Gestione salone

1 Introduzione 3

1.1 Informazioni sul progetto 3

1.2 Abstract 3

2 Analisi 4

2.1 Analisi del dominio 4

2.2 Analisi e specifica dei requisiti 4

2.3 Analisi dei mezzi 6

2.3.1 Software 6

2.3.2 Hardware 6

3 Progettazione 7

3.1 Design dei dati e database 7

4 Implementazione 8

5 Test 9

5.1 Protocollo di test 9

5.2 Risultati test 10

6 Conclusioni 11

6.1 Sviluppi futuri 11

6.2 Considerazioni personali 11

7 Glossario 12

8 Sitografia 13

# Introduzione

## Informazioni sul progetto

**Titolo**: Gestione salone

**Allievi coinvolti nel progetto**:

* Andrea Curti, [andrea.curti@samtrevano.ch](mailto:andrea.curti@samtrevano.ch)
* Mattia Pasquini, [mattia.pasquini@samtrevano.ch](mailto:mattia.pasquini@samtrevano.ch)

**Classe**: I4AA Scuola Arti e Mestieri Trevano, Informatica

**Formatore**: Massimo Sartori, [massimo.sartori@samtrevano.ch](mailto:massimo.sartori@samtrevano.ch)

**Data** **inizio**: 05.10.2022

**Data** **fine**: 21.12.2022

## Abstract

All’inizio di questo progetto c’è stato chiesto di realizzare un applicativo web che permettesse di facilitare la gestione dei clienti di un salone di parrucchiere. Con questo prodotto è possibile gestire i servizi effettuati da un cliente con i loro relativi acquisti dei prodotti. I pagamenti sono suddivisi in tipologie, come contanti, bonifico…

# Analisi

## Analisi del dominio

Questo prodotto verrà utilizzato dalle commesse del salone.

Serve per gestire e visualizzare i propri clienti con i relativi servizi effettuati e i prodotti comprati.

L’applicativo è semplice da utilizzare con un’interfaccia grafica intuitiva.

## Analisi e specifica dei requisiti

|  |  |
| --- | --- |
| ID: REQ-001 | |
| Nome | Creazione database |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessitano i permessi di root. |
| Sotto requisiti | |
| 001 | Installazione MySQL. |

|  |  |
| --- | --- |
| ID: REQ-002 | |
| Nome | Pagina di login |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessita di una pagina dove effettuare il login. |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login. |
| 002 | Si necessita un bottone di conferma. |

|  |  |
| --- | --- |
| ID: REQ-003 | |
| Nome | Pagina home |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessita una pagina home per gli utenti |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login. |

|  |  |
| --- | --- |
| ID: REQ-004 | |
| Nome | Pagine gestione clienti |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessitano delle pagine per la gestione dei clienti. |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login. |
| 002 | Si necessita una pagina per l’aggiunta di un cliente. |
| 003 | Si necessita una pagina per l’eliminazione di un cliente. |
| 004 | Si necessita una pagina per la modifica di un cliente. |

|  |  |
| --- | --- |
| ID: REQ-005 | |
| Nome | Pagine gestione servizi |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessitano delle pagine per la gestione dei servizi. |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login. |
| 002 | Si necessita una pagina per l’aggiunta di un servizio. |
| 003 | Si necessita una pagina per l’eliminazione di un servizio. |
| 004 | Si necessita una pagina per la modifica di un servizio. |

|  |  |
| --- | --- |
| ID: REQ-006 | |
| Nome | Pagine gestione prodotti |
| Priorità | 1 |
| Versione | 1.0 |
| Note | Si necessitano delle pagine per la gestione dei prodotti. |
| Sotto requisiti | |
| 001 | Si necessita una maschera di login. |
| 002 | Si necessita una pagina per l’aggiunta di un prodotto. |
| 003 | Si necessita una pagina per l’eliminazione di un prodotto. |
| 004 | Si necessita una pagina per la modifica di un prodotto. |

## Analisi dei mezzi

### Software

Programmi e software:

* Apache 2.4
* MySQL 8
* PHP 8.1.6
* HTML 5
* CSS 3
* PhpStorm 2022.1.1

### Hardware

Macchina di sviluppo:

* Intel Core I7-7700 3.60 GHz
* 16GB RAM

Macchina di produzione:

* Host Infomaniak fornito dalla scuola

# Progettazione

## Design dell’architettura del sistema

### Model View Controller

Model View Controller (MVC) è una struttura che permette di gestire al meglio tutti i file php. Viene utilizzato nell’ambito web con il linguaggio di scripting PHP. In ogni cartella vengono salvati dei file con compiti differenti.

* **Model:** in questa cartella vengono messe tutte le classi che elaborano i dati e che comunicano con il database
* **View:** in questa cartella vengono messi tutti i file contenenti script html. Non viene messo codice php. Si può usare php in questi file solo per stampare file, quindi è concesso l’utilizzo di “for” e semplici “if”.
* **Controller:** in questa cartella vengono messi tutte le classi che ricevono le richieste da parte del browser. Queste classi elaborano le richieste, prendono i dati necessari dal database grazie ai file nella cartella “model”. Dopo passano gli eventuali dati ai file nella cartella “view” e mandando al browser il file giusto dalla cartella “view”.

### Divisione dei ruoli in MVC

# Implementazione

In questo capitolo dovrà essere mostrato come è stato realizzato il lavoro. Questa parte può differenziarsi dalla progettazione in quanto il risultato ottenuto non per forza può essere come era stato progettato.

Sulla base di queste informazioni il lavoro svolto dovrà essere riproducibile.

In questa parte è richiesto l’inserimento di codice sorgente - Print Screen - di maschere solamente per quei passaggi particolarmente significativi e/o critici.

Inoltre, dovranno essere descritte eventuali varianti di soluzione o scelte di prodotti con motivazione delle scelte.

Non deve apparire nessuna forma di guida d’uso di librerie o di componenti utilizzati. Eventualmente questa va allegata.

Per eventuali dettagli si possono inserire riferimenti ai diari.

# Test

## Protocollo di test

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-001  REQ-001 | **Nome:** | Database funzionante |
| **Descrizione:** | Si controlla se è possibile il collegamento al database | | |
| **Prerequisiti:** | * Avere apache funzionante sulla porta 80 * Avere php funzionante * Avere mysql installato | | |
| **Procedura:** | 1. Visitare localhost 2. Premere invio | | |
| **Risultati attesi:** | Si apre una pagina con scritto “Connected successfully” | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-002  REQ-002 | **Nome:** | Pagina di login funzionante |
| **Descrizione:** | Si controlla se il login viene effettuato correttamente per un amministratore | | |
| **Prerequisiti:** | * Avere il database funzionante * Avere un utente amministratore | | |
| **Procedura:** | 1. Aprire la pagina per il login 2. Inserire il nome utente e le credenziali dell’amministratore 3. Premere il tasto login | | |
| **Risultati attesi:** | Viene mostrata la pagina home dell’amministratore | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-003  REQ-003 | **Nome:** | Pagina home utente funzionante |
| **Descrizione:** | Si controlla se la pagina home dell’utente funziona correttamente | | |
| **Prerequisiti:** | * Avere il database funzionante * Avere un utente | | |
| **Procedura:** | 1. Aprire la pagina per il login 2. Inserire il nome utente e le credenziali 3. Premere il tasto login | | |
| **Risultati attesi:** | Viene mostrata la pagina home | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-004  REQ-004 | **Nome:** | Creare un cliente |
| **Descrizione:** | Si controlla che vengano creati clienti correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per aggiungere un cliente 2. Inserire i vari dati 3. Premere il bottone per aggiungere il cliente | | |
| **Risultati attesi:** | Si riesce a vedere il nuovo cliente nella lista di clienti | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-005  REQ-004 | **Nome:** | Modificare un cliente |
| **Descrizione:** | Si controlla che vengano modificati clienti correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per modificare un cliente 2. Inserire i vari dati 3. Premere il bottone per modificare il cliente | | |
| **Risultati attesi:** | Si riesce a vedere il cliente con le nuove modifiche apportate | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-004 | **Nome:** | Eliminare un cliente |
| **Descrizione:** | Si controlla che vengano eliminati clienti correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per eliminare un cliente 2. Scegliere il cliente da eliminare 3. Premere il bottone per eliminare il cliente | | |
| **Risultati attesi:** | Il cliente non è più presente tra la lista di clienti disponibili | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-006  REQ-004 | **Nome:** | Aggiungere un servizio |
| **Descrizione:** | Si controlla che vengano aggiunti servizi correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per aggiungere un servizio 2. Inserire i vari dati 3. Premere il bottone per aggiungere il servizio | | |
| **Risultati attesi:** | Il nuovo servizio è presente nella lista | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-007  REQ-004 | **Nome:** | Modificare un servizio |
| **Descrizione:** | Si controlla che vengano modificati servizi correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per modificare un servizio 2. Inserire i vari dati 3. Premere il bottone per modificare il servizio | | |
| **Risultati attesi:** | Il servizio ha le informazioni appena modificare nella lista | | |

|  |  |  |  |
| --- | --- | --- | --- |
| **Test Case:**  **Riferimento**: | TC-008  REQ-004 | **Nome:** | Eliminare un servizio |
| **Descrizione:** | Si controlla che vengano eliminati servizi correttamente | | |
| **Prerequisiti:** | * Fare login con un utente | | |
| **Procedura:** | 1. Navigare fino alla pagina per eliminare un servizio 2. Inserire i vari dati 3. Premere il bottone per eliminare il servizio | | |
| **Risultati attesi:** | Il servizio non è più presente nella lista | | |

## Risultati test

Tabella riassuntiva in cui si inseriscono i test riusciti e non del prodotto finale. Se un test non riesce e viene corretto l’errore, questo dovrà risultare nel documento finale come riuscito (la procedura della correzione apparirà nel diario), altrimenti dovrà essere descritto l’errore con eventuali ipotesi di correzione.

# 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.

# Glossario

Inserite una semplice tabella con due colonne che spieghi i termini specifici del progetto (lista dei termini in ordine alfabetico A-Z)

Esempio:

|  |  |
| --- | --- |
| **Termine** | **Descrizione** |
| AJAX | **Asynchronous JavaScript And XML**: una tecnica che permette di eseguire richieste ed ottenere dati da una pagina web in modo asincrono. |
| CSS | **Cascading Style Sheets**: linguaggio che permette di definire il layout e la grafica di una pagina web. |

# 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.