



**Scuola Arti e Mestieri Trevano**

**Sezione informatica**

# **Documentazione Progetto 1**

**Titolo del progetto:** Progetto 1  
**Alunno/a:** Carlo Pezzotti  
**Classe:** I3AC  
**Anno scolastico:** 2018/2019  
**Docente responsabile:** Francesco Mussi, Luca Muggiasca, Adriano Barchi, Elisa Nannini

1	Introduzione .....	3
1.1	Informazioni sul progetto .....	3
1.2	Abstract .....	3
1.3	Scopo .....	3
	Analisi .....	3
1.4	Analisi del dominio .....	3
1.5	Analisi e specifica dei requisiti .....	3
1.6	Analisi dei costi .....	7
1.7	Pianificazione .....	7
1.8	Analisi dei mezzi .....	7
1.8.1	Software .....	7
1.8.2	Hardware .....	7
2	Progettazione .....	8
2.1	Design dell'architettura del sistema .....	8
2.2	Design dei dati e database .....	8
2.3	Design delle interfacce .....	8
2.4	Design procedurale .....	9
3	Implementazione .....	10
4	Test .....	12
4.1	Protocollo di test .....	12
4.2	Risultati test .....	2
4.3	Mancanze/limitazioni conosciute .....	2
5	Consuntivo .....	2
6	Conclusioni .....	3
6.1	Sviluppi futuri .....	3
6.2	Considerazioni personali .....	3
7	Bibliografia .....	3
7.1	Bibliografia per articoli di riviste .....	3
7.2	Bibliografia per libri .....	3
7.3	Sitografia .....	3
8	Allegati .....	3

## 1 Introduzione

### 1.1 Informazioni sul progetto

**Alunno:** Carlo Pezzotti  
**Sede:** Informatico, SAM Trevano, I3AC  
**Docenti:** Luca Muggiasca, Francesco Mussi, Adriano Barchi

### 1.2 Abstract

The project I made is a registration page to win an iPhone XS. The project was created in language, HTML, CSS and PHP. Thanks to the use of a web server installed by me I managed to make the project work in a complete way. In circulation there are many sites similar to mine, but mine is easy, fast and simple. User data is then saved in CSV files

### 1.3 Scopo

Lo scopo di questo progetto è di realizzare una pagina di registrazione, compresa di pagina di benvenuto, che salvi i dati degli utenti che partecipano al sondaggio in file csv. La pagina dovrà avere tutti i controlli necessari per far sì che nessun utente esterno possa inserire dati non validi, o sbagliati. In caso che un utente avesse notato che dei dati inseriti non sono validi avrà la possibilità di modificare il dato

## Analisi

### 1.4 Analisi del dominio

Il cliente vuole una pagina semplice che gli permetta di inserire dei dati e in seguito scriverli all'interno di un file CSV. Nel mercato esistono parecchi form di registrazione, in molteplici linguaggi di programmazione. Nel nostro caso però il form deve essere il più semplice possibile perché i possibili utenti che userà il sito non avrà per forza delle competenze informatiche elevate.

### 1.5 Analisi e specifica dei requisiti

Si tratta di sviluppare un sito web che permetta la registrazione dei dati di un utente, la registrazione deve avvenire dopo la lettura di una pagina riassuntiva dei dati.

I dati devono essere tutti convalidati.

I dati saranno memorizzati in due file di testo in formato CSV

L'aspetto grafico del sito e la disposizione dei campi nel form, nella pagina riassuntiva e nella pagina di riletture è a discrezione degli sviluppatori.

ID REQ 001	
<b>Nome</b>	Computer con sistema operativo
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Il computer deve avere la possibilità di avviare un web server per fare sì che possa girare il sito.
Sotto requisiti	
<b>001</b>	Web server

ID REQ 002	
<b>Nome</b>	Pagina di benvenuto
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	La pagina di benvenuto la sarà la prima pagina internet che l'utente dovrà visualizzare. Verrà mostrata la possibilità di registrarsi Nella pagina home sarà presente un tasto con la possibilità di registrarsi.
Sotto requisiti	
<b>001</b>	La pagina implementerà la possibilità di registrarsi o accedere al sito

ID REQ 003	
<b>Nome</b>	Pagina di registrazione
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	Nella pagina di registrazione verranno mostrati tutti i campi obbligatori e non che l'utente dovrà compilare per poter iscriversi al concorso. Se per caso l'utente non avesse compilato un campo obbligatorio la pagina si refresherà da sola mostrando come messaggio di errore il campo / i campi non compilati.
Sotto requisiti	
<b>001</b>	La pagina di registrazione impemerterà un form di registrazione
<b>003</b>	Una volta aver schiacciato il tasto "Avanti" l'utente verrà portto in una pagina di conferma dati. Dove l'utente potrà cambiare il valore di un campo. O "Cancella" taasto che cancellerà tutti i dati inseriti.
<b>004</b>	Alla fine di tutto ci sarà la pagina di conferma per la registrazione,che porterà l'utente ad una home, o il tasto per modificare i dati..
<b>005</b>	NOME – Obbligatorio – 255 caratteri max
<b>006</b>	COGNOME - Obbligatorio – 255 caratteri max
<b>007</b>	DATA DI NASCITA – Obbligatorio – DD.MM.YYYY
<b>008</b>	N. CIVICO – Obbligatorio – Numeri e Lettere
<b>009</b>	CITTà – Obbligatorio – 255 caratteri max
<b>010</b>	NAP – Obbligatorio – numero
<b>011</b>	NO TELEFONO – Obbligatorio
<b>012</b>	EMAIL – Obbligatorio – <a href="mailto:testo@testo.testo">testo@testo.testo</a>
<b>013</b>	SESSO – Obbligatorio – F o M
<b>014</b>	HOBBY – Falcotativo
<b>015</b>	Professione - Facoltativo

ID REQ 004	
<b>Nome</b>	Validatore
<b>Priorità</b>	1

<b>Versione</b>	1.0
<b>Note</b>	Script lato server che verrà chiamato solo una volta che l'utente dia l'okay di registrazione. Se i campi obbligatori non sono compilati verranno riproposti refreshando la pagina.
<b>Sotto requisiti</b>	
<b>001</b>	Script che verifica ogni volta la validità dei dati

ID REQ 005	
<b>Nome</b>	Conferma dati e pagina di visualizzazione dati
<b>Priorità</b>	2
<b>Versione</b>	1.0
<b>Note</b>	Verranno mostrati all'utente i propri dati inseriti con la possibilità di correggere i dati sbagliati. Se tutto dovesse andare bene allora verrà portato ad una pagina con tutti i dati mostrati.
<b>Sotto requisiti</b>	
<b>001</b>	Pagina che mostrerà tutti i dati di un utente
<b>002</b>	Bottone per tornare alla home

ID REQ 006	
<b>Nome</b>	File Csv
<b>Priorità</b>	1
<b>Versione</b>	1.0
<b>Note</b>	I file csv saranno 2, 1 con tutte le registrazioni legate alla giornata e l'altro con tutte le registrazioni avvenute durante l'arco di vita del sito internet.
<b>Sotto requisiti</b>	
<b>001</b>	File CSV con al suo interno tutti gli utenti. Formattazione "Registrazione_tutte.csv"
<b>002</b>	File CSV con al suo interno tutti gli utenti registrati in un giorno. Formattazione "Registrazione_aaaa-mm-gg.csv"

## 1.6 Analisi dei costi

Categoria	Costo all'ora	Ore Lavrate	Costo Totale
Manodopera	62 CHF		

## 1.7 Pianificazione

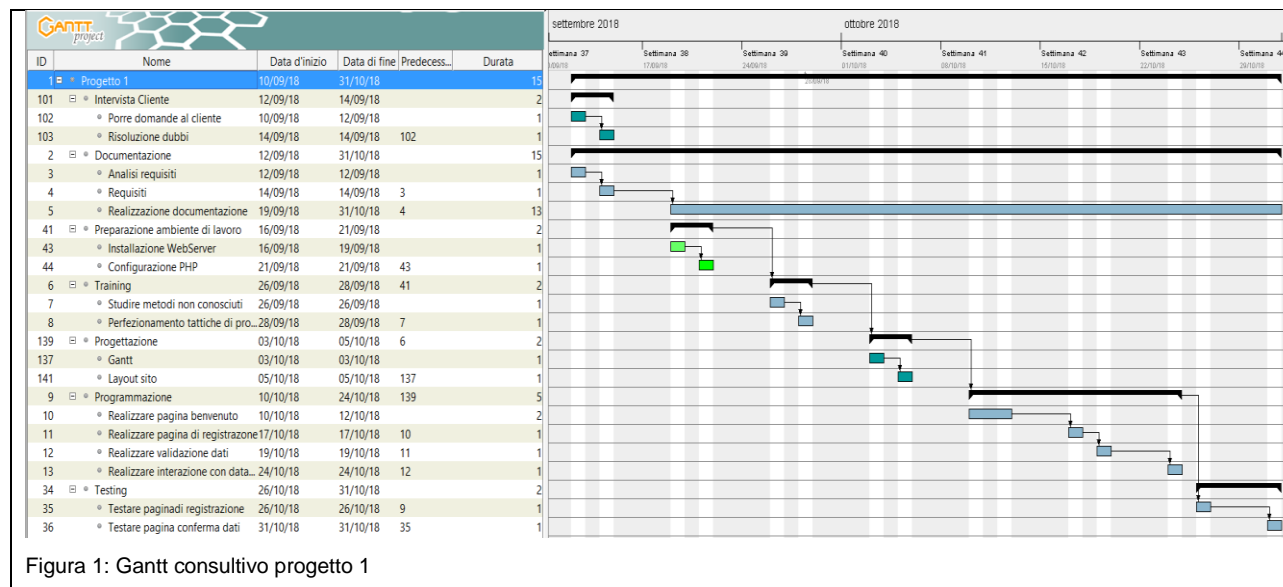


Figura 1: Gantt consultivo progetto 1

L'immagine riportata sopra è il gantt realizzato per la gestione del tempo dell'intero progetto.

## 1.8 Analisi dei mezzi

Per la realizzazione del progetto ho utilizzato un WebServer(Apache24) e php7.

### 1.8.1 Software

I software utilizzati sono Apache 24 per la gestione del webserver e PHP7 per la gestione della pagina lato server. Per la realizzazione del gantt ho utilizzato GanttProject

- **Apache 2.4.27:** Server HTTP che gira su qualsiasi SO (Sistema operativo).
- **PHP 7.1.9:** Linguaggio utilizzato da Apache24 per la gestione di tutte operazioni lato server.
- **GanttProject 2.8.5:** Programma limitato per fare progetti gantt.
- **Visual Code** per la programmazione del codice

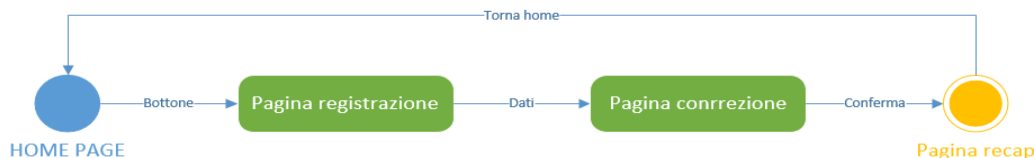
### 1.8.2 Hardware

Il prodotto finale girerà su un sistema *window 10*, verrà gestito però da apache24, software multiplatforma. Per la realizzazione del progetto ho la possibilità di utilizzare le macchine messe a disposizione dall'azienda. La macchina che ho utilizzato per la creazione del progetto è un *HP OMEN*:

- **RAM:** 16 gb
- **Spazio (GB):** 1T;
- **Processore:** Intel(R) Core(TM) i7-6700HQ CPU @ 2.60GHz, 2592 Mhz, 4 core, 8 processori logici
- **Scheda Grafica:** Nvidia Geforce 985M

## 2 Progettazione

### 2.1 Design dell'architettura del sistema



### 2.2 Design dei dati e database

Al posto di utilizzare un database, verrà utilizzato un file CSV:

- File CSV globale con tutti gli utenti registrati durante l'arco di funzionamento del sito
- File CSV giornaliero con il nome degli utenti registrati durante l'arco di una giornata.

Il file CSV verrà formattato con il seguente modo:

*NOME;COGNOME;DATA DI NASCITA;N. CIVICO;CITTÀ;NAP;NO TELEFONO;EMAIL;SESSO;HOBBY;PROFESSIONE*

### 2.3 Design delle interfacce

#### • PAGINA DI BENVENUTO

TITOLO

button

#### • PAGINA DI REGISTRAZIONE

TITOLO

nome	<input type="text"/>
cognome	<input type="text"/>
Sesso	<input type="radio"/> F <input type="radio"/> M
email	<input type="text"/>
N. civico	<input type="text"/>
città	<input type="text"/>
nap	<input type="text"/>
data di nascita	<input type="text"/>
num. telefono	<input type="text"/>
hobby	<input type="text"/>
professione	<input type="text"/>



• **PAGINA DI RECAP**

**TITOLO**

nome  
nome (obbligatorio)

cognome  
cognome (obbligatorio)

sex  
sex (obbligatorio)

email  
email (obbligatorio)

N civico  
N civico (obbligatorio)

città  
città (obbligatorio)

nap  
nap (obbligatorio)

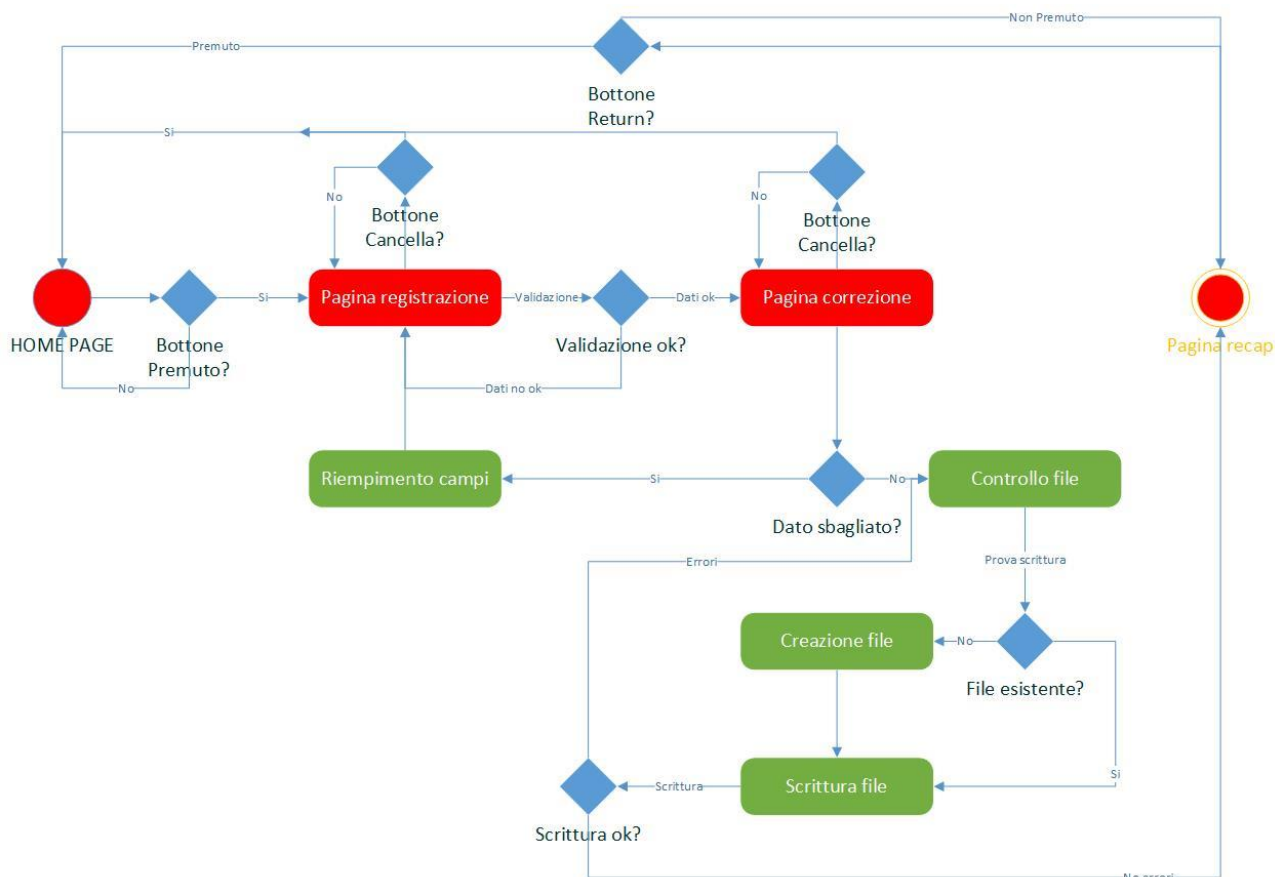
data di nascita  
data di nascita (obbligatorio)

num. telefono  
num. telefono (obbligatorio)

hobby  
hobby (obbligatorio)

professione  
professione (obbligatorio)

**2.4 Design procedurale**



### 3 Implementazione[s1]

Per la realizzazione del progetto ho suddiviso il lavoro in pagine diverse.

#### Benvenuto:

All'inizio viene mostrata una pagina di benvenuto che serve semplicemente a far procedere l'utente alla pagina di registrazione.

#### Registrazione:

```
function checkName(object){
    var regex = /^[a-zA-Zàáâäåãäççèéíîñóôùüöø÷-']+$;/;
    if(object.value.match(regex)){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkData(object){
    var birthdayDate = new Date(object.value);
    var age = getAge(birthdayDate);
    if(age > 64 && age < 100){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function getAge(DOB) {
    var today = new Date();
    var birthDate = new Date(DOB);
    var age = today.getFullYear() - birthDate.getFullYear();
    var m = today.getMonth() - birthDate.getMonth();
    if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
        age--;
    }
    return age;
}
function checkNumber(object){
    object.value = object.value.replace(/ /g, "");
    object.value = object.value.replace(/-/g, "");
    var regexNumber = /^[0-9-]{10,14}$/;
    if(object.value.match(regexNumber)){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkEmail(object){
    var email = /^[a-zA-Z0-9_!#$%&'*+,-./:;=?@^_`{|}~]+@[a-zA-Z0-9_!#$%&'*+,-./:;=?@^_`{|}~]+$/;
    if(object.value.match(email)){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkMap(object){
    var napRegex = /^[0-9]{4,5}$/;
    if(object.value.match(napRegex) && (object.value.length >= 4 && object.value.length <= 5)){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkHouseNumber(object){
    var houseRegexWithText = /^[0-9]{1,4}[A-Z]{1,15}$/;
    var houseRegexOnlyNumber = /^[0-9]{1,4}$/;
    if(object.value.match(houseRegexWithText) || object.value.match(houseRegexOnlyNumber)){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkBox(object){
    if(object.value.length <= 500 && !object.value.includes(";")){
        object.style.backgroundColor = "#33ff33";
        return true;
    }
    else{
        object.style.backgroundColor = "#cc0000";
        return false;
    }
}
function checkAll(){
    if( !checkName(document.getElementById('name')) ||
        !checkName(document.getElementById('surname')) ||
        !checkData(document.getElementById('birthday')) ||
        !checkName(document.getElementById('city')) ||
        !checkHouseNumber(document.getElementById('ncivico')) ||
        !checkName(document.getElementById('via')) ||
        !checkMap(document.getElementById('nap')) ||
        !checkEmail(document.getElementById('email')) ||
        !checkNumber(document.getElementById('telefono')) ||
        !checkBox(document.getElementById('hobby')) ||
        !checkBox(document.getElementById('professione'))
    ){
        alert("Alcuni dati non sono corretti");
    }
    else{
        document.getElementById("formRegistration").submit();
    }
}
```

Il seguente codice è il palo che gestisce la pagina di registrazione. Serve per controllare tutti i campi che vengano inseriti correttamente. Questi controlli vengo fatti interamente lato client. I vari metodi vengono chiamati ogni volta che l'utente scatena l'evento del tasto premuto. Per realizzare i metodo mi sono fatto aiutare dalla pagina online stackoverflow.

### Validazione lato server:

```
$flag = false;
if(checkAll()){
    $flag = true;
    $values = array("name","surname","date","ncivico","city","nap","ntelefono","email","gender","hobby","professione","via");
    echo "<form action='confirm.php' method='POST' id='hiddenForm'>";
    for($i = 0; $i < sizeof($values); $i++){
        echo "<input type='hidden' name='".$values[$i]."' value='\".htmlspecialchars($_POST[$values[$i]]).\"'>";
    }
    echo "</form>";
    echo "<script>document.getElementById('hiddenForm').submit()</script>";
}else
    header("Location: register.php");
```

In questa pagina momentanea vengono semplicemente riproposti gli stessi controlli fatti lato client solo tradotti in un altro linguaggio, ovvero PHP.

Se tutti i controlli vanno a buon fine allora creo un form invisibile che invia i dati ad un'altra pagina, quella di conferma dei dati.

### Conferma dei dati:

In questa pagina non c'è nesso tipo di codice nuovo, questa pagina serve per stampare i dati inseriti dall'utente e in caso che siano sbagliati viene reindirizzato sulla pagina di registrazione con i dati già inseriti.

### Scrittura dei dati:

```
function writeCsv($text,$destination){
    if(!file_exists($destination)){
        $myfile = fopen("newfile.txt", "w") or die("Unable to open file!");
    }
    $myfile = fopen($destination, "a+") or die("Unable to open file!");
    file_put_contents($destination,$text."\\r",FILE_APPEND);
    fclose($myfile);
}
```

Pagina anch'essa momentanea che serve per scrivere all'interno dei file csv che sono presenti nella cartella registrazione. Ho suddiviso i diversi

campi con un ; tra di loro. Nel caso che il file non esista ancora lo creo.

### Pagina finale di ringraziamento:

La logica che sta dietro questa pagina è sempre la stessa che ho addotato nella stampa dei campi di conferma nella pagina di conferma. Solo che ripropongo il ciclo per ogni riga che trovo nel file così da stampare ogni iscrizione che c'è stata giornalmente.

## 4 Test

### 4.1 Protocollo di test

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Controlli lato client
<b>Riferimento:</b>	REQ-003		
<b>Descrizione:</b>	Test per verificare che i dati inseriti dall'utente siano corretti e che rispettano la forma desiderata.		
<b>Prerequisiti:</b>	L'utente deve avere schiacciato il bottone nella pagina di benvenuto.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Provare ad inserire dei dati giusti e fasulli. Nel caso siano fasulli allora la cassella di desto dovrebbe colorarsi di rosso.</li> <li>2. Provare ad inserire tutti dati giusti e procedere con l'iscrizione.</li> <li>3. Provare ad inserire dati fasulli e procedere con l'iscrizione.</li> </ol>		
<b>Risultati attesi:</b>	Entri in atto la pagina di validazione lato sever. E di seguito la pagina di recapilazione dei file.		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Controlli lato client
<b>Riferimento:</b>	REQ-004		
<b>Descrizione:</b>	Test per verificare che i dati siano stati inseriti correttamente.		
<b>Prerequisiti:</b>	-		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Provare ad iscriversi da un altra pagina puntando come la pagina di validazione dei dati.</li> <li>2. Se i dati sono giusti allora si verrà reindirizzati nelle prossima pagina.</li> <li>3. Se i dati sono sbagliati allora verrà riproposto di reinserire i dati.</li> </ol>		
<b>Risultati attesi:</b>	E di seguito la pagina di recapilazione dei file.		

<b>Test Case:</b>	TC-003	<b>Nome:</b>	File csv
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Test per verificare l'inserimento dei dati sia fatto secondo le regole scritte nel codice e che vengano salvati sul file CSV.		
<b>Prerequisiti:</b>	I dati devono essere tutti controllati e validati per la scrittura.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Recarsi nella pagina di registrazione e inserire I dati</li> <li>2. Cliccare il bottone per la registrazione</li> <li>3. Verificare che non si riscontrino errori nella pagina</li> <li>4. Controllare che il file sia stato riempito e in caso non esisteva controlloare che sia stato creato e riempito. NOME;COGNOME;DATA DI NASCITA;N. CIVICO;CITTÀ;NAP;NO TELEFONO;EMAIL;SESSO;HOBBY;PROFESSIONE</li> </ol>		
<b>Risultati attesi:</b>	Tutti I dati nel file csv		

## 4.2 Risultati test

### Test case 1:

Nome:\* Carlo  
Cognome:\* Pezzotti  
Data Di Nascita:\* 01/12/2001  
Città:\* Capolago  
Via:\* Via Lavaggio  
Numero civico:\* 9  
Nap:\* 6825  
Email:\* carlo.pezzotti@samtrevano.ch  
Sesso:\* ☒ Maschio ☐ Femmina  
Numero di telefono:\* +41788159957  
Hobby:  
Professione:

Risultato test 1 lato client. Tutti i dati sono corretti quindi le caselle di testo vengono colorate di verde come specificato nel test case.

### Test case 2:

Purtroppo il test case non ha dei risultati grafici evidenti quindi non posso illustrare il corretto funzionamento. Però dato che la realizzazione del modulo è stata una semplice traduzione dei controlli effettuati lato client sono certo del suo funzionamento.

### Test case 3:

	Registrazione_20181024.csv	26/10/2018 13:13
	Registrazione_20181026.csv	26/10/2018 14:20
	Registrazione_20181029.csv	29/10/2018 13:23
	Registrazione_20181101.csv	01/11/2018 13:28
	Registrazioni_tutte.csv	01/11/2018 13:28

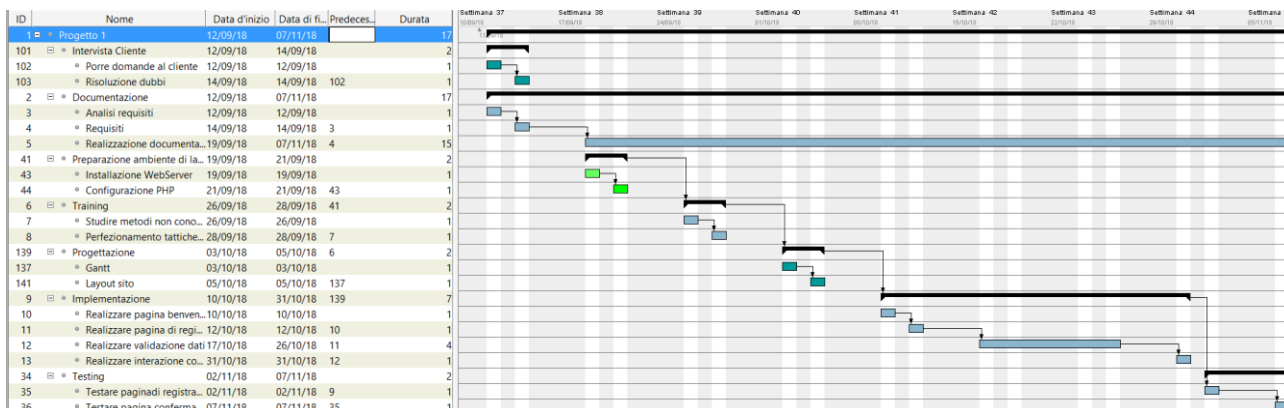
Il file è stato creato correttamente e i campi al suo interno pure.

1 Carlo;Pezzotti;2001-12-01;9;Capolago;Via Lavaggio;6825;+41788159957;carlo.pezzotti@samtrevano.ch;Femmina;

## 4.3 Mancanze/limitazioni conosciute

Il prodotto non funziona se utilizzato come browser Internet Explorer.

## 5 Consuntivo



## 6 Conclusioni

La realizzazione di questo mio primo progetto serio, mi ha fatto capire che dare il via ad un progetto non è un cosa molto facile perchè bisogna organizzarsi parecchio su tutto il da fare nel periodo di vita della doc. Una cosa che non avrei mai preso in considerazione è stato l'utilizzo del diario giornaliero, che mi ha fatto capire come fossi messo un po' con il lavoro, e se avrei dovuto accelerare o restare costante con la realizzazione. Trovo ancora difficile utilizzare il diagramma di gant, perrò anch'esso mi ha dato una grandissima mano.

Infine sono maturato parecchio a livello produttivo e questo mi fa molto piacere.

### 6.1 Sviluppi futuri

Molto probabilmente in futuro non ci saranno sviluppi perchè, secondo me, il prodotto è già concluso. Nel caso che degli utenti finali dovessero trovare dei bug all'interno del progetto allora potrei pensare di mettere a posto. Quindi solo bug fixing.

### 6.2 Considerazioni personali

Ho imparato a gestire il tempo per la realizzazione di un progetto

## 7 Bibliografia

### 7.1 Bibliografia per articoli di riviste:

1. Pezzotti Carlo
2. "Documentazione Progetto 1"
3. *Documentazione Progetto 1*
4. 01/11/18

### 7.2 Bibliografia per libri

5. Pezzotti Carlo
6. *Documentazione Progetto 1*
1. Edizione 1
2. Carlo Pezzotti
3. 2018

### 7.3 Sitografia

1. <https://www.w3schools.com/>
2. Nel periodo di progettazione del sito
1. <https://stackoverflow.com/>
2. Nel periodo di progettazione del sito
1. <http://www.php.net/>
2. Nel periodo di progettazione del sito

## 8 Allegati

Elenco degli allegati:

- Diari di lavoro
- Codici sorgente
- Mandato e/o Qdc

## Diari

# Diario 05.09.2018

Luogo: Lugano  
Data: 05.09.2018

Oggi abbiamo visto in classe le basi per poter fare un progetto a regola d'arte.  
Le cose fondamentali che dovrebbe avere un progetto sono: "Un Diario", "Una Documentazione",...



# PROGETTO1 | Diario di lavoro - 07/09/2018

**Lavoro svolto:**

Oggi in classe abbiamo visto nei dettagli la documentazione che dovrò svolgere parallelamente al progetto che mi sarà assegnato. Inoltre ho iniziato a sviluppare la bozza del mio progetto. Verso fine lezione abbiamo iniziato a studiare della teoria legata al modulo 306.

Orario	Lavoro svolto
13:15 – 15:30	Descrizione della documentazione
15:30 – 15:45	Inizio bozza scheletro progetto
15:45 – 16:30	Teoria modulo 306

**Problemi riscontrati e soluzioni adottate:**

Nessun problema riscontrato

**Punto della situazione rispetto alla pianificazione:**

Pianificazione non ancora iniziata

**Programma di massima per la prossima giornata di lavoro:**

Aggiungeremo conoscenze alla teoria svolta oggi in classe

Inizierò a scrivere del codice per la realizzazione della bozza del progetto.

# PROGETTO1 | Diario di lavoro - 12/09/2018

**Lavoro svolto:**

In classe abbiamo iniziato a fare le analisi e i requisiti del progetto

Orario	Lavoro svolto
13:15 – 14:30	Descrizione analisi

**Problemi riscontrati e soluzioni adottate:**

Nessun problema

**Punto della situazione rispetto alla pianificazione:**

Nessuna pianificazione fatta

**Programma di massima per la prossima giornata di lavoro:**

Concludere le analisi e i requisiti

# PROGETTO1 | Diario di lavoro - 14/09/2018

**Lavoro svolto:**

Oggi in classe ho concluso le analisi e i requisiti legati al progetto

Orario	Lavoro svolto
13:15 – 14:45	Completamento delle analisi e requisiti
15 – 16:30	Inizio progettazione gant

**Problemi riscontrati e soluzioni adottate:**

Complicazione nel capire il formato dei requisiti

**Punto della situazione rispetto alla pianificazione:**

Nessuna pianificazione fatta

**Programma di massima per la prossima giornata di lavoro:**

Pianificazione

# PROGETTO1 | Diario di lavoro - 19/09/2018

**Lavoro svolto:**

Durante la lezione di oggi ho realizzato il gantt preventivo. E mi sono stati corretti i requisiti fatti la lezione precedente

Orario	Lavoro svolto
13:15-14:50	Progettazione gantt

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

Metà del lavoro

**Programma di massima per la prossima giornata di lavoro:**

Inizio progettazione

# PROGETTO1 | Diario di lavoro - 21/09/2018

**Lavoro svolto:**

Oggi in classe ho iniziato a fare la documentazione, ho finito completamente il gantt consuntivo.  
Inoltre abbiamo fatto un po' teori legata alla progettazione.  
Installazione del WebServer;

Orario	Lavoro svolto
13:15-14:45	Teoria progettazione
15:16:30	Documentazione e WebServer PHP

**Problemi riscontrati e soluzioni adottate:**

Nessuno problema riscontrato

**Punto della situazione rispetto alla pianificazione:**

Circa 50% del progetto

**Programma di massima per la prossima giornata di lavoro:**

Continuazione della documentazione

# PROGETTO1 | Diario di lavoro - 26/09/2018

**Lavoro svolto:**

Il lavoro svolto oggi in classe è stato l'espe teorico che riguarda lo sviluppo del progetto

Orario	Lavoro svolto
13:15-14:45	Espe

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

50%

**Programma di massima per la prossima giornata di lavoro:**

Continuo progettazione

# PROGETTO1 | Diario di lavoro - 28/09/2018

**Lavoro svolto:**

Completazione e progettazione. Inoltre ho concluso il diagramma di flusso e procedurale

Orario	Lavoro svolto
13:15-16:30	Diagramma di flusso-procedurale e progettazione

**Problemi riscontrati e soluzioni adottate:**

Incompetenza nella creazione del diagramma di flusso

**Punto della situazione rispetto alla pianificazione:**

Più della metà

**Programma di massima per la prossima giornata di lavoro:**

Documentazione

# PROGETTO1 | Diario di lavoro - 03/10/2018

**Lavoro svolto:**

Oggi in classe abbiamo svolto la teoria sulla progettazione

Orario	Lavoro svolto
13:15-14:45	Teoria progettazione

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

Progettazione

**Programma di massima per la prossima giornata di lavoro:**

Progettazione pagina benvenuto



# PROGETTO1 | Diario di lavoro - 05/10/2018

**Lavoro svolto:**

[TEXT]

Orario	Lavoro svolto
13:15-13:45	Creazione primo test case
13:45-16:30	Progettazione grafica sito

**Problemi riscontrati e soluzioni adottate:**

Nessun problema

**Punto della situazione rispetto alla pianificazione:**

Progettazione

**Programma di massima per la prossima giornata di lavoro:**

Progettazione

# PROGETTO1 | Diario di lavoro - 10/10/2018

**Lavoro svolto:**

Miglioramenti estetici ll'interfaccia grafica

Orario	Lavoro svolto
13:15-14:45	Miglioramenti estetici

**Problemi riscontrati e soluzioni adottate:**

no

**Punto della situazione rispetto alla pianificazione:**

In orario con la progettazione

**Programma di massima per la prossima giornata di lavoro:**

Validazione server

# PROGETTO1 | Diario di lavoro - 12/10/2018

## Lavoro svolto:

Completamento grafica siti internet

Implementazione controlli lato client utilizzando l'utilizzo del seguente sito:

<https://stackoverflow.com/questions/2385701/regular-expression-for-first-and-last-name>

Orario	Lavoro svolto
13:15-14:45	Completamento pagine grafiche Implementazione controlli lato client
15:16:30	Implementazione controlli lato client

## Problemi riscontrati e soluzioni adottate:

Nessuno problema

## Punto della situazione rispetto alla pianificazione:

In tempo

## Programma di massima per la prossima giornata di lavoro:

Implementazione controlli lato server

```
function checkName(object){
  var regex = /^[a-zA-Zàáâãäåæçèéêëìíîïðñóôûü ,.'-]+$ /u;
  var newString = "";
  if(object.value == ""){
    object.style.backgroundColor = "white";
    return;
  }
  if(!regex.test(object.value)){
    for (var i = 0; i < object.value.length-1; i++) {
      newString += object.value[i];
    }
    console.log(newString);
    object.value = newString;
  }
}

function checkData(object){
  var birthdayDate = new Date(object.value);
  var age = getAge(birthdayDate);
  if(age > 5 && age < 100){
    object.style.backgroundColor = "rgb(50,150,50)";
  } else {
    object.style.backgroundColor = "rgb(200,50,50)";
  }
}

function getAge(DOB) {
  var today = new Date();
  var birthDate = new Date(DOB);
  var age = today.getFullYear() - birthDate.getFullYear();
  var m = today.getMonth() - birthDate.getMonth();
  if (m < 0 || (m === 0 && today.getDate() < birthDate.getDate())) {
    age--;
  }
  return age;
}

function checkNumber(object){
  var phone = /^(?([0-9]{3})\)?[-. ]?([0-9]{3})[-. ]?([0-9]{4})$/;
  console.log(object.value);
  if(object.value.match(phone)){
    object.style.backgroundColor = "rgb(50,150,50)";
  } else {
    object.style.backgroundColor = "rgb(200,50,50)";
  }
}
```

Codice piu importante scritto oggi

# PROGETTO1 | Diario di lavoro - 17/10/2018

**Lavoro svolto:**

Ho implementato l'email checker lato client

Orario	Lavoro svolto
3:15-14:45	Implementazione email checker

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

In orario

**Programma di massima per la prossima giornata di lavoro:**

Implementazione

# PROGETTO1 | Diario di lavoro - 24/10/2018

**Lavoro svolto:**

Oggi ho svolto, gran parte del progett. Ho finito i controlli lato server e client e ho iniziato ad implementare la scrittura dei file.

Orario	Lavoro svolto
13:15-14:45	Controlli e inizio scrittura file

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

Orario

**Programma di massima per la prossima giornata di lavoro:**

Scrittura file

# PROGETTO1 | Diario di lavoro - 26/10/2018

**Lavoro svolto:**

Ho concluso l'implementazione, ultimando così l'ultima pagina web.

Orario	Lavoro svolto
13:15-14:45	Implementazione pagina conclusiva
15-16:30	Documentazione

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

In orario

**Programma di massima per la prossima giornata di lavoro:**

Documentazione

# PROGETTO1 | Diario di lavoro - 07/11/2018

**Lavoro svolto:**

Ho terminato la documentazione

Orario	Lavoro svolto
13:15-14:45	Completamento della documentazione

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

In orario

**Programma di massima per la prossima giornata di lavoro:**

Consegna

# PROGETTO1 | Diario di lavoro - 09/11/2018

**Lavoro svolto:**

Concluso il progetto.

Orario	Lavoro svolto
13:15-16:30	Conclusione progetto

**Problemi riscontrati e soluzioni adottate:**

Nessuno

**Punto della situazione rispetto alla pianificazione:**

Finito

**Programma di massima per la prossima giornata di lavoro:**

Niente