

## **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	
<b>Nome</b>	Verifica dell'hosting
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	È necessario verificare che l'hosting fornito dalla scuola funzioni correttamente

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

ID: REQ-003	
<b>Nome</b>	Creazione menu amministratori gerenti

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

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

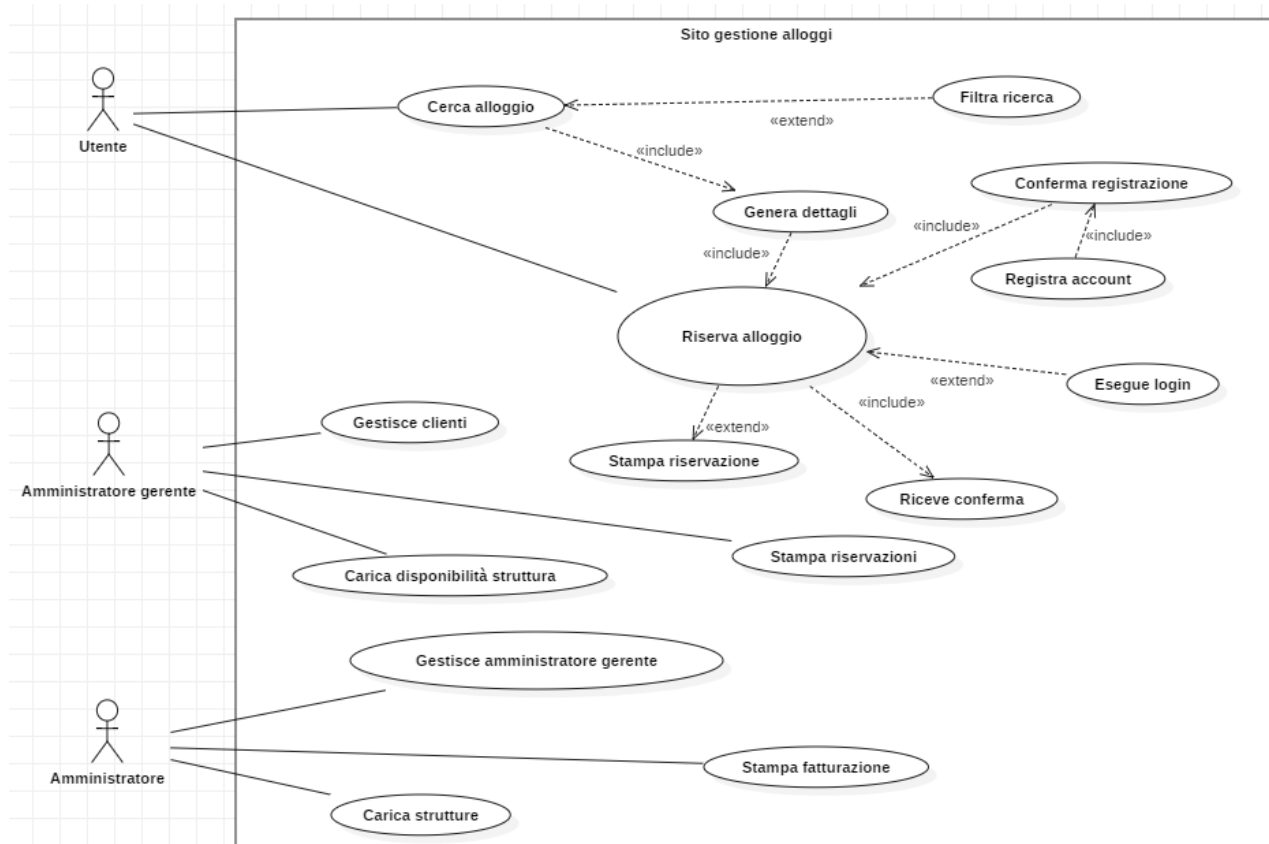
<b>ID: REQ-005</b>	
<b>Nome</b>	Possibilità di svolgere una prenotazione
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	L'utente avrà la possibilità di svolgere delle prenotazioni
<b>Sotto requisiti</b>	
<b>001</b>	Sarà obbligatorio registrarsi per poter completare una prenotazione
<b>002</b>	Bisognerà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

<b>ID: REQ-006</b>	
<b>Nome</b>	Gestire il sistema di stampa

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

<b>ID: REQ-007</b>	
<b>Nome</b>	Gestire le fatturazioni
<b>Priorità</b>	2
<b>Versione</b>	1.0
<b>Note</b>	Il sito deve prevedere di poter gestire la fatturazione delle strutture
<b>Sotto requisiti</b>	
<b>001</b>	Il sito guadagna tramite le riservazioni
<b>002</b>	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:

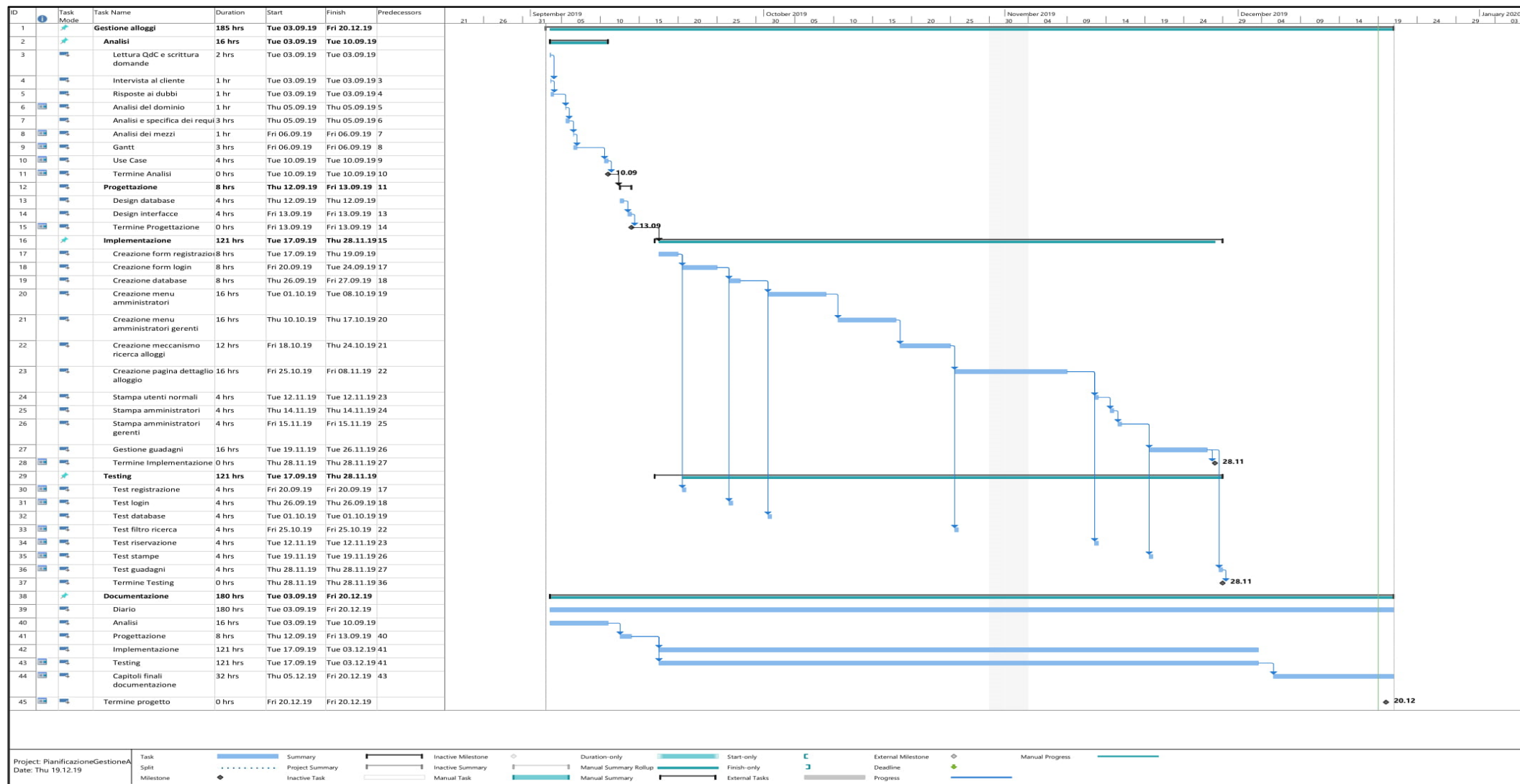


Diagramma di Gantt rappresentante la pianificazione del progetto



#### **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.

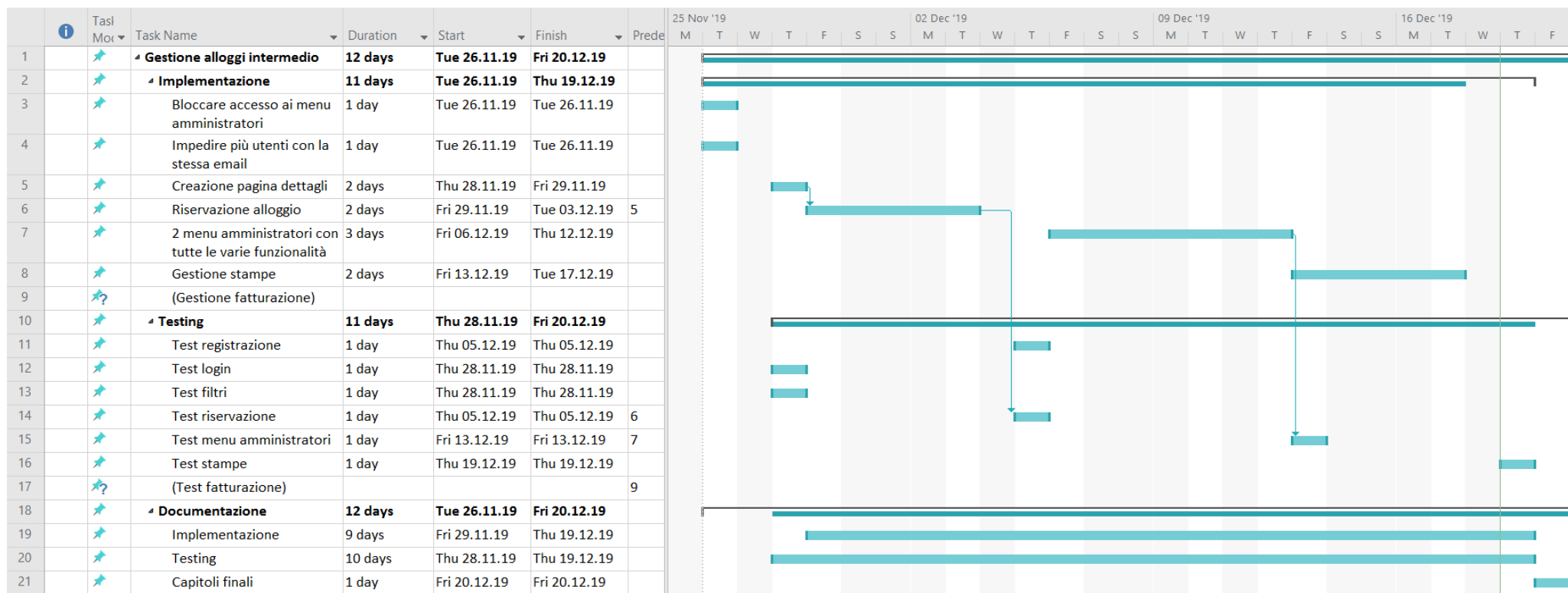


Diagramma di Gantt rappresentante la pianificazione intermedia del progetto

## **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

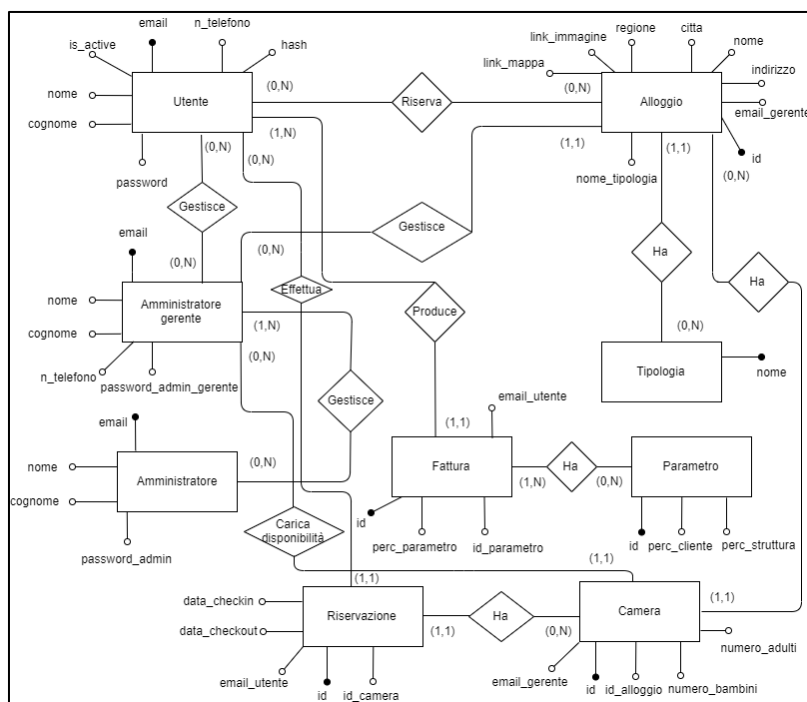
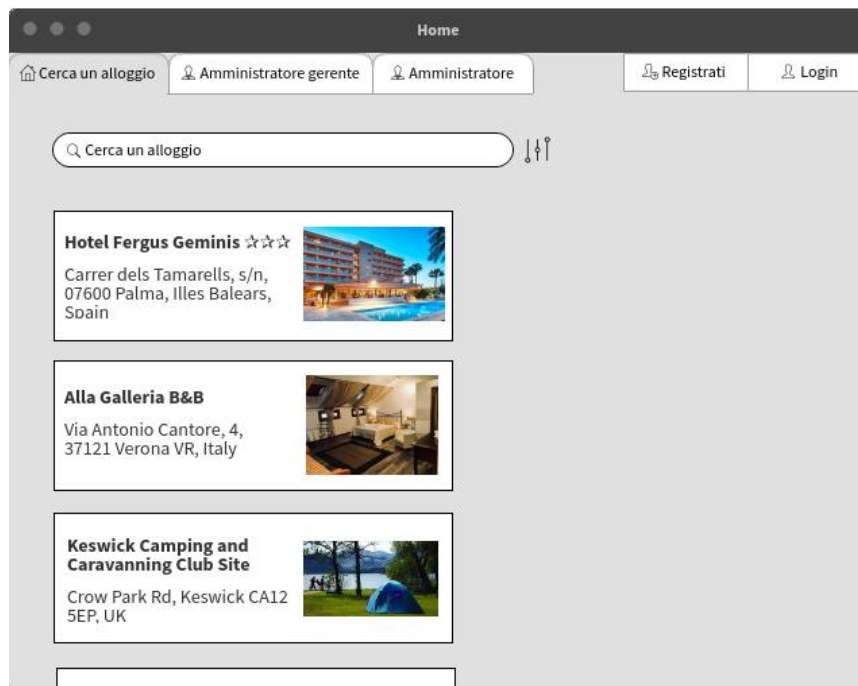


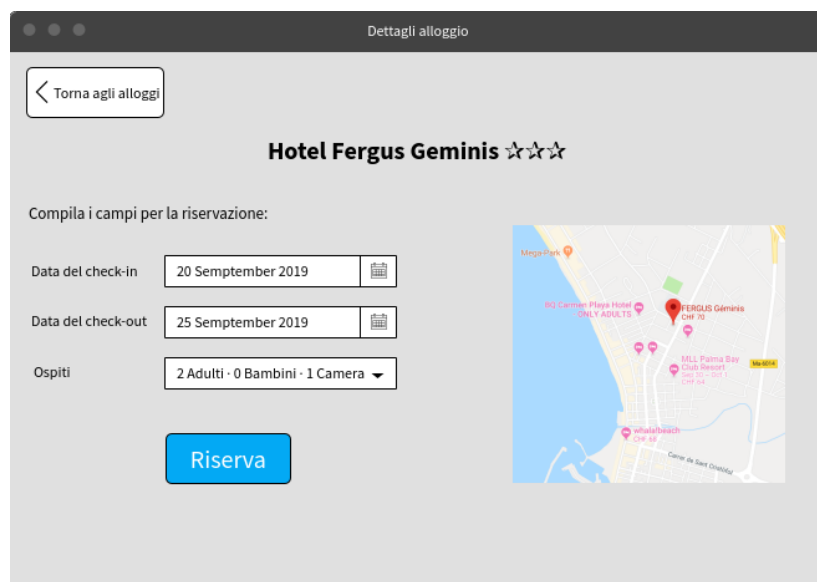
Diagramma E-R rappresentante la progettazione del Database

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

The mockup shows a window titled "Registrazione". Inside, the heading "Registrati" is displayed. Below it, there are five input fields with labels: "Nome:", "Cognome:", "E-mail:", "Password:", and "Numero di telefono:". At the bottom right of the form area is a blue button labeled "Registrati".

Mockup della pagina di registrazione

The mockup shows a window titled "Login". Inside, the heading "Login" is displayed. Below it, there are two input fields with labels: "E-mail:" and "Password:". At the bottom right of the form area is a blue button labeled "Login".

Mockup della pagina di login



Mockup della pagina riservata agli amministratori



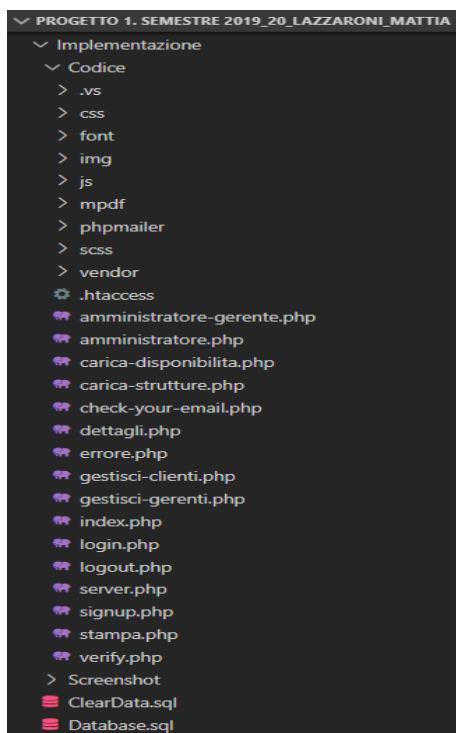
Mockup della pagina riservata agli amministratori gerenti

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 aggiunta 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 riserve 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 riserve 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.

```
//Istanzio 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 = 'ssl';
$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.

```
//Istanzio 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 efef_i16lazmat;

use efef_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://samtinio.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.samtinio.ch/gestionealloggi2019/blabla.php](http://www.samtinio.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 bot
tone 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 imposto 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);
//Eseguo 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"];
${"citta" . $i} = $row["citta"];
${"nome_tipologia" . $i} = $row["nome_tipologia"];
${"link_mappa" . $i} = $row["link_mappa"];
${"email_gereente" . $i} = $row["email_gereente"];
//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>
</form>';
```

#### 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);
$params_password = password_hash($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! <br>
        Il tuo account è stato creato, puoi accedere con le seguenti credenziali dopo aver attivato l'account. <br>
```

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

```
        Fai clic su questo link per attivare il tuo account: <br>
        <a href='http://saminfo.ch/gestionealloggi2019/verify.php?email=$email&hash=$hash'>http://saminfo.ch/gestionealloggi2019/verify.php?email=$email&hash=$hash</a>";
```

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 st
ampo un messaggio di riuscita all'utente.
if ($match > 0) {
    $update = $db-
>prepare("UPDATE utente SET is_active='1' WHERE email='" . $email . "' AND hash='" . $h
ash . "' AND is_active='0'");
    $update->execute();
}
```

Se viene trovato un utente, vado essenzialmente a modificargli 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&apos;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 “&apos;”, 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 = :em
ail";

//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 reindirizzato alla pagina principale.

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

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

//Distraggo 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 riservezioni 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 prenotazioni </button>
    </div> <!-- form-group// -->
</form>
</article>

```

Si riesce quindi a capire che quello che l'utente vede è un form contenente dei bottoni, 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 prenotazioni. 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 riserve effettuate su camere di sua proprietà

Una volta entrati in questa pagina, si presenta una tabella che stampa le riserve 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 riserve 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 riserva in base alla riga specificata da rownum  
r.

```
$email_query = \"SELECT * FROM
(SELECT row_number() OVER (ORDER BY id ASC) AS rownumber, id
FROM riserva)
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);
    $riserva_da_eliminare = $row[1];

```

//Query che disattiva i controlli.

```
$disable_checks = \"SET FOREIGN_KEY_CHECKS=0\";

```

//Query che elimina la riserva.

```
$delete_query = \"DELETE FROM riserva
WHERE id = \" . $riserva_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 riserva cliccata. Per far questo, utilizzo \"\$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 riserva da eliminare e andiamo ad eseguire una semplice query che elimina la riserva 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 riserva 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 prenotazioni. Ovviamente queste date verranno controllate validate. Se le date saranno valide, la prenotazione verrà aggiornata nel database per merito del seguente codice:

```
//Query per modificare la prenotazione.
$modifica_query = "UPDATE prenotazione
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 prenotazione. Il modo in cui viene preso l'id della prenotazione da modificare dalla tabella è lo stesso medesimo di prima, non andrò dunque a spiegarlo.

#### 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 riserve fatte nel range di date inserito prima e/o fatte dall'utente di cui si è inserita l'email.

```
$_SESSION['stampa_multipla'] = "La tua ricerca ha restituito le seguenti riserve:<br><br>";
<table class='table table-<striped'>
    <tr>
        <th>Email utente</th>
        <th>ID Camera</th>
        <th>Adulti</th>
        <th>Bambini</th>
        <th>Check-in</th>
        <th>Check-out</th>
    </tr>";
while ($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 riserve (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 riserve". Cliccando questo bottone si passerà a PHPMailer e all'invio della tabella per email:

```
//Se l'utente decide di inviare la riserva.
if (isset($_POST['invia'])) {
    //Istanzio 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'ammi-
    nistratore aggiunge.
    $nextId = $row[0] + 1;
    $insert_alloggio_query = "INSERT INTO alloggio values (" . $nextId . ", '" . $n-
    ome . "', '" . $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'&grave; amministratore gerente con email '" . $email . "'
            &grave; 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 . '', password_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](mailto: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 p
agina.
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    //Stampo l'alloggio cliccato prendendo tutte le informazioni della pagina "inde
x.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_scelta'] = $row[0];
    ${ "id" . $i } = $row[0];
    //La query che controlla le date delle prenotazioni 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 una.
$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 riservazi
one.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 p
er " . $_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 nella riservaione nel database. Dopo aver salvato la riservaione nel database, semplicemente vado a mandare un email all'utente sempre con PHPMailer e in più gli allego un pdf con le informazioni della riservaione 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

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Verifica del funzionamento di: form di registrazione, conferma tramite email e form di login.
<b>Riferimento:</b>	REQ-005		
<b>Descrizione:</b>	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.		
<b>Prerequisiti:</b>	-		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure localhost/index.php</li> <li>2. Premere il bottone "Registrati" nella navbar.</li> <li>3. Riempire il form con dei dati validi (es.: "Mattia", "Lazzaroni", "lazza.yt@gmail.com", "+41761234567", "prova123", "prova123", "Utente base")</li> <li>4. 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).</li> <li>5. Recarsi nell'inbox dell'email inserita durante la registrazione e cliccare il link presente nella nuova email appena arrivata.</li> <li>6. Si verrà reindirizzati alla pagina che conferma che l'account è stato attivato, premere "Vai al login".</li> <li>7. Inserire l'email e la password inserite in precedenza nel form di registrazione e premere "Accedi".</li> </ol>		
<b>Risultati attesi:</b>	Si deve venir reindirizzati alla homepage e deve uscire un messaggio tipo "Ciao Mattia, hai effettuato correttamente l'accesso come utente. Cerca un alloggio."		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Verifica dei controlli e dei messaggi di errore del form di registrazione.
<b>Riferimento:</b>	REQ-005		
<b>Descrizione:</b>	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.		
<b>Prerequisiti:</b>	È necessario che nel database ci sia almeno un utente dall'email conosciuta.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure localhost/index.php</li> <li>2. Premere il bottone "Registrati" nella navbar.</li> <li>3. Riempire il form con dei dati non validi o duplicati (nel caso dell'email).</li> <li>4. Premere "Crea Account".</li> </ol>		

	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).
<b>Risultati attesi:</b>	Dovranno venire mostrati degli errori in base al dato sbagliato o vuoto e la registrazione risulterà non completata.

<b>Test Case:</b>	TC-003	<b>Nome:</b>	Verifica dei messaggi di errore del form di login.
<b>Riferimento:</b>	REQ-005		
<b>Descrizione:</b>	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		
<b>Prerequisiti:</b>	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.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a></li> <li>2. Premere il bottone “Login” nella navbar.</li> <li>3. Riempire il form con un email che non sia quella conosciuta e una password a scelta.</li> <li>4. Premere “Accedi”.</li> <li>5. Riempire il form con l’email conosciuta e presente nel database e la password che coincida con l’email inserita.</li> <li>6. Premere ancora “Accedi”.</li> <li>7. Attivare l’utente dal link ricevuto per email.</li> <li>8. Riempire il form con l’email conosciuta e la password che non coincida con l’email.</li> <li>9. Premere un’ultima volta “Accedi”.</li> </ol>		
<b>Risultati attesi:</b>	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.		

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

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

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

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

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

<b>Descrizione:</b>	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.
<b>Prerequisiti:</b>	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.
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca.</li> <li>3. Inserire il nome di un gerente nel campo di testo “Nome gerente della struttura”.</li> <li>4. Premere il bottone blu “Applica”.</li> <li>5. Premere il bottone blu con la lente.</li> </ol>
<b>Risultati attesi:</b>	Tutti gli alloggi dovranno sparire, tranne quelli inseriti dal gerente scelto precedentemente.

<b>Test Case:</b>	TC-008	<b>Nome:</b>	Verifica del funzionamento della ricerca tramite nome
<b>Riferimento:</b>	REQ-004		
<b>Descrizione:</b>	Sarà necessario verificare che inserendo un nome della barra di ricerca, vengano mostrati solo gli alloggi che contengono una parte del nome inserito.		
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Inserire il nome di un alloggio o una parte del nome (ad esempio “Fergus” oppure “ergu”) nella barra di ricerca.</li> <li>3. Premere il bottone blu con la lente.</li> </ol>		
<b>Risultati attesi:</b>	Dovranno scomparire tutti gli alloggi, tranne quello/i che soddisfa/no la ricerca (in questo caso l’Hotel Fergus Geminis).		

<b>Test Case:</b>	TC-009	<b>Nome:</b>	Verifica del funzionamento dei filtri combinati
<b>Riferimento:</b>	REQ-004		
<b>Descrizione:</b>	Sarà necessario verificare che scegliendo più filtri, essi continuino a funzionare.		
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Schiacciare la freccia nel bottone bianco presente di fianco alla barra di ricerca.</li> <li>3. Scegliere o inserire almeno un valore nei quattro filtri presenti (ad esempio tipologia “Albergo” e nome gerente della struttura “Mattia”).</li> <li>4. Premere il bottone blu “Applica”.</li> </ol>		

	<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>
<b>Risultati attesi:</b>	Dovranno scomparire tutti gli alloggi, tranne quello/i che soddisfa/no la ricerca (in questo caso l’Hotel Fergus Geminis e il Katikies Hotel).

<b>Test Case:</b>	TC-010	<b>Nome:</b>	Verifica del funzionamento della riservazione con email di conferma e PDF da stampare
<b>Riferimento:</b>	REQ-005		
<b>Descrizione:</b>	Sarà necessario verificare che funzioni la riservazione di un alloggio e che arrivi un email con le informazioni della riservazione.		
<b>Prerequisiti:</b>	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.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Scegliere ad esempio l'alloggio “Hotel Fergus Geminis” dalla lista e premere “Mostra dettagli”.</li> <li>3. Inserire 2 adulti e 0 bambini nei campi sotto.</li> <li>4. Scegliere una data di check-in, ad esempio 02.02.2020.</li> <li>5. Scegliere una data di check-out, ad esempio 09.02.2020.</li> <li>6. Premere il bottone blu “Cerca disponibilità”.</li> <li>7. Usciranno 2 camere disponibili, scegliere ad esempio la camera 36 selezionando il secondo radio button sulla destra.</li> <li>8. Premere il bottone blu “Riserva la camera”.</li> </ol>		
<b>Risultati attesi:</b>	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.		

<b>Test Case:</b>	TC-011	<b>Nome:</b>	Verifica del funzionamento della pagina del caricamento delle disponibilità per i gerenti.
<b>Riferimento:</b>	REQ-003		
<b>Descrizione:</b>	Sarà necessario verificare che funzioni correttamente il caricamento di una camera.		
<b>Prerequisiti:</b>	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.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Nella barra di navigazione sopra selezionare “Amministratore gerente”.</li> <li>3. Premere il bottone blu “Carica disponibilità”.</li> <li>4. Scegliere il numero di adulti e di bambini, ad esempio 2 e 3.</li> </ol>		

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

<b>Test Case:</b>	TC-012	<b>Nome:</b>	Verifica del funzionamento della pagina della gestione dei clienti.
<b>Riferimento:</b>	REQ-003		
<b>Descrizione:</b>	Sarà necessario verificare che funzioni correttamente la gestione di un cliente.		
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi, camere e prenotazioni. Bisognerà inoltre aver fatto l'accesso come amministratore gerente, se non si sa come fare, riferirsi al TC-001.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Nella barra di navigazione sopra selezionare "Amministratore gerente".</li> <li>3. Premere il bottone blu "Gestisci clienti".</li> <li>4. Premere il bottone azzurro "Modifica", ad esempio nella riga della prenotazione 22.</li> <li>5. Nel form, cambiare le date di check-in e check-out, ad esempio in 15.01.2020 e 20.11.2019</li> <li>6. Premere "Salva modifiche"</li> </ol>		
<b>Risultati attesi:</b>	Dovrà venire mostrato un messaggio di riuscita della modifica e controllando il database, dovrebbe esserci le nuove date della prenotazione appunto nella tabella "Prenotazione".		

<b>Test Case:</b>	TC-013	<b>Nome:</b>	Verifica del funzionamento della stampa per gli amministratori gerenti
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Sarà necessario verificare che gli amministratori gerenti possano stampare correttamente le prenotazioni fatte in un certo periodo e/o da un certo utente.		
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi, camere e prenotazioni. Bisognerà inoltre aver fatto l'accesso come amministratore gerente, se non si sa come fare, riferirsi al TC-001.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Nella barra di navigazione sopra selezionare "Amministratore gerente".</li> <li>3. Premere il bottone blu "Stampa prenotazioni".</li> <li>4. Nel form, inserire le date di check-in e check-out per cui si vogliono vedere le prenotazioni, ad esempio dal 01.11.2019 al 25.12.2019. Inserire eventualmente anche l'email dell'utente, ad esempio <a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>.</li> <li>5. Premere "Cerca".</li> <li>6. Verrà mostrata una lista di prenotazioni, premere "Invia le prenotazioni".</li> </ol>		



<b>Risultati attesi:</b>	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 prenotazioni stampate precedentemente a schermo.
--------------------------	--

<b>Test Case:</b>	TC-014	<b>Nome:</b>	Verifica del funzionamento della pagina del caricamento delle strutture per gli amministratori.
<b>Riferimento:</b>	REQ-002		
<b>Descrizione:</b>	Sarà necessario verificare che funzioni correttamente il caricamento di un alloggio.		
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che nel database ci siano un po' di alloggi, camere e prenotazioni. Bisognerà inoltre aver fatto l'accesso come amministratore, se non si sa come fare, riferirsi al TC-001.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Nella barra di navigazione sopra selezionare "Amministratore".</li> <li>3. Riempire il form come si preferisce, con dei dati però consoni a quanto indica l'input box.</li> <li>4. Premere il bottone blu "Aggiungi".</li> </ol>		
<b>Risultati attesi:</b>	Dovrà venire mostrato un messaggio di riuscita dell'aggiunta e tornando nella home page e scorrendo infondo alla pagina dovrebbe esserci il nuovo alloggio.		


<b>Test Case:</b>	TC-015	<b>Nome:</b>	Verifica del funzionamento della pagina della gestione dei gerenti.
<b>Riferimento:</b>	REQ-003		
<b>Descrizione:</b>	Sarà necessario verificare che funzioni correttamente la gestione di un gerente.		
<b>Prerequisiti:</b>	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.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Aprire con un browser l'url <a href="http://www.samtinfo.ch/gestionealloggi2019">www.samtinfo.ch/gestionealloggi2019</a> oppure <a href="http://localhost/index.php">localhost/index.php</a>.</li> <li>2. Nella barra di navigazione sopra selezionare "Amministratore".</li> <li>3. Premere il bottone blu "Gestisci gerenti".</li> <li>4. Premere il bottone azzurro "Modifica", ad esempio nella riga del gerente "<a href="mailto:mattia.lazza@gmail.com">mattia.lazza@gmail.com</a>".</li> <li>5. Nel form, cambiare dei valori come si preferisce, nel mio caso andrò a modificare il numero di telefono in "+41764650110".</li> <li>6. Premere "Salva modifiche".</li> </ol>		
<b>Risultati attesi:</b>	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".		

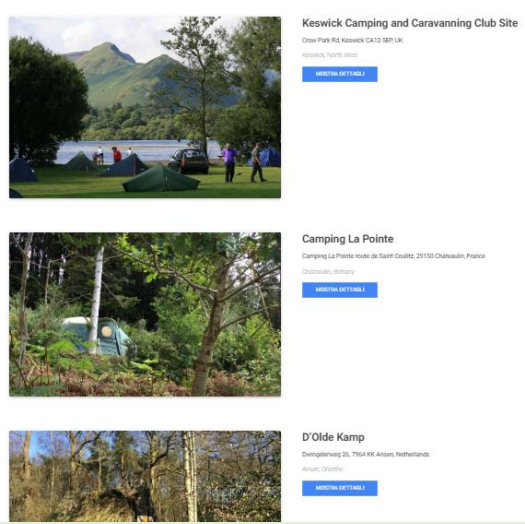
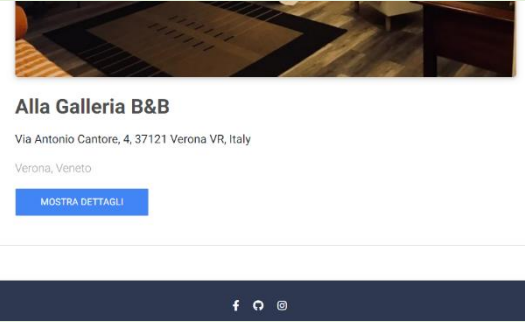
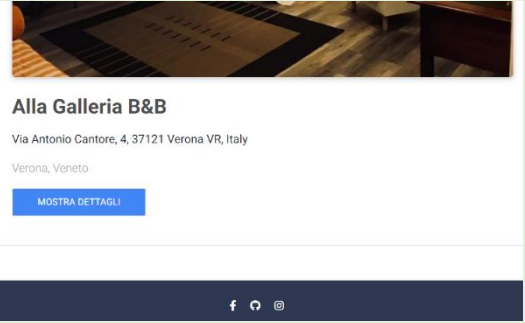
<b>Test Case:</b>	TC-016	<b>Nome:</b>	Verifica del funzionamento della gestione della fatturazione del sito
<b>Riferimento:</b>	REQ-007		




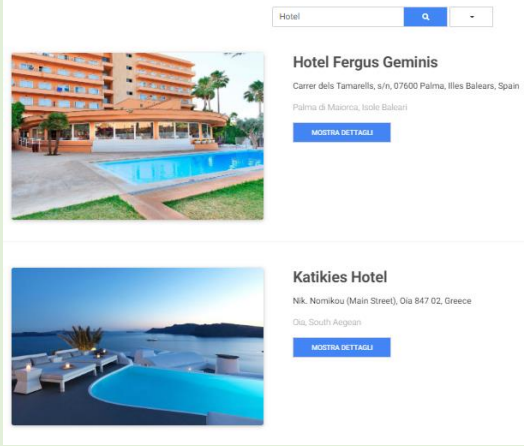
<b>Descrizione:</b>	Sarà necessario verificare che funzioni correttamente la gestione della fatturazione del sito web.
<b>Prerequisiti:</b>	È necessario che il sito venga aperto (in locale o online) e che venga effettuata qualche riservazione.
<b>Procedura:</b>	-
<b>Risultati attesi:</b>	-


## 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</p> <div> <p>Il nome non può contenere valori numerici</p> <p>Il cognome è richiesto</p> <p>Il formato dell'email non è valido</p> <p>Il formato per il numero di telefono non è valido</p> <p>Le due password non coincidono</p> </div> <p>mostrati i seguenti errori:</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</p> <div> <p>prova@gmail.com</p> <p>.....</p> <p>Nessun utente trovato con questa email</p> </div> <p>risultato:</p> <p>Successivamente, ho provato con l'email presente nel database ma con la password errata:</p> <div> <p>lazza.yt@gmail.com</p> <p>.....</p> <p>Non hai verificato l'account, controlla la tua email!</p> </div> <p>Infinte, ho inserito le credenziali corrette dell'utente non attivo:</p> <div> <p>lazza.yt@gmail.com</p> <p>...</p> <p>La password che hai inserito non è corretta</p> </div>	29.11.19








	<b>SAMT – Sezione Informatica</b> <b>Gestione disponibilità alloggi</b>	Pagina 52 di 61
--	--	-----------------


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> 	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> 	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> 	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 data-bbox="726 398 1252 891">  <p><b>Hotel Fergus Geminis</b> Carrer dels Tamarrells, s/n, 07600 Palma, Illes Balears, Spain Palma di Maiorca, Isole Baleari <a href="#">MOSTRA DETTAGLI</a></p>  <p><b>Alla Galleria B&amp;B</b> Via Antonio Cantoni, 4, 37121 Verona VR, Italy Verona, Veneto <a href="#">MOSTRA DETTAGLI</a></p>  <p><b>Keswick Camping and Caravanning Club Site</b> Cresk Park Rd, Keswick CA12 5BP UK Keswick, North West <a href="#">MOSTRA DETTAGLI</a></p> </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 data-bbox="726 1097 1252 1552">  <p>Hotel</p> <p><b>Hotel Fergus Geminis</b> Carrer dels Tamarrells, s/n, 07600 Palma, Illes Balears, Spain Palma di Maiorca, Isole Baleari <a href="#">MOSTRA DETTAGLI</a></p> <p><b>Katikies Hotel</b> Nik. Nomikou (Main Street), Oia 847 02, Greece Oia, South Aegean <a href="#">MOSTRA DETTAGLI</a></p> </div>	01.12.19

	<b>SAMT – Sezione Informatica</b>	Pagina 54 di 61
	<b>Gestione disponibilità alloggi</b>	

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> <div><div>Club Site</div><div></div></div>	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> <div><div>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.</div></div> <p>Andando poi a controllare le email, ho trovato</p> <div><div>Stampa riservazione <div>Inbox x</div><div></div></div></div> <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&amp;B”. Mi è stato mostrato il seguente messaggio:</p> <div><div>La camera è stata aggiunta con successo.</div></div> <p>Andando a controllare la tabella “Camera” del database ho potuto notare la nuova camera:</p> <table><tr><th>id</th><th>numero_bambini</th><th>numero_adulti</th><th>id_alloggio</th><th>email_gerente</th></tr><tr><td>37</td><td>3</td><td>2</td><td>2</td><td>mattia.lazza@gmail.com</td></tr></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</p> <div><div>La riservazione con ID '22' è stata modificata con successo.</div></div> <p>messaggio:</p>	19.12.19										

		<p>Andando a controllare la tabella “Riservazione” e cercando quella con ID “22”, ho potuto notare che le date erano quello nuove appena inserite:</p> <table><tr><td>id</td><td>data_checkin</td><td>data_checkout</td><td>id_camera</td><td>email_utente</td></tr><tr><td>22</td><td>2020-01-15</td><td>2020-01-20</td><td>1</td><td>lazza.yt@gmail.com</td></tr></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 prenotazioni così che egli possa stamparle. Questo è il messaggio una volta premuto il bottone per inviare l'email:</p> <p>L'email con le prenotazioni è stata inviata a 'mattia.lazza@gmail.com' con successo.</p> <p>Controllando la posta elettronica del gerente, ho notato l'email:</p> <div><p>Stampa prenotazioni <span>Posta in arrivo x</span>  </p><p><b>Gestione Alloggi</b> &lt;gestion...  11:51 (4 minuti fa)   </p><p>a me ▾</p><p>La tua ricerca ha restituito le seguenti prenotazioni:</p><table><tr><th>Email utente</th><th>ID Camera</th><th>Adulti</th><th>Bambini</th><th>Check-in</th><th>Check-out</th></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>1</td><td>0</td><td>2</td><td>25-11-2019</td><td>18-12-2019</td></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>1</td><td>0</td><td>2</td><td>16-12-2019</td><td>22-12-2019</td></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>36</td><td>0</td><td>2</td><td>17-12-2019</td><td>24-12-2019</td></tr></table><p>In allegato è possibile trovare un pdf da stampare.</p><div><p>Per le tue prenotazioni hai richiesto le seguenti informazioni:</p><table><tr><td>Email utente</td><td>ID Camera</td><td>Adulti</td><td>Bambini</td><td>Check-in</td><td>Check-out</td></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>1</td><td>0</td><td>2</td><td>25-11-2019</td><td>18-12-2019</td></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>1</td><td>0</td><td>2</td><td>16-12-2019</td><td>22-12-2019</td></tr><tr><td><a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a></td><td>36</td><td>0</td><td>2</td><td>17-12-2019</td><td>24-12-2019</td></tr></table></div><p> <b>riservazioni.pdf</b></p></div>	Email utente	ID Camera	Adulti	Bambini	Check-in	Check-out	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	25-11-2019	18-12-2019	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	16-12-2019	22-12-2019	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	36	0	2	17-12-2019	24-12-2019	Email utente	ID Camera	Adulti	Bambini	Check-in	Check-out	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	25-11-2019	18-12-2019	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	16-12-2019	22-12-2019	<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	36	0	2	17-12-2019	24-12-2019	19.12.19
Email utente	ID Camera	Adulti	Bambini	Check-in	Check-out																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	25-11-2019	18-12-2019																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	16-12-2019	22-12-2019																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	36	0	2	17-12-2019	24-12-2019																																														
Email utente	ID Camera	Adulti	Bambini	Check-in	Check-out																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	25-11-2019	18-12-2019																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	1	0	2	16-12-2019	22-12-2019																																														
<a href="mailto:lazza.yt@gmail.com">lazza.yt@gmail.com</a>	36	0	2	17-12-2019	24-12-2019																																														
TC-014	OK	<p>Ho provato a riempire il form nel seguente modo:</p> <div><p>Aggiungi una struttura</p><p>Hotel Milano</p><p>Piazza della Stazione, 6850 Mendrisio</p><p><a href="https://q-cf.bstatic.com/images/hotel/max1024x768/208/2">https://q-cf.bstatic.com/images/hotel/max1024x768/208/2</a></p><p><a href="https://www.google.com/maps/embed?pb=!1m18!1m12!1n">https://www.google.com/maps/embed?pb=!1m18!1m12!1n</a></p><p>Ticino</p><p>Mendrisio</p><p>mattia.lazza@gmail.com</p><p>Albergo ▾</p></div> <p>Premendo poi il bottone “Aggiungi”, mi è stato mostrato il seguente messaggio:</p> <p>La struttura è stata aggiunta con successo.</p>	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> <div><p><b>Hotel Milano</b> Piazza della Stazione, 6850 Mendrisio Mendrisio, Ticino <a href="#">MOSTRA DETTAGLI</a></p></div>											
TC-015	OK	<p>Ho provato a modificare il numero di telefono dell’amministratore gerente con email “<a href="mailto:mattia.lazza@gmail.com">mattia.lazza@gmail.com</a>” in “+41764650110”.</p> <div>L’amministratore gerente con email 'mattia.lazza@gmail.com' è stato modificato con successo.</div> <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 nuove appena inserito:</p> <table><tr><td>email</td><td>nome</td><td>cognome</td><td>password_admin_gente</td><td>n_telefono</td></tr><tr><td>mattia.lazza@gmail.com</td><td>Mattia</td><td>Lazzaroni</td><td>\$2y\$10\$UwB.V5xhQAxhXKZycFydeTbs9j9AIS...</td><td>+41764650110</td></tr></table>	email	nome	cognome	password_admin_gente	n_telefono	mattia.lazza@gmail.com	Mattia	Lazzaroni	\$2y\$10\$UwB.V5xhQAxhXKZycFydeTbs9j9AIS...	+41764650110	19.12.19
email	nome	cognome	password_admin_gente	n_telefono									
mattia.lazza@gmail.com	Mattia	Lazzaroni	\$2y\$10\$UwB.V5xhQAxhXKZycFydeTbs9j9AIS...	+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 riserve 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 sprecate 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 trovato 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.

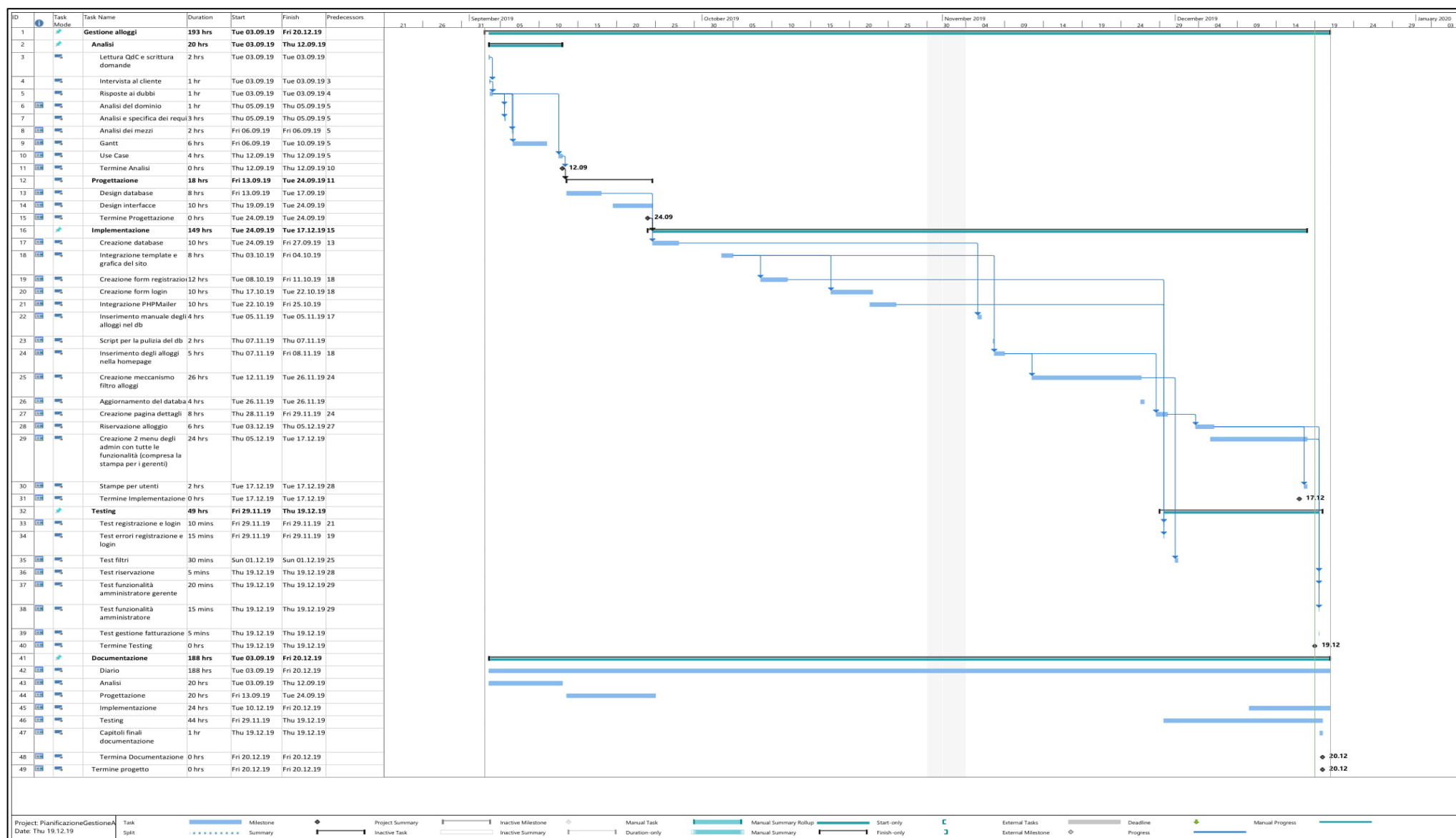


Diagramma di Gantt rappresentante il consuntivo del progetto

## 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 prenotazioni in modo da evitare più prenotazioni 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 fornitoci 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