ho inserito “test@gmail.com” e “prova123”. Questo è il risultato:</p> <p>Successivamente, ho provato con l'email presente nel database ma con la password errata:</p> <p>Nessun utente trovato con questa email!</p> <p>Non hai verificato l'account, controlla la tua email!</p> <p>Infine, ho inserito le credenziali corrette dell'utente non attivo:</p> <p>La password che hai inserito non è corretta</p>	29.11.19

TC-004	OK	<p>Ho provato a scegliere la tipologia “Camping” nella sezione “filtri”, ho applicato il filtro, ho premuto la lente e tutti gli alloggi, tranne quelli di tipologia “Camping”, sono spariti:</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Keswick Camping and Caravanning Club Site Crose Park Rd, Keswick CA12 5EP UK Keswick, North West MOstra dettagli</p> </div> <div style="text-align: center;"> <p>Camping La Pointe Camping La Pointe route de Saint Coulitz, 29150 Châteaulin France Châteaulin, Brittany MOstra dettagli</p> </div> <div style="text-align: center;"> <p>D'Olde Kamp Dwingelooweg 26, 7954 KX Aanzen, Netherlands Aanzen, Drenthe MOstra dettagli</p> </div> </div>	01.12.19
TC-005	OK	<p>Ho provato a inserire la regione “Veneto” nella sezione “filtri”, ho applicato il filtro, ho premuto la lente e tutti gli alloggi, tranne quello con regione “Veneto”, sono spariti:</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Alla Galleria B&B Via Antonio Cantore, 4, 37121 Verona VR, Italy Verona, Veneto MOstra dettagli</p> </div> <div style="text-align: center;"> <p>f o @</p> </div> </div>	01.12.19
TC-006	OK	<p>Ho provato a inserire la città “Verona” nella sezione “filtri”, ho applicato il filtro, ho premuto la lente e tutti gli alloggi, tranne quello con città “Verona”, sono spariti:</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;"> <p>Alla Galleria B&B Via Antonio Cantore, 4, 37121 Verona VR, Italy Verona, Veneto MOstra dettagli</p> </div> <div style="text-align: center;"> <p>f o @</p> </div> </div>	01.12.19

TC-007	OK	<p>Ho provato a inserire il nome del gerente “Mattia” nella sezione “filtri”, ho applicato il filtro, ho premuto la lente e tutti nessun alloggio è sparito in quanto gli alloggi che esistono per ora sono stati</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Hotel Fergus Geminis Carrer dels Tamarells, s/n, 07600 Palma, Illes Balears, Spain Palma di Maiorca, Isole Baleari</p> <p>MOSTRA DETTAGLI</p> </div> <div style="text-align: center;">  <p>Alla Galleria B&B Via Antonio Cantore, 4, 37121 Verona VR, Italy Verona, Veneto</p> <p>MOSTRA DETTAGLI</p> </div> <div style="text-align: center;">  <p>Keswick Camping and Caravanning Club Site Cone Park Rd, Keswick CA12 5EP, UK Keswick, Cumbria, UK</p> <p>MOSTRA DETTAGLI</p> </div> </div> <p>tutti aggiunti da me:</p>	01.12.19
TC-008	OK	<p>Ho provato inserire il nome dell’alloggio “Hotel” nella barra di ricerca, ho premuto la lente e tutti gli alloggi sono spariti, tranne i due con “Hotel” nel nome:</p> <div style="display: flex; justify-content: space-around;"> <div style="text-align: center;">  <p>Hotel Fergus Geminis Carrer dels Tamarells, s/n, 07600 Palma, Illes Balears, Spain Palma di Maiorca, Isole Baleari</p> <p>MOSTRA DETTAGLI</p> </div> <div style="text-align: center;">  <p>Katikies Hotel Nik. Nomikou (Main Street), Oia 847 02, Greece Oia, South Aegean</p> <p>MOSTRA DETTAGLI</p> </div> </div>	01.12.19

TC-009	OK	<p>Nella sezione “filtri” ho provato a scegliere la tipologia “Camping”, la città “Keswick”, nessuna regione e il nome del gerente “Mattia”. Ho applicato poi tutti i filtri, inserito il nome “Club Site” nella barra di ricerca, ho premuto la lente e tutti gli alloggi, tranne quelli di tipologia “Camping”, città “Keswick” e nome gerente “Mattia”, sono spariti:</p> 	01.12.19										
TC-010	OK	<p>Ho provato a riservare la camera 36 dell’Hotel Fergus Geminis, questo è stato il messaggio di riuscita:</p> <p>La riservazione della camera 36, dal 02-02-2020 al 09-02-2020, per 2 adulti/o e per 0 bambini/o è stata aggiunta con successo. Controlla la tua email se vuoi stampare la riservazione.</p> <p>Andando poi a controllare le email, ho trovato</p>  <p>questo:</p>	19.12.19										
TC-011	OK	<p>Ho provato ad inserire una camera per 2 adulti e 3 bambini nell’alloggio “Alla Galleria B&B”. Mi è stato mostrato il seguente messaggio:</p> <p>La camera è stata aggiunta con successo.</p> <p>Andando a controllare la tabella “Camera” del database ho potuto notare la nuova camera:</p> <table border="1"> <thead> <tr> <th>id</th> <th>numero_bambini</th> <th>numero_adulti</th> <th>id_alloggio</th> <th>email_gerente</th> </tr> </thead> <tbody> <tr> <td>37</td> <td>3</td> <td>2</td> <td>2</td> <td>mattia.lazza@gmail.com</td> </tr> </tbody> </table>	id	numero_bambini	numero_adulti	id_alloggio	email_gerente	37	3	2	2	mattia.lazza@gmail.com	19.12.19
id	numero_bambini	numero_adulti	id_alloggio	email_gerente									
37	3	2	2	mattia.lazza@gmail.com									
TC-012	OK	<p>Ho provato a modificare le date della riservazione con ID “22”. Mi è stato mostrato il seguente messaggio:</p> <p>La riservazione con ID '22' è stata modificata con successo.</p>	19.12.19										

		<p>Andando a controllare la tabella “Riservazione” e cercando quella con ID “22”, ho potuto notare che le date erano quelle nuove appena inserite:</p> <table border="1"> <thead> <tr> <th>id</th><th>data_checkin</th><th>data_checkout</th><th>id_camera</th><th>email_utente</th></tr> </thead> <tbody> <tr> <td>22</td><td>2020-01-15</td><td>2020-01-20</td><td>1</td><td>lazza.yt@gmail.com</td></tr> </tbody> </table>	id	data_checkin	data_checkout	id_camera	email_utente	22	2020-01-15	2020-01-20	1	lazza.yt@gmail.com	
id	data_checkin	data_checkout	id_camera	email_utente									
22	2020-01-15	2020-01-20	1	lazza.yt@gmail.com									
TC-013	OK	<p>Ho provato a mandare per email al gerente una serie di riservazioni così che egli possa stamparle. Questo è il messaggio una volta premuto il bottone per inviare l'email:</p> <div style="background-color: #f0f0f0; padding: 5px;"> L'email con le riservazioni è stata inviata a 'mattia.lazza@gmail.com' con successo. </div> <p>Controllando la posta elettronica del gerente, ho notato l'email:</p> <p>In allegato è possibile trovare un pdf da stampare.</p>	19.12.19										
TC-014	OK	<p>Ho provato a riempire il form nel seguente modo:</p> <p>Premendo poi il bottone “Aggiungi”, mi è stato mostrato il seguente messaggio:</p> <div style="background-color: #f0f0f0; padding: 5px;"> La struttura è stata aggiunta con successo. </div>	19.12.19										

		<p>Tornando poi nella homepage, (schiacciando “Tutte le strutture” nella navbar in alto) ho potuto notare il nuovo alloggio infondo alla pagina:</p>  <p>Hotel Milano Piazza della Stazione, 6850 Mendrisio Mendrisio, Ticino MOSTRA DETTAGLI</p>											
TC-015	OK	<p>Ho provato a modificare il numero di telefono dell'amministratore gerente con email "mattia.lazza@gmail.com" in "+41764650110".</p> <p>L'amministratore gerente con email 'mattia.lazza@gmail.com' è stato modificato con successo.</p> <p>Questo è stato il messaggio mostratomi una volta premuto "Salva modifiche":</p> <p>Andando a controllare la tabella "Amministratore gerente" e cercando quello con email "mattia.lazza@gmail.com", ho potuto notare che il numero di telefono era quello nuovo appena inserito:</p> <table border="1"> <thead> <tr> <th>email</th> <th>nome</th> <th>cognome</th> <th>password_admin_gerente</th> <th>n_telefono</th> </tr> </thead> <tbody> <tr> <td>mattia.lazza@gmail.com</td> <td>Matta</td> <td>Lazzaroni</td> <td>\$2y\$10\$UwB.V5xhQAxhXRKZycFydeTbs9j9AlS...</td> <td>+41764650110</td> </tr> </tbody> </table>	email	nome	cognome	password_admin_gerente	n_telefono	mattia.lazza@gmail.com	Matta	Lazzaroni	\$2y\$10\$UwB.V5xhQAxhXRKZycFydeTbs9j9AlS...	+41764650110	19.12.19
email	nome	cognome	password_admin_gerente	n_telefono									
mattia.lazza@gmail.com	Matta	Lazzaroni	\$2y\$10\$UwB.V5xhQAxhXRKZycFydeTbs9j9AlS...	+41764650110									
TC-016	NOK	<p>Il test non è passato in quanto la parte per la gestione delle fatturazioni non è stata implementata.</p>	19.12.19										

5.3 Mancanze/limitazioni conosciute

Purtroppo, per mancanza di tempo non ho potuto implementare tutto quello che riguarda il requisito REQ-007, ovvero la gestione della fatturazione del sito. Inoltre, la parte della stampa riguardante gli amministratori non è stata implementata in quanto richiedeva il sistema di gestione della fatturazione. La query che si occupa di controllare che in un certo alloggio, per una certa data di check-in e di check-out e per un certo numero di adulti e di bambini, ci sia una camera libera non funziona al 100%. Infatti, esegue tutti i controlli correttamente, tranne quelli per la data di check-in e di check-out. È quindi possibile svolgere più riservazioni della stessa camera nello stesso giorno.

6 Consuntivo

Sicuramente, andando a confrontare la pianificazione iniziale con il Gantt consuntivo, si possono notare subito delle grandi differenze. Questo perché purtroppo non ho calcolato bene i tempi di tutte le varie attività e soprattutto non ho pensato alle cose extra, come ad esempio la configurazione di PHPMailer che mi ha preso ben 10 ore.

Nell'analisi non sono rimasto particolarmente indietro, tutta via i primi rallentamenti sono stati riscontrati nella progettazione, in cui pensavo di impiegare circa una lezione per il design delle interfacce, bensì ce ne ho spurate due e mezzo. Probabilmente sono andato a realizzare dei mockup troppo complicati e dettagliati che mi hanno fatto perdere del tempo prezioso.

Durante l'implementazione un altro elemento che mi ha fatto rimanere indietro rispetto alla pianificazione, oltre a PHPMailer come citato in precedenza, è stato l'integrazione del template Bootstrap e quindi tutta la parte di front-end. Queste sono quindi altre 8 ore che non avevo pianificato ma che ho comunque voluto investire nel mio progetto. In generale posso dire che nell'implementazione quasi per qualsiasi cosa ho impiegato più tempo di quanto ne avevo pensato inizialmente, questo mi ha costretto a svolgere del lavoro a casa ad esempio durante i weekend.

Per quanto riguarda il testing, a settembre avevo pensato di aggiornare costantemente i test in parallelo con l'implementazione (ad esempio: creo il form di registrazione e scrivo i test appena l'ho finito). Tuttavia, ho quasi subito abbandonato quest'idea in quanto l'implementazione mi prendeva sempre più di quanto pensassi e quindi non trovavo il tempo per scrivere anche i test. Ho quindi dovuto lasciare i test verso la fine del progetto e dividerli in 2/3 giorni.

Per la scrittura dell'implementazione nella documentazione è stata la stessa cosa che con i test: mi aspettavo di aggiornarla in parallelo con l'implementazione, tuttavia mi mancava il tempo.

L'analisi e la progettazione non sono state un problema in fatto di documentazione in quanto per svolgere queste due fasi lavoravo direttamente sulla documentazione.

Anche i diari non sono stati un problema e li ho sempre realizzati per tempo.

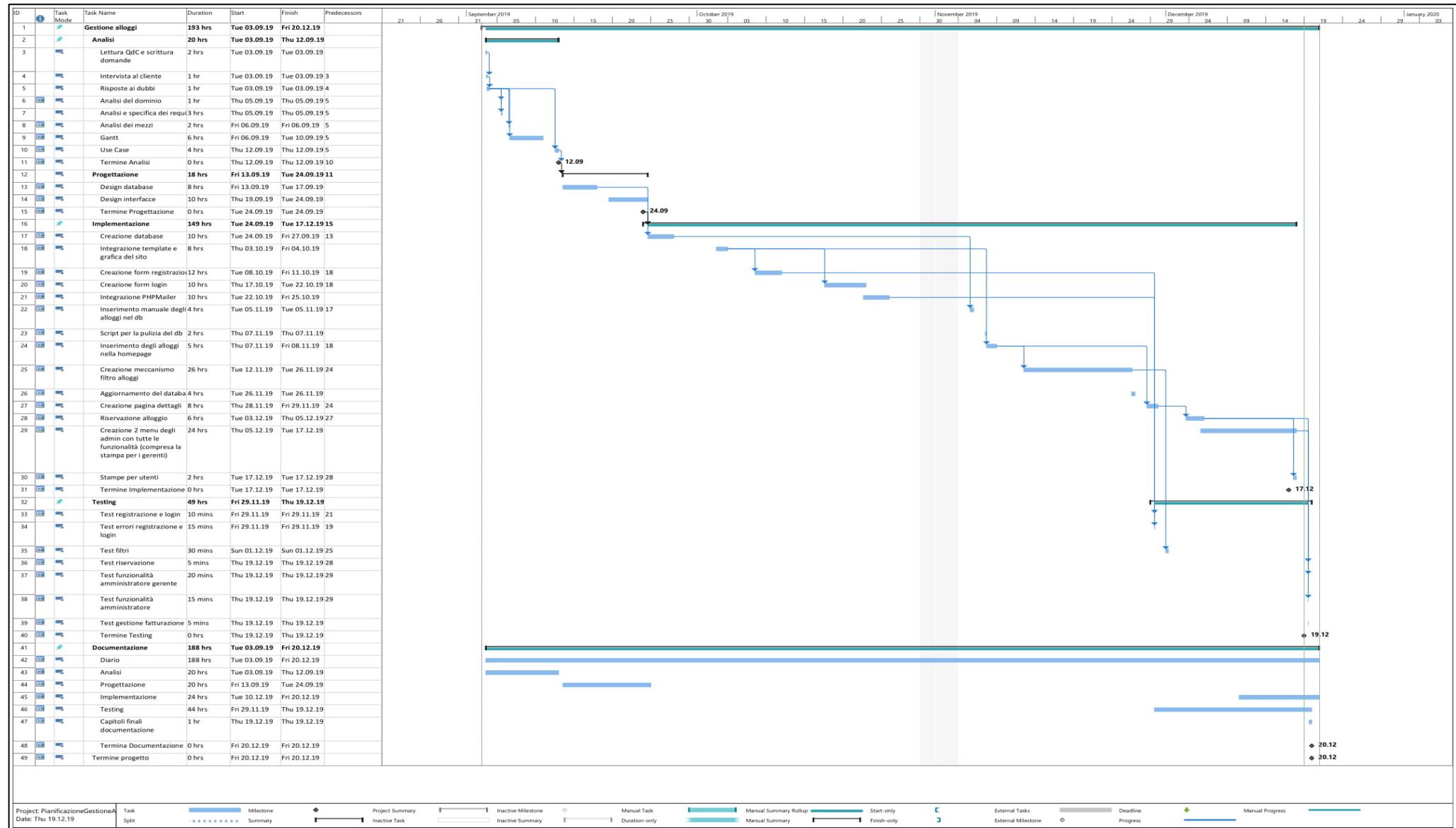


Diagramma di Gantt rappresentante il consuntivo del progetto

7 Conclusioni

Sicuramente il mio prodotto non cambierà il mondo in quanto si tratta di una semplice sito per la gestione delle disponibilità degli alloggi. Tuttavia, può servire per dimostrare che per riservare una camera non sono per forza necessari siti incredibilmente complicati ed avanzati. Probabilmente nessuno userà mai questo prodotto, ma comunque non credo sia giusto definirlo come una perdita di tempo, in quanto a me ha permesso di imparare molte cose nuove, sia in ambito tecnico che in ambito della gestione dei tempi del progetto. Infatti, spero che questo progetto mi possa servire da lezione per far sì che la prossima volta non debba rimanere con delle parti del prodotto non implementate o comunque non terminare del tutto.

7.1 Sviluppi futuri

Sicuramente nel futuro potrei pensare di implementare quanto tralasciato, ovvero la gestione delle fatturazioni. Potrei inoltre sistemare la query per le riservazioni in modo da evitare più riservazioni della stessa camera nello stesso periodo.

Inoltre, volendo, si potrebbe anche pensare a delle aggiunte per rendere più performante il sito, come un sistema di paginazione per dividere gli alloggi in più pagine. Sarebbe anche utile riscrivere il codice in maniera OOP (Object Oriented Programming), magari tramite il template MVC fornитоci dal professore Massimo Sartori, così da rendere più semplice la gestione e la manutenzione del sito web.

7.2 Considerazioni personali

Personalmente ho trovato questo progetto troppo lungo, ripetitivo e alle volte con requisiti troppo complicati. Avrei trovato più utile realizzare un progetto simile a questo, ma più corto e meno ripetitivo. Per esempio, per poter completare il progetto erano richieste troppe tabelle e una marea di query diverse. La maggior parte del tempo del progetto l'ho passata a realizzare query, il che mi è sembrato un po' monotono. Il progetto mi ha comunque permesso di rafforzare le mie conoscenze di PHP e di SQL e di imparare cose nuove e sicuramente utili, come PHPMailer e mPDF.

8 Bibliografia

8.1 Sitografia

- <https://mockflow.com/>, MockFlow - Online Wireframe Tools, Prototyping Tools, Design tool, UI Mockups, UX Suite, Desktop software, UX Planning, 19.09.2019.
- <https://mdbootstrap.com/freebies/jquery/1-column-listing/>, 1 Column Listing Template - Material Design for Bootstrap, 01.10.2019.
- <https://bootsnipp.com/>, Home of free code snippets for Bootstrap, 01.10.2019.
- <https://fontawesome.com/>, Font Awesome, 03.10.2019
- <http://codewithawa.com/posts/complete-user-registration-system-using-php-and-mysql-database>, Complete user registration system using PHP and MySQL database | CodeWithAwa, 08.10.2019
- <https://www.tutorialrepublic.com/php-tutorial/php-mysql-login-system.php>, Creating a User Login System with PHP and MySQL - Tutorial Republic, 17.10.2019
- <https://github.com/PHPMailer/PHPMailer>, PHPMailer/PHPMailer: The classic email sending library for PHP, 26.10.2019
- <https://www.theguardian.com/travel/2018/mar/24/20-best-campsites-europe-camping-greece-france-spain>, 20 of the best campsites in Europe | Travel | The Guardian, 05.11.2019
- <https://www.travelandleisure.com/worlds-best/hotels-top-100-overall>, Best 100 Hotels: World's Best Hotels 2019 | Travel + Leisure, 05.11.2019
- <https://www.bedandbreakfast.com/info/travelers/awards/amenities-awards/top-luxury-inns>, Top 10 Luxurious B&Bs | BedandBreakfast.com, 05.11.2019
- <https://github.com/mpdf/mpdf>, mpdf/mpdf: PHP library generating PDF files from UTF-8 encoded HTML, 15.12.2019

9 Allegati

Elenco degli allegati:

- Abstract
- Diari di lavoro
- File per gli accessi
- Quaderno dei compiti

ALLEGATI

Gestione alloggi | Diario di lavoro - 03.09.2019

Mattia Lazzaroni

Canobbio, 03.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nel corso delle prime due ore della lezione ci sono stati assegnati i progetti.
15:00 - 16:30	In seguito ho iniziato a scrivere le domande da fare al docente responsabile (domande che ho poi fatto alle 16:30). Dopo di questo ho iniziato a creare la repository su github (https://github.com/mattialazzaroni/Gestione-alloggi/) e ho iniziato a creare il README.md, la documentazione (in cui ho già inserito alcuno delle informazioni principali) e il primo diario.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

Pianificazione ancora non iniziata.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro dovrei iniziare l'analisi e la pianificazione del progetto.

Gestione alloggi | Diario di lavoro - 05.09.2019

Mattia Lazzaroni

Canobbio, 05.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nel corso delle prime due ore della lezione ho inizialmente inserito le risposte alle domande che ho posto alla fine della scorsa lezione al superiore professionale. Successivamente, ho scritto nella documentazione lo scopo del progetto, ovvero il capitolo 1.3. Ho poi iniziato la parte di analisi (dal capitolo 1.4 in poi).
15:00 - 16:30	Dopo la pausa ho dovuto installare VMware Workstation per poter usare MS Project, che mi serve per il gantt di pianificazione. Inoltre sono andato avanti con la documentazione, in particolare il punto 1.5, ovvero "Analisi e specifica dei requisiti".

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

Pianificazione ancora non iniziata.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro devo in primis realizzare il gantt della pianificazione, successivamente continuerò il lavoro di analisi e di progettazione sulla documentazione.

Gestione alloggi | Diario di lavoro - 06.09.2019

Mattia Lazzaroni

Canobbio, 06.09.2019

Lavori svolti

Orario	Lavori svolti
13:15	Nel corso di tutta la giornata ho lavorato solamente sulla pianificazione del progetto tramite MS Project. La pianificazione non è ancora completamente terminata dato che ho dovuto investire del tempo per imparare ad usare MS Project tramite dei tutorial online.
-	
16:30	

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In linea con la pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò sicuramente la pianificazione e inizierò la parte di progettazione (design architettura-database-interfacce).

Gestione alloggi | Diario di lavoro - 10.09.2019

Mattia Lazzaroni

Canobbio, 10.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Durante le prime due ore della giornata di oggi, mi sono occupato di terminare il gantt iniziato la scorsa lezione.
15:00 - 16:30	Dopo aver sistemato un problema che non mi permetteva di utilizzare la rete, ho continuato l'analisi dei requisiti e ho terminato il capitolo 1.4 della documentazione. Ho infine scaricato StarUML, che mi servirà per realizzare gli use case.

Problemi riscontrati e soluzioni adottate

Dopo la pausa ho riscontrato un problema con la rete che mi ha impedito di procedere con il mio lavoro per circa 1 ora.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, dato che mi mancano gli use case prima di poter cominciare la progettazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro farò gli use case, terminerò l'analisi dei requisiti e comincerò la progettazione.

Gestione alloggi | Diario di lavoro - 12.09.2019

Mattia Lazzaroni

Canobbio, 12.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Durante le prime due ore della giornata di oggi ho realizzato lo use case del sito. Questo è il risultato: Use Case Diagram
15:00 - 16:30	Dopo la pausa mi sono concentrato sulla documentazione, in particolare ho sistemato la struttura generale e ho aggiunto la descrizione del Gantt e degli Use Case. In generale nella giornata di oggi ho finito l'analisi sulla documentazione.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto dovrei essere nel mezzo della progettazione ma ho appena finito l'analisi.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro inizierò la progettazione, a partire dal design dell'architettura.

Gestione alloggi | Diario di lavoro - 13.09.2019

Mattia Lazzaroni

Canobbio, 13.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	All'inizio della giornata di lavoro di oggi ho avuto una discussione con il responsabile del progetto, che mi ha detto delle cose da modificare e ci siamo accordati riguardo alcuni requisiti. Prima della pausa ho sistemato le richieste del responsabile.
15:00 - 16:30	Dopo la pausa ho cominciato il design del database che dovrò terminare nella prossima lezione

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto oggi dovrei aver terminato la progettazione ma oggi l'ho solamente iniziata (design database).

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò il design del database.

Gestione alloggi | Diario di lavoro - 17.09.2019

Mattia Lazzaroni

Canobbio, 17.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 16:30	All'inizio della giornata di lavoro di oggi ho lavorato sulla documentazione. Ho modificato il capitolo dell'analisi dei requisiti, aggiungendone di nuovi e sistemandone quelli già esistenti. Fatto ciò ho aggiunto all'inizio del capitolo una descrizione per ogni tabella dei requisiti. Finito di lavorare sulla documentazione mi sono dedicato alla realizzazione del design del database. Dopo averlo terminato ho cominciato il design del database per la gestione della fatturazione, che non so se è necessario e che in caso non serva andrà ad eliminare durante la prossima lezione.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto dovrei essere in fase di implementazione ma sono ancora in fase di design.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò il design del secondo database, se il mio responsabile lo riterrà necessario, altrimenti comincerò il design delle interfacce.

Gestione alloggi | Diario di lavoro - 19.09.2019

Mattia Lazzaroni

Canobbio, 19.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Iniziata la lezione, sono andato a controllare le risposte del docente responsabile alle domande postegli durante la scorsa lezione. Sono andato dunque a modificare il database secondo le sue specifiche ed ho eliminato il database inutile. In seguito sono incappato in un problema con il tool "GitHub Desktop" che non mi permetteva di pullare da GitHub e ho perso molto tempo online per trovare una soluzione che non ho trovato. Ho poi sistemato il problema semplicemente riavviando il computer.
14:45 - 15:00	Durante la pausa mi sono recato dal docente responsabile per porgli qualche domanda riguardo l'integrazione della tabella "Fattura" all'interno del database e riguardo la creazione dello script per l'inserimento degli alloggi.
15:00 - 16:30	Dopo la pausa ho iniziato il design delle interfacce. Prima di tutto ho dovuto investire del tempo per imparare ad usare il sito WireFrame per poter realizzare i design. Per oggi ho realizzato le interfacce per la registrazione e per il login. Sign up Login

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto dovrei essere in fase di implementazione ma sono ancora in fase di design.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò il design del secondo database, se il mio responsabile lo riterrà necessario, altrimenti comincerò il design delle interfacce.

Gestione alloggi | Diario di lavoro - 20.09.2019

Mattia Lazzaroni

Canobbio, 20.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nel corso delle prime due ore ho realizzato solamente il mockup della homepage. Home
15:00 - 16:30	Dopo la pausa ho invece realizzato il mockup della pagina di dettaglio di un alloggio. Questo è quanto ho fatto per ora: Dettagli Alla fine della lezione ho inserito tutti i mockups nella documentazione con le relative didascalie.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto dovrei essere in fase di implementazione ma sono ancora in fase di design.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò il design delle interfacce e magari comincerò già l'implementazione.

Gestione alloggi | Diario di lavoro - 24.09.2019

Mattia Lazzaroni

Canobbio, 24.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nelle due ore prima della pausa ho inizialmente lavorato sulla documentazione, sistemando alcune cose e inserendo le descrizioni dei mockups della scorsa lezione. Dopo di questo sono andato a progettare gli ultimi 2 mockups: quello per la pagina degli amministratori gerenti e quello per la pagina degli amministratori. Questi sono i risultati: Amministratore Gerente Amministratore
15:00 - 16:30	Dopo la pausa ho iniziato a lavorare sull'implementazione, in particolare sulla creazione del Database. Prima di questo ho dovuto però installare MySQL Workbench, attività che mi ha preso tanto tempo in quanto il servizio di mysql non era inizialmente in esecuzione e successivamente non mi permetteva di accedere con lo user 'root'@'localhost'. Sono quindi dovuto andare a modificare i privilegi di questo utente.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto oggi dovrei avere finito i form di registrazione e di login, ma ho solamente creato il Database. Attenzione, l'ordine di implementazione era: form di registrazione, form di login e Database. Tuttavia, ho preferito realizzare prima il Database rispetto ai form.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro invece di continuare con l'implementazione dovrò lavorare sulla documentazione per inserire le descrizioni del design del Database e dei mockups delle interfacce.

Gestione alloggi | Diario di lavoro - 27.09.2019

Mattia Lazzaroni

Canobbio, 27.09.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:15	Nelle due ore prima della pausa sono incappato in un problema che sono andato a spiegare nella sezione "Problemi riscontrati e soluzioni adottate" e che mi ha preso 1 ora di orologio per poterlo sistemare.
14:15 - 16:30	Una volta sistemato il problema mi sono dedicato alla realizzazione del Database. Purtroppo alle 16:15 mi sono accorto di un problema nel design del Database: in alcune tabella mancavano delle foreign key per le cardinalità (1,1), di conseguenza ho deciso che questo weekend mi prenderò del tempo a casa per sistemare il design e terminare ufficialmente il Database.

Problemi riscontrati e soluzioni adottate

andando ad inserire sul terminale "mysql -u root -p", ottenevo l'errore: "ERROR 1045 - Access denied for user 'root'@'localhost' (using password: YES)". Per sistemare questo problema ho disinstallato completamente MySQL e l'ho reinstallato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto oggi dovrei avere finto i form di registrazione e di login, ma ho solamente creato il Database. Attenzione, l'ordine di implementazione era: form di registrazione, form di login e Database. Tuttavia, ho preferito realizzare prima il Database rispetto ai form.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro invece di continuare con l'implementazione dovrò lavorare sulla documentazione per inserire le descrizioni del design del Database e dei mockups delle interfacce.

Gestione alloggi | Diario di lavoro - 01.10.2019

Mattia Lazzaroni

Canobbio, 01.10.2019

Lavori svolti

Come scritto nell'ultimo diario, questo weekend mi sono preso del tempo a casa per sistemare il design del Database e per creare ufficialmente il Database.

Orario	Lavori svolti
13:15 - 14:15	Inizialmente mi sono preso del tempo per cercare un template gratuito. Quello che ho trovato è realizzato per MDBBootstrap e link per il template è il seguente: https://mdbootstrap.com/freebies/jquery/1-column-listing/ . Dopo aver scelto il template ho realizzato uno snippet, modificandone uno già esistente, per poter ricercare un alloggio e per poter aggiungere dei filtri alla ricerca. È possibile vedere un'anteprima dello snippet tramite il seguente link: https://bootsnipp.com/user/snippets/gXoaQ .
14:15 - 16:30	Dopo la pausa, ho iniziato a lavorare sul template scaricato nelle prime due ore. Ho iniziato ad implementare la "Home" del sito. Ho già realizzato la navbar, il titolo e sto creando la barra di ricerca con i vari filtri.

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto oggi avrei dovuto iniziare a realizzare la pagina per l'amministratore.

Programma di massima per la prossima giornata di lavoro

Iniziare il form di login e di registrazione.

Gestione alloggi | Diario di lavoro - 03.10.2019

Mattia Lazzaroni

Canobbio, 03.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:15	Nel corso delle prime due ore ho proceduto solamente con l'implementazione della homepage, questo è quanto ho realizzato fino ad ora: Homepage
14:15 - 16:30	Dopo la pausa ho iniziato a creare la pagina di registrazione e di login. Nell'ultima mezz'ora ho dovuto configurare il Web Server per poter utilizzare php nel progetto. Per farlo ho modificato la DocumentRoot in httpd.conf indicando la cartella contenente il codice:

```
DocumentRoot "C:\Users\matti\OneDrive\Desktop\Progetti individuali\Gestione alloggi\Impl  
<Directory "C:\Users\matti\OneDrive\Desktop\Progetti individuali\Gestione alloggi\Impl
```

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto oggi dovrei aver terminato la pagina di login e di registrazione, il database e dovrei essere in fase di realizzazione del menu degli amministratori, tuttavia ho terminato solo il database e un pezzo della homepage.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro proverò a terminare il form di login e di registrazione e se dovesse avanzare del tempo comincerò il menu degli amministratori.

Gestione alloggi | Diario di lavoro - 04.10.2019

Mattia Lazzaroni

Canobbio, 04.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:15	Nel corso delle prime due ore ho lavorato esclusivamente sul front-end del mio sito, in particolare ho terminato la di pagina di registrazione e la pagina di login. Questi sono gli screenshot del risultato finale: Signup Login
14:15 - 16:30	Dopo la pausa quello che ho fatto è stato andare a creare tutta la parte di front-end per la pagina riservata agli amministratori, la pagina per gli amministratori gerenti e la pagina che mostra un errore per mancanza di permessi (pagina che viene utilizzata quando un utente prova ad accedere alle pagine riservate ai vari amministratori).

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto per tutte le pagine dovrei aver già terminato sia il front-end che il back-end. Tuttavia, per ogni pagina realizzata al momento è solo presente il front-end.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro al back-end delle varie pagine.

Gestione alloggi | Diario di lavoro - 08.10.2019

Mattia Lazzaroni

Canobbio, 08.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 16:30	Nella lezione di oggi ho lavorato esclusivamente sul back-end del form di registrazione. In particolare gestisco la validazione dei campi (con i rispettivi messaggi di errore), e salvo ad ogni creazione di utente tutti i dati nel database. Inoltre ho dovuto implementare un meccanismo per far sì che all'invio del form i dati inseriti dall'utente non scompaiano dai campi in caso di errore.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in quanto dovrei aver già terminato il back-end della pagina di login tuttavia devo ancora iniziarlo.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro terminerò un paio di cose per quanto riguarda il form di registrazione e sicuramente inizierò e terminerò la parte di back-end della pagina di login.

Gestione alloggi | Diario di lavoro - 10.10.2019

Mattia Lazzaroni

Canobbio, 10.10.2019

Lavori svolti

Orario	Lavori svolti
13:15	Durante la giornata di lavoro mi sono concentrato sul login tramite gli utenti presenti nel database. Purtroppo, sono incappato in un problema che non ho ancora risolto. Il problema mi si presenta quando provo a loggarmi con un utente presente nel Database.
- 16:30	

Problemi riscontrati e soluzioni adottate

Oggi ho riscontrato un problema nel seguente codice:

```
$query = "SELECT * FROM utenti WHERE email='$email' AND password='$password'";
#echo $query;
$result = mysqli_query($db, $query);
if (mysqli_num_rows($result) == 1) {
    $_SESSION['nome'] = $nome;
    $_SESSION['success'] = "Hai effettuato il login correttamente";
    header('location: index.php');
} else {
    array_push($errors, "Combinazione email/password errata");
}
```

L'errore è il seguente: Warning: mysqli_num_rows() expects parameter 1 to be mysqli_result, boolean given

Ho provato in tutti i modi a sistemarlo ma senza risultati.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro proverò a chiedere al responsabile una mano per sistemare il problema e nel caso che non riuscisse dovrò assolutamente trovare un'alternativa funzionante per il login.

Gestione alloggi | Diario di lavoro - 10.10.2019

Mattia Lazzaroni

Canobbio, 10.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nelle prime due ore il professore Valsangiacomo ci ha esposto la presentazione LPI.
15:00 - 16:30	Dopo la pausa ho lavorato sul codice della registrazione/login. Essenzialmente quello che ho fatto oggi è stato cambiare tutto il codice da mysqli a pdo.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro testerò la connessione al server ftp con le credenziali fornite dal professore responsabile e lavorerò probabilmente sulla pagina dell'amministratore.

Gestione alloggi | Diario di lavoro - 15.10.2019

Mattia Lazzaroni

Canobbio, 15.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Oggi nel corso delle due ore ho lavorato sul login. Inizialmente ho finito di adattare il login e la registrazione con PDO, successivamente ho lavorato per sistemare alcuni problemi che mi impedivano di loggarmi correttamente. Ad esempio nella query per accedere alle informazioni della tabella del Database, andavo a scrivere "utenti" per il nome della tabella, al posto di "utente". Questo mi scatenava un errore che non riuscivo a sistemare a ci ho messo un po' a capire. Inoltre andavo a limitare la lunghezza dell'password nel database ad un varchar(30), in questo modo la hash della password non veniva inserita completamente e al login risultava il problema che le password non coincidevano. Ho quindi messo il campo della password a varchar(255) per non aver problemi.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro sicuramente dovrò aggiungere la conferma della registrazione tramite email e comincerò la pagina dell'amministratore.

Gestione alloggi | Diario di lavoro - 17.10.2019

Mattia Lazzaroni

Canobbio, 17.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nel corso delle prime due ore ho lavorato sui messaggi personalizzati da mostrare all'utente quando completa la registrazione tramite le sessioni e ho implementato un nuovo bottone per permettere all'utente di fare logout.
15:00 - 16:30	Dopo la pausa ho inizialmente creato la pagina "check-your-email.php", che è la pagina in cui viene reindirizzato l'utente quanto completa una registrazione. Dopo aver realizzato il front-end di questa pagina, ho cominciato ad informarmi su internet per come implementare il sistema di verifica tramite email. Ho scoperto di aver bisogno di un nuovo campo nella mia tabella utente (hash). Questo è il modo in cui vado a creare l'hash che sarà sempre univoco:

```
hash = md5(time());
```

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro sicuramente terminerò se possibile la conferma tramite email.

Gestione alloggi | Diario di lavoro - 18.10.2019

Mattia Lazzaroni

Canobbio, 18.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 16:30	Nel corso della lezione di oggi ho inizialmente lavorato sulla validazione dei campi come la password, il nome e il cognome. In seguito mi sono spostato di aula per fare il punto della situazione con il superiore professionale responsabile del mio progetto. Inoltre, mi ha assegnato dei compiti che dovranno essere terminati per venerdì prossimo. Per tutto il resto della giornata ho lavorato sulla conferma della registrazione tramite email. Purtroppo non sono riuscito a far funzionare il metodo "mail()" per mandare le mail, quindi online ho trovato una libreria chiamata "PHPMailer" per poter impostare l'indirizzo di posta e per poter mandare le email agli utenti.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Probabilmente nel weekend proverò la libreria "PHPMailer" per vedere se riesco a mandare l'email. Inoltre creerò un email aziendale da google che sarà poi quella che userò per mandare appunto le email agli utenti.

Gestione alloggi | Diario di lavoro - 22.10.2019

Mattia Lazzaroni

Canobbio, 22.10.2019

Lavori svolti

Durante il weekend ho trovato del tempo per portarmi avanti col progetto: ho installato PHPMailer, l'ho testato e mi sono informato riguardo il funzionamento.

Orario	Lavori svolti
13:15	Nel corso della giornata di oggi sono riuscito a mandare la conferma tramite email tramite la libreria PHPMailer. Per farlo ho creato un nuovo account google con le seguenti credenziali:
- 16:30	gmail = gestionealloggi2019@gmail.com e password = Password&1

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro metterò il sito sul server.

Gestione alloggi | Diario di lavoro - 24.10.2019

Mattia Lazzaroni

Canobbio, 24.10.2019

Lavori svolti

Orario	Lavori svolti
13:15 - 14:45	Nel corso delle prime due ore ho riscontrato problemi con il mio account gmail nell'invio delle email a causa della sicurezza e della verifica dell'account. Avendo cercato una soluzione per più di un'ora ma senza successo, ho deciso di creare un account hotmail (email = gestionealloggi@hotmail.com e password = Password&3), in quanto Outlook non va a creare tutti i problemi relativi alla sicurezza e alla verifica dell'account. Dopo aver creato il nuovo account, andando a modificare qualche parametro nel codice, sono riuscito subito a far funzionare l'invio delle email. Questi sono i nuovi parametri funzionanti:

```
$mail->isSMTP();  
$mail->SMTPDebug = 2;  
$mail->Host      = 'smtp.live.com';  
$mail->SMTPAuth  = true;  
$mail->Username   = 'gestionealloggi@hotmail.com';  
$mail->Password   = 'Password&3';  
$mail->SMTPSecure = 'tsl';    ^ PHPMailer::ENCRYPTION_SMTPS` also  
$mail->Port       = 587;  
$mail->setFrom('gestionealloggi@hotmail.com', 'Gestione alloggi');  
$mail->addAddress($email, $name . ' ' . $surname);  
$mail->addReplyTo('gestionealloggi@hotmail.com', 'Gestione alloggi');
```

Orario	Lavori svolti
15:00 - 16:30	Dopo la pausa invece ho lavorato sulla gestione dell'utente dal momento che clicca il link dalle email. Quando l'utente clicca il link nell'email, nel database il parametro "active" presente nella tabella "utente" viene modificato da "0" a "1" e l'utente può premere un bottone per tornare al login.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro caricicherò il sito sul server e inizierò lo script.

Gestione alloggi | Diario di lavoro - 25.10.2019

Mattia Lazzaroni

Canobbio, 25.10.2019

Lavori svolti

Orario	Lavori svolti
13:15	Nel corso della giornata ho sistemato un problema di PHPMailer, reinstallandolo con Composer.
- 16:30	Successivamente sono riuscito a caricare il sito sul server e ad integrarlo con il database online.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro determinerò cosa fare in base a quanto riuscirò a fare a casa.

Gestione alloggi | Diario di lavoro - 05.11.2019

Mattia Lazzaroni

Canobbio, 05.11.2019

Lavori svolti

| Orario | Lavori svolti | | ----- | ----- || 13:15 - 14:45 | Nel corso delle prime due ore mi sono occupato unicamente di inserire i 20 alloggi nella tabella "alloggio" del database. Per trovare tutti i vari alloggi mi sono appoggiato ai seguenti 3 siti: <https://www.theguardian.com/travel/2018/mar/24/20-best-campsites-europe-camping-greece-france-spain>

<https://www.travelandleisure.com/worlds-best/hotels-top-100-overall>

<https://www.bedandbreakfast.com/info/travelers/awards/amenities-awards/top-luxury-inns>

Orario	Lavori svolti
15:00 - 16:30	Nel corso delle due ore successive ho iniziato a guardare come fare per inserire i dati sul sito dal database e mi sono informato sulla realizzazione di script legati ad un database.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Terminare lo script e iniziare il meccanismo di ricerca degli alloggi.

Gestione alloggi | Diario di lavoro - 07.11.2019

Mattia Lazzaroni

Canobbio, 07.11.2019

Lavori svolti

Orario	Lavori svolti
13:15	Nel corso delle prime due ore ho inizialmente realizzato lo script (.sql) per poter pulire tutte le tabelle e reinserire successivamente tutti gli alloggi. Dopodiché ho iniziato a lavorare sull'inserimento degli alloggi nella homepage. Questo è il risultato ottenuto 10 minuti prima della fine della lezione: Alloggi inseriti (da sistemare)
- 16:30	Come si può notare ci sono ancora delle parti da sistemare per quanto riguarda il front-end. Per queste parti lavorerò nella prossima giornata di lavoro

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro sistemerò lo stile degli alloggi, sistemando i piccoli problemi riscontrati durante l'ultima lezione (problemi di front-end).

Diario di lavoro

Luogo	Canobbio
Data	08.11.19

Lavori svolti

Nella lezione di oggi ho inizialmente sistemato il codice della scorsa lezione che riguarda l'inserimento degli alloggi. Ora gli alloggi vengono mostrati correttamente. In seguito, ho tenuto una discussione con il responsabile del progetto che mi ha assegnato dei nuovi compiti per la settimana prossima. Finita la discussione, ho iniziato a lavorare sul filtro per la tipologia dell'alloggio. Questo è il codice più importante che ho scritto oggi:

```
//Prendo il valore del filtro della tipologia.
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    $value = $_POST["tipologia"];
}

ob_start();
include('server.php');
ob_end_clean();
//Preparo la query per prendere l'id più grande così da sapere il numero di alloggi da stampare.
$get_max_id = "SELECT id FROM alloggio ORDER BY id DESC LIMIT 1";
$stmt = $db->prepare($get_max_id);
$stmt->execute();
$max_id = implode($stmt->fetch(PDO::FETCH_ASSOC));
//Ciclo che viene eseguito in base al numero di alloggi presenti che stampa gli alloggi.
for ($i = 1; $i < $max_id; $i++) {
    //Se viene utilizzato un filtro per la tipologia utilizzo una certa query.
    if(isset($value)){
        $accomodation_query = "SELECT id FROM alloggio WHERE id = $i AND nome_tipologia = '$value' LIMIT 1";
        $stmt = $db->prepare($accomodation_query);
        $stmt->execute();
        $id = implode($stmt->fetch(PDO::FETCH_ASSOC));
        if($id == $i){
            $accomodation_query = "SELECT * FROM alloggio WHERE id = $i LIMIT 1";
        }
        else{
            break;
        }
    }
    //Altrimenti uso la query di "default".
    else{
        $accomodation_query = "SELECT * FROM alloggio WHERE id = $i LIMIT 1";
    }
    $stmt = $db->prepare($accomodation_query);
```

```
//Eseguo la query.  
$stmt->execute();  
$row = $stmt->fetch();
```

Questo codice non produce ancora un filtro funzionante al 100%, in quanto quando vado a selezionare una tipologia di alloggio, mi viene mostrato un errore come si

Cerca un alloggio

Warning: implode(): Argument must be an array in C:\Users\matti\OneDrive\Desktop\Progetti individuali\Gestione alloggi\Implementazione\Bootstrap-4-template-one-column-listing\one-column-listing\index.php on line 193



Hotel Fergus Geminis

Carrer dels Tamarells, s/n, 07600 Palma, Illes Balears, Spain

[MOSTRA DETTAGLI ➔](#)



Alla Galleria B&B

Via Antonio Cantore, 4, 37121 Verona VR, Italy

[MOSTRA DETTAGLI ➔](#)

Warning: implode(): Argument must be an array in C:\Users\matti\OneDrive\Desktop\Progetti individuali\Gestione alloggi\Implementazione\Bootstrap-4-template-one-column-listing\one-column-listing\index.php on line 193



Keswick Camping and Caravanning Club Site

Crow Park Rd, Keswick CA12 5EP, UK

[MOSTRA DETTAGLI ➔](#)

Situazione nella quale sono andato a selezionare tipologia = Bed & Breakfast può vedere qua sotto.

Come si può notare, il primo e il terzo alloggio, che dovrebbero sparire, rimangono con un errore. Il secondo alloggio invece viene mostrato correttamente.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Gestione disponibilità alloggi

2/3

Nella prossima giornata di lavoro sistemerò innanzitutto il problema del filtro della tipologia. Successivamente, lavorerò come richiesto dal responsabile sulla documentazione e sui commenti del codice. Se mi avanzerà del tempo, comincerò ad implementare gli altri filtri (regione, città,...).

Diario di lavoro

Luogo	Canobbio
Data	12.11.19

Lavori svolti

Nella lezione di oggi mi sono occupato dei compiti assegnati dal docente responsabile durante la scorsa lezione. Dunque, per cominciare mi sono recato a commentare tutto il codice scritto fino ad ora. I file con più commenti sono quelli principali e quelli su cui per adesso ho lavorato di più, ovvero: "index.php", "login.php", "signup.php", "server.php" e "verify.php". Finito di commentare il codice, ho sistemato il problema del filtro della volta scorsa. Il problema consisteva in questo: la tipologia può essere di 3 tipi (Albergo, B&B, Camping). Il primo alloggio, di tipologia "Albergo", possiede l'id "1". Quando andavo a inserire un filtro per la tipologia, nel caso che il filtro fosse "Albergo", mi veniva mostrato solo il primo alloggio. Nei casi invece che il filtro fosse "B&B" o "Camping", non mi veniva mostrato nessun alloggio. Per sistemare il problema, mi sono accorto che andavo a controllare se la query "SELECT id FROM alloggio WHERE id = \$i AND nome_tipologia = '\$value' LIMIT 1" producesse un risultato, e nel caso che non lo producesse, andavo ad interrompere completamente il for tramite un "break". Quello che mi serviva era invece un "continue", che mi permettesse di terminare la seguente iterazione del ciclo e passare a quella successiva.

```
//Ciclo che viene eseguito in base al numero di alloggi presenti e che stampa gli alloggi.
for ($i = 1; $i < $max_id + 1; $i++) {
    //Se viene utilizzato un filtro per la tipologia utilizzo una certa query.
    if (isset($value)) {
        $accomodation_query = "SELECT id FROM alloggio WHERE id = $i AND nome_tipologia = '$value' LIMIT 1";
        $stmt = $db->prepare($accomodation_query);
        $stmt->execute();
        //Se eseguendo la query viene trovata una riga, preparo una nuova query.
        if ($stmt->rowCount() > 0) {
            $id = implode($stmt->fetch(PDO::FETCH_ASSOC));
            $accomodation_query = "SELECT * FROM alloggio WHERE id = $id LIMIT 1";
        }
        //Altrimenti interrompo la corrente iterazione e passo alla prossima.
        else {
            continue;
        }
    }
    //Altrimenti uso la query di "default".
    else {
        $accomodation_query = "SELECT * FROM alloggio WHERE id = $i LIMIT 1";
    }
    $stmt = $db->prepare($accomodation_query);
    //Esegue la query.
```

Cambiando il contenuto dell'else, da "break" a "continue", il filtro funziona correttamente.

In seguito, sono andato a creare due nuovi file “credenzialiDatabase.php” e “emailBody.php” che sono i due file che devo modificare ogni volta prima di caricare online il sito. In entrambi i file basta scommentare il codice per lavorare online e commentare quello per lavorare localmente.

Ho anche creato un file “.htaccess” nella DocumentRoot del progetto (ovvero la cartella “one-column-listing”), in cui ho inserito il seguente codice:

```
RewriteEngine On  
RewriteCond %{REQUEST_FILENAME} !-f  
RewriteCond %{REQUEST_FILENAME} !-d  
RewriteRule . /errore.php [L]
```

Questo codice fa sì che se l’utente inserisce un URL inesistente, ad esempio “localhost/aaa.php”, egli viene reindirizzato alla pagina “localhost/errore.php”

Successivamente, sono andato a caricare su GitHub, come richiesto, il file “.mpp” contenente il gantt.

Al termine della lezione, prima di scrivere il diario sono andato a caricare il sito sul server (cosa che devo fare ogni martedì sera).

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro dovrei lavorare sulla documentazione, in particolare sui test case. Inoltre, potrei procedere con l’implementazione dei filtri.

Gestione disponibilità alloggi

2/2

Diario di lavoro

Luogo	Canobbio
Data	14.11.19

Lavori svolti

Nella lezione di oggi ho inizialmente aggiornato il database sul sito <https://h2-phpmyadmin.infomaniak.com/>. Successivamente, volevo modificare due file sul server tramite FileZilla, tuttavia ho riscontrato il problema del solito “Errore: Impossibile collegarsi al server**”, che non sono ancora riuscito a sistemare. Nel corso delle successive due ore ho semplicemente lavorato sulla documentazione, in particolare sui Test Case che riguardano il filtro della tipologia.**

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro dovrei terminare i Test Case e lavorare sulla parte di implementazione della documentazione.

Gestione disponibilità alloggi

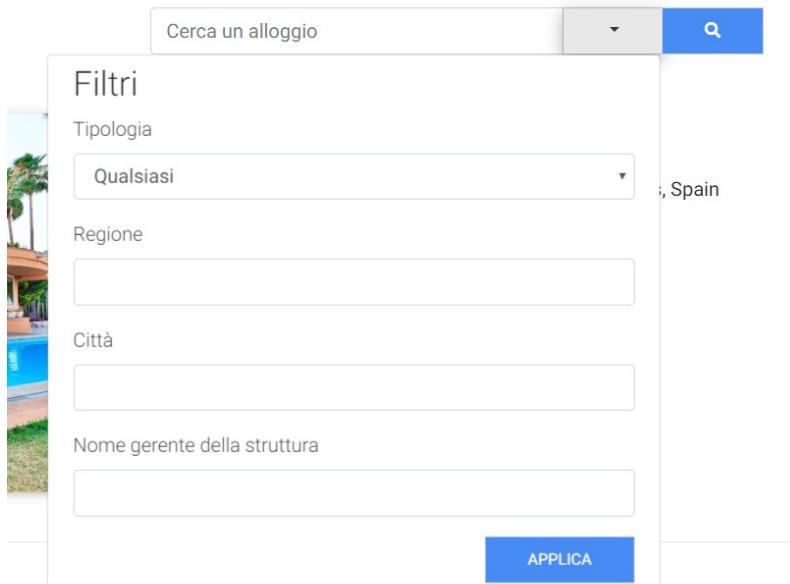
1/1

Diario di lavoro

Luogo	Canobbio
Data	15.11.19

Lavori svolti

Nella lezione di oggi ho tenuto una discussione con il responsabile del mio progetto, che ha controllato i compiti che mi vennero assegnati la settimana scorsa e successivamente mi ha assegnato dei nuovi compiti. Il primo, che ho realizzato



subito, era quello di inserire un bottone per confermare i filtri.
Successivamente, ho dovuto inserire l'impedimento per un utente che non ha confermato l'account di loggarsi nel sito (non ho ancora caricato online questa modifica). Per il resto della lezione ho lavorato sui Test Case che riguardano i filtri.

			
	4. Premere il bottone blu "Applica".		
	5. Premere il bottone blu con la lente.		
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli con la città scelta precedentemente.		
Test Case:	TC-005	Nome:	Verifica del funzionamento del filtro nome gerente della struttura.
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che scegliendo il nome del gerente di un alloggio nella sezione "Filtri", rimangano soltanto gli alloggi inseriti da quel gerente e gli altri spariscano.		
Prerequisiti:	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi inseriti da diversi gerenti.		
Procedura:	6. Aprire con un browser l'url www.samtinfo.ch/gestionealloggi2019 oppure localhost/index.php . 7. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca. 8. Inserire il nome di un gerente nel campo di testo "Nome gerente della struttura". 9. Premere il bottone blu "Applica". 10. Premere il bottone blu con la lente.		
Risultati attesi:	Tutti gli alloggi dovranno sparire, tranne quelli inseriti dal gerente scelto precedentemente.		

4.2 Risultati test

TC-001	NOK	Ancora da terminare l'implementazione.
TC-002	NOK	Ancora da terminare l'implementazione.
TC-003	NOK	Ancora da terminare l'implementazione.
TC-004	NOK	Ancora da terminare l'implementazione.
TC-005	NOK	Ancora da terminare l'implementazione.
TC-???	OK	...

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro lavorerò solamente sui compiti che mi ha assegnato oggi il responsabile del progetto. Uno di questi è quello di togliere il fatto di dover ogni volta commentare e scommentare i due file.

Diario di lavoro

Luogo	Canobbio
Data	19.11.19

Lavori svolti

Inizialmente, come avevo pensato la volta scorsa, ho sistemato il “problema” del locale/online, che mi costringeva ogni volta ad andare a commentare e scommentare del codice per permettere la connessione al database. Ho scoperto dell’array “\$_SERVER”, che contiene le informazioni tipo headers, paths e posizioni degli script. Tramite un semplice if, vado a controllare se l’indirizzo del web server è “127.0.0.1” oppure se il nome è “localhost”. Se una delle due condizioni si avvera, vuol dire che sto aprendo il sito da locale, di conseguenza apro il database locale.

```
//Controllo se sto lavorando in locale.  
if [$_SERVER["SERVER_ADDR"] == '127.0.0.1' || $_SERVER["SERVER_NAME"] == 'localhost']{  
    //Credenziali per lavorare in locale.  
    $servername = "localhost";  
    $username = "root";  
    $password = "";  
    $database = "efof_i16lazmat";  
}  
//Altrimenti sono online.  
else{  
    //Credenziali per lavorare online.  
    $servername = "efof.myd.infomaniak.com";  
    $username = "efof_i16lazmat";  
    $password = "Alloggi_Admin2019";  
    $database = "efof_i16lazmat";  
}
```

Altrimenti mi conetto al database online.

Successivamente, ho perso un po’ di tempo per riuscire a far funzionare il bottone “Applica” del filtro. Per farlo andare, ho aggiunto `onclick="this.form.submit()"` al bottone del filtro (a cui ho dato il name “applica”). In seguito, controllo se il bottone viene premuto e in quel caso vado a impostare i parametri inseriti nel filtro nelle rispettive variabili.

```
//Se viene premuto il bottone applica.  
if (isset($_POST['applica'])) {  
    //Imposto la tipologia selezionata in una variabile se la tipologia non è "Qualsiasi".  
    if ($_POST["tipologia"] != "Qualsiasi") {  
        $tipologia = $_POST["tipologia"];  
    }  
    //Imposto la regione in una variabile se viene inserita dall'utente.  
    if (isset($_POST["regione"])) {  
        $regione = $_POST["regione"];  
    }  
    //Imposto la città in una variabile se viene inserita dall'utente.  
    if (isset($_POST["citta"])) {  
        $citta = $_POST["citta"];  
    }  
    //Imposto il nome del gerente in una variabile se viene inserita dall'utente.  
    if (isset($_POST["nome_gerente"])) {  
        $nome_gerente = $_POST["nome_gerente"];  
    }  
}
```

Per terminare, ho iniziato a scrivere il codice per far sì che il filtro funzioni effettivamente quando l'utente schiaccia la lente. Tuttavia, questo codice non funziona ancora ed è da terminare.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, se non avrò terminato a casa il filtro, lo terminerò.

Per il resto della lezione invece lavorerò solo sulla documentazione.

Gestione disponibilità alloggi

2/2

Diario di lavoro

Luogo	Canobbio
Data	21.11.19

Lavori svolti

Al termine della scorsa lezione pensavo che durante la lezione di oggi sarei riuscito a terminare i filtri e lavorare un po' sulla documentazione. Tuttavia, i filtri mi hanno preso più tempo del previsto e sono riuscito finalmente a terminarli, senza però

```
//Se viene premuto il bottone applica.
if (isset($_POST['applica'])) {
    //Imposto la tipologia selezionata in una variabile se la tipologia non è "Qualsiasi".
    if ($_POST["tipologia"] != "Qualsiasi") {
        $_SESSION['filtro_tipologia'] = $_POST["tipologia"];
    }
    //Imposto la sessione a null se la tipologia selezionata dall'utente è "Qualsiasi".
    else if($_POST["tipologia"] == "Qualsiasi"){
        $_SESSION['filtro_tipologia'] = null;
    }
    //Imposto la regione in una variabile se viene inserita dall'utente.
    if (!empty($_POST["regione"])){
        $_SESSION['filtro_regione'] = $_POST["regione"];
    }
    //Imposto la sessione a null se l'utente non inserisce nessuna regione.
    else if(empty($_POST["regione"])){
        $_SESSION['filtro_regione'] = null;
    }
    //Imposto la città in una variabile se viene inserita dall'utente.
    if (!empty($_POST["citta"])){
        $_SESSION['filtro_citta'] = $_POST["citta"];
    }
    //Imposto la sessione a null se l'utente non inserisce nessuna città.
    else if(empty($_POST["citta"])){
        $_SESSION['filtro_citta'] = null;
    }
    //Imposto il nome del gerente in una variabile se viene inserita dall'utente.
    if (!empty($_POST["nome_gerente"])){
        $_SESSION['filtro_nome_gerente'] = $_POST["nome_gerente"];
    }
    //Imposto la sessione a null se l'utente non inserisce nessun nome.
    else if(empty($_POST["nome_gerente"])){
        $_SESSION['filtro_nome_gerente'] = null;
    }
}
```

lavorare sulla documentazione.

L'immagine qua sopra rappresenta forse il codice più importante scritto oggi. A dipendenza del filtro inserito dall'utente vado ad impostare una sessione. Se invece non viene inserito nessun filtro (per esempio per la città), vado ad impostare a null la sessione (della città).



```
if (isset($_SESSION['filtro_nome_gerente'])) {  
    $get_nome_gerente = "SELECT alloggio.email_gerente FROM alloggio JOIN amministratore_gerente  
    ON alloggio.email_gerente = amministratore_gerente.email WHERE amministratore_gerente.nome = '".$_SESSION['filtro_nome_gerente']."'";  
    $stmt = $db->prepare($get_nome_gerente);  
    $stmt->execute();  
    $row = $stmt->fetch();  
    $email_gerente = $row['email_gerente'];  
    $where .= " AND email_gerente = '" . $email_gerente . "'";  
}  
In questo pezzo di codice, invece, vado a preparare un query con anche un “join” per poter accedere alla tabella “amministratore gerente” e controllare se il nome del gerente che l’utente ha inserito, è presente nel database e coincide con l'email corretta.
```

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Se non dovessi finire i commenti del codice di oggi a casa, li terminerò all'inizio della prossima giornata di lavoro, che per il resto sarà dedicata solamente alla documentazione.

Diario di lavoro

Luogo	Canobbio
Data	22.11.19

Lavori svolti

Nel corso della lezione di oggi ho lavorato sulla barra di ricerca per il nome dell'alloggio. Questa è la parte di html che rappresenta la barra di ricerca, in cui vado a reimpostare il valore ogni qual volta che un form viene inviato.

```
<form action="" role="form" method="post" id="formCerca">
  <input type="text" name="nome" class="form-control" size="4.5em" >?php if (isset($_POST['nome'])) {
    echo 'value=' . $_POST['nome'] . '';
  } else if (isset($_SESSION['nome'])) {
    echo 'value=' . $_SESSION['nome'] . '';
  } >? placeholder="Cerca un alloggio" />
  <button name="cerca" value="cerca" type="submit" /> //Se viene cliccato il tasto di ricerca.
</form>
```

Il codice seguente, invece, serve per la query che va a ricercare un alloggio tramite nome nel caso in cui venga appunto inserito un nome.

Dopo aver fatto ciò, ho tenuto la solita discussione con il responsabile del progetto che mi ha assegnato dei nuovi compiti per settimana prossima.
Nelle restanti due ore ho lavorato un po' sullo stile del sito e un po' sui test case della documentazione.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro svolgerò le attività che mi sono state assegnate dal responsabile.

Gestione disponibilità alloggi

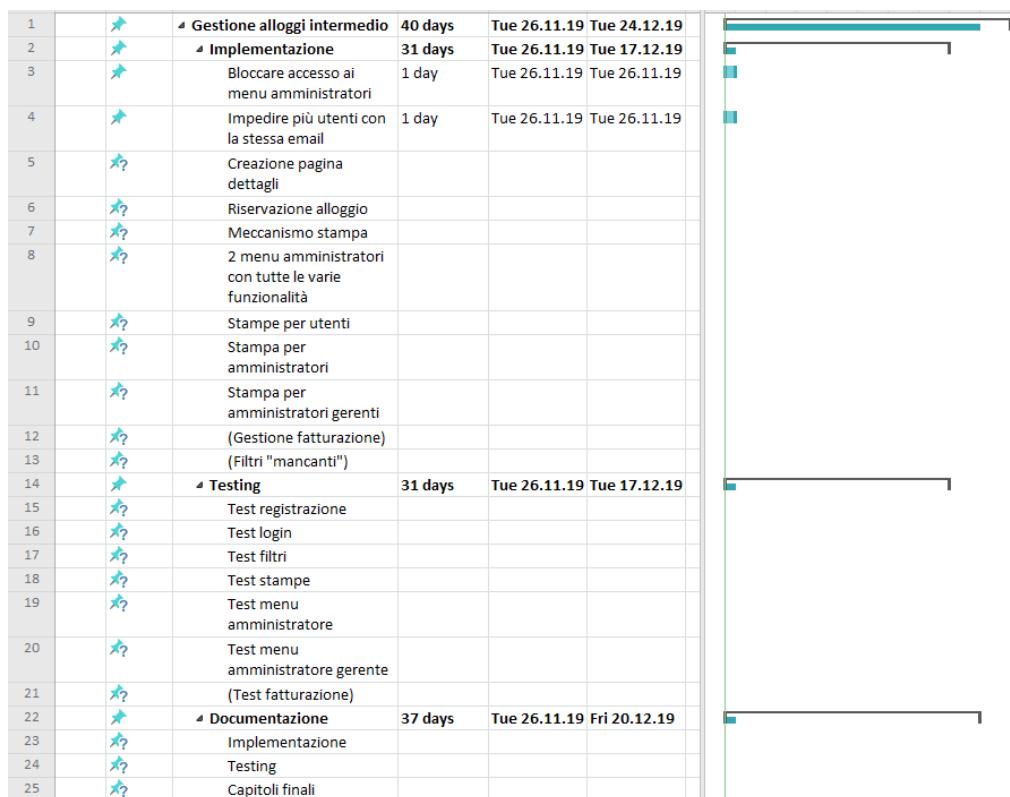
1/1

Diario di lavoro

Luogo	Canobbio
Data	26.11.19

Lavori svolti

Per iniziare, ho cominciato a realizzare la pianificazione intermedia, che non è ancora terminata e che mi occuperò di terminare al termine della giornata in base a quanto avrò fatto oggi durante la lezione. Questo è quanto fatto fino ad ora:



L'attività successiva è stata quella di bloccare l'accesso al menu degli amministratori agli utenti (come si può vedere dalla pianificazione). Ho poi realizzato che per poter mostrare o nascondere i vari bottoni, devo far sì che nel mio database non ci possa essere un utente con l'email di un amministratore. Di conseguenza sono andato a lavorare un po' sul database, definendo che da ora l'amministratore avrà l'email "mattia.lazzaroni@samtrevano.ch" e l'amministratore gerente "mattia.lazza@gmail.com". Nessun utente potrà più registrarsi con queste due email grazie ai controlli che ho inserito in precedenza:

```
//Query per controllare che un utente non esista già con quella email.
$user_check_query = "SELECT * FROM utente WHERE email=:email LIMIT 1";
//Query per controllare che un amministratore non esista già con quella email.
$admin_check_query = "SELECT * FROM amministratore WHERE email=:email LIMIT 1";
//Query per controllare che un amministratore gerente non esista già con quella email.
$admin_manager_check_query = "SELECT * FROM amministratore_gerente WHERE email=:email LIMIT 1";
//Preparo la query.
$stmt1 = $db->prepare($user_check_query);
$stmt2 = $db->prepare($admin_check_query);
$stmt3 = $db->prepare($admin_manager_check_query);
$stmt1->bindParam(":email", $param_email, PDO::PARAM_STR);
```

Gestione disponibilità alloggi

```

$stmt2->bindParam(":email", $param_email, PDO::PARAM_STR);
$stmt3->bindParam(":email", $param_email, PDO::PARAM_STR);
$param_email = trim($email);
//Eseguo la query del controllo degli utente.
$stmt1->execute();
//Eseguo la query del controllo degli amministratore.
$stmt2->execute();
//Eseguo la query del controllo degli amministratore gerente.
$stmt3->execute();
//Se la query trova un utente, un amministratore o un amministratore gerente con quell'email stampo un errore

if ($stmt1->rowCount() == 1 || $stmt2->rowCount() == 1 || $stmt3->rowCount() == 1) {
    array_push($errors, "La seguente email è già in uso");
}

```

Dato che stavo già lavorando sul database mi è tornato in mente che nella schermata del dettaglio di un alloggio deve venire mostrata una mappa con la posizione dell'alloggio. Di conseguenza, ho aggiunto una colonna nella tabella

`select * from alloggio;`

	link_immagine	link_mappa
a, Illes B...	https://www.fergushotels.com/backoffice/imag...	https://www.google.com/maps/embed?pb=!1m...
VR, Italy	http://alla-galleria-bb.verona-hotel.org/data/Ph...	https://www.google.com/maps/embed?pb=!1m...
	https://mb.cision.com/Public/6265/9664585/b6...	https://www.google.com/maps/embed?pb=!1m...
Chengdu...	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
Iletti, Fas...	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
United States	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
ited States	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
, Greece	https://cdn-image.travelandleisure.com/sites/d...	https://www.google.com/maps/embed?pb=!1m...
510, Unit...	https://www.bedandbreakfast.com/files/live/sit...	https://www.google.com/maps/embed?pb=!1m...
ited States	https://www.bedandbreakfast.com/files/live/sit...	https://www.google.com/maps/embed?pb=!1m...

“alloggio”:

Per recuperare tutti i link per le mappe mi sono recato sul sito

[“https://www.google.com/maps/”.](https://www.google.com/maps/)

Ho poi scritto il codice per poter nascondere i bottoni agli utenti:

```

<ul class="navbar-nav mr-auto" style="margin-bottom:5px;">
    <li class="nav-item active">
        <a class="nav-link waves-effect" href="#">Tutte le strutture
            <span class="sr-only">(current)</span>
        </a>
    </li>
    <?php
        if(isset($_SESSION['amministratore']) || isset($_SESSION['amministratore_gerente'])) :
    ?>
    <li class="nav-item">
        <a class="nav-link waves-effect" href="amministratore-gerente.php">Amministratore gerente</a>
    </li>
    <?php
        elseif(isset($_SESSION['amministratore'])) :
    ?>

```

Gestione disponibilità alloggi

```
<li class="nav-item">
  <a class="nav-link waves-effect" href="amministratore.php">Amministratore</a>
</li>
<?php endif; ?>
</ul>
```

Questo codice guarda se vengono settate delle sessioni prima di mostrare i vari bottoni. Le sessioni vengono settate al momento del login, a dipendenza dell'email che si inserisce e quindi a dipendenza del tipo di utente che esegue login (utente normale -> nessuna sessione, amministratore -> `$_SESSION['amministratore']`, amministratore gerente -> `$_SESSION['amministratore_gerente']`). Queste sessioni verranno poi utilizzate nelle varie pagine degli amministratori per controllare che sia stato realizzato effettivamente il login correttamente e non che un utente abbia provato ad accedere inserendo il link della pagina a mano.

Per terminare la giornata di lavoro ho implementato la parte nel login che va a controllare il tipo di utente dell'email inserita. Ora a dipendenza dell'email che si usa per loggarsi vengono mostrati o nascosti i bottoni per recarsi nelle pagine degli amministratori.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in linea rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

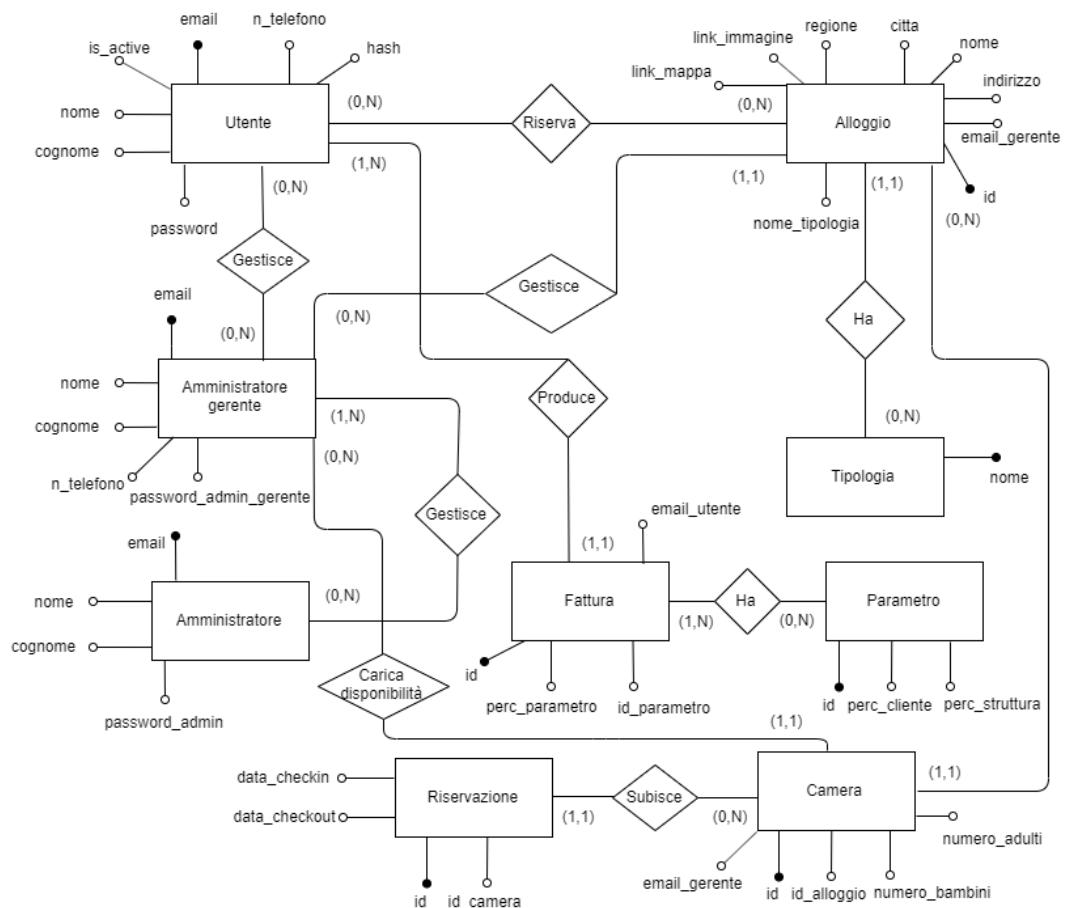
Nella prossima giornata di lavoro implementerò il fatto che un utente qualsiasi non può accedere alle pagine degli amministratori tramite il link diretto, poi mi recherò ad aggiornare il design del database e infine continuerò un po' di test case sulla doc.

Diario di lavoro

Luogo	Canobbio
Data	28.11.19

Lavori svolti

Per iniziare, ho cominciato a sistemare il design del database in base a quanto discusso con il responsabile del progetto:



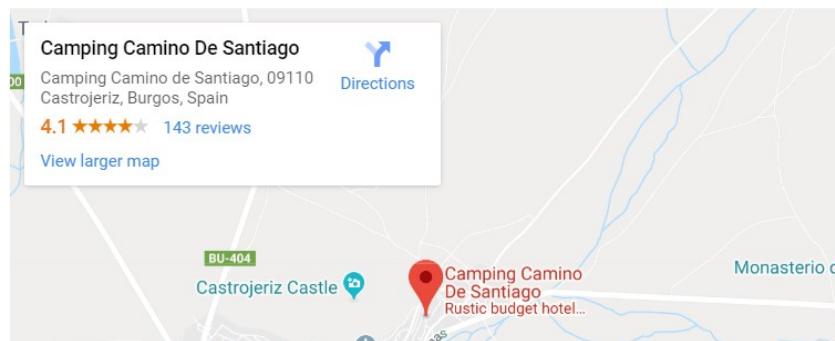
Ho poi cambiato il database da MySQL Workbench in base al design qua sopra. Dopo aver fatto ciò, ho iniziato a lavorare sulla pagina di dettaglio di un alloggio, che per ora mostra le solite informazioni più la mappa dell'alloggio:



Camping Camino de Santiago

Camping Camino de Santiago, 09110 Castrojeriz, Burgos, Spain

Burgos, Castile and Leon



Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione, in linea rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro implementerò il fatto che un utente qualsiasi non può accedere alle pagine degli amministratori tramite il link diretto, poi mi recherò ad aggiornare il design del database e infine continuerò un po' di test case sulla doc.

Gestione disponibilità alloggi

2/2

Diario di lavoro

Luogo	Canobbio
Data	29.11.19

Lavori svolti

Oggi per quasi tutta la durata della lezione, ho lavorato sulla scrittura dei test case e dei risultati di alcuni di essi. Questo è quanto ho fin'ora:

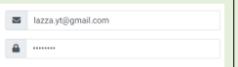
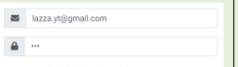
Test Case:	TC-009	Nome:	Verifica del funzionamento della ricerca tramite nome
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che inserendo un nome nella barra di ricerca, vengano mostrati solo gli alloggi che contengono una parte del nome inserito.		
Prerequisiti:			
Procedura:	1.		
Risultati attesi:			

Test Case:	TC-010	Nome:	Verifica del funzionamento dei filtri combinati
Riferimento:	REQ-004		
Descrizione:	Sarà necessario verificare che scegliendo più filtri, essi continuino a funzionare.		
Prerequisiti:			
Procedura:	2.		
Risultati attesi:			

4.2 Risultati test

Test Case	Funzionamento	Commento	Data
TC-001	OK	Si riesce correttamente a registrarsi, l'email arriva, il link attiva l'account e si riesce a fare il login.	29.11.19
TC-002	OK	Ho provato ad inserire il nome "Mattia123", l'email "mattia.lazzaroni@antevane.ch", il numero "0764650110", la password "prova123", la seconda password "prova12" e ho lasciato il cognome vuoto. Il risultato è stato che sono stati mostrati i seguenti errori: Il nome non può contenere valori numerici Il cognome è richiesto Il formato dell'email non è valido Il formato per il numero di telefono non è valido Le due password non coincidono	29.11.19
TC-003	OK	Credenziali presenti nel database: "lazza.yt@gmail.com:prova123".	29.11.19

Inizialmente nel form di login ho inserito "test@gmail.com" e "prova123". Questo è il risultato:

Successivamente, ho provato con l'email presente nel database ma con la password errata:

Infine, ho inserito le credenziali corrette dell'utente non attivo:


Come è possibile notare, ci sono ancora due Test Case (009 e 010) che sono da scrivere. Inoltre, tutti i test dal "004" al "010", sono ancora da testare. Mi occuperò di terminare queste due mancanze appena citate nel weekend.

Lavorando sui Test Case del form di registrazione, mi sono accorto che c'erano dei controlli non completamente corretti sui dati inseriti dall'utente (formato email e password) che sono andato a sistemare velocemente.

Nell'ultima mezz'ora rimanente, ho lavorato sul mostrare correttamente tutti gli elementi della pagina di dettagli. In precedenza, avevo il problema che aprendo la pagina di dettagli di un qualsiasi alloggio, mi venivano mostrati i dettagli dell'ultimissimo alloggio presente. Oggi sono riuscito (in un modo non molto



Hotel Fergus Geminis

Camping Camino de Santiago, 09110 Castrojeriz, Burgos, Spain

Burgos, Castile and Leon

ottimizzato) a stampare il nome dell'alloggio cliccato.

Per quanto riguarda l'ottimizzazione del codice e il funzionamento non solo del nome, ma anche dell'immagine, dell'indirizzo, ecc. lavorerò a casa questo weekend.

Gestione disponibilità alloggi

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In leggero ritardo rispetto alla pianificazione intermedia, ma mi rimetterò in linea senza problemi durante il weekend.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro comincerò a lavorare sulla riservazione di un alloggio.

Diario di lavoro

Luogo	Canobbio
Data	03.12.19

Lavori svolti

Nella giornata di oggi ho lavorato unicamente sulla validazione dei 4 campi che inserisce l'utente (data check-in, data check-out, numero bambini, numero adulti). Questo lavoro mi ha impiegato tutta la giornata in quanto inizialmente avevo dei problemi a stampare gli errori correttamente nella pagina.

I controlli principali che ho implementato sono i seguenti:

	21-02-2019	23-04-2020
	Inserisci il numero di adulti	
	2	
CERCA DISPONIBILITÀ		

Non puoi riservare nel passato

Il numero di adulti è richiesto (in caso di assenza, inserire "0")

Il numero di adulti è richiesto (in caso di assenza, inserire "0")

In questa prima immagine si può vedere che vado a controllare la data del check-in non sia nel passato e che venga inserito il numero di adulti.

In questa seconda immagine invece vado a controllare che la data del check-in sia prima della data del check-out e anche qui che venga inserito il numero di adulti.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia, ma mi rimetterò in linea senza problemi durante il weekend.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, lavorerò sulla riservazione di un alloggio.

Gestione disponibilità alloggi

1/1

Diario di lavoro

Luogo	Canobbio
Data	05.12.19

Lavori svolti

Oggi la giornata di lavoro è saltata in quanto è stata effettuata la visita alla SUP di Friborgo. Tuttavia, sul bus per il ritorno ho iniziato a lavorare un po' sulla riservazione degli alloggi, provando a scrivere la query che si occupa di cercare le camere disponibili con un certo numero di bambini e adulti e disponibili in una certa data.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, continuerò a lavorare sulla riservazione di un alloggio.

Gestione disponibilità alloggi

1/1

Diario di lavoro

Luogo	Canobbio
Data	06.12.19

Lavori svolti

Nella giornata di oggi ho inizialmente sistemato un po' l'aspetto grafico del sito. Più tardi, il responsabile del progetto mi ha chiamato e abbiamo tenuto la solita discussione, in cui mi ha spiegato come realizzare il meccanismo di stampa delle camere libere. Dopo di questo, ho iniziato a realizzare una tabella di prova in cui stampare poi le informazioni della camera (in questo caso vado a stampare tutte le

Per quella data sono disponibili le seguenti camere:

Numero	Adulti	Bambini	Gerente	Riserva
1	0	2	mattia.lazza@gmail.com	<input type="checkbox"/>
2	1	1	mattia.lazza@gmail.com	<input type="checkbox"/>
3	0	1	mattia.lazza@gmail.com	<input type="checkbox"/>
4	1	2	mattia.lazza@gmail.com	<input type="checkbox"/>
5	2	2	mattia.lazza@gmail.com	<input type="checkbox"/>

(camere esistenti):

Per il resto della giornata, ho lavorato solamente sulla query per riuscire correttamente a vedere le camere libere. La parte di riservazione di un alloggio, dunque, non è ancora finita e la finirò nel weekend.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, continuerò a lavorare in base alla pianificazione intermedia.

Gestione disponibilità alloggi

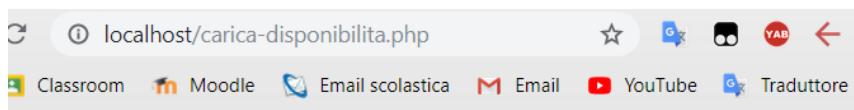
1/1

Diario di lavoro

Luogo	Canobbio
Data	10.12.19

Lavori svolti

Durante il weekend ho provato a terminare la parte di riservazione in quanto la query essendo abbastanza complicata non mi funziona e non riesco a sistemarla in alcun modo. Di conseguenza, avendo già perso tanto tempo a caso, oggi a lezione ho deciso di mettere questa attività da parte e di procedere con altro. Il passo successivo era quello di realizzare le due pagine di gestione degli amministratori. Prima di fare questo però, ho lavorato 1 oretta sulla documentazione, più specificatamente nella parte di implementazione (che avevo già iniziato durante il weekend). Dopodiché, ho iniziato a lavorare sulla pagina di gestione dell'amministratore gerente. Egli dispone 2 funzionalità: caricamento di una camera e gestione dei clienti. Quello che avevo iniziato a fare era mettere tutto in una pagina, andando a nascondere o mostrare dei form in base a quale dei due button veniva cliccato. Tuttavia, ho iniziato a riscontrare dei problemi con i vari form, di



Aggiungi una camera

Inserisci il numero di adulti

Inserisci il numero di bambini

Inserisci il nome del tuo alloggio

AGGIUNGI

conseguenza ho deciso di creare 2 nuovi file per semplificare il tutto. Un file per il caricamento di una camera e uno per la gestione dei clienti. Questo è quanto ho realizzato finora per la parte dell'aggiunta di una camera. Stasera a casa andrò avanti, lavorando sulla query che va ad aggiungere la camera nel database.

Problemi riscontrati e soluzioni adottate

Problema con i form. Citato nella parte dei lavori svolti.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Gestione disponibilità alloggi

1/2

Nella prossima giornata di lavoro, continuerò a lavorare sulle pagine degli amministratori.

Diario di lavoro

Luogo	Canobbio
Data	12.12.19

Lavori svolti

Nella lezione di oggi ho lavorato unicamente sulla pagina di gestione degli amministratori gerenti da parte degli amministratori. Questo è quello che ho fatto

Gestione disponibilità alloggi

Tutte le strutture Amministratore

Gestisci gli amministratori gerenti

<input type="text"/>	Inserisci l'email del gerente
<input type="text"/>	Inserisci il nome del gerente
<input type="text"/>	Inserisci il cognome del gerente
<input type="text"/>	Crea la password del gerente
<input type="text"/>	Ripeti la password del gerente
<input type="text"/>	Inserisci il numero del gerente
AGGIUNGI / SALVA MODIFICHE	

Email	Nome	Cognome	Numero di telefono	Modifica	Elimina
mattia.lazza@gmail.com	Mattia	Lazzaroni	+41764650110	MODIFICA	ELIMINA
prova1@gmail.com	Prova1	Prova	+123456789	MODIFICA	ELIMINA
 amministratore-gerente.php			12.12.2019 16:09	File PHP	5 KB
 amministratore.php			12.12.2019 16:09	File PHP	5 KB
 carica-disponibilita.php			12.12.2019 16:24	File PHP	10 KB
 gestisci-clienti.php			12.12.2019 16:25	File PHP	9 KB
 carica-strutture.php			12.12.2019 16:25	File PHP	13 KB
 gestisci-gerenti.php			12.12.2019 16:28	File PHP	15 KB

nati 45.1 KB

finora:

È possibile notare un form in cui inserire le informazioni del nuovo amministratore gerente da aggiungere e una tabella contenente tutti gli amministratori già esistenti. Da questa tabella è possibile modificare oppure eliminare un amministratore di quelli già presenti. Nel caso che si decidesse di modificarne uno, tutte le sue informazioni verrebbero riportate nel form di inserimento da cui sarà poi possibile modificare le informazioni e salvarle nel database. Nel corso degli ultimi 15 minuti ho aggiornato nella documentazione la prima parte riguardante la struttura della mia cartella, andando ad aggiungere i quattro file per i vari menu degli amministratori. Questi sono i 4 file nuovi:

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Gestione disponibilità alloggi

1/2

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, continuerò a lavorare sulle pagine degli amministratori.

Diario di lavoro

Luogo	Canobbio
Data	13.12.19

Lavori svolti

Nella lezione di oggi ho lavorato unicamente sulla pagina di gestione degli amministratori gerenti da parte degli amministratori. Ora è possibile eliminare correttamente un amministratore gerente, modificarne uno esistente o aggiungerne un nuovo. Per riuscire a terminare questa pagina ho perso molto tempo, in quanto non riuscivo a passare correttamente le informazioni dell'amministratore che veniva cliccato e di conseguenza riuscivo a eliminare o modificare soltanto l'ultimo amministratore presente nella tabella. Per sistemare questa cosa ho aggiunto nel "value" dei due button un identificatore che dipende dal ciclo che cicla tutti gli amministratori.

```
for ($i = 1; $i < $stmt->rowCount() + 1; $i++) {
    //${"id" . $i} = $row[0];
    //${"email_gerente" . $i} = $row[1];
    $row = $stmt->fetch(PDO::FETCH_NUM);
    echo "<tr>
        <td><input type='hidden' name='email' value="" . $i . "">" .
            $row[0] .
        "</td>
        <td><input type='hidden' name='nome' value="" . $i . "">" .
            $row[1] .
        "</td>
        <td><input type='hidden' name='cognome' value="" . $i . "">" .
            $row[2] .
        "</td><input type='hidden' name='password' value="" . $row[3] . "">
        <td><input type='hidden' name='telefono' value="" . $i . "">" .
            $row[4] .
        "</td>
        <td>
            <button id='modifica' value="" . $i . "" type='submit' name='modifica' class='btn btn-info btn-block'>Modifica</button>
        </td>
        <td>
            <button id='elimina' value="" . $i . "" type='submit' name='elimina' class='btn btn-danger btn-block'>Elimina</button>
        </td>
    </tr>";
}
```

Sistemata questa faccenda, ho iniziato a implementare la pagina in cui gli amministratori gerenti possono modificare un utente.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, continuerò a lavorare in base alla pianificazione intermedia.

Diario di lavoro

Luogo	Canobbio
Data	17.12.19

Lavori svolti

A casa durante il weekend ho terminato l'implementazione. Questo mi ha permesso oggi di sistemare alcuni commenti nel codice, sistemare alcuni piccoli bug, inserire dei leggeri miglioramenti grafici e per tutto il resto del tempo lavorare sulla documentazione, più nello specifico sulla parte di implementazione.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, continuerò a lavorare unicamente sulla doc, probabilmente realizzando i test case mancanti e il gantt consuntivo.

Gestione disponibilità alloggi

1/1

Diario di lavoro

Luogo	Canobbio
Data	19.12.19

Lavori svolti

Durante la giornata di oggi ho lavorato unicamente sulla documentazione. In particolare, ho terminato il Gantt consuntivo e l'ho aggiunto alla doc con tanto di descrizione. Ho anche aggiunto la pianificazione intermedia con la rispettiva spiegazione. Ho poi terminato di scrivere l'abstract e ho terminato tutti test con i vari risultati. Alla fine della giornata ho scritto i capitoli conclusivi della documentazione, quindi le mancanze/limitazioni conosciuti, le conclusioni, gli sviluppi futuri, le considerazioni personali e gli allegati.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In leggero ritardo rispetto alla pianificazione. In orario rispetto alla pianificazione intermedia.

Programma di massima per la prossima giornata di lavoro

Nella prossima giornata di lavoro, se sarà necessario terminerò l'implementazione sulla documentazione. Passerò poi alla rilegatura della documentazione e alla consegna del progetto. Se mi dovesse avanzare del tempo potrei iniziare la presentazione PPT.

Gestione disponibilità alloggi

1/1

Diario di lavoro

Luogo	Canobbio
Data	20.12.19

Lavori svolti

Nella giornata di oggi ho inizialmente dato una rilettura veloce alla documentazione. Dopo aver fatto ciò, sono andato a stampare tutti i fogli necessari per la consegna. Al termine della consegna ho iniziato a realizzare la presentazione power point che consegnerò nei prossimi giorni.

Problemi riscontrati e soluzioni adottate

Nessun problema riscontrato.

Punto della situazione rispetto alla pianificazione

In orario rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

-

Gestione disponibilità alloggi

1/1

Parametri per l'accesso all'hosting samtinfo per il progetto:

FTP:

- Host
 - efof.ftp.infomaniak.com port: 21
- User
 - efof_i16lazmat
- Pass
 - Alloggi_Admin_2019
- Path
 - /users/gestionealloggi2019

MySql:

- Server
 - efof.myd.infomaniak.com
- User
 - efof_i16lazmat
- Pass
 - Alloggi_Admin2019
- Database
 - efof_i16lazmat

phpmyadmin:

- <https://h2-phpmyadmin.infomaniak.ch/MySQLAdmin/>

URL:

- <http://samtinfo.ch/gestionealloggi2019>

Versione PHP: 7.2

Credenziali utenti:

- Amministratore → mattia.lazzaroni@samtrevano.ch:prova123
- Amministratore gerente → [mattia.lazza@gmail.com:prova123](mailto:mattia.lazza@gmail.com)
- Utente normale → Crearlo tramite il form di registrazione, oppure → lazza.yt@gmail.com:prova123

1 INFORMAZIONI GENERALI

Candidato	Nome:	Cognome:
	 @samtrevano.ch	
Luogo di lavoro	Scuola Arti e Mestieri / CPT Trevano-Canobbio	
Orientamento	<input type="checkbox"/> 88601 Sviluppo di applicazioni <input checked="" type="checkbox"/> 88602 Informatica aziendale <input type="checkbox"/> 88603 Tecnica dei sistemi	
Superiore professionale	Nome: Guido  guido.montalbetti@edu.ti.ch	Cognome: Montalbetti 
Perito 1	Nome: 	Cognome: 
Perito 2	Nome: 	Cognome: 
Periodo	3 settembre 2019 – 20 dicembre 2019 (presentazioni: 7-17 gennaio 2020)	
Orario di lavoro	Secondo orario scolastico 1° semestre	
Numero di ore	174	
Pianificazione 80h (in ore o %)	Analisi: 10% Implementazione: 50% Test: 10% Documentazione: 30%	

2 PROCEDURA

- Il candidato realizza il lavoro autonomamente sulla base del quaderno dei compiti ricevuto il 1 ° giorno.
- Il quaderno dei compiti è approvato dai periti. È anche presentato, commentato e discusso con il candidato. Con la sua firma, il candidato accetta il lavoro proposto.
- Il candidato ha conoscenza della scheda di valutazione prima di iniziare il lavoro.
- Il candidato è responsabile dei suoi dati.
- In caso di problemi gravi, il candidato o il superiore professionale avvertono immediatamente il perito.
- Il candidato ha la possibilità di chiedere aiuto, ma deve menzionarlo nella documentazione.
- Alla fine del tempo a disposizione per la realizzazione del LPI, il candidato deve inviare via e-mail il progetto al superiore professionale e al perito 1. In parallelo, una copia cartacea della documentazione dovrà essere fornita in duplice copia (superiore professionale e perito). Quest'ultima deve essere in tutto identica alla versione elettronica.

3 TITOLO

Gestione disponibilità alloggi

4 HARDWARE E SOFTWARE DISPONIBILE

1 PC fornito dalla scuola con gli strumenti necessari per lo svolgimento del progetto (Apache, MySql, PHP, ecc.)

1 Accesso presso l'hosting interno messo a disposizione dalla scuola per caricare il progetto

5 PREREQUISITI

Conoscenze per lo sviluppo di gestionali Web, acquisite presso la SAMT

6 DESCRIZIONE DEL PROGETTO

Il progetto vuole creare un sito Web dove poter gestire la disponibilità degli alloggi.

Le persone potranno ricercare la disponibilità degli alloggi per un determinato giorno, riservare e stampare la loro riservazione.

Specifiche dell'applicativo:

- Il sito Web dovrà avere una parte del menu riservata agli amministratori del sito, che potranno caricare le strutture legate al sito e gestire i gerenti delle strutture
- Vi sarà un secondo livello di amministratori (gerenti strutture) che con un menu riservato, potranno caricare le disponibilità della struttura e gestire i clienti che avranno riservato gli alloggi
- I clienti finali potranno in primo luogo ricercare per:
 - Regione
 - Città
 - Tipologia (Albergo, Camping, B&B, ...)
 - Nome gerente della struttura
- La lista risultato fornirà le descrizioni essenziali delle strutture per permettere ai clienti di scegliere e selezionare una struttura dalla lista
- Dalla scelta della struttura vi sarà una schermata di dettaglio della stessa
 - Da un lato sarà evidenziata una mappa del punto preciso del luogo dell'alloggio scelto
 - In seconda battuta vi sarà la possibilità di inserire le date (obbligatorio) e il numero di persone (facoltativo) che necessitano dell'alloggio
 - Il risultato della disponibilità potrà essere filtrato e ordinato sulla quantità delle persone e sulla tipologia dell'alloggio
 - Inoltre, la lista potrà essere ordinata per il prezzo e/o la tipologia
- Per la riservazione di un alloggio, le persone dovranno registrarsi al sito ed inserire i propri dati principali. I campi minimi da compilare saranno: Nome, Cognome, email e un recapito telefonico
- L'applicativo dovrà prevedere un meccanismo di conferma della registrazione per l'identificazione della persona, p.es. l'invio di un link all'indirizzo di posta elettronica dalla persona impostato in fase di registrazione
- Al fronte di una riservazione, i clienti riceveranno una conferma tramite email

Stampe

- Le persone che avranno riservato l'alloggio, potranno stampare la loro riservazione con tutti i dettagli essenziali
- Gli amministratori del sito potranno:
 - Stampare la fatturazione per un determinato periodo e per una specifica struttura
- Gli amministratori, gerenti, potranno:
 - Stampare le riservazioni per un determinato periodo e/o persona

Gestione fatturazione

- Il sito deve prevedere di poter gestire la fatturazione delle strutture
- Il sito guadagna il 15% su ogni riservazione eseguita tramite il suo "portale", la suddivisione dei costi viene ripartita in questo modo:
 - Il 10% viene pagato dal cliente finale
 - Il 5% dalla struttura - alloggi
- Vi deve essere la possibilità di modificare le percentuali percepite dal sito in modo celere, quindi i valori devono essere salvati nella banca dati
 - Vi possono essere delle strutture che pagano meno rispetto ad altre, questo a dipendenza del volume degli affari

IMPORTANTE

- La convalida dei dati immessi è una funzionalità dell'applicativo, p.es. l'email e il numero di telefono devono avere un formato corretto
- La fase di test deve essere preparata anticipatamente all'inizio dello sviluppo dell'applicativo; i "Test Case" fanno da garanti al corretto funzionamento del software prodotto
- Ogni volta che i test case saranno eseguiti, si dovranno protocollare, indicando il loro esito in modo esaustivo, un semplice "OK, NOK" non sarà sufficiente
- Prevedere un meccanismo di concorrenzialità per evitare che una persona in procinto di riservare l'alloggio, si veda portar via la sua riservazione da un'altra persona
- **L'applicativo dovrà essere messo a disposizione sul sito Web dopo un mese dall'inizio del progetto e dovrà essere aggiornato settimanalmente**
- Nel caso in cui, tutte le specifiche siano state coperte in modo completo, compresa la fase dei "Test Case", il docente si riserva la facoltà di aggiungere all'applicativo delle nuove funzionalità

7 RISULTATI FINALI

L'allievo è responsabile della consegna al docente responsabile di:

- Una pianificazione iniziale (entro la prima settimana)
- Una documentazione del progetto e su un DVD, tutti i file del progetto (Documentazione, file sorgenti, pianificazioni, ...)
- Un diario di lavoro
- Implementazione dell'Applicativo Web

8 PUNTI TECNICI SPECIFICI VALUTATI

La griglia di valutazione definisce i criteri generali secondo cui il lavoro dell'allievo sarà valutato (documentazione, diario, rispetto degli standard, della qualità, ...).

Inoltre, il lavoro sarà valutato sui seguenti 7 punti specifici (punti da A14 a A20):

- 135 – Documentazione DB, tabelle, ecc.
- 237 – Analisi della sicurezza (Applicazione Web)
- 240 – Sicurezza di base di dati
- 148 – Solidità verifica dei dati dei dati, intercettazione degli errori di inserimento
- 193 – Design del GUI
- 254 – Responsive Web Design
- 232 – Programmazione web professionale

9 FIRMA

Candidato

Canobbio, 03.09.2019

Superiore professionale

Canobbio, 03.09.2019

Perito 1

(luogo e data)

Perito 2

(luogo e data)