Progetto 1

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

# Introduzione

## 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.

## 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.

## 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.

## 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.

## 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 |

## Pianificazione

|  |
| --- |
|  |

## Analisi dei mezzi

### 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

### Hardware

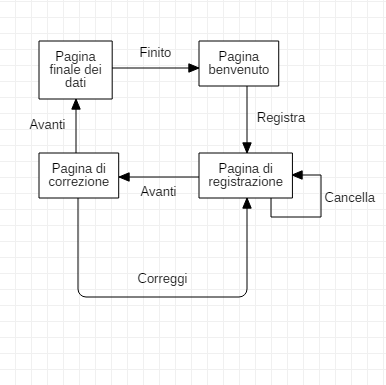
HP Omen

i7-6700HQ CPU 2.6 GHz

RAM 16 GB

# Progettazione

## Design dell’architettura del sistema

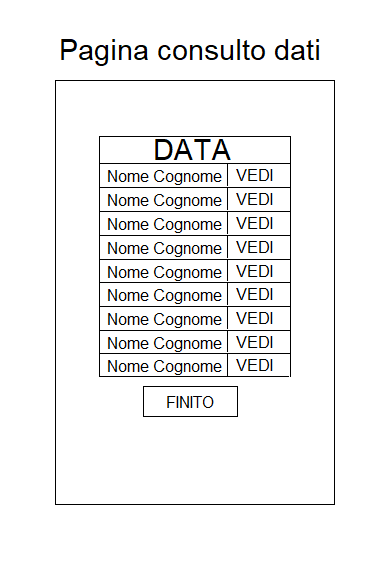
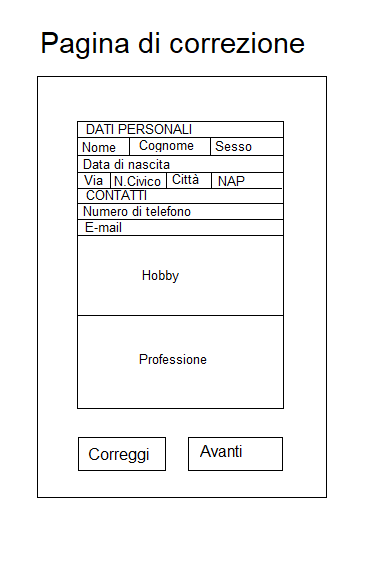


## Design dei dati e database

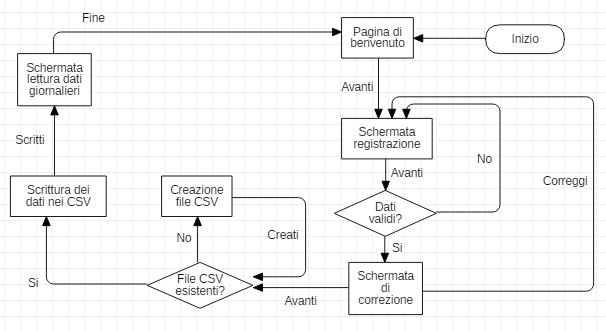
Campi del form:

|  |  |  |  |
| --- | --- | --- | --- |
|  | 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 |

## Design delle interfacce



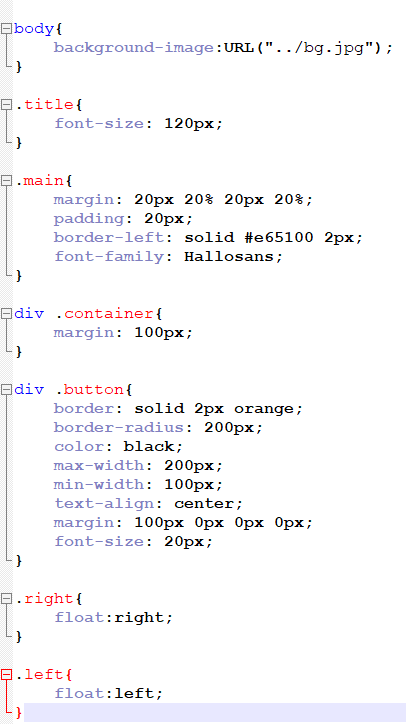
## Design procedurale



# Implementazione

## 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.



## 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 = "";

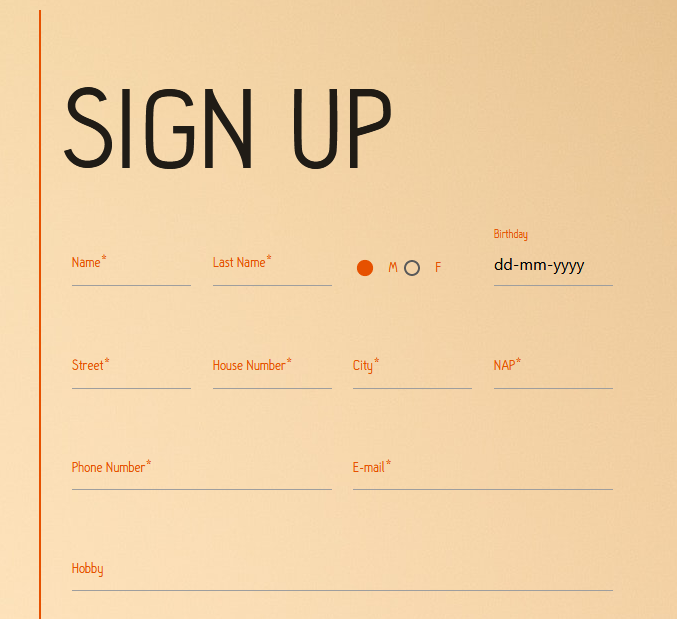
document.getElementById("email").value = "";

document.getElementById("hobby").value = "";

document.getElementById("pro").value = "";

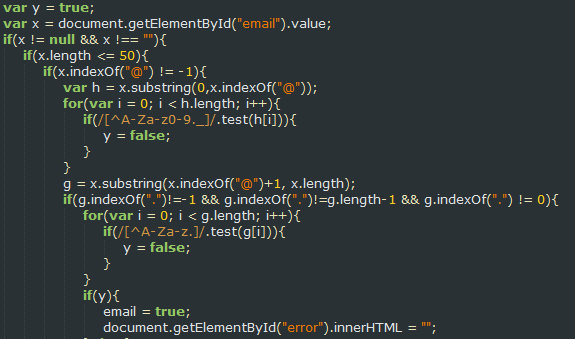
}

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 sovvrascrivere il colore già esistente.



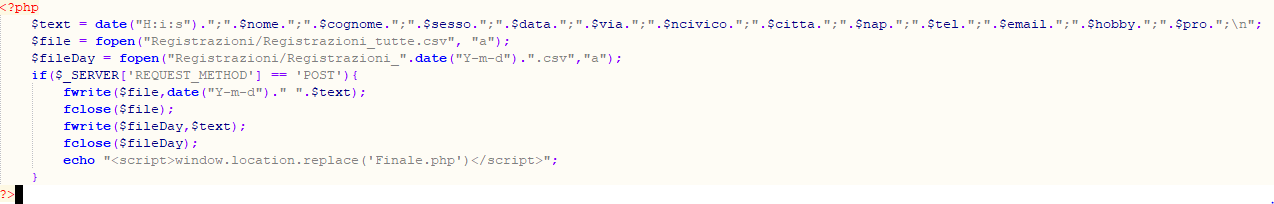
### Controllo validità dei dati

Per sviluppare i controlli ho usufruito di [www.W3School.com](http://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.



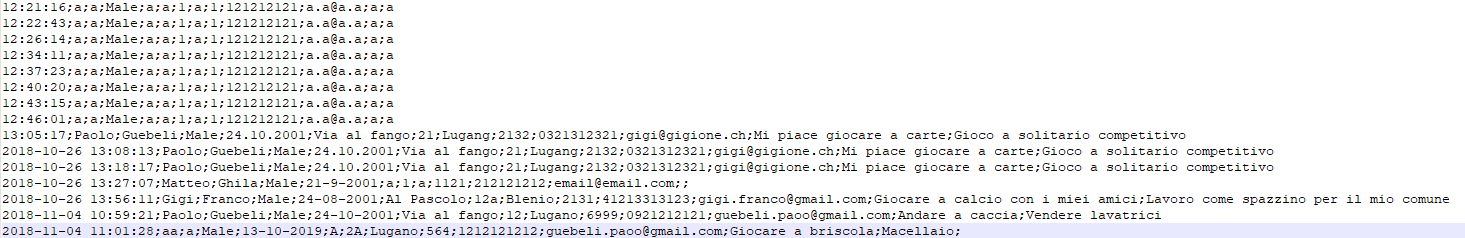
## Pagina di correzione

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 d 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”**.

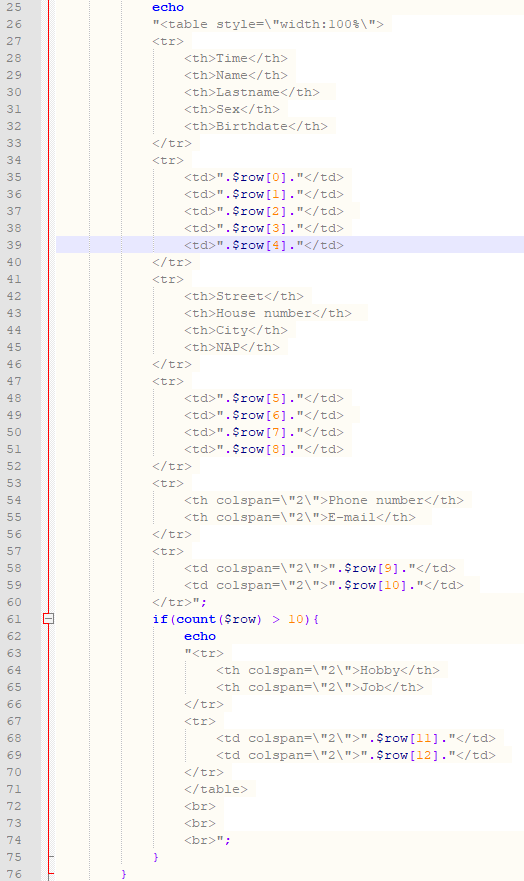


## Pagina lettura dati giornalieri

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.







# Test

## 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:**  **Riferimento**: | TC-001  REQ-01 | **Nome:** | Controllo dei dati |
| **Descrizione:** | Controllare che la pagina di registrazione verifichi la validità dei dati | | |
| **Prerequisiti:** | Devono esserci i campi del form editabili eil pulsante avanti clickabile | | |
| **Procedura:** | 1. Compilare il from 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:**  **Riferimento**: | TC-002  REQ-03 | **Nome:** | Test del pulsante CORREGGI |
| **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:** | 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:**  **Riferimento**: | TC-003  REQ-03 | **Nome:** | Controllo della creazione e scrittura/lettura dei file CSV |
| **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:** | 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 idati 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. | | |

## 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. |

## Mancanze/limitazioni conosciute

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

# Consuntivo

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

# 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.

## 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

# Bibliografia

## Sitografia

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

# Allegati

* Diari di lavoro
* Presentazione del prodotto
* Codice sorgente è carcicato su GitHub