Lavoro temporaneo

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

1.3 Scopo 4

2 Analisi 5

2.1 Scopo 5

2.2 Requisiti 5

2.3 Pianificazione 8

2.4 Analisi dei mezzi 9

2.4.1 Software 9

2.4.2 Hardware 9

3 Progettazione 10

3.1 Design della struttura dell’applicativo web 10

3.2 Design dei dati e database 11

3.2.1 Progettazione logica 11

3.2.2 Progettazione fisica 12

3.3 Design delle interfacce 13

3.3.1 Progettazione pagina principale 13

3.3.2 Progettazione form di registrazione 13

3.3.3 Progettazione form di accesso 14

3.3.4 Progettazione pagina amministratore 14

3.3.5 Progettazione pagina crea fatturazione 15

3.3.6 Progettazione pagina visualizza fatturazione 15

3.3.7 Progettazione pagina datore di lavoro 16

3.3.8 Progettazione pagina datore di lavoro (creazione offerta) 16

3.3.9 Progettazione pagina Lavoratore 17

3.3.10 Progettazione pagina Lavoratore (creazione domanda) 17

3.4 Design procedurale 17

4 Implementazione 18

4.1 Database 18

4.1.1 Account per l’accesso 18

4.1.2 Implementazione fisica banca dati 18

4.1.3 Implementazione grafica banca dati 19

4.2 Applicativo Web 20

4.2.1 Struttura 20

4.3 Pagina Index 20

5 Test 21

5.1 Protocollo di test 21

5.2 Risultati test 26

5.3 Mancanze/limitazioni conosciute 31

6 Consuntivo 32

7 Conclusioni 33

7.1 Sviluppi futuri 33

7.2 Considerazioni personali 33

8 Bibliografia 34

8.1 Bibliografia per articoli di riviste: 34

8.2 Bibliografia per libri 34

8.3 Sitografia 34

9 Allegati 35

# Introduzione

## Informazioni sul progetto

* Allievo:
  + Samuele Abbà
* Docente:
  + Guido Montalbetti
* Scuola:
  + SAMT Trevano
* Sezione:
  + Informatica
* Materia:
  + M306
* Inizio:
  + 16.09.2021
* Fine:
  + 23.12.2021

## Abstract

* **Situazione iniziale**
  + All’inizio di questo progetto, mi è stato chiesto di realizzare un sistema che possa gestire la domanda e l’offerta di posti di lavoro temporanei. Questo progetto nasce da un bisogno ben specifico. Fino ad ora, non esiste un prodotto o servizio che offre la possibilità di mettere in contatto due o più persone che hanno bisogno di un aiuto o che possono dare un aiuto. In breve: questo progetto è incentrato sull’elaborazione di un portale web che possa gestire comodamente le domande e le offerte per un lavoro temporaneo.
* **Attuazione**
  + Per questo progetto, saranno necessarie buone conoscenze nella programmazione web, poiché complesso nella sua struttura e nelle sue funzionalità. Seguendo i requisiti dati, per questo progetto è stato quindi necessario adoperare una banca dati per immagazzinare delle informazioni, compresa l’implementazione di tutta la sua sicurezza. Sono stati implementati una serie di sistemi che permettano l’aggiunta, la modifica e la rimozione delle varie offerte di lavoro all’interno della banca dati. Questo permettendo agli utenti presenti nel sistema di essere il più comodi, veloci e sicuri nell’utilizzo del prodotto. Vi sarà inoltre la possibilità di caricare dei documenti, quali p.es. Curriculum Vitae, Diplomi, ad ogni richiesta e una volta completata la candidatura, i clienti riceveranno tramite eMail la conferma della loro iscrizione al posto in offerta. L’offerta scadendo permette di fa calcolare al sistema i soldi che il datore di lavoro deve pagare al lavoratore e all’agenzia, inoltre ci saranno degli amministratori che potranno eseguire delle stampe, con inserimento del periodo desiderato, per gli stipendi da consegnare ai collaboratori e la fatturazione al datore di lavoro da parte dell’Agenzia.
* **Risultati**
  + Questo sito è realizzato da zero con i principali linguaggi web (HTML, CSS, PHP). Ho quindi avuto possibilità di lavorare su argomenti già consolidati e argomenti non ancora consolidati, questo mi ha permesso di ampliare il mio bagaglio delle conoscenze durante il progetto. Uno dei punti fondamentali tenuti in considerazione durante il progetto riguarda lo studio per raggiungere la facilità di utilizzo per un utente medio. Questo portale web è difatti utilizzabile e configurabile anche da un’utenza che non possiede conoscenze informatiche. In conclusione, questo progetto è ora l’unico sito funzionante che permetta di dare o prendere lavori temporanei.

## Scopo

* Vedi QdC in allegato

# Analisi

## Scopo

Spesso per poter trovare un lavoro 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 trovare un lavoro in pochi minuti.

Per poter accedere al mio sito sarà necessario come minimo un dispositivo con accesso ad Internet e un’email e password nel caso in qui si vuole accettare un lavoro e una conoscenza basilare per poter usare un sito Web.

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 del sito web, che funzioni correttamente e senza essere loggati, questo ti permette di vedere le varie offerte di lavoro.

Come secondo requisito ho messo la verifica del form di registrazione e anche la verifica del form di accesso per (amministratori, datori di lavoro e clienti).

Dopodiché come quarto requisito ho messo la verifica del sito web per i clienti nell’ accettare le offerte di lavoro.

Dopo come quinto requisito ho messo la verifica del sito web per i datori di lavoro nell’ aggiungere, togliere o modificare le offerte di lavoro.

Poi come sesto requisito ho messo la fatturazione per i datori di lavoro e per i collaboratori.

Successivamente i requisiti per le stampe; per gli stipendi dei collaboratori e per le fatture dei datori di lavoro.

I prossimi due requisiti concernono la gestione delle richieste dei clienti e delle offerte dei datori di lavoro registrati, è importante definire gli attributi delle richieste di lavoro e delle offerte, come anche la conferma tramite e-mail.

Come decimo requisito vi sarà inoltre la possibilità di caricare dei documenti, quali p.es. Curriculum Vitae, Diplomi, ad ogni offerta e una volta completata la candidatura, i clienti riceveranno tramite eMail la conferma della loro iscrizione al posto in offerta.

## Requisiti

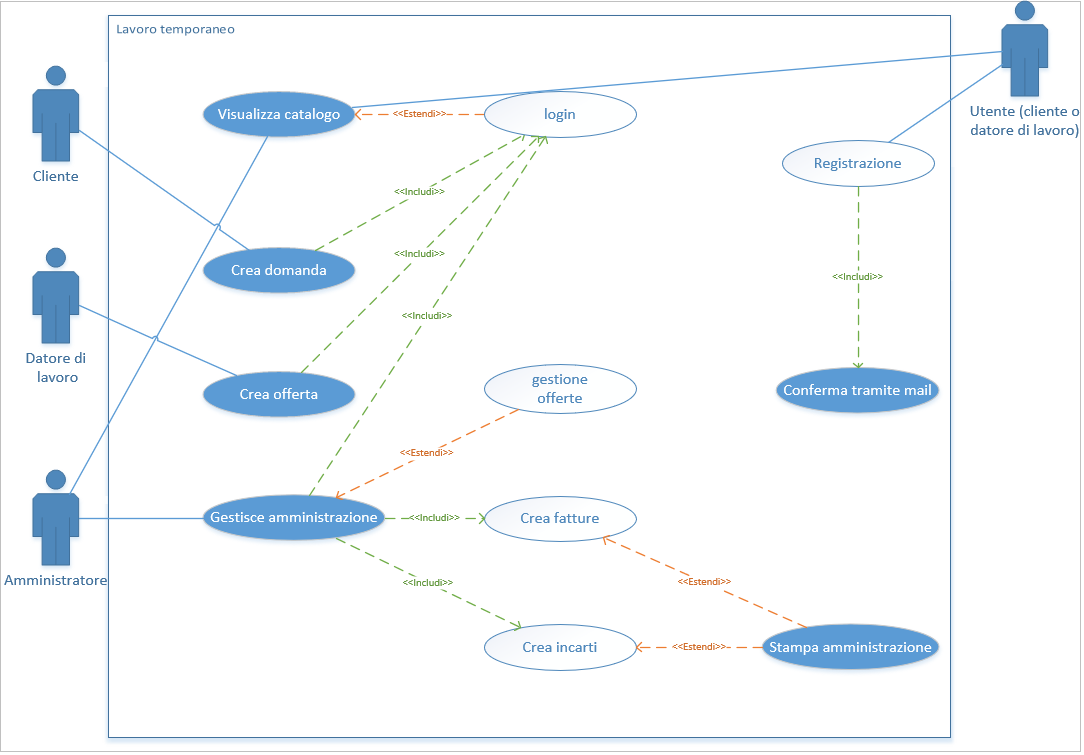
|  |  |
| --- | --- |
|  | **ID: REQ-001** |
| **Nome** | Verifica del sito web senza login |
| **Priorità** | 1 |
| **Versione** | 1.0 |
| **Note** | È necessario verificare che il sito web funzioni correttamente e senza essere loggati visualizzare i lavori disponibili. |

|  |  |  |
| --- | --- | --- |
|  | | **ID: REQ-002** |
| **Nome** | | Verifica del sito web con login |
| **Priorità** | | 1 |
| **Versione** | | 1.0 |
| **Note** | | È necessario verificare che il sito web funzioni correttamente. |
| **Sotto Requisiti** | | |
| **0021** | Verifica del funzionamento di: form di registrazione, conferma tramite email | |
| **0022** | Verifica del funzionamento di: form di accesso. | |

|  |  |  |
| --- | --- | --- |
|  | | **ID: REQ-003** |
| **Nome** | | Verifica risultato del login |
| **Priorità** | | 1 |
| **Versione** | | 1.0 |
| **Note** | | È necessario verificare il corretto risultato del login |
| **Sotto Requisiti** | | |
| **0031** | Si possano creare richieste. (se si è clienti) | |
| **0032** | Si possano creare offerte. (se si è datori di lavoro) | |
| **0033** | Si possano gestire le offerte. (se si è amministratori) | |

|  |  |  |
| --- | --- | --- |
|  | | **ID: REQ-004** |
| **Nome** | | Verifica gestione delle spese |
| **Priorità** | | 1 |
| **Versione** | | 1.0 |
| **Note** | | È necessario verificare che le fatture funzionino correttamente. |
| **Sotto Requisiti** | | |
| **0041** | Verifica fine del lavoro. | |
| **0042** | Calcolo spese del datore di lavoro. | |
| **0043** | Calcolo stipendio dei collaboratori. | |

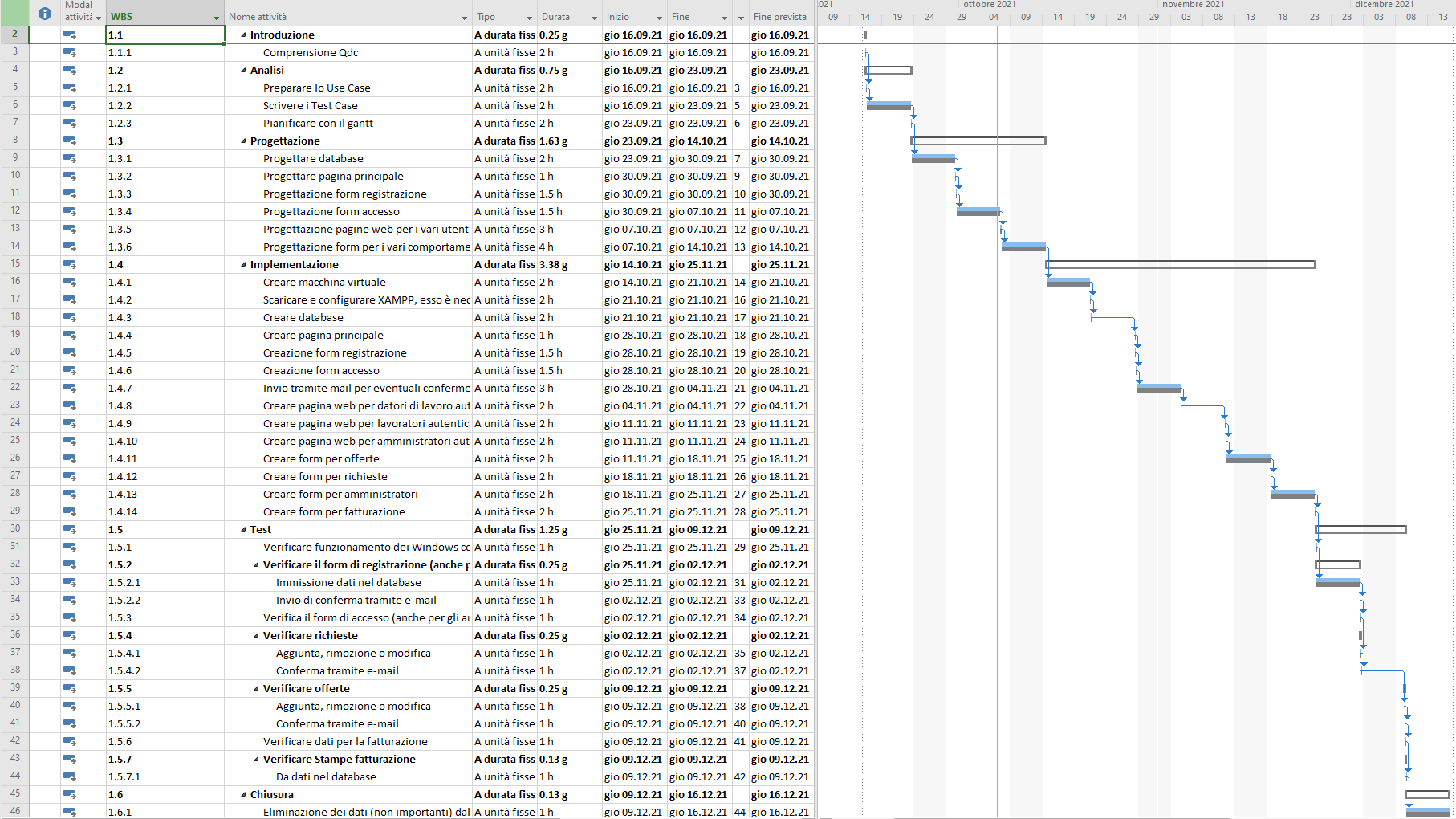
Use case



All’interno dell’applicativo ho inserito 3 tipi diversi di utente, più specificatamente:

* Amministratore: Gli utenti appartenenti a questo gruppo sono presenti nel database l’unico compito che avranno sarà quello di preparare le fatture, gli incarti e gestire il sistema. Per fare ciò gli amministratori dovranno loggarsi e la loro aggiunta avverrà nel database alla sua creazione.
* Datore di lavoro e cliente: essi dovranno obbligatoriamente eseguire l’accesso tramite un account per interagire con il sito. Essi potranno se datori di lavoro creare, modificare e cancellare le offerte oppure, se clienti accettare le varie richieste.
* In ogni caso se non si è loggati si può comunque guardare il catalogo.

## Pianificazione



## Analisi dei mezzi

Per questo progetto mi sarà fornito 1 PC della scuola, sul quale ci sarà un sistema operativo Windows10Pro.

### Software

I software utilizzati per questo progetto sono XAMPP (Apache, MySQL, Php e Perl) per la creazione del server web e la sua gestione, bisogna anche installare le VC16 per il corretto funzionamento di Apahce.

Per questo progetto utilizzerò inoltre i software Firefox/Google e notepad/notepad++ e MySQL Workbench.

### Hardware

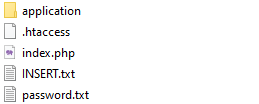
L’hardware coinvolto nel progetto è un PC fornito dalla scuola, esso ha ottime prestazioni ma come limitazioni ha il fatto che è collegato in una rete interna quindi il sito potrà venir utilizzato unicamente nella rete scolastica. Per questo progetto utilizzerò inoltre una macchina virtuale, un SSD e una serie di chiavette.

# Progettazione

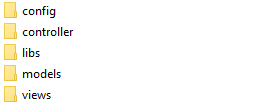
## Design della struttura dell’applicativo web

La struttura dell’applicativo WEB si baserà su MVC, un framework base ma che basterà per il progetto.

La struttura delle cartelle è descritta di seguito e all’interno sono presenti i seguenti elementi:



E all’interno della cartella application sono contenute le cartelle seguenti



* **Application**: Cartella contenente tutte le cartelle e i file necessari per il funzionamento del servizio.
* **Config**: Dentro questa cartella è presente un file di configurazione contenente il codice per mostrare

tutti gli errori sulla pagina web e delle costanti che verranno utilizzate durante lo sviluppo, come l’URL, essa serve per raggiungere l’applicativo WEB (in locale).

* **Controller**: Questa cartella, come si può facilmente intuire, conterrà tutti i controller. Essi sono i file

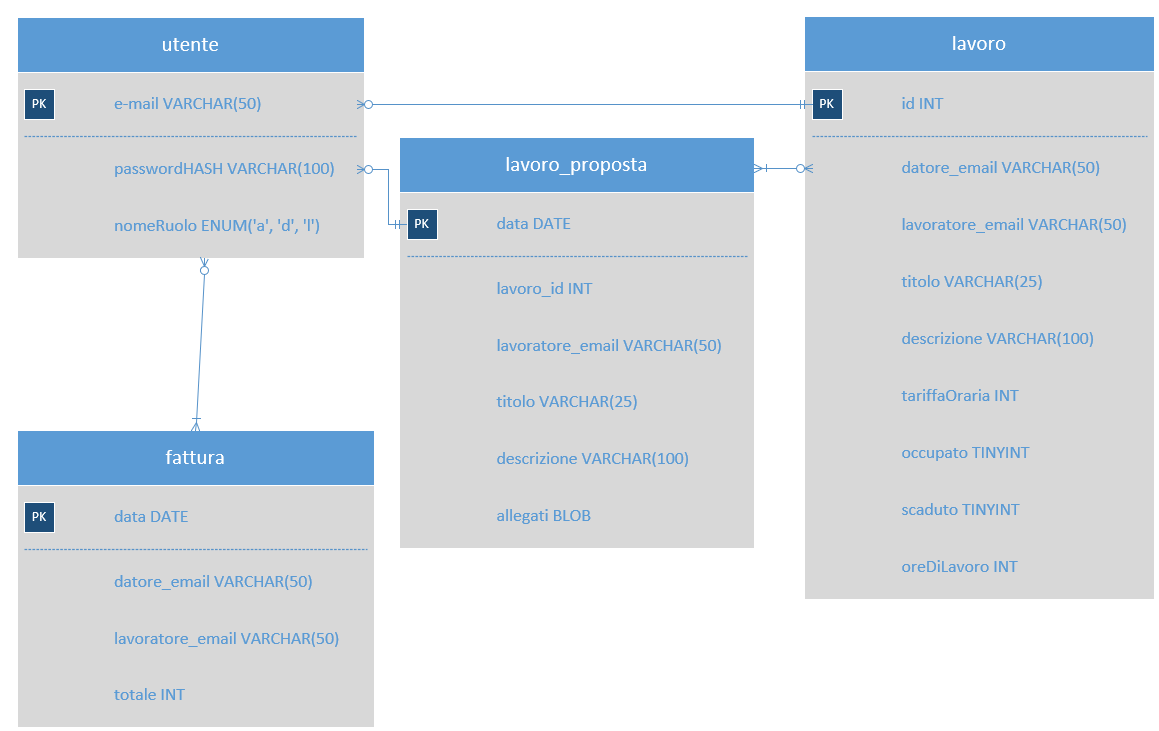
che si occupano di prendere i dati dalle classi e darli alle views. Questi file sono pensati per avere una comunicazione specifica tra i dati effettivi e quello che vede l’utente, avendo così un codice

estremamente più ordinato.

* **Libs**: In questa cartella ci saranno tutte le librerie che verranno utilizzate e saranno necessarie ai Controller.
* **Models**: All’interno ci saranno tutte le classi che si occuperanno di gestire i dati, esse si interfacceranno con il database ed estrapoleranno i dati dal suo interno, oltre alle funzioni più complesse.
* **Views**: Qui ci sono i file dove l’utente s’interfaccia, ad esempio la pagina principale.

## Design dei dati e database

### Progettazione logica



Di seguito una descrizione di tutte le tabelle del database.

* Utente: questa tabella contiene tutti gli utenti del database, ogni utente avrà una e-mail, una passwordHash e un nomeRuolo.
* Lavoro\_Proposta: questa tabella contiene tutte le proposte che verranno fatte dai lavoratori alle offerte di lavoro dei datori di lavoro esse sono caratterizzate da (data, lavoro\_id, lavoratore\_email, titolo, descrizione e allegati).
* Lavoro: questa tabella contiene tutti i lavori offerti dai datori di lavoro essi sono caratterizzate da (id, datore\_email, lavoratore\_email, titolo, descrizione, oreDiLavoro, tariffaOraria, se occupato e se scaduto).
* Fattura: questa tabella contiene tutte le fatture, che si fanno dai datori di lavoro ai lavoratori e ai collaboratori, questo grazie ai campi data, datore\_email, lavoratore\_email e totale.

### Progettazione fisica

CREATE DATABASE lavoro\_temporaneo;

USE DATABASE lavoro\_temporaneo;

tabella 1

CREATE TABLE utente(

email VARCHAR(50) PRIMARY KEY,

passwordHash VARCHAR(255) NOT NULL,

nomeRuolo ENUM('amministratore', 'datore', 'lavoratore') NOT NULL

);

tabella 2

CREATE TABLE fattura(

data DATETIME PRIMARY KEY,

datore\_email VARCHAR(50) NOT NULL,

lavoratore\_email VARCHAR(50) NOT NULL,

totale INT,

FOREIGN KEY (datore\_email)

REFERENCES utente(email)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email)

REFERENCES utente(email)

ON DELETE CASCADE

ON UPDATE CASCADE

);

tabella3

CREATE TABLE lavoro(

id INT AUTO\_INCREMENT PRIMARY KEY,

datore\_email VARCHAR(50) NOT NULL,

lavoratore\_email VARCHAR(50),

titolo VARCHAR(25) NOT NULL,

descrizione TEXT NOT NULL,

tariffaOraria INT NOT NULL,

occupato TINYINT NOT NULL,

scaduto TINYINT NOT NULL,

oreDiLavoro INT NOT NULL,

FOREIGN KEY (datore\_email)

REFERENCES utente(email)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email)

REFERENCES utente(email)

ON DELETE CASCADE

ON UPDATE CASCADE

);

tabella 4

CREATE TABLE lavoro\_proposta(

data DATETIME PRIMARY KEY,

lavoro\_id INT NOT NULL,

lavoratore\_email VARCHAR(50) NOT NULL,

titolo VARCHAR(25) NOT NULL,

descrizione TEXT NOT NULL,

allegati BLOB NOT NULL,

FOREIGN KEY (lavoro\_id)

REFERENCES lavoro(id)

ON DELETE CASCADE

ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email)

REFERENCES utente(email)

ON DELETE CASCADE

ON UPDATE CASCADE

);

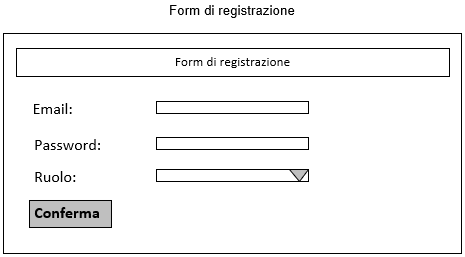
## Design delle interfacce

### Progettazione pagina principale



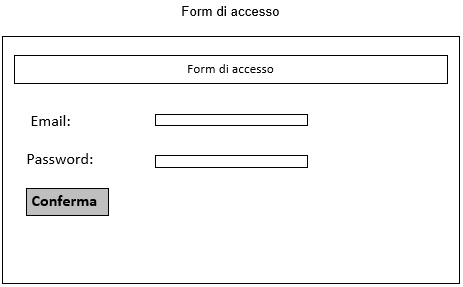
Come primissima interfaccia riporto quella che si visualizza non appena si apre il sito web senza essersi loggati, dove potremo accederci, registrarci o visualizzare semplicemente tutte le offerte di lavoro.

### Progettazione form di registrazione



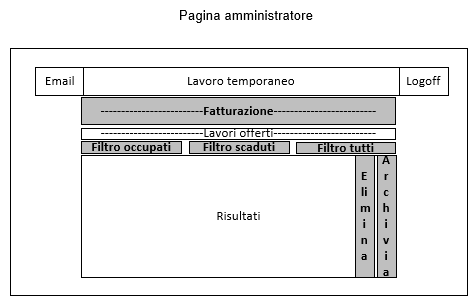
Come seconda interfaccia ho riportato quella della registrazione dove tramite di essa è possibile creare un account (e-mail e password) e definire un ruolo (datore di lavoro o lavoratore).

### Progettazione form di accesso



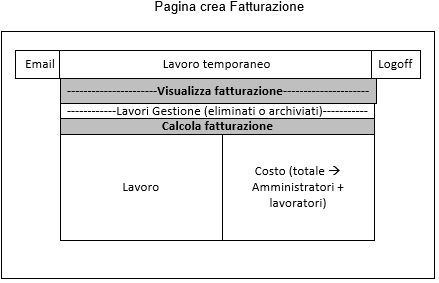
Come terza interfaccia ho riportato quella dell’accesso dove tramite di essa è possibile accedere al proprio account (e-mail e password).

### Progettazione pagina amministratore



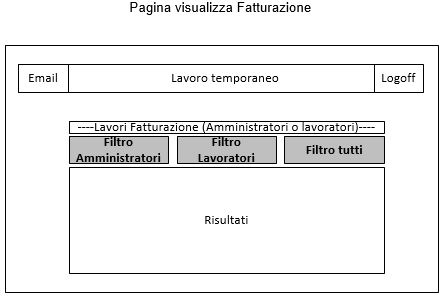
Come quarta interfaccia ho riportato quella degli amministratori una volta effettuato l’accesso, tramite di essa è possibile visualizzare tutte le offerte di lavoro e filtrarle (occupato, scaduto, tutti), è anche possibile eliminarle o archiviarle e infine da questa interfaccia si può accedere all’interfaccia della fatturazione.

### Progettazione pagina crea fatturazione



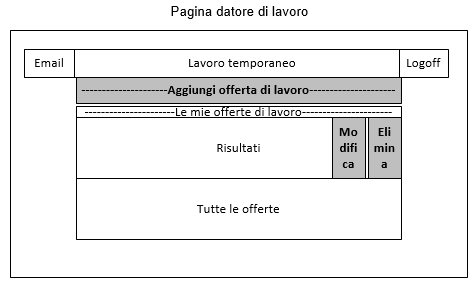
Come quinta interfaccia ho riportato quella per creare le fatture, da essa si possono vedere i lavori archiviati o eliminati e di conseguenza si può calcolare il costo per gli amministratori e per i lavoratori. Inoltre sarà possibile collegarsi alla pagina che permetterà di vedere tutte le varie fatture per i datori di lavoro e per i collaboratori.

### Progettazione pagina visualizza fatturazione



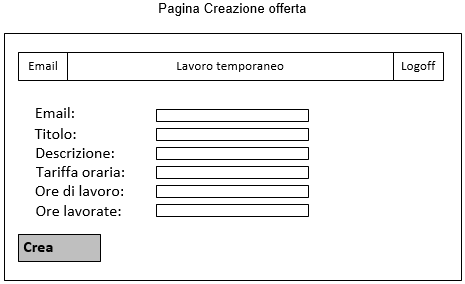
Come sesta interfaccia ho riportato quella per visualizzare le fatture, da essa si possono vedere tutte le fatture e grazie ad alcuni filtri è possibile visualizzare le paghe destinate agli amministratori e ai lavoratori date dai datori di lavoro.

### Progettazione pagina datore di lavoro



Come settima interfaccia ho riportato quella che visualizza il datore di lavoro una volta loggato nel sito, esso può vedere le sue offerte, queste ultime possono essere modificate o eliminate, oltre alle proprie offerte di lavoro si vedono anche quelle già presenti nel sito, infine cliccando sul bottone aggiungi offerta di lavoro si viene reindirizzati alla pagina per la creazione delle offerte e li è possibile crearne delle nuove.

### Progettazione pagina datore di lavoro (creazione offerta)



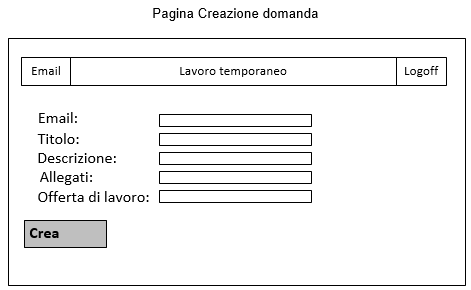
Come ottava interfaccia ho riportato quella che permette di creare un’offerta di lavoro, per farlo bisogna inserire l’e-mail del datore di lavoro, un titolo, una descrizione, una tariffa oraria, le ore di lavoro e successivamente le ore lavorate dal lavoratore. Cliccando su Crea viene creata l’offerta di lavoro.

### Progettazione pagina Lavoratore



Come nona interfaccia ho riportato quella che visualizza il lavoratore una volta loggato nel sito, esso può visualizzare le sue proposte di lavoro, queste ultime possono essere modificate o eliminate, inoltre si vede anche quelle già presenti nel sito, infine cliccando sul bottone aggiungi proposta di lavoro si viene reindirizzati alla pagina per proporsi alle offerte e li è possibile creare delle nuove domande di lavoro.

### Progettazione pagina Lavoratore (creazione domanda)



Come decima interfaccia ho riportato quella che permette di creare una domanda ad una offerta di lavoro, per farlo bisogna inserire l’e-mail del lavoratore, un titolo, una descrizione, allegare file se necessario, selezionare il lavoro interessato. Cliccando su Crea viene creata la domanda all’offerta di lavoro.

## Design procedurale

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

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

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

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato. Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile. In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici. Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte. Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata. Per eventuali dettagli si possono inserire riferimenti ai diari.

Attenzione: mettere parte dove viene spiegato come impostare i file php.ini e mailsender.php

## Database

### Account per l’accesso

L’account di accesso al database è root e non ha la password. Esso può creare un database e su di esso ha permessi completi sul db e sulle tabelle contenute in esso, inoltre può anche dare i permessi ad altri utenti.

### Implementazione fisica banca dati

Il database è stato implementato seguendo lo schema ER creato nel capitolo di progettazione, anche se purtroppo sono stato costretto a modificare quasi tutte le tabelle e qualche campo, poiché non avevo pensato a conservare i dati se rimossi a mettere una data per riuscire in seguito a fare delle verifiche. Di seguito c’è lo script per creare il db completo.

Tabella utente:

CREATE TABLE utente (

email varchar(50) NOT NULL,

passwordHash varchar(255) NOT NULL,

nomeRuolo enum('amministratore','datore','lavoratore') NOT NULL,

PRIMARY KEY (`email`)

)

Tabella lavoro:

CREATE TABLE lavoro (

id int(11) NOT NULL AUTO\_INCREMENT,

datore\_email varchar(50) NOT NULL,

lavoratore\_email varchar(50) DEFAULT NULL,

titolo varchar(25) NOT NULL,

descrizione text NOT NULL,

tariffaOraria int(11) NOT NULL,

occupato tinyint(4) NOT NULL,

scaduto tinyint(4) NOT NULL,

oreDiLavoro int(11) NOT NULL,

data datetime NOT NULL,

archiviato tinyint(4) NOT NULL,

PRIMARY KEY (id),

FOREIGN KEY (datore\_email) REFERENCES utente(email) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email) REFERENCES utente(email) ON DELETE CASCADE ON UPDATE CASCADE

)

Tabella lavoro\_proposta:

CREATE TABLE lavoro\_proposta (

data datetime NOT NULL,

lavoro\_id int(11) NOT NULL,

lavoratore\_email varchar(50) NOT NULL,

titolo varchar(25) NOT NULL,

descrizione text NOT NULL,

allegati blob NOT NULL,

archiviato tinyint(4) DEFAULT NULL,

PRIMARY KEY (data),

FOREIGN KEY (lavoro\_id) REFERENCES lavoro(id) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email) REFERENCES utente(email) ON DELETE CASCADE ON UPDATE CASCADE

)

Tabella fattura:

CREATE TABLE fattura (

data datetime NOT NULL,

datore\_email varchar(50) NOT NULL,

lavoratore\_email varchar(50) NOT NULL,

totale int(11) DEFAULT NULL,

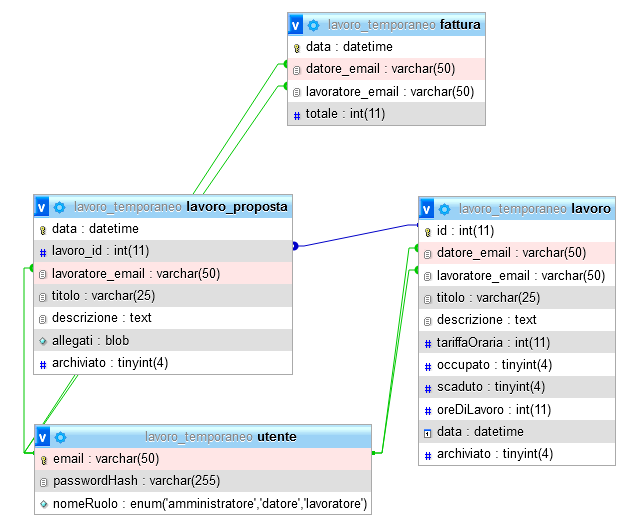
PRIMARY KEY (data),

FOREIGN KEY (datore\_email) REFERENCES utente(email) ON DELETE CASCADE ON UPDATE CASCADE,

FOREIGN KEY (lavoratore\_email) REFERENCES utente(email) ON DELETE CASCADE ON UPDATE CASCADE

)

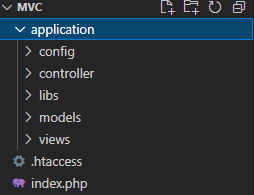
### Implementazione grafica banca dati



## Applicativo Web

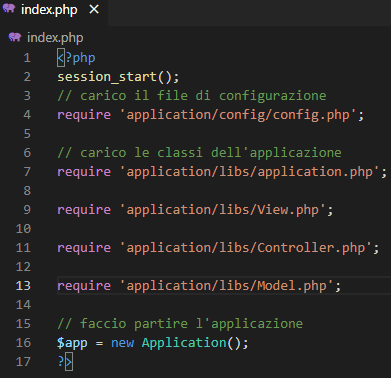
### Struttura cartelle

Per funzionare questo applicativo si basa un MVC, esso necessita di alcuni file di configurazione molto importanti e una struttura ben precisa. Deve contenere una struttura di cartelle (config, controller, libs, models e views) e avere i file .htaccess e index.php al di fuori della struttura di cartelle (x applicativo).



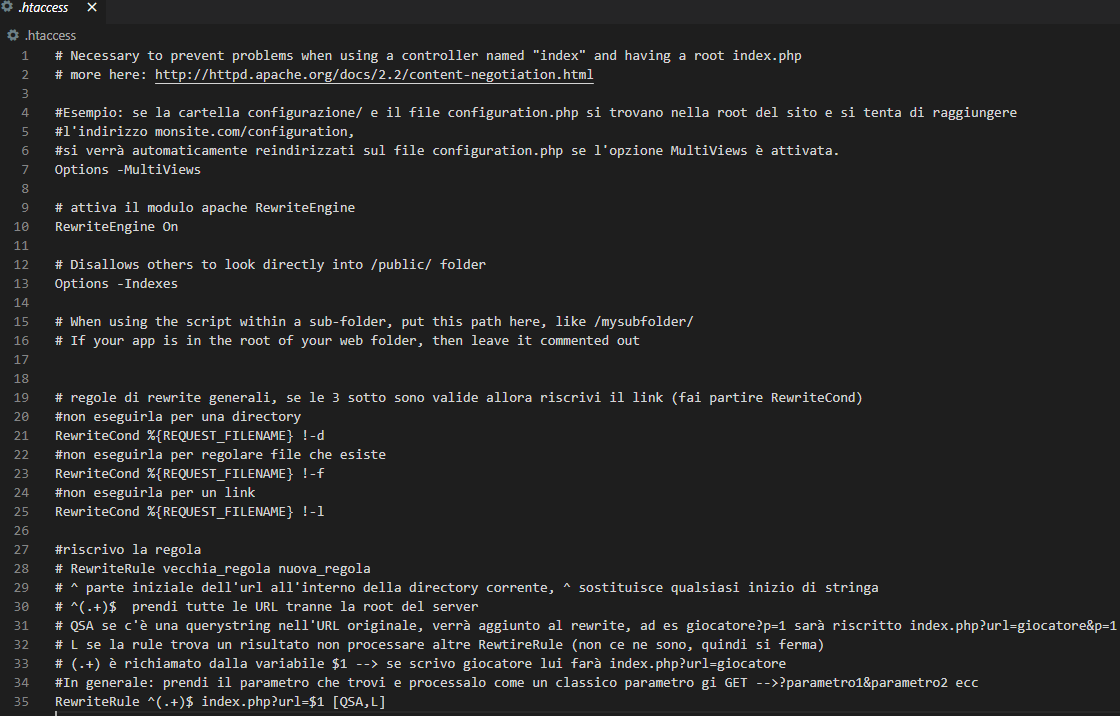
### File index.php

Il file index.php permette di creare il file base sul quale si basa poi tutto l’applicativo, infatti esso permette di includere tutte le librerie e i file di configurazione, inizializza la sessione e inoltre crea l’oggetto applicazione.



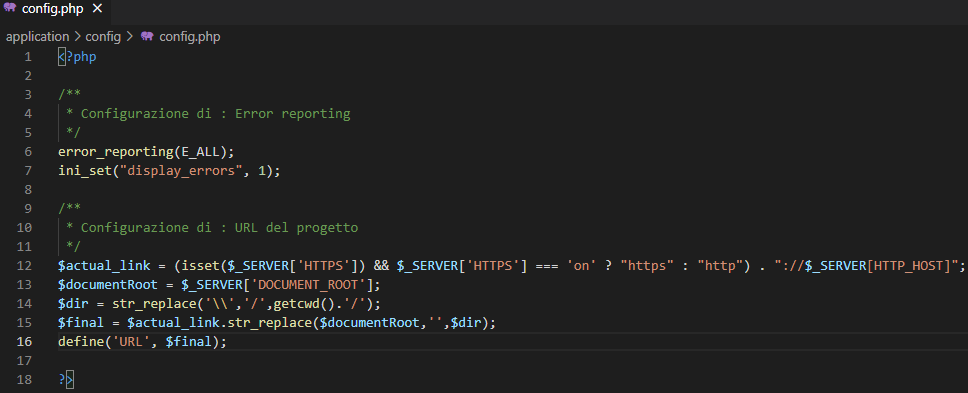
### File .htaccess

Il file .htaccess è necessario per prevenire problemi quando si utilizza un controller chiamato "index" e con un root index.php.



### Cartella config

Nella cartella config c’è il file config.php, dove in esso è contenuto il codice necessario alla configurazione degli errori e del URL. URL è una variabile super globale che viene utilizzata per tutti i riferimenti interni.



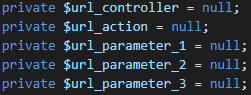
### Cartella libs

Nella cartella libs ci sono tutti i file che contengono le varie classi che necessita un sistema basato su MVC ovvero application, model, view e controller. In questa cartella sono contenuti i file seguenti.

#### File application.php

Il file application.php contiene la classe Application, essa viene inizializzata nel file index.php visto in precedenza. Questa classe è necessaria per splittare l’url e tramite questo sistema richiamare un determinato controller, un metodo e da uno a tre parametri, inoltre se non dovesse esistere quello che si cerca nella barra di ricerca si verrà direttamente reindirizzati al controller della pagina index.php. Dalla classe Application si possono descrivere i metodi e gli attributi che seguono:

* l’attributo url\_controller è utile per conservare il controller al quale ci riferiamo.
* l’attributo url\_action è utile per conservare il metodo del controller.
* I 3 attributi parametri sono utili da passare al metodo del controller.



* Il metodo costruttore richiama il metodo splitUrl, che permette di dividere l’url e assegnare alle variabili citate in precedenza il valore corrispettivo, il costruttore invece serve per chiamare fisicamente il controller, il metodo e gli attributi che sono assegnati alle variabili e in un caso di errore esso ricaricherebbe il controller index.



* Il metodo splitUrl che permette di dividere l’url correttamente le informazioni date, validarle e assegnare alle variabili citate in precedenza il valore corrispettivo.



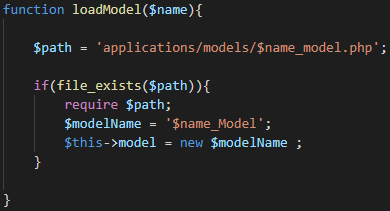
#### File Controller.php

Il file Controller.php contiene la classe Controller, essa è necessaria da estendere in ogni Controller che si va a creare. Questa classe contiene due metodi:

* Il metodo costruttore che permette di creare una nuova view.

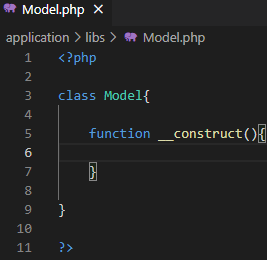


* Il metodo loadModel che permette di implementare un model, dato il nome come parametro.



#### File Model.php

Il file Model.php contiene la classe Model, essa è necessaria da estendere in ogni Model che si va a creare. Questa classe contiene un costruttore vuoto, essa è comunque esistente per il futuro in caso di aggiornamenti.



#### File View.php

Il file View.php contiene la classe View, essa è non necessaria da estendere in ogni View ma viene invece utilizzata nei controller per richiedere la View alla quale ci si riferisce. Questa classe contiene un costruttore vuoto e un metodo chiamato render utile per permettere di includere nella pagina del controller la view fisica.

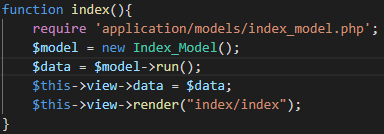


### Cartella Controller

Nella cartella Controller ci sono tutti i file che contengono le varie classi controller che servono per prendere i dati da una view e darli ad un model, per poi riprenderli dal model e ridarli alla view. In questa cartella sono contenuti i file seguenti.

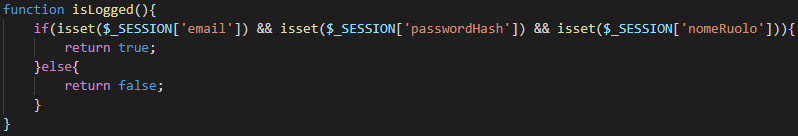
#### File Index.php

Questo file contiene la classe Index che estende la classe controller. Questa classe ha due metodi uno è il costruttore che fa riferimento al costruttore della super classe e l’altro è il metodo index che include il file contenente la classe Index\_model, successivamente viene creato un oggetto di tipo Index\_model, dopo viene richiamato il metodo run del Index\_model, poi viene passato il suo risultato ad una variabile contenuta nella view e infine viene reindirizzata la pagina index.



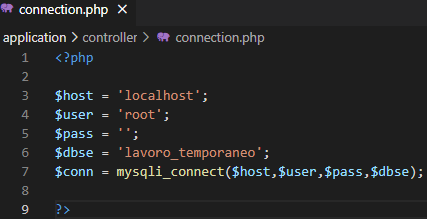
#### File session.php

Questo file contiene la classe session che estende la classe controller. Questa classe ha due metodi uno è il costruttore vuoto e l’altro è il metodo isLogged, questo metodo ritorna true se sono settate le variabili session email, passwordHash e nomeRuolo.



#### File connection.php

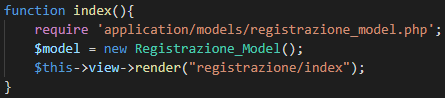
Questo file contiene tutte le variabili necessarie per creare la connessione al database e un oggetto contenente la connessione fisica.



#### File registrazione.php

Questo file contiene la classe Registrazione che estende la classe controller. E i metodi index, rerender, test e testinput.

* Il metodo index è utile per reindirizzare la pagina registrazione.php.



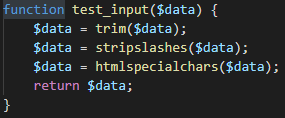
* Il metodo rerender è utile per reindirizzare al controller Accesso.



* Il metodo test è utile per testare e validare i dati immessi per la registrazione, se sono corretti  
  viene chiamato un metodo per inserirli nel database, nell’altro caso vengono dati degli errori.



* Il metodo testinput è utile per testare e validare il parametro stringa passato al metodo.



### Struttura

La struttura delle cartelle è rimasta identica a quella della progettazione (MVC).

## Pagina Index

La pagina Index.php è contenuta nella cartella Index la quale contiene anche il file style.css per lo stile della pagina index.php. Questa pagina viene aperta tramite la classe controller Index, che oltre ad aprire la pagina mette in comunicazione quest’ultima detta View e il model\_index che è una classe che ha tutti i metodi necessari per comunicare con il database. Di seguito ci saranno tutte le spiegazioni necessarie per capire il funzionamento delle varie classi e file.

Continuare da <file:///C:/Users/samuele.abba/Downloads/Documentazione%20-%20Ritardi%20Allievi%20SAMT.pdf> pagina 26

# Test

## Protocollo di test

|  |  |
| --- | --- |
| **Prerequisiti base:** | 1. Essere al punto 1.5 del Gantt 2. Avere XAMPP 3. Avere il database 4. Avere le pagine web |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Verifica del sito web senza login |
| **Descrizione:** | È necessario verificare che il sito web funzioni correttamente e senza essere loggati visualizzare i lavori disponibili. | | |
| **Prerequisiti:** | 1. Prerequisiti base | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che si possono vedere tutte le offerte di lavoro. | | |
| **Risultati attesi:** | 1. Vedere tutte le offerte di lavoro senza essere loggati. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-0020 | **Nome:** | Verifica del sito web con login |
| **Descrizione:** | È necessario verificare che il sito web funzioni correttamente essendo loggati. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale e le pagine per l’amministratore, il datore di lavoro ed il cliente | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che si possa effettuare il login nel ruolo di amministratore, datore di lavoro o cliente. | | |
| **Risultati attesi:** | 1. Vedere in base al ruolo la rispettiva pagina web. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-0021 | **Nome:** | Verifica del funzionamento di: form di registrazione |
| **Descrizione:** | È necessario verificare che il sito web permetta ad un nuovo possibile utente di registrarsi. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di registrazione | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone registrati si apra una pagina con il form di registrazione 5. Controllare che la registrazione sia avvenuta correttamente con e-mail | | |
| **Risultati attesi:** | 1. Possibilità di fare la registrazione e conferma tramite e-mail. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-0022 | **Nome:** | Verifica del funzionamento di: form di accesso |
| **Descrizione:** | È necessario verificare che il sito web permetta ad un utente di accedere. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. | | |
| **Risultati attesi:** | 1. Riuscire ad entrare nella pagina corretta. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-003 | **Nome:** | Verifica risultato del login |
| **Descrizione:** | È necessario verificare che dopo il login l’utente possa accedere alla sua corretta sezione. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta | | |
| **Risultati attesi:** | 1. Riuscire ad entrare nella pagina corretta. 2. Vedi per REQ-0031, REQ-0032, REQ-0033. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-0031 | **Nome:** | Si possano creare richieste. (se si è lavoratori) |
| **Descrizione:** | È necessario verificare che dopo il login l’utente con il ruolo di cliente possa accettare le offerte. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta 7. Possibilità di accettare le offerte di lavoro | | |
| **Risultati attesi:** | 1. Riuscire a fare una richiesta per un lavoro. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-0032 | **Nome:** | Si possano creare offerte. (se si è datori di lavoro) |
| **Descrizione:** | È necessario verificare che dopo il login l’utente con il ruolo di datore di lavoro possa creare le offerte. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta 7. Possibilità di creare le offerte di lavoro | | |
| **Risultati attesi:** | 1. Riuscire a creare le offerte di lavoro. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-0033 | **Nome:** | Si possano gestire le offerte. (se si è amministratori) |
| **Descrizione:** | È necessario verificare che dopo il login l’utente con il ruolo di amministratore possa gestire le offerte di lavoro. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta 7. Possibilità di gestire le offerte di lavoro | | |
| **Risultati attesi:** | 1. Riuscire a gestire le offerte di lavoro. | | |
| **glossario** | 1. Gestire: possibilità di modificare. | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-0041 | **Nome:** | Verifica fine del lavoro. |
| **Descrizione:** | È necessario verificare che dopo il login l’utente con il ruolo di amministratore possa gestire le offerte di lavoro. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta 7. Possibilità di gestire le offerte di lavoro 8. Possibilità di verificare che un lavoro sia finito | | |
| **Risultati attesi:** | 1. Vedere i lavori scaduti occupati oppure tutti i lavori | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-0042 | **Nome:** | Calcolo spese del datore di lavoro. |
| **Descrizione:** | È necessario verificare che dopo il login l’utente con il ruolo di amministratore possa gestire le spese per il datore di lavoro. | | |
| **Prerequisiti:** | 1. Prerequisiti base 2. Avere la pagina web principale 3. Aver creato il form di accesso 4. Aver creato la pagina per il cliente 5. Aver creato la pagina per il datore di lavoro 6. Aver creato la pagina per gli amministratori | | |
| **Procedura:** | 1. Accendere tutti i servizi XAMPP (…) 2. Recarsi sulla pagina web principale 3. Controllare di non essere loggati 4. Verificare che cliccando sul bottone accedi si apra una pagina con il form di accesso 5. Controllare che l’accesso sia avvenuto correttamente. 6. Avere accesso alla pagina corretta 7. Possibilità di calcolare le spese per il datore di lavoro | | |
| **Risultati attesi:** | 1. Riuscire a calcolare le spese di lavoro per i datori e i collaboratori | | |

## Risultati test

|  |  |  |  |
| --- | --- | --- | --- |
| Test Case | Funzionamento | Commento | Data |
| TC-001 | OK | Quando accedo al sito web senza effettuare il login vedo tutti i lavori proposti dal sito web. | 09.12.2021 |
| TC-002  REQ-0020 | OK | Quando effettuo il login al sito web, a dipendenza del ruolo al quale appartengo vedrò la pagina corrispettiva.  Nel caso dell’amministratore vedrò:    Nel caso del datore vedrò:    Nel caso del lavoratore vedrò: | 09.12.2021 |
| TC-002  REQ-0021 | OK | Quando si registra una nuova persona, le viene inviata una mail che conferma di aver creato il suo profilo per il sito del lavoro temporaneo. | 09.12.2021 |
| TC-002  REQ-0022 | OK | Quando accede una persona, le viene la rispettiva pagina per il sito del lavoro temporaneo, vedi risultato test TC-002 REQ-0020. | 09.12.2021 |
| TC-003  REQ-0031 | OK | Quando si accede con il ruolo lavoratore, si può creare una richiesta per un determinato lavoro. | 09.12.2021 |
| TC-003  REQ-0032 | OK | Quando si accede con il ruolo datore di lavoro, si può creare un’offerta di lavoro. | 09.12.2021 |
| TC-003  REQ-0033 | OK | Quando si accede con il ruolo di amministratore, si possono gestire tutti i lavori (modificarli) se non già archiviati, in quel caso sarà possibile visualizzarli ma non modificarli. | 09.12.2021 |
| TC-004  REQ-0041 | OK | Quando si accede con il ruolo di amministratore, si possono vedere tutti i lavori scaduti occupati oppure tutti i lavori:  Nel caso dei lavori occupati vedrò:    Nel caso dei lavori scaduti vedrò:    Nel caso di tutti i lavori vedrò: | 09.12.2021 |
| TC-004  REQ-0042 | OK | Quando si clicca su crea fattura, potremo vedere tutti i lavori che bisogna fatturare. Cliccando successivamente sul bottone calcola fatturazione, verrà creata la fattura per il datore e per l’amministratore. | 09.12.2021 |

## Mancanze/limitazioni conosciute

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

# Consuntivo

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

# Conclusioni

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

## Sviluppi futuri

Migliorie o estensioni che possono essere sviluppate sul prodotto.

## Considerazioni personali

Cosa ho imparato in questo progetto? ecc.

# Bibliografia

## Bibliografia per articoli di riviste:

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

## Bibliografia per libri

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

## Sitografia

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

**Esempio:**

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

# Allegati

Elenco degli allegati, esempio:

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