

Progetto 1

Titolo del progetto: Progetto 1
Alunno/a: Gübeli Paolo
Classe: Info 3AA
Anno scolastico: 2018/2019
Docenti responsabili: Mussi Francesco, Barchi Adriano, Muggiasca Luca.

1	Introduzione	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
	Analisi	Error! Bookmark not defined.
1.4	Analisi del dominio	4
1.5	Analisi e specifica dei requisiti	4
1.6	Use case	Error! Bookmark not defined.
1.7	Pianificazione	6
1.8	Analisi dei mezzi	6
1.8.1	Software	6
1.8.2	Hardware	6
2	Progettazione	7
2.1	Design dell'architettura del sistema	7
2.2	Design dei dati e database	7
2.3	Design delle interfacce	8
2.4	Design procedurale	9
3	Implementazione	9
4	Test	13
4.1	Protocollo di test	13
4.2	Risultati test	14
4.3	Mancanze/limitazioni conosciute	14
5	Consuntivo	14
6	Conclusioni	15
6.1	Sviluppi futuri	15
6.2	Considerazioni personali	Error! Bookmark not defined.
7	Bibliografia	Error! Bookmark not defined.
7.3	Sitografia	15
8	Allegati	15

1 Introduzione

1.1 Informazioni sul progetto

Questo progetto è stato fatto da Paolo Gübeli allievo della Scuola Arti e Mestiere Trevano sezione Informatica con l'aiuto dei docenti Luca Muggiasca, Francesco Mussi, Adriano Barchi e Elisa Nannini. È stato fatto per imparare come fare un progetto uno dei requisiti del Modulo 306. È stato iniziato il 5.9.2018 Ed è finito il 09-11-2018.

1.2 Abstract

Today everyone has a website, and everyone wants a registration site for the costumers. I created this beautiful registration site that takes the costumers' data and writes it in a .csv file. This site checks if the data is correct with a client side javascript code. This package of websites contains a welcome site, a registration site, a correction site and a last website to check the registration made that day. There are a lot of websites like this one but mine is faster, lighter and has a unique style.

1.3 Scopo

Lo scopo del progetto è di migliorare la nostra conoscenza sui progetti, quindi ci assegnato un lavoro che consiste nel creare un sito di registrazione. Il sito è composto da una pagina di benvenuto con un pulsante registrati che ti porta a una pagina con dei form obbligatori e facoltativi.

1.4 Analisi del dominio

La mancanza di un sito di registrazione rende difficile la gestione dei clienti. Il prodotto funzionerà sul web dove i clienti si potranno registrare. Per questo problema ci sono già molte soluzioni su internet che consiglio visto che sono gratuite.

1.5 Analisi e specifica dei requisiti

ID: REQ-001	
Nome	Pagina di registrazione
Priorità	1
Versione	1.0
Note	I tipi di campo sono a scelta del programmatore
Sotto requisiti	
001	Deve contenere dei campi, quelli con * sono obbligatori. Vedi tabella campi
002	Deve contenere un pulsante Cancella che svuota tutti i campi
003	Deve contenere un pulsante avanti che passa al form di correzione
004	Deve validare i dati quando viene premuto il pulsante avanti

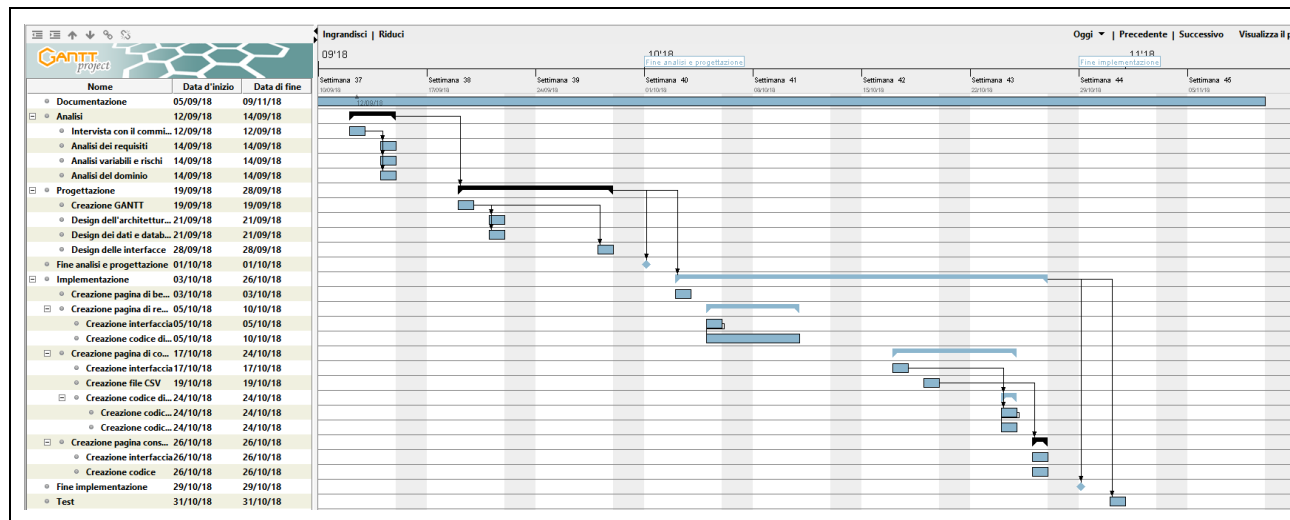
ID: REQ-002	
Nome	Pagina di controllo
Priorità	1
Versione	1.0
Note	La disposizione dei dati è a scelta
Sotto requisiti	
001	Ci dev'essere un pulsante Coreggi che riporta alla pagina del form
002	Ci dev'essere un pulsante Registra che scrive i dati nei file e porta alla pagina riassuntiva
003	Nel momento della registrazione viene aggiunta la data e l'ora

ID: REQ-003	
Nome	Creazione e gestione file CSV
Priorità	1
Versione	1.0
Note	Ogni giorno la prima registrazione crea il file del giorno
Sotto requisiti	
001	I file CSV saranno nella cartella "Registrazioni" nella cartella del sito
002	Bisogna controllare se i CSV sono già esistenti
003	Ci sarà un file con tutte le registrazione di nome "Registrazioni_tutte.csv"
004	Ci sarà un file con le registrazione della giornata di nome "Registrazioni_aaaa-mm-gg.csv"

ID: REQ-005	
Nome	Pagina riassuntiva
Priorità	1
Versione	1.0
Note	
Sotto requisiti	
001	Deve contenere tutti i dati raccolti nella giornata

ID: REQ-006	
Nome	Design dei siti
Priorità	2
Versione	1.0
Note	Design delle pagine a scelta
Sotto requisiti	
001	Ci dev'essere una pagina d'introduzione con un pulsante registrati

1.6 Pianificazione



1.7 Analisi dei mezzi

1.7.1 Software

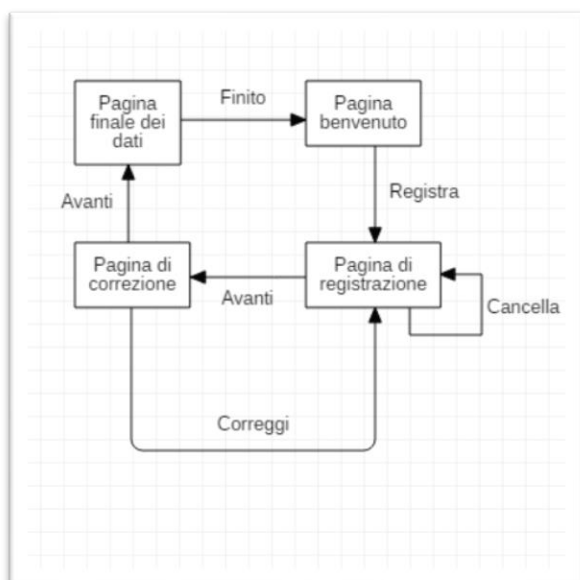
GanttProject 2.8.5 Pilsen (build 2179),
 Microsoft Office Word 2016
 Notepad++ 7.5.1
 VMware Workstation 12 Pro 12.5.7 build-5813279

1.7.2 Hardware

HP Omen
 i7-6700HQ CPU 2.6 GHz
 RAM 16 GB

2 Progettazione

2.1 Design dell'architettura del sistema



2.2 Design dei dati e database

Campi del form:

	Obbligatorio	Tipo	Lunghezza Max.
Nome	Si	Testo	50
Cognome	Si	Testo	50
Data di nascita	Si	Data	Data valida
Via	Si	Testo	50
No. Civico	Si	Testo	4
Città	Si	Testo	50
Nap	Si	Numerico	5
No. Telefono	Si	Testo	12
E-mail	Si	Testo	50
Genere	Si	Testo (M o F)	1
Hobby	No	Testo	500
Professione	No	Testo	500

2.3 Design delle interfacce

Pagina di benvenuto

TESTO DI BENVENUTO

REGISTRATI

Pagina di registrazione

Dati Personali

Nome
 Cognome
 Sesso

Data di nascita

Via
 N. Civico
 Città
 NAP

Contatti

No. Telefono

E-mail

Altro

Hobby

Professione

Pagina di correzione

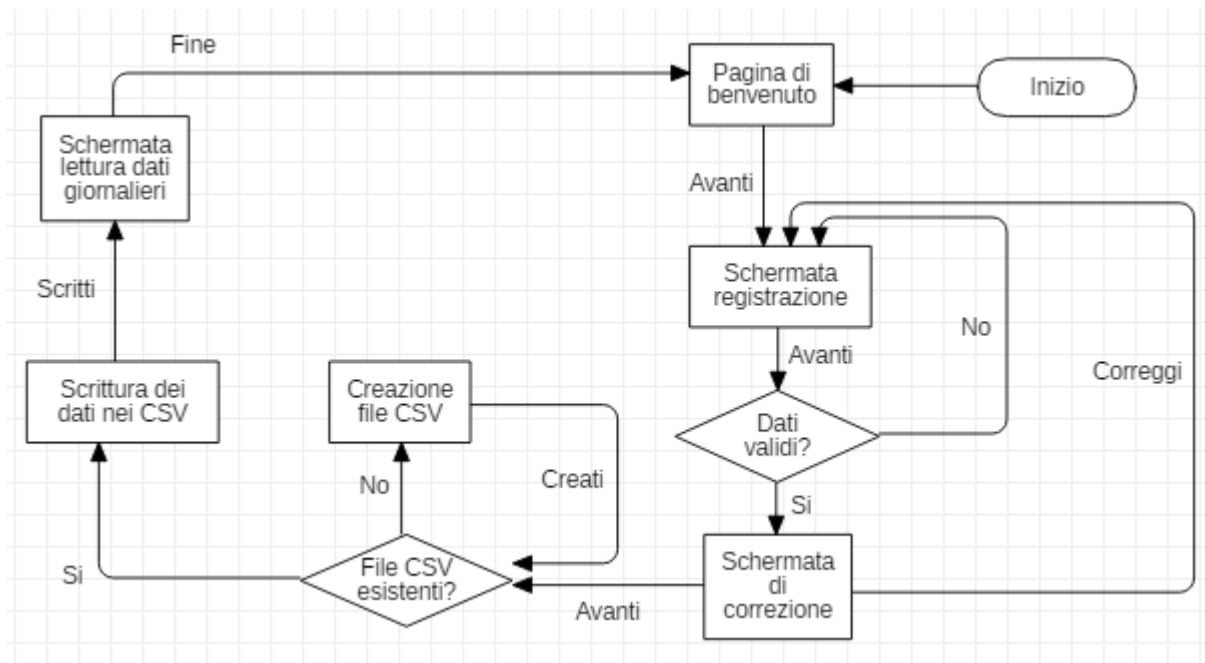
DATI PERSONALI			
Nome	Cognome	Sesso	
Data di nascita			
Via	N. Civico	Città	NAP
CONTATTI			
Numero di telefono			
E-mail			
Hobby			
Professione			

Pagina consulto dati

DATA

Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI
Nome Cognome	VEDI

2.4 Design procedurale



3 Implementazione

3.1 Pagina di benvenuto

La pagina di Benvenuto è molto semplice quindi ho usato solo CSS e Html per implementarla, questa la parte di CSS che definisce lo stile della pagina e il file html che definisce il formato.

```

body{
  background-image:URL("../bg.jpg"); <html>
  <head>
    <link href="https://fonts.googleapis.com/icon?family=Material+Icons" rel="stylesheet">
    <link href="css/style.css" rel="stylesheet">
    <link type="text/css" rel="stylesheet" href="css/materialize.min.css" media=
    "screen,projection">
    <style>
      @font-face {
        font-family: Hallosans;
        src: url("Hallo.otf");
      }
    </style>
  </head>
  <body>
    <div class="main">
      <h1 class="title">WELCOME</h1>
      <a href="Registrazione.php">
        <div class="button">
          <p>SIGN UP</p>
        </div>
      </a>
    </div>
    <script type="text/javascript" src="js/materialize.min.js"></script>
  </body>
</html>
}
.title{
  font-size: 120px;
}
.main{
  margin: 20px 20% 20px 20%;
  padding: 20px;
  border-left: solid #e65100 2px;
  font-family: Hallosans;
}
div .container{
  margin: 100px;
}
div .button{
  border: solid 2px orange;
  border-radius: 200px;
  color: black;
  max-width: 200px;
  min-width: 100px;
  text-align: center;
  margin: 100px 0px 0px 0px;
  font-size: 20px;
}
.right{
  float:right;
}
.left{
  float:left;
}

```

3.2 Pagina di registrazione

Per la pagina di registrazione ho scelto di fare i controlli sulla parte client quindi usando Javascript. Per resettare i valori della pagina c'è il pulsante CANCELLA in basso a sinistra della pagina che richiama un metodo che cancella i valori.

```
<a onclick="deleteAll()">
  <div class="button left">
    <p>DELETE</p>
  </div>
</a>

function deleteAll(){
  document.getElementById("nome").value = "";
  document.getElementById("cognome").value = "";
  document.getElementById("m").checked = true;
  document.getElementById("nome").value = "";
  document.getElementById("via").value = "";
  document.getElementById("ncivico").value = "";
  document.getElementById("citta").value = "";
  document.getElementById("nap").value = "";
  document.getElementById("tel").value = "";
  document.getElementById("email").value = "";
  document.getElementById("hobby").value = "";
  document.getElementById("pro").value = "";
}
```

Per la grafica della pagina ho usato una libreria online di materialize che è disponibile gratuitamente su <https://www.materializecss.com>, questa libreria permette di modificare molte parti della propria pagina io ho solo modificato lo stile degli input creando dei input interattivi e più belli da vedere. Nell'implementazione della libreria ho avuto problemi a gestire il colore dei form ma poi ho scoperto grazie alla documentazione che bisogna usare "important" per sovrascrivere il colore già esistente.

SIGN UP

Name* Last Name* ☒ M ☐ F Birthday dd-mm-yyyy

Street* House Number* City* NAP*

Phone Number* E-mail*

Hobby

3.2.1 Controllo validità dei dati

Per sviluppare i controlli ho usufruito di www.W3School.com per certi metodi particolari. Per esempio per suddividere una stringa in diverse parti per poi estrapolare i numeri ho trovato il metodo `testo.split(char divisore)`. Per verificare che i dati sono corretti ho assegnato una variabile ad ogni parametro e se tutte le variabili sono true compare il pulsante AVANTI che permette di avanzare alla pagina di correzione. Ogni volta che un valore viene modificato controllo se è valido e se lo è controllo se tutte le variabili sono a true. Ogni controllo a una regular expression personalizzata, un controllo se è null o spazi, e un controllo per la lunghezza.

```
var y = true;
var x = document.getElementById("email").value;
if(x != null && x != ""){
    if(x.length <= 50){
        if(x.indexOf("@") != -1){
            var h = x.substring(0,x.indexOf("@"));
            for(var i = 0; i < h.length; i++){
                if(/^[A-Za-z0-9._]/.test(h[i])){
                    y = false;
                }
            }
            g = x.substring(x.indexOf("@")+1, x.length);
            if(g.indexOf(".") != -1 && g.indexOf(".") != g.length-1 && g.indexOf(".") != 0){
                for(var i = 0; i < g.length; i++){
                    if(/^[A-Za-z]/.test(g[i])){
                        y = false;
                    }
                }
            }
            if(y){
                email = true;
                document.getElementById("error").innerHTML = "";
            }
        }
    }
}
```

3.3 Pagina di correzione

La pagina di correzione è stata l'unica parte del progetto per cui ci ho messo meno tempo del previsto, per trasferire i dati dalla registrazione a questa pagina ho usato PHP usando il metodo `$_GET` in seguito la pagina di correzione prende i dati li assegna a delle variabili e le inserisce in una tabella che permette una semplice lettura da parte del utente. Nella pagina in più sono presenti due pulsanti: uno che porta alla pagina precedente e l'altro che scrive i dati nel file CSV e poi apre la pagina finale di lettura dei dati giornalieri. La parte di scrittura è molto breve visto che basta convertire tutti i valori in testo e unirli in un'unica stringa suddivisi dal “;”. In seguito basta usare l'add sul file CSV e aggiungere la nuova riga chiudendo con “\n” per andare a capo e aggiunge all'inizio la data e l'ora della registrazione nel totale e nel giornaliero solo l'ora. Un caso particolare è il maschio e femmina perché ho usato dei radio button quindi mi ritorna o true o false, di conseguenza ho dovuto tradurre dei booleani in “Male/Female”. Infine richiama la pagina finale usando “`window.location.replace`”.

```
<?php
$text = date("H:i:s").";".$nome.";".$cognome.";".$ Sesso.";".$data.";".$via.";".$civico.";".$citta.";".$cap.";".$tel.";".$email.";".$hobby.";".$pro.";
$file = fopen("Registrazioni/Registrazioni_tutte.csv", "a");
$fileDay = fopen("Registrazioni/Registrazioni_".$date("Y-m-d").".csv","a");
if($_SERVER['REQUEST_METHOD'] == 'POST'){
    fwrite($file,date("Y-m-d")."; ".$text);
    fclose($file);
    fwrite($fileDay,$text);
    fclose($fileDay);
    echo "<script>window.location.replace('Finale.php')</script>";
}
```

3.4 Pagina lettura dati giornalieri

Per leggere i dati ho richiamato il CSV in modalità read e ho suddiviso i dati per righe per poi ogni campo, in seguito gli ho inseriti nella casella apposita della tabella che viene creata per ogni utente. E poi ho semplicemente aggiunto un pulsante in fondo alla pagina che riporta all'index.

```
12:21:16;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:22:43;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:26:14;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:34:11;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:37:23;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:40:20;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:43:15;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
12:46:01;a;a;Male;a;a;1;a;l;121212121;a.a@a.a;a;a
13:05:17;Paolo;Guebeli;Male;24.10.2001;Via al fango;21;Lugang;2132;0321312321;gigi@gigione.ch;Mi piace giocare a carte;Gioco a solitario competitivo
2018-10-26 13:08:13;Paolo;Guebeli;Male;24.10.2001;Via al fango;21;Lugang;2132;0321312321;gigi@gigione.ch;Mi piace giocare a carte;Gioco a solitario competitivo
2018-10-26 13:18:17;Paolo;Guebeli;Male;24.10.2001;Via al fango;21;Lugang;2132;0321312321;gigi@gigione.ch;Mi piace giocare a carte;Gioco a solitario competitivo
2018-10-26 13:27:07;Matteo;Ghila;Male;21-9-2001;a;l;a;l;121;212121212;email@email.com;
2018-10-26 13:56:11;Gigi;Franco;Male;24-08-2001;Al Pascolo;12a;Blenio;2131;41213313123;gigi.franco@gmail.com;Giocare a calcio con i miei amici;Lavoro come spazzino per il mio comune
2018-11-04 10:59:21;Paolo;Guebeli;Male;24-10-2001;Via al fango;12;Lugano;6999;0921212121;guebeli.paolo@gmail.com;Andare a caccia;Vendere lavatrici
2018-11-04 11:01:28;aa;a;Male;13-10-2019;A;2A;Lugano;564;1212121212;guebeli.paolo@gmail.com;Giocare a briscola;Macellaio;
```

RIEPILOGO GIORNATA

Time	Name	Lastname	Sex	Birthdate
13:56:11	Gigi	Franco	Male	21-08-2001
Street	House number	City	NAP	
Al Pascolo	12a	Blenio	2131	
Phone number		E-mail		
4121313123		gigi.franco@gmail.com		
Hobby		Job		
Giocare a calcio con i miei amici		Lavoro come spazzino per il mio comune		

FINISH

```

25 echo
26 "<table style=\"width:100%\">
27 <tr>
28     <th>Time</th>
29     <th>Name</th>
30     <th>Lastname</th>
31     <th>Sex</th>
32     <th>Birthdate</th>
33 </tr>
34 <tr>
35     <td>". $row[0]. "</td>
36     <td>". $row[1]. "</td>
37     <td>". $row[2]. "</td>
38     <td>". $row[3]. "</td>
39     <td>". $row[4]. "</td>
40 </tr>
41 <tr>
42     <th>Street</th>
43     <th>House number</th>
44     <th>City</th>
45     <th>NAP</th>
46 </tr>
47 <tr>
48     <td>". $row[5]. "</td>
49     <td>". $row[6]. "</td>
50     <td>". $row[7]. "</td>
51     <td>". $row[8]. "</td>
52 </tr>
53 <tr>
54     <th colspan=\"2\">Phone number</th>
55     <th colspan=\"2\">E-mail</th>
56 </tr>
57 <tr>
58     <td colspan=\"2\">". $row[9]. "</td>
59     <td colspan=\"2\">". $row[10]. "</td>
60 </tr>;
61 if(count($row) > 10){
62     echo
63     "<tr>
64         <th colspan=\"2\">Hobby</th>
65         <th colspan=\"2\">Job</th>
66     </tr>
67     <tr>
68         <td colspan=\"2\">". $row[11]. "</td>
69         <td colspan=\"2\">". $row[12]. "</td>
70     </tr>
71 </table>
72 <br>
73 <br>
74 <br>;
75 }
76 }

```

4 Test

4.1 Protocollo di test

Definire in modo accurato tutti i test che devono essere realizzati per garantire l'adempimento delle richieste formulate nei requisiti. I test fungono da garanzia di qualità del prodotto. Ogni test deve essere ripetibile alle stesse condizioni.

Test Case:	TC-001	Nome:	Controllo dei dati
Riferimento:	REQ-01		
Descrizione:	Controllare che la pagina di registrazione verifichi la validità dei dati		
Prerequisiti:	Devono esserci i campi del form editabili e il pulsante avanti clickabile		
Procedura:	1. Compilare il form non rispettando le regole. Quindi inserendo numeri in campi di testo e testo in campi di n		

	2. Provare a proseguire alla prossima pagina. 3. Ripetere questa procedura una decina di volte.
Risultati attesi:	Il sito non dovrebbe permettere di avanzare

Test Case:	TC-002	Nome:	Test del pulsante CORREGGI
Riferimento:	REQ-03		
Descrizione:	Controllare che se si preme il pulsante CORREGGI si ritorna alla pagina precedente.		
Prerequisiti:	Devono esserci le pagine di registrazione e correzione funzionanti		
Procedura:	<ol style="list-style-type: none"> 1. Compilare correttamente il form di registrazione. 2. Premere AVANTI in basso a destra. 3. Premere il pulsante CORREGGI. 		
Risultati attesi:	Il pulsante dovrebbe alla pagina precedente.		

Test Case:	TC-003	Nome:	Controllo della creazione e scrittura/lettura dei file CSV
Riferimento:	REQ-03		
Descrizione:	Controllare la scrittura/lettura dei file CSV e che se non sono ancora esistenti i file CSV la creazione.		
Prerequisiti:	Devono esserci le pagine di registrazione e correzione funzionanti		
Procedura:	<ol style="list-style-type: none"> 1. Cancellare tutti i file nella cartella Registrazioni. 2. Compilare correttamente il form di registrazione. 3. Premere AVANTI in basso a destra. 4. Controllare se i dati sono corretti e premere finito. 5. Controllare che la pagina finale mostra i dati giornalieri. 6. Confrontare i dati con quelli presenti nel file nella cartella Registrazioni di nome Registrazioni_<data>.csv. 		
Risultati attesi:	I dati dovrebbero essere identici.		

4.2 Risultati test

TC-001	Il test è andato a buon fine il sito controlla i dati correttamente.
TC-002	Se viene premuto il pulsante CORREGGI si ritorna al sito precedente.
TC-003	I dati vengono scritti nei file CSV.

4.3 Mancanze/limitazioni conosciute

Il programma non gestisce nel hobby e nella professione gli html special chars, quando si preme il pulsante correggi il form viene svuotato.

5 Consuntivo

Ho rispettato i tempi definiti nel GANTT precedente. Non ho avuto nessun tipo di ritardo.

6 Conclusioni

Secondo me questo lavoro è stato solo d'aiuto a livello didattico visto che online ci sono milioni di siti del genere probabilmente migliori del mio che fanno lo stesso lavoro magari con una validazione lato server.

6.1 Sviluppi futuri

Per sviluppare questa applicazione si potrebbe aggiungere una validazione dei dati lato server così da renderlo più sicuro e magari non salvare i dati su dei CSV ma su dei server appositi così da rendere più facile la ricerca di certi dati. In più si dovrebbe migliorare il pulsante correggi che cancella i dati quando torna indietro

7 Bibliografia

7.1 Sitografia

- <http://www.stackoverflow.com> 11-10-2018
- <http://www.w3schools.com> 13-10-2018

8 Allegati

- Diari di lavoro
- Presentazione del prodotto
- Codice sorgente è caricato su GitHub