

Progetto 1

Titolo del progetto: Progetto 1
Alunno/a: Bryan Beffa
Classe: Info I3AA
Anno scolastico: 2018/2019
Docente responsabile: Adriano Barchi, Luca Muggiasca, Francesco Mussi

1	Introduzione.....	3
1.1	Informazioni sul progetto	3
1.2	Abstract	3
1.3	Scopo	3
Analisi		4
1.4	Analisi del dominio	4
1.5	Analisi e specifica dei requisiti.....	4
1.6	Pianificazione	8
1.7	Analisi dei mezzi	9
1.7.1	Software	9
1.7.2	Hardware	9
2	Progettazione	10
2.1	Design dell'architettura del sistema.....	10
2.2	Design dei dati e database.....	11
2.3	Design delle interfacce	12
2.4	Design procedurale	14
3	Implementazione	15
3.1	Pagina di benvenuto	15
3.1.1	FadeDivMask().....	15
3.2	Pagina di registrazione.....	16
3.2.1	EnableNextButton().....	16
3.3	Pagina di riassunto dati	17
3.3.1	PrintValues()	17
3.3.2	ModifyData()	18
3.4	Classe Validator.php	19
3.4.1	CheckTextFields().....	19
3.5	Classe Csv.php	20
3.5.1	GetDataTable()	20
3.5.2	GetDataRows()	21
4	Test	22
4.1	Protocollo di test.....	22
4.2	Risultati test.....	30
4.3	Mancanze/limitazioni conosciute.....	30
5	Consuntivo	31
6	Conclusioni.....	1
Sviluppi futuri		1
6.1	Considerazioni personali	1
7	Bibliografia.....	1
7.1	Sitografia	1
8	Allegati	1

1 Introduzione

1.1 Informazioni sul progetto

Allievo coinvolto nel progetto: Beffa Bryan

Classe: Informatica 3AA presso la sede Scuola Arti e Mestieri Trevano

Docenti responsabili: Adriano Barchi, Luca Muggiasca e Francesco Mussi

Data inizio: 05:09:2018

Data fine: 09:11:2018

1.2 Abstract

The teachers need a website with a form where users can register. When a user writes his personal information in the right fields and he confirm them, all the information will be saved in csv files. There is a file with the information about all users and if a user is the first registration of the day they will be created a daily csv file. The programming languages will be used in this project will be: php, html, css and javascript.

1.3 Scopo

Lo scopo principale è di prepararci al progetto che dovrà essere consegnato a fine apprendistato dove ci viene richiesto di gestirne uno. Ha anche lo scopo di combinare i vari moduli che ci sono stati proposti durante i due primi anni di formazione professionale. La realizzazione del prodotto finale in questo progetto è considerato un scopo secondario, ma comunque importante.

Analisi

1.4 Analisi del dominio

A scuola ci è stato richiesto per il modulo di progetti di realizzare un form per la registrazione di utenti. Lo scopo di questo progetto è quello di salvare su file esterni di formato CSV tutte le registrazioni che sono state effettuate. Le registrazioni devono essere salvate in due file distinti. Il primo file contiene tutte le registrazioni effettuate, mentre il secondo file che viene creato è giornaliero. Il primo utente del giorno che si registra verrà inserito nel file odierno appena creato, oltre al file contenente tutte le registrazioni.

1.5 Analisi e specifica dei requisiti

Il committente richiede una struttura di pagine web che permettano la registrazione di utenti. La registrazione deve essere svolta in un form dove vengono richiesti campi obbligatori (Nome, Cognome, Data di nascita, Indirizzo, No. Civico, Città, Nap, No Telefono, E-mail, Genere) e non obbligatori (Hobby, Professione). Se tutti i campi obbligatori non sono compilati non si può procedere. Una volta che tutti i campi obbligatori sono stati riempiti si può cliccare sul bottone "avanti". Una volta cliccato su "avanti" verranno visualizzati tutti i dati appena immessi e si avrà la possibilità di modificare le informazioni tramite il bottone "correggi" oppure procedere e confermare quindi i dati tramite il bottone "avanti". A questo punto i dati verranno salvati in due file distinti di formato csv. Il primo file "Registrazioni_tutte.csv" contiene tutte le registrazioni. Il secondo file "Registrazione_aaaa-mm-dd.csv" e viene creato una volta al giorno. Entrambi i file, se non esistono già, vengono creati e sono contenuti nella cartella "Registrazioni". Successivamente l'utente verrà riportato ad una pagina che mostra i dati che sono stati inseriti nel file giornaliero. Questa pagina contiene inoltre un bottone "Home" per tornare alla pagina iniziale di benvenuto. L'utente dovrebbe interagire in modo semplice ed efficace senza necessitare l'intervento di terzi. Il prodotto dovrà avere un interfaccia che permetta all'utente di muoversi in modo autonomo. L'intero progetto dovrà avere una struttura di cartelle ordinata che permette una semplice gestione.

ID: REQ-01	
Nome	Pagina di benvenuto
Priorità	2
Versione	1.0
Note	La pagina deve presentare un bottone che permette lo spostamento alla pagina di registrazione
Sotto requisiti	
001	Bottone “registrai” per il cambio pagina

ID: REQ-02	
Nome	Pagina di registrazione
Priorità	1
Versione	1.0
Note	Si necessita una pagina che contenga i campi di registrazione
Sotto requisiti	
001	Campi obbligatori (Nome, Cognome, Data di nascita, Indirizzo, No. Civico, Città, Nap, No Telefono, E-mail, Genere)
002	I campi obbligatori devono essere segnalati con il carattere asterisco “*”
003	Campi non obbligatori (Hobby, Professione)
004	Bottone “avanti” per procedere al controllo dei dati inseriti
005	Bottone “cancella” per cancellare tutti i dati inseriti
006	Non si può procedere se tutti i campi obbligatori non sono compilati

ID: REQ-03	
Nome	Convalida dati
Priorità	1
Versione	1.0
Note	Convalidazione dei dati lato client. È necessaria la pagina di registrazione
Sotto requisiti	
001	Il campo Nome, Cognome, Città, Indirizzo, No Civico, deve essere di tipo testo
002	Il campo E-mail deve contenere la struttura testo@testo.testo
003	I campi non obbligatori hobby e professione sono di tipo testo
004	Il campo No di telefono è di tipo testo ma deve contenere solo cifre spazi e trattini
005	Il campo Nap deve essere di tipo numerico e composto da 5 cifre
006	Il genere deve essere un campo di tipo testo che accetta solo "M" o "F"

ID: REQ-04	
Nome	Conferma dati
Priorità	1
Versione	1.0
Note	Visualizzazione dei dati appena inseriti ed eventuale modifica o conferma
Sotto requisiti	
001	Bottone "correggi" per modificare i dati
002	Bottone "avanti" per procedere al salvataggio dei dati
003	I dati sono visualizzati come stringa

ID: REQ-05	
Nome	Codice lato server
Priorità	1
Versione	1.0
Note	Scrittura del codice lato server
Sotto requisiti	
001	Struttura completa del sito
002	Si necessita il codice per scrivere nei file csv (lato server)
003	Si necessita il codice per leggere dai file csv (lato server)

ID: REQ-06	
Nome	Scrittura dati
Priorità	1
Versione	1.0
Note	Scrittura dopo la conferma dei dati dell'utente nei file csv
Sotto requisiti	
001	File CSV dove salvare i dati
002	Se i file csv non esistono bisogna crearli
003	File CSV per salvare tutte le registrazioni di nome "Registrazioni_tutte.csv"
003	File CSV per salvare le registrazioni giornaliero di nome "Registrazione_aaaa-mm-gg.csv"
005	Il file giornaliero deve essere creato ogni giorno
006	I dati sono visualizzati come stringa

ID: REQ-07	
Nome	Creazione pagina di ricapitolazione dei dati inseriti
Priorità	1
Versione	2.0
Note	I dati inseriti vanno salvati in due file CSV distinti
Sotto requisiti	
001	File csv da cui estrarre i dati inerenti all'utente
002	Bottone "Home" per tornare alla pagina di benvenuto

1.6 Pianificazione

Ho diviso il progetto in 4 fasi principali. La prima fase è quella di progettazione dove vengono svolte attività incentrate sull'analisi del progetto. La seconda fase è quella di sviluppo, dove avviene la stesura effettiva del codice delle pagine, script. La terza attività è il momento in cui vengono testati le varie parti di progetto. L'ultima fase, che copre tutta la durata del progetto, è la scrittura della documentazione. Ho utilizzato un diagramma di Gantt per gestire il mio progetto.

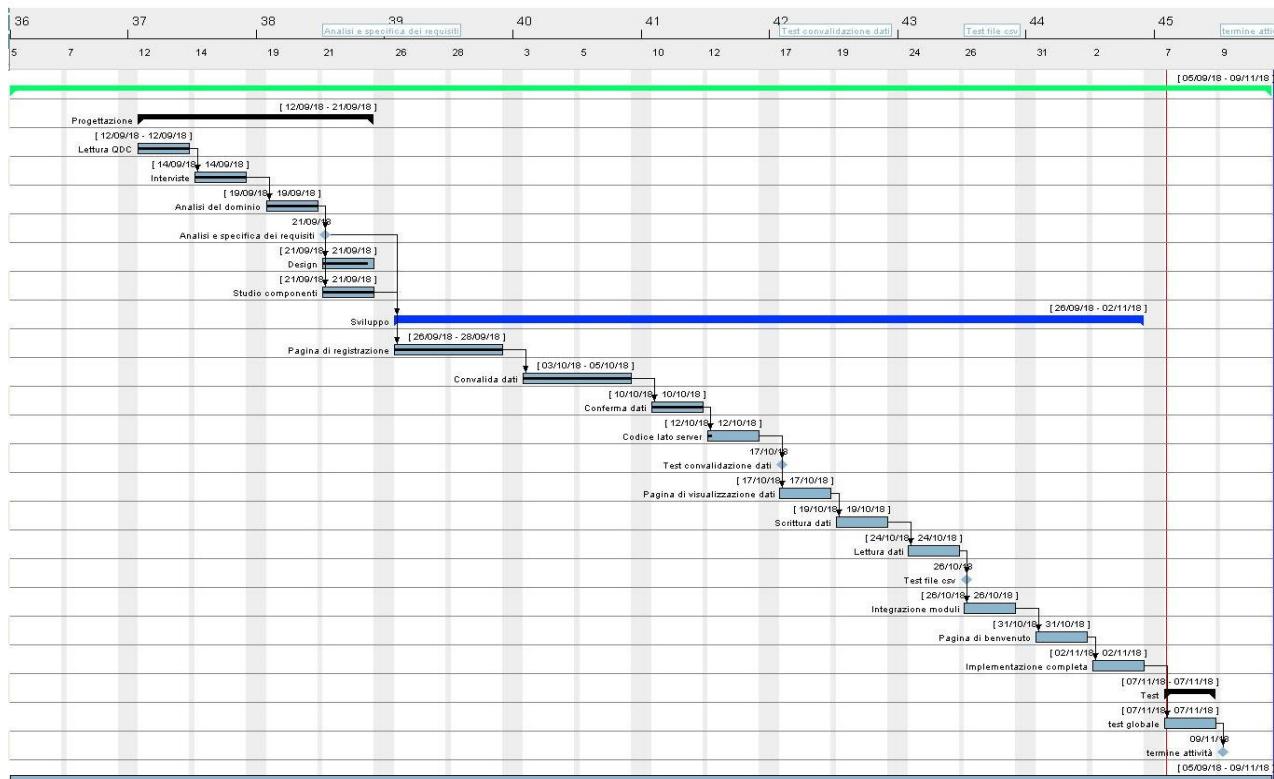


Figura 1 Diagramma di Gantt

1.7 Analisi dei mezzi

1.7.1 Software

I software che sono stati utilizzati per la realizzazione di questo progetto sono:

- Sublime Text 3
- GanttProject
- Microsoft Word 2013
- Microsoft Powerpoint 2013
- Microsoft Excel 2013
- Apache Webserver 2.4.27
- Php 7.1.9
- Strumento di cattura

Il progetto è stato testato sui seguenti browser:

- Google Chrome
- Mozilla Firefox
- Microsoft Edge
 - Con Microdoft Edge c'è un piccolo problema con la tabella dei dati giornalieri, in particolare con lo sfondo della tabella che risulta trasparente
- Browser Opera

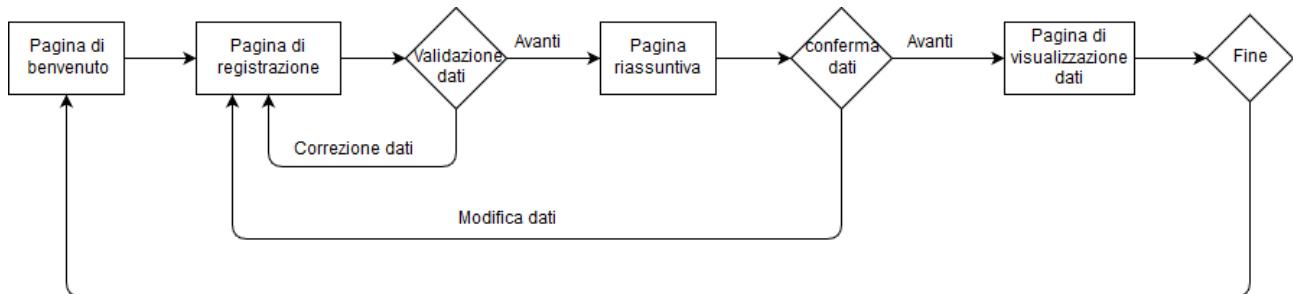
1.7.2 Hardware

L'intero progetto è un sito web, quindi non necessitavo di hardware specifico ed ho eseguito il tutto sul mio computer portatile Asus con sistema operativo windows 10. È stato pubblicato sul sito di hosting che ci ha fornito la scuola admin2.infomaniak.com/ftp ed è raggiungibile effettuando la ricerca del sito <http://www.samtinfo.ch/i16befbry/web/progetto1>

2 Progettazione

In questo capitolo è rappresentato e spiegato come ho deciso di progettare l'intero sistema. Ho progettato tutto il sistema nel modo che ritenevo più efficace e coerente per garantire un prodotto che superasse le aspettative

2.1 Design dell'architettura del sistema



Dopo aver inserito i dati nella pagina di registrazione viene fatta una verifica locale della sintassi dei vari campi. Se i campi non rispettano la forma corretta non è possibile proseguire fino alla modifica corretta della sintassi. Se la sintassi è corretta si può proseguire con il bottone "avanti". Successivamente viene fatto un secondo controllo lato server. Se il controllo lato server è negativo si possono modificare i dati sbagliati, se il controllo va a buon fine invece si arriva alla pagina riassuntiva. Alla pagina riassuntiva si può confermare i dati o decidere di modificarli e tornare quindi alla pagina di registrazione. Se avviene una conferma dei dati si può tornare alla pagina di benvenuto tramite un bottone.

2.2 Design dei dati e database

I dati vengono salvati in due file di formato csv distinti:

- Il file contiene tutte le registrazioni degli utenti di nome “Registrazioni_tutte.csv”.
- Il file contiene tutte le registrazioni giornaliere degli utenti di nome “Registrazione_aaaa-mm-yy.csv”

In entrambi i file di formato csv ho deciso di aggiungere un campo denominato “id”, così da rendere univoco ogni record del file. Questa aggiunta è pensata anche per una futura implementazione al progetto di un database.

Design dei dati nel file “Registrazioni_tutte.csv”.

Nome colonna	Formato valori (esempi)
Id	“1”
Nome	“Nome”
Cognome	“Cognome”
Data_nascita	“YYYY/MM/dd”
Indirizzo	“Indirizzo”
No_civico	“10A”
Citta	“Città”
Nap	“000 000 00 00”
No_telefono	“0000”
Email	“esempio@esempio.esempio”
Genere	“M”
Hobby	“Hobby”
Professione	“Professione”
Data_iscrizione	“YYYY-MM-dd – hh:mm:ss”

Design dei dati nel file “Registrazione_aaaa-mm-yy.csv”

Nome colonna	Formato valori (esempi)
Id	“1”
Nome	“Nome”
Cognome	“Cognome”
Data_nascita	“aaaa-mm-dd”
Indirizzo	“Indirizzo”
No_civico	“10A”
Citta	“Città”
Nap	“000 000 00 00”
No_telefono	“0000”
Email	“esempio@esempio.esempio”
Genere	“M”
Hobby	“Hobby”
Professione	“Professione”
Data_iscrizione	“YYYY-MM-dd – hh:mm:ss”

2.3 Design delle interfacce

La pagina di benvenuto presenta un titolo con sotto una breve introduzione testuale. Sotto all'introduzione vi è il bottone per accedere alla pagina di registrazione.

The screenshot shows a simple web interface. At the top center is the word "Titolo". Below it is the word "Introduzione". At the bottom center is a rectangular button with the text "Registrati" inside. The entire interface is contained within a large rectangular frame.

La pagina di registrazione presenta tutti i campi. Quelli segnati con il carattere “*” sono obbligatori. Alla destra del nome del campo vi è un input in cui poter immettere il valore. Il bottone “cancella” e il bottone avanti sono situati sotto

The screenshot shows a registration form with five fields labeled "Campo 1" through "Campo 5". Each label is followed by a rectangular input field. At the bottom left is a button labeled "Cancella" and at the bottom right is a button labeled "Avanti". The entire form is contained within a large rectangular frame.

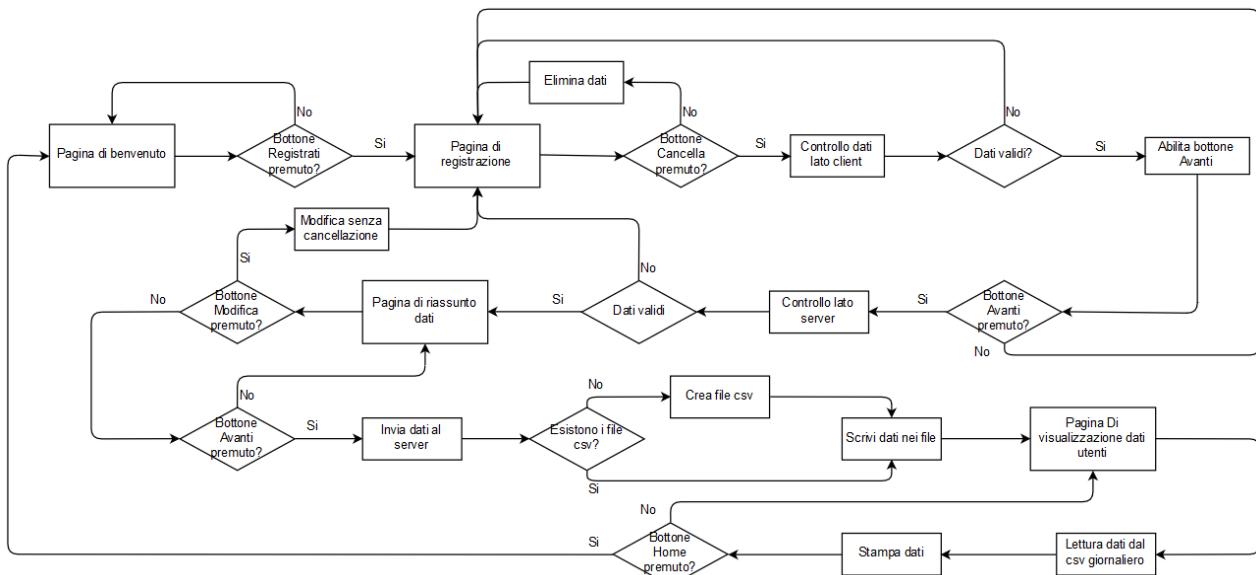
La pagina di riassunto stampa i campi con i rispettivi valori inseriti dall'utente in precedenza. In basso vi è il bottone "modifica" e il bottone "avanti".

Nome*	valore
Cognome*	valore
Data_nascita*	valore
Indirizzo*	valore
No_civico*	valore
Citta*	valore
Nap*	valore
No_telefono*	valore
Email*	valore
Genere*	valore
Hobby	valore
Professione	valore

L'ultima pagina di visualizzazione dati giornalieri presenta una griglia con le varie registrazioni giornaliere e un bottone per ritornare alla pagina di benvenuto.

utente 1	valore	valore
utente 2	valore	valore
utente 3	valore	valore
utente 4	valore	valore
utente 5	valore	valore
utente 6	valore	valore
utente 7	valore	valore
utente 8	valore	valore
utente 9	valore	valore
utente 10	valore	valore
utente 11	valore	valore
utente 12	valore	valore
...

2.4 Design procedurale



3 Implementazione

3.1 Pagina di benvenuto

Nella pagina di benvenuto non ci sono parti di codice complicate e l'interfaccia risulta semplice. Il tema dell'intera struttura del progetto era a nostra scelta.

Qui di seguito vi è uno screen della pagina di benvenuto.

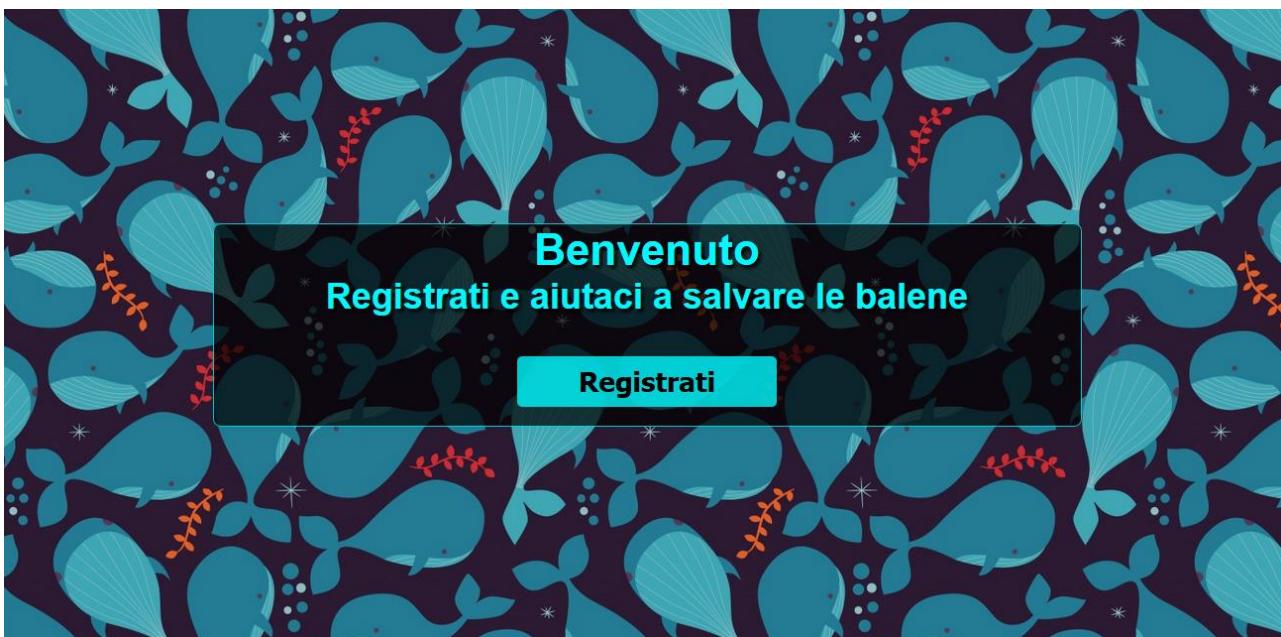


Figura 2 Pagina di benvenuto

3.1.1 FadeDivMask()

Nella pagina di benvenuto vi è la funzione fadeDivMask(). Questa funzione serve a creare un'animazione in cui la maschera contenente il testo di benvenuto e il bottone "Registrati" compare in modo graduale. Per creare questo animazione ho utilizzato ajax.

```
//Metodo che fa comparire gradualmente il divMask.  
function fadeDivMask(){  
    divMask.style.display = "none";  
    divMask.style.visibility = "visible";  
    $(document).ready(function(){  
        $("#divMask").fadeIn(2000)  
    });  
}
```

Figura 3 Funzione FadeDivMask()

3.2 Pagina di registrazione

La pagina di registrazione presenta un file javascript con molti metodi. Molte funzioni servono per la convalida dei dati e funzionano tutti nello stesso modo, ovvero se il valore, durante la digitazione, è corretto il bordo inferiore dell'input si diventa verde, altrimenti rosso. Ho utilizzato questo approccio in modo da far capire all'utente che sta inserendo un carattere non consentito.

A lato di ogni input vi è comunque un'icona di informazione sulle restrizioni del campo adiacente.

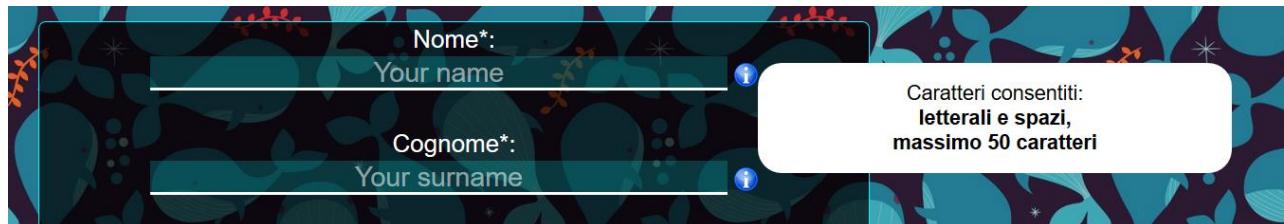


Figura 4 Aiuto restrizioni campo

3.2.1 EnableNextButton()

La funzione enableNextButton() ha due scopi:

1. Informare l'utente di compilare tutti i campi obbligatori in modo corretto.
2. Inviare tramite URL i valori, se quest'ultimi presentano la sintassi corretta e l'utente conferma l'invio dei dati, e spostarsi sulla pagina di riassunto dati

I due comportamenti avvengono in base all'esito del controllo globale dei dati tramite la funzione checkMainInputs(). Il metodo enableNextButton() costruisce un URL contenente i dati dell'utente che verranno letti successivamente nella pagina di riassunto dati e stampati a schermo.

```
function enableNextButton(){
    if(checkMainInputs()){
        if(confirm("Sicuro di voler procedere?")){
            var queryString = "?name=" + inputList[0].value
            + "&surname=" + inputList[1].value
            + "&date=" + inputList[2].value
            + "&address=" + inputList[3].value
            + "&civicNumber=" + inputList[4].value
            + "&city=" + inputList[5].value
            + "&nap=" + inputList[6].value
            + "&phoneNumber=" + inputList[7].value
            + "&email=" + inputList[8].value
            + "&gener=" + genderValue
            + "&hobby=" + inputList[12].value
            + "&profession=" + inputList[13].value;
            window.location.href = 'riassuntoDati.php' + queryString;
        }
    }else{
        alert("Devi compilare tutti i campi obbligatori")
    }
}
```

Figura 5 Funzione enableNextButton()

Ho deciso di utilizzare questa metodologia per il semplice fatto che l'utente non inserisce dati sensibili come ad esempio una password. Inoltre ho ritenuto più opportuno non utilizzare codice lato server fino alla conferma definitiva dei dati da parte dell'utente.

3.3 Pagina di riassunto dati

Nella pagina di riassunto vengono stampati a schermo i dati inseriti dall'utente in precedenza per una rapida rilettura.

Nome*:	Bryan
Cognome*:	Beffa
Data di nascita*:	1999-02-20
Indirizzo*:	Via cantonale

Figura 6 Pagina riassunto dati

Nell'immagine precedente è raffigurata l'interfaccia grafica.

3.3.1 PrintValues()

Questo funzione serve a prendere i dati dall'URL, inviati in precedenza dal metodo enableNextButton() nella pagina di registrazione, e stamparli a schermo. I valori vengono inseriti in degli input che sono impostati sulla modalità lettura per non permettere all'utente di modificare direttamente da questa pagina i valori.

```
function printValues(){
    var queryString = decodeURIComponent(window.location.search);
    queryString = queryString.substring(1);
    var queries = queryString.split("&");
    for (var i = 0; i < queries.length; i++){
        var realValue = queries[i].split "=";
        inputList[i].value = realValue[realValue.length-1];
    }
}
```

Figura 7 Funzione printValues()

I valori potrebbero venir corrotti dall'utente tramite l'URL, ma avendo creato i controlli lato server, quando si clicca sul bottone Avanti, se questi valori non sono accettabili si viene rimandati alla pagina di registrazione.

3.3.2 ModifyData()

Questa funzione viene richiamata se l'utente clicca il bottone Correggi. Utilizza lo stesso approccio utilizzato in precedenza dal metodo enableNextButton(), ovvero rinvia alla pagina di registrazione tramite URL i dati inseriti. Nella pagina di registrazione vi è un metodo che esegue la stessa procedura della funzione printValues().

```
function modifyData(){
    if(confirm("Vuoi modificare i dati?")){
        var queryString = "?name=" + inputList[0].value
        + "&surname=" + inputList[1].value
        + "&date=" + inputList[2].value
        + "&address=" + inputList[3].value
        + "&civicNumber=" + inputList[4].value
        + "&city=" + inputList[5].value
        + "&nap=" + inputList[6].value
        + "&phoneNumber=" + inputList[7].value
        + "&email=" + inputList[8].value
        + "&gener=" + inputList[9].value
        + "&hobby=" + inputList[10].value
        + "&profession=" + inputList[11].value;
        window.location.href = 'registrazione.html' + queryString;
    }
}
```

Figura 8 Funzione modifyData()

3.4 Classe Validator.php

Siccome ho deciso di effettuare i controlli anche lato server ho creato la classe Validator.php contenente tutti i metodi per effettuare i controlli lato server.

3.4.1 CheckTextFields()

La funzione checkTextFields() controlla che i valori inseriti nei campi testuali (nome, cognome, indirizzo, città) rispetti la sintassi corretta ovvero contenga solo caratteri letterali e spazi e non superi il massimo di 50 caratteri. Per svolgere questa operazione ho utilizzato un'espressione regolare. Il metodo ritorna “true” se la sintassi è corretta altrimenti “false”.

```
/**  
 * Metodo che controlla che i campi di testo con gli stessi criteri abbiano dei valori validi.  
 * Numero massimo di caratteri 50.  
 * Caratteri consentiti: letterali e spazio.  
 */  
public function checkTextFields($text){  
    $text = trim($text);  
    if(preg_match("/^[\a-zA-Z \ééÈÈÍÍàáÀÀòóöóÙÙÜÜ]+$/", $text) && strlen($text) <= 50){  
        return true;  
    }  
    return false;  
}
```

Figura 9 Funzione checkTextFields()

Tutti questi metodi vengono successivamente utilizzati nella pagina di riassunto dati una volta che si preme il bottone “avanti” per effettuare un unico controllo su tutti i dati inseriti. Come si vede nell’immagine viene creato l’oggetto Validator e successivamente vengono utilizzati i suoi metodi per il controllo totale dei campi.

```
$validator = new Validator();  
if($validator->checkTextFields($name)  
    && $validator->checkTextFields($surname)  
    && $validator->checkDate($date)  
    && $validator->checkTextFields($address)  
    && $validator->checkCivicNumber($civicNumber)  
    && $validator->checkTextFields($city)  
    && $validator->checkNap($nap)  
    && $validator->checkPhoneNumber($phoneNumber)  
    && $validator->checkEmail($email)  
    && $validator->checkGender($gender)  
    && $validator->checkHobbyAndProfession($hobby)  
    && $validator->checkHobbyAndProfession($profession)){
```

Figura 10 Controllo campi lato server

Se tutti questi controlli sono positivi si viene spostati alla pagina finale, mentre se uno o più valori non superano questo controllo si viene rimandati alla pagina di registrazione.

3.5 Classe Csv.php

La classe Csv.php contiene metodi utili per i file csv. Vi sono alcuni metodi per leggere, scrivere nei file, ricevere una tabella già formattata dei dati di un file. Questa classe è stata concepita per mantenere un codice più pulito, file php, della pagina riassunto dati, quando si scrive nei file .csv e nella pagina finale in cui vengono letti i dati dai file.

3.5.1 GetDataTable()

L'immagine sottostante ritrae il codice della funzione getDataTable(). Questa funzione, come annunciato in precedenza ritorna una stringa contenente una tabella con tutti i dati di un csv. Ho ritenuto utile creare questo metodo per utilizzarlo nella pagina finale in cui stampo tutte le registrazioni giornaliere. Il codice php della pagina finale non deve far altro che stampare la stringa ritornata da questa funzione.

```
public function getDataTable(){
    //controllo se il file esiste
    if(file_exists($this->getPath())){
        //salvo i dati del file
        $data = file_get_contents($this->getPath());
        $table = "<table>";
        $rows = explode('/', $data);
        //conto il numero di righe del file csv e tolgo 1, perché l'ultimo campo dell'array è vuoto
        $rowsLength = count($rows)-1;

        //creo il nome delle colonne
        $table .= "<tr>
                    <th>id</th>
                    <th>Nome</th>
                    <th>Cognome</th>
                    <th>Data di nascita</th>
                    <th>Indirizzo</th>
                    <th>No. Civico</th>
                    <th>Città</th>
                    <th>Nap</th>
                    <th>No. Telefono</th>
                    <th>E-mail</th>
                    <th>Sesso</th>
                    <th>Hobby</th>
                    <th>Professione</th>
                    <th>Data iscrizione</th>
                </tr>";

        //ripeto per il numero totale di utenti
        for ($i=0; $i < $rowsLength; $i++) {
            $table .= "<tr>";
            $fields = explode(';', $rows[$i]);
            $fieldsLength = count($fields);
            //viene aggiunto anche
            for ($j=0; $j < $fieldsLength; $j++) {
                $table .= "<td>" . $fields[$j] . "</td>";
            }
            $table .= "</tr>";
        }

        //termino la struttura
        $table .= "</table>";
        return $table;
    }
}
```

Figura 11 Funzione getDataTable()

3.5.2 GetDataRow()

Questa funzione ho deciso di implementarla in un secondo tempo, quando, per un eventuale sviluppo futuro di un database, ho inserito il campo ID nei file csv.

```
public function getDataRows(){
    //salvo i dati del file
    $data = file_get_contents($this->getPath());
    $rows = explode('//', $data);
    //conto il numero di righe del file csv e tolgo 1, perché l'ultimo campo dell'array è vuoto
    $rowsLength = count($rows)-1;
    return $rowsLength;
}
```

Figura 12 Funzione getDataRows()

Questo metodo torna utile per calcolare l'id dell'utente perché ritorna il numero di righe di un file, infatti viene utilizzato due volte (per il file giornaliero e per quello contenente tutte le registrazioni) nella pagina di riassunto dati. Ho utilizzato i caratteri "://" per concludere ogni riga di dati, utilizzando la funzione creo un'array contenente riga per riga tutti gli utenti registrati.

```
//salvo l'id giornaliero e generale
$dailyId = $dayCsv->getDataRows();
$id = $mainCsv->getDataRows();
```

Figura 13 Utilizzo metodo getDataRows()

4 Test

4.1 Protocollo di test

Test Case:	TC-001	Nome:	Validazione campi testuali lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura dei campi testuali (nome, cognome, indirizzo, città)		
Prerequisiti:	Pagina di registrazione contenente i campi testuali		
Procedura:	<ol style="list-style-type: none">1. Aprire la pagina di Registrazione.php2. Cliccare sul campo testuale (nome, cognome, indirizzo, città)3. Inserire un valore		
Risultati attesi:	Se il valore inserito presenta una struttura corretta il bordo del campo testuale diventa verde. Se la struttura del valore non è corretta il bordo dell'input deve colorarsi di rosso.		

Test Case:	TC-002	Nome:	Validazione campo "data di nascita" lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura del valore inserito nel campo "data di nascita" lato client		
Prerequisiti:	Pagina di registrazione contenente il campo "data di nascita"		
Procedura:	<ol style="list-style-type: none">1. Aprire la pagina di Registrazione.php2. Cliccare sul campo "data di nascita"3. Inserire una data		
Risultati attesi:	Se la data inserita è coerente (non futura) il bordo del campo "data di nascita" diventa verde. Se la data è incoerente il bordo dell'input deve colorarsi di rosso.		

Test Case:	TC-003	Nome:	Validazione campo “numero civico” lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura del valore inserito nel campo “numero civico” lato client		
Prerequisiti:	Pagina di registrazione contenente il campo “numero civico”		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Cliccare sul campo “numero civico” 3. Inserire un valore 		
Risultati attesi:	Se il valore inserito presenta una struttura corretta (numeri, lettere, da una a 5 cifre) il bordo del campo “numero di civico” diventa verde. Se la struttura del valore non è corretta il bordo dell’input deve colorarsi di rosso.		

Test Case:	TC-004	Nome:	Validazione campo “nap” lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura del valore inserito nel campo “nap” lato client		
Prerequisiti:	Pagina di registrazione contenente il campo “nap”		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Cliccare sul campo “nap” 3. Inserire un valore 		
Risultati attesi:	Se il valore inserito presenta una struttura corretta (4 o 5 numeri) il bordo del campo “nap” diventa verde. Se la struttura del valore non è corretta il bordo dell’input deve colorarsi di rosso.		

Test Case:	TC-005	Nome:	Validazione campo “numero di telefono” lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura del valore inserito nel campo “numero di telefono” lato client		
Prerequisiti:	Pagina di registrazione contenente il campo “ numero di telefono”		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Cliccare sul campo “numero di telefono” 3. Inserire un valore 		
Risultati attesi:	Se il valore inserito presenta una struttura corretta (“+123456789” o “012345689”) il bordo del campo “numero di telefono” diventa verde. Se la struttura del valore non è corretta il bordo dell’input deve colorarsi di rosso.		

Test Case:	TC-006	Nome:	Validazione "email" lato client
Riferimento:	REQ-03		
Descrizione:	Controllo struttura del valore inserito nel campo "e-mail" lato client		
Prerequisiti:	Pagina di registrazione contenente il campo "e-mail"		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Cliccare sul campo "E-mail" 3. Inserire un indirizzo email 		
Risultati attesi:	Se l'indirizzo inserito presenta una struttura corretta (x@x.xx) il bordo del campo "email" diventa verde. Se la struttura del valore non è corretta il bordo dell'input deve colorarsi di rosso.		

Test Case:	TC-007	Nome:	Funzionalità bottone cancella
Riferimento:	REQ-03		
Descrizione:	Controllo della funzionalità del bottone cancella		
Prerequisiti:	Pagina di registrazione contenente tutti i campi		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina 3. Cliccare il bottone cancella 		
Risultati attesi:	Quando il bottone viene cliccato tutti i valori inseriti negli input devono essere cancellati		

Test Case:	TC-008	Nome:	Controllo campi testuali lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nei campi testuali (nome, cognomen, indirizzo, città) lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere "=" situato dopo un qualsiasi campo testuale (es. "surname") 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito è corretto si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-009	Nome:	Controllo campo “data nascita” lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo “data di nascita” lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull’URL e sostituire il valore indicato dopo il carattere “=” situato dopo il campo “date” 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito è corrisponde ad una data corretta e coerente si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-010	Nome:	Controllo campo “numero civico” lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo “numero civico” lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull’URL e sostituire il valore indicato dopo il carattere “=” situato dopo il campo “civicNumber” 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito rispetta la struttura corretta del numero civico(numero o lettere, massimo 5 caratteri) si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-011	Nome:	Controllo campo “nap” lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo “nap” lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere “=” situato dopo il campo “nap” 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito rispetta la struttura corretta del nap (4 o 5 numeri) si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-012	Nome:	Controllo campo “numero telefono” lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo “numero telefono” lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere “=” situato dopo il campo “nap” 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito rispetta la struttura corretta del numero di telefono (1234567890, +123456789) si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-013	Nome:	Controllo campo "email" lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo "email" lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere "=" situato dopo il campo "email" 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore inserito rispetta la struttura corretta dell'email (x@x.xx) si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-014	Nome:	Controllo campo "genere" lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo "genere" lato server		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere "=" situato dopo il campo "gener" 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore è corretto (m, f o altro) si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-015	Nome:	Controllo campo "hobby" e "professione" lato server
Riferimento:	REQ-05		
Descrizione:	Controllo del valore inserito nel campo "hobby" o "professione"		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Cliccare sull'URL e sostituire il valore indicato dopo il carattere "=" situato dopo il campo "hobby" o "professione" 5. Inserire un valore 6. Cliccare su avanti 		
Risultati attesi:	Se il valore presenta solo caratteri consentiti ed un massimo di 500 caratteri si viene spostati sulla pagina finale, altrimenti si viene riportati alla pagina di registrazione		

Test Case:	TC-016	Nome:	Controllo creazione cartella Registrazioni, file csv e scrittura file
Riferimento:	REQ-06		
Descrizione:	Controllo creazione della cartella Registrazioni, file csv e scrittura dati corretta nei file		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati		
Procedura:	<ol style="list-style-type: none"> 1. Aprire la pagina di Registrazione.php 2. Riempire tutti i campi della pagina con dei valori accettabili 3. Cliccare sul bottone Avanti Confermare di voler procedere 4. Aprire la cartella src 5. Se ci sono i file .csv aprirli 		
Risultati attesi:	All'interno della cartella src ci sarà la cartella Registrazioni con all'interno I file csv, quello Registrazione_aaaa-mm-dd.csv e Registrazioni_tutte.csv. Se vi sono I file corretti all'interno dovrebbero esserci I dati corretti separate dal carattere ";"		

Test Case:	TC-017	Nome:	Controllo lettura e stampa dal file csv giornaliero
Riferimento:	REQ-06		
Descrizione:	Controllo della lettura e stampa del file giornaliero sull'ultima pagina del sito		
Prerequisiti:	Pagina di registrazione, pagina di riassunto dati e pagina finale		
Procedura:	<ol style="list-style-type: none">1. Aprire la pagina di Registrazione.php2. Riempire tutti i campi della pagina con dei valori accettabili3. Cliccare sul bottone Avanti Confermare di voler procedere4. Premere sul bottone Avanti nella pagina di riassunto dati Confermare		
Risultati attesi:	Sulla pagina finale verranno stampate tutte le registrazioni giornaliere in una tabella		

4.2 Risultati test

Codice-Test	Risultato	Note
TC-001	Passato	Nessuna
TC-002	Passato	Nessuna
TC-003	Passato	Nessuna
TC-004	Passato	Nessuna
TC-005	Passato	Nessuna
TC-006	Passato	Nessuna
TC-007	Passato	Nessuna
TC-008	Passato	Nessuna
TC-009	Passato	Nessuna
TC-010	Passato	Nessuna
TC-011	Passato	Nessuna
TC-012	Passato	Nessuna
TC-013	Passato	Nessuna
TC-014	Passato	Nessuna
TC-015	Passato	Nessuna
TC-016	Passato	Nessuna
TC-017	Passato	Nessuna

4.3 Mancanze/limitazioni conosciute

Il campo telefonico non presenta come concordato la possibilità di includere il carattere “-“ e gli spazi, ma accetta solo cifre e (facoltativamente il carattere “+” per il prefisso).

Inoltre per concludere ogni riga di dati nei file csv ho aggiunto i caratteri “//”, in modo da riuscire a dividere riga per riga e poterle successivamente stampare sulla pagina finale. Nel qdc non vi è nessuna condizione sul carattere di chiusura, ma avendo avuto dei problemi e successivamente una mancanza di tempo non sono riuscito ad implementare la divisione delle righe con il carattere “a capo (\n)”.

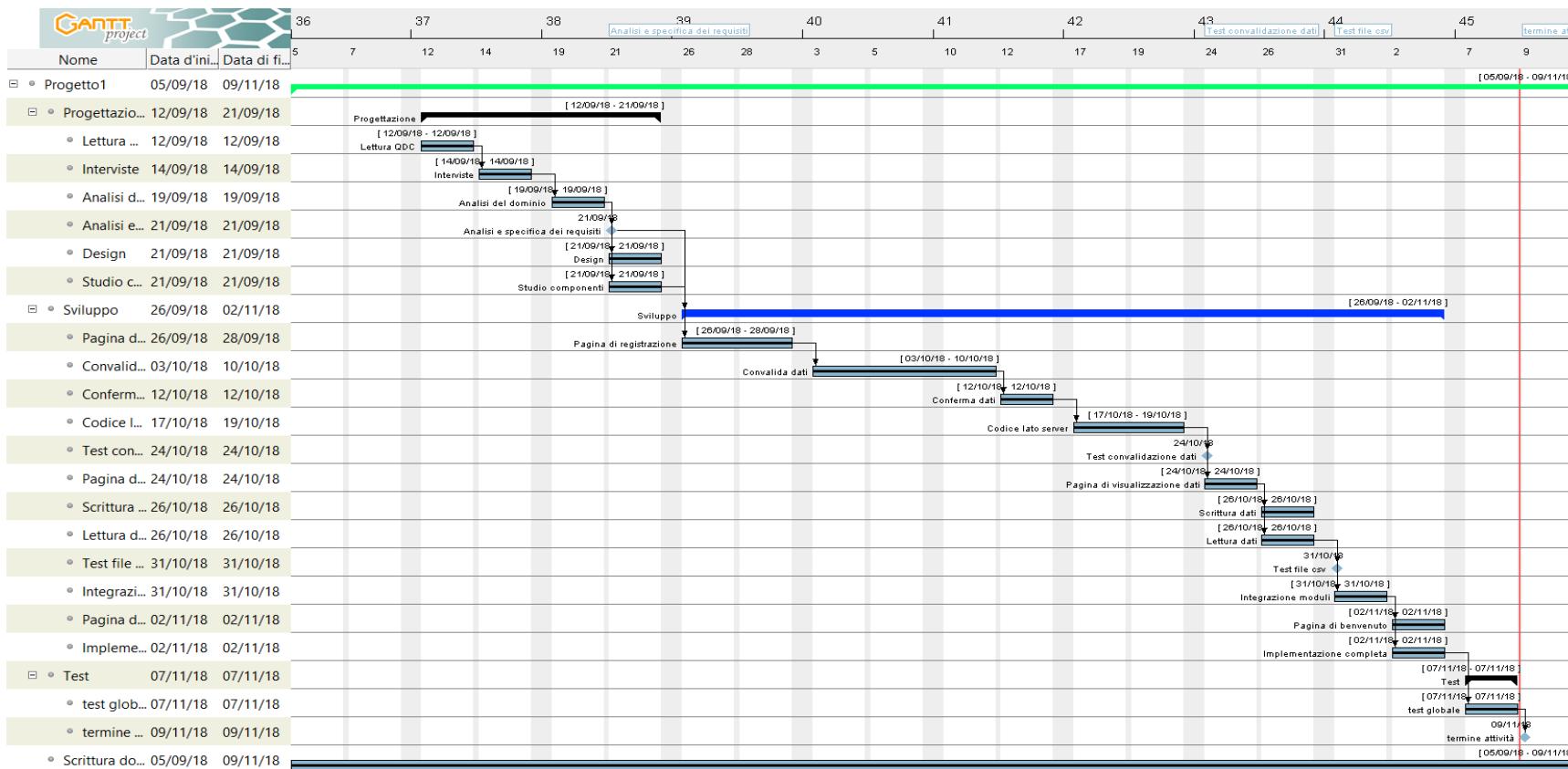
5 Consuntivo

Il consuntivo finale non è molto diverso da quello che ho pianificato nella fase di analisi, ci sono state attività che sono durate più del previsto ed altre di meno. Mi sono trovato in leggero ritardo durante la fase di sviluppo, ma ho recuperato nella parte di integrazione dei moduli e implementazione. Sono riuscito a finire il progetto e tutto il materiale che deve essere consegnato per la data prestabilita.

Nella pagina successiva ho inserito il diagramma di Gantt del consuntivo

Scuola Arti e Mestieri Trevano

Sezione informatica



Titolo del progetto:

Progetto 1

Alunno/a:

Bryan Beffa

Classe:

Info I3AA

Anno scolastico:

2018/2019

Docente responsabile:

Adriano Barchi, Luca Muggiasca, Francesco Mussi

6 Conclusioni

Questo prodotto offre un sistema di registrazione dati non sensibili. Questo progetto può essere esteso per creare una pagina di login, un sito di statistiche di dati. Non esisteva una situazione precedente per cui è stato richiesto questo progetto.

Sviluppi futuri

Sulla base di questo prodotto può essere sviluppato un database per la sostituzione dei file csv. Può essere implementato anche un sistema di login ed essere utilizzato per accedere ad aree riservate ad un singolo utente. Oppure come detto in precedenza come sito di statistica dati in cui i dati finali vengono stampati. I sistemi di registrazione sono molto frequenti e utilizzati. In questo caso si tratta di dati non sensibili, ma in futuro potrebbero essere aggiunti anche dei campi come password.

6.1 Considerazioni personali

In questo progetto ho imparato a implementare le conoscenze che ho appreso nei vari moduli e combinarle tra loro. Ho approfondito la mia conoscenza per quanto riguarda php. Ho trovato interessante questa combinazione di più moduli insieme, che hanno fruttato un lavoro più completo rispetto a quelli che ci sono stati assegnati in questi primi due anni di formazione. Inoltre sono soddisfatto del risultato ottenuto. La pianificazione che ho effettuato ha aiutato molto buona realizzazione del prodotto.

7 Bibliografia

Per la realizzazione di questo progetto non ho consultato riviste, libri, manuali ma solo siti web.

7.1 Sitografia

- <https://www.draw.io/>, Draw.io, dal 21.09.2018 al 28.09.2018
- <https://www.w3schools.com/>, W3schools dal 05.10.2018 al 24.10.2018

8 Allegati

Elenco degli allegati:

- Diari di lavoro
- Prodotto
- Quaderno dei compiti
- Diagrammi di Gantt
- Codici sorgente, git <https://github.com/bryanbeffa/progetto1.git>

Titolo del progetto:	Progetto 1
Alunno/a:	Bryan Beffa
Classe:	Info I3AA
Anno scolastico:	2018/2019
Docente responsabile:	Adriano Barchi, Luca Muggiasca, Francesco Mussi

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-07

Lavori svolti

Le prime due ore ci è stato illustrato nel dettaglio come deve essere strutturata la documentazione dei progetti che dovremo affrontare. I test sono molto importanti da fare e riportare per controllare il perfetto funzionamento del progetto. La qualità del risultato dipende dalla profondità dei test effettuati. Vi deve essere il procedimento accurato per effettuare il test nel modo corretto. Per ogni test deve essere fatta una tabella dei risultati ed eventuali annotazioni su cosa funziona e cosa no. Bisogna fare un consuntivo in cui si annotano le differenze che riguardano tra la pianificazione fatta ad inizio progetto e il tempo di lavoro effettivo per ogni fase. A fine documentazione bisogna annotare tutte le fonti suddividendole in tre categorie diverse. Alla documentazione bisogna allegare tutti i diari, i codici sorgenti. Una volta finita la dimostrazione del modello di documentazione abbiamo iniziato con la parte teorica. Ci è stata proposta una presentazione sul ciclo di vita di un progetto. Abbiamo parlato dei vari modelli di gestione di un progetto. Successivamente abbiamo guardato una seconda presentazione sul come va fatta un'analisi di un progetto.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

Porre domande sulle parti che troviamo poco chiare del progetto.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-12

Lavori svolti

Ad inizio lezione abbiamo avuto tempo per porre i nostri quesiti riguardanti il “qdc”. Sono state poste molte domande e successivamente ci è stato indicato il file contenente le risposte. Nella parte successiva di lezione abbiamo avuto tempo per iniziare la parte di analisi. Ho iniziato con il punto 1.5. Inoltre ho sfruttato un buco di 5 minuti nella lezione per scaricare il programma “GanttProject” un software che gratuito che permette la creazione di un diagramma gant. La parte di analisi che stavo affrontando non è però giunta al termine e dovrà essere quindi continuata in un secondo momento.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

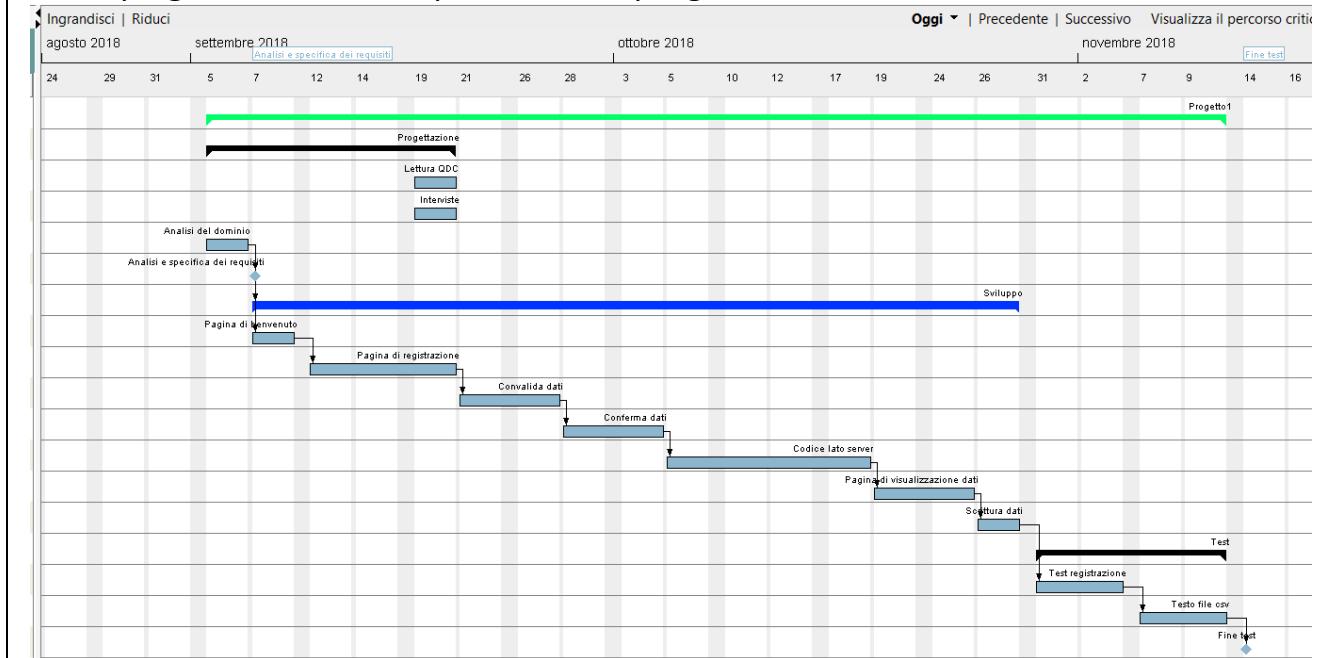
Ho in programma per la prossima lezione di terminare la parte di analisi della documentazione.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-14

Lavori svolti

Le prime due abbiamo continuato la documentazione, nello specifico ho scritto i requisiti del progetto. Ho diviso il progetto in 7 requisiti. Nella seconda parte di lezione partendo dai requisiti scritti in precedenza ho iniziato a progettare il gantt con il sw GanttProject. Nel gantt ho inserito 3 fasi principali che si possono riassumere in Progettazione, Sviluppo, Test. Inoltre ho terminato la parte di analisi in cui si spiega cosa e come è presentato il progetto.



Problemi riscontrati e soluzioni adottate

Ho dovuto riguardare e dovrò ancora riguardare come è strutturato un Gantt.

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

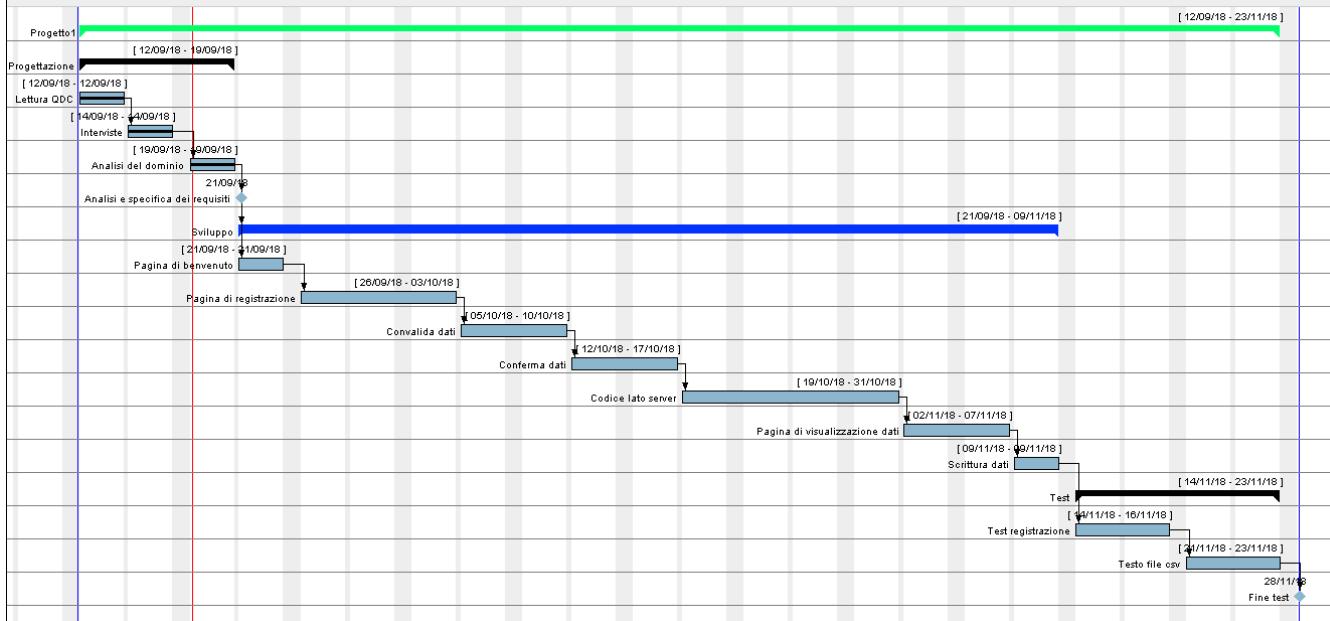
Ho in programma di terminare il Gantt e continuare tutta la prima parte di documentazione inerente all'analisi.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-19

Lavori svolti

Durante la prima ora ci sono stati controllati i requisiti e abbiamo potuto correggerli nel modo consigliato. Successivamente ci è stato assegnato del tempo per migliorare



e modificare il diagramma di Gantt iniziato la scorsa lezione.

Nella seconda parte di lezione ho scritto nella documentazione la parte di analisi del dominio (punto 1.4 della doc).

Problemi riscontrati e soluzioni adottate

Punto della situazione rispetto alla pianificazione

Rispetto al diagramma Gannt che ho creato risulta in leggero anticipo sulle giornate di lavoro.

Programma di massima per la prossima giornata di lavoro

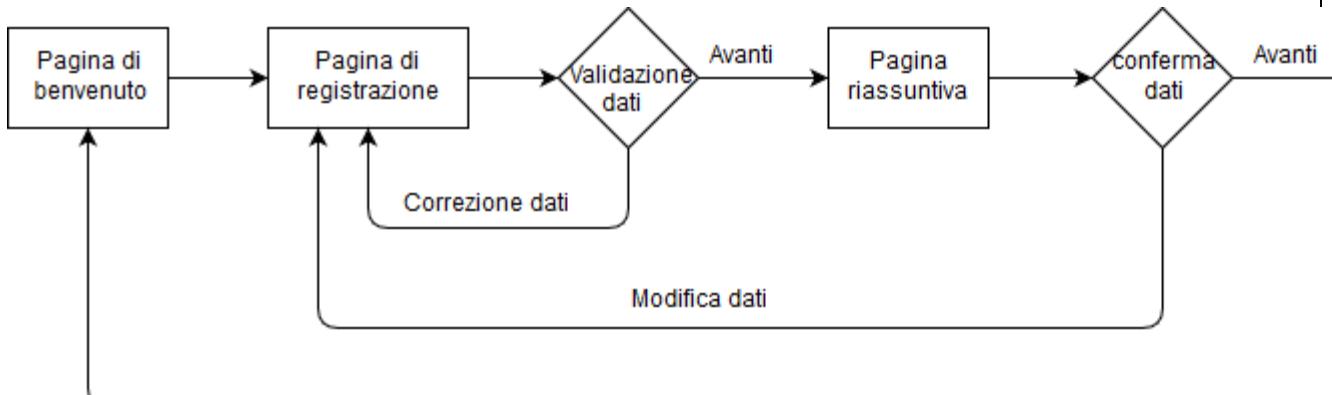
Nella prossima giornata di lavoro dovrei iniziare la pagina di benvenuto e dovrei riuscire a finirla in una sola lezione e iniziare anche la pagina di registrazione.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-21

Lavori svolti

Le prime due ore ci è stata proposta una presentazione sulla progettazione. Successivamente ho modificato nuovamente il gantt, ma non è ancora una versione definitiva. Le seguenti due ore ho iniziato la parte di progettazione (capitolo 2 della documentazione). Ho terminato il Design dell'architettura del sistema aggiungendo un diagramma di flusso. Ho iniziato il design dei dati e aggiunto i sw utilizzati fino ad adesso.



Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

Per la prossima settimana ho il programma di continuare la parte di progettazione.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-26

Lavori svolti

Oggi abbiamo svolto la verifica teorica del modulo.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

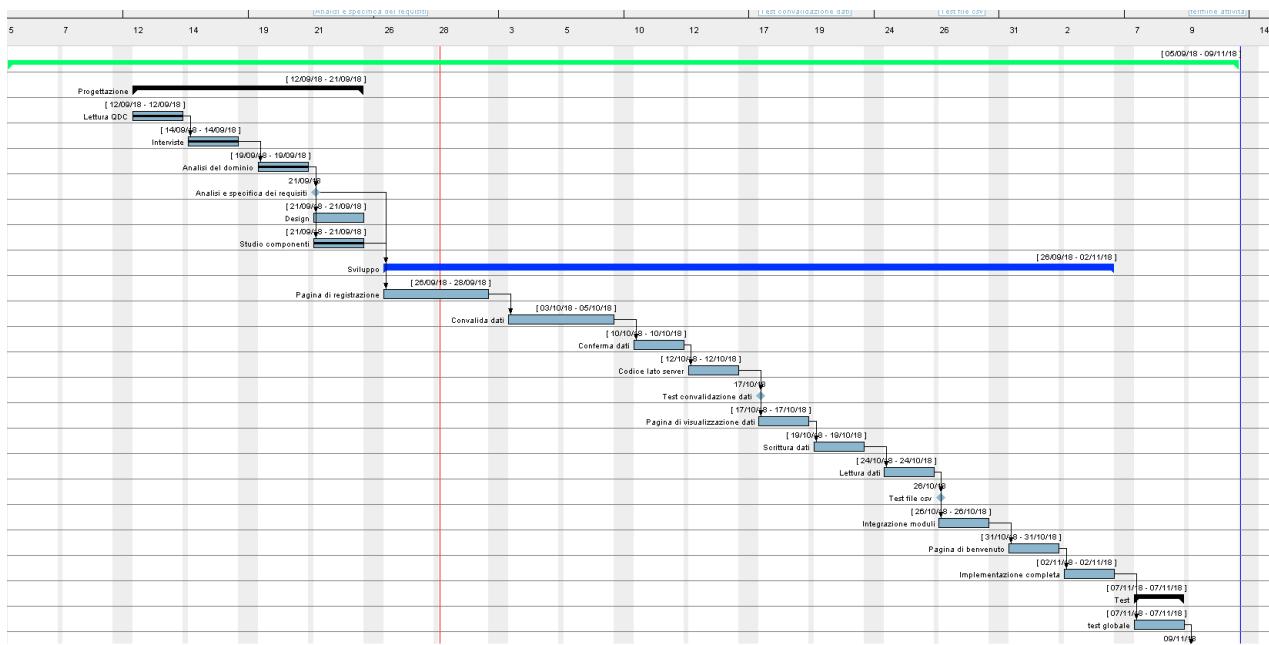
Per la prossima settimana ho il programma di continuare la parte di progettazione.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-28

Lavori svolti

Oggi abbiamo lavorato per quattro ore sulla documentazione. Inizialmente ho apportato delle modifiche al diagramma di gantt aggiungendo alcune attività, modificando l'ordine, le precedenze e l'aggiunta di alcune milestone.



Successivamente ho fatto il capitolo di design dei dati, delle interfacce tramite excel (bozza) e iniziato il diagramma di flusso completo del capitolo 2.4.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Rispetto alla mia pianificazione mi trovo un po' in ritardo ma con la prossima lezione dovrei riuscire a recuperare.

Programma di massima per la prossima giornata di lavoro

La prossima giornata di lavoro devo continuare il diagramma di flusso e terminarlo. Inoltre devo accurare la parte di analisi.

Bryan Beffa

modulo 306

progetto 1

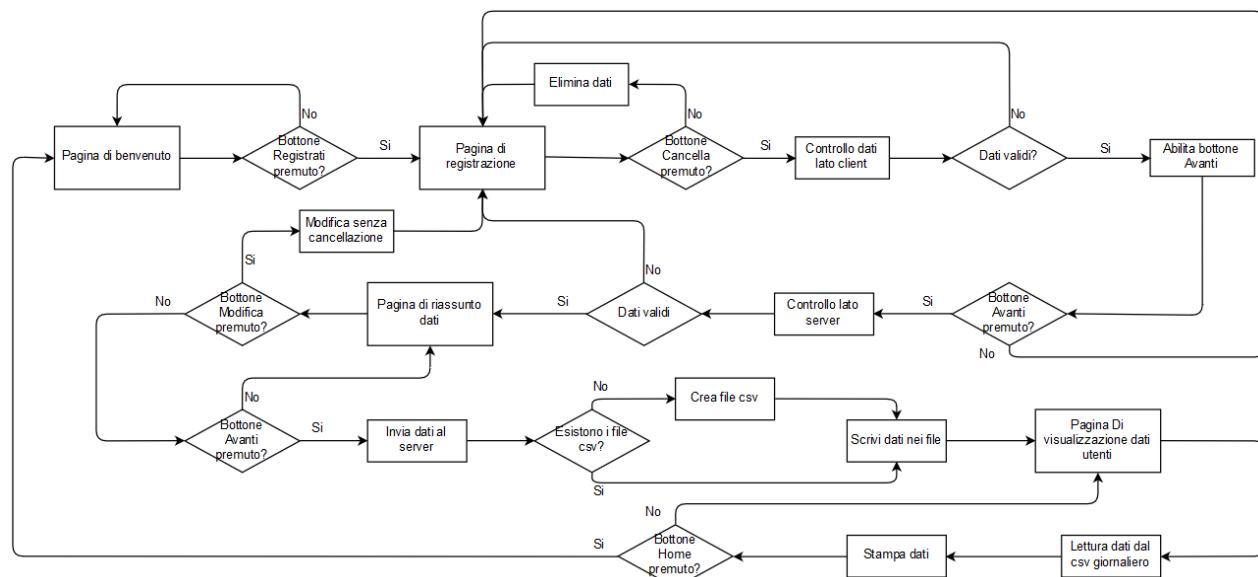
Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-03

Lavori svolti

La prima ora ci è stata proposta una presentazione sulla fase d'implementazione in un progetto. Ci è stato riferito quanto sia importante la fase di test e che può anche occupare fino al 30% del tempo impiegato nel progetto. Ogni test deve essere riportato in una tabella in cui si spiega cosa viene testato e come. Alla fine si mette una tabella con i test totali, il risultato ed eventualmente una nota (commento).

Nell'ultima seconda parte di lezione abbiamo continuato con la nostra documentazione.



Ho terminato il diagramma di flusso del capitolo 2.4. Ho iniziato la classe CsvFile.php.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

Bryan Beffa

modulo 306

progetto 1



Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-05

Lavori svolti

Ho separato tutti i file .css e .js dalle pagine .html e .php. Ho creato una struttura delle cartelle nella cartella src. Nella cartella src ci sono i file .html e .php, la cartella script con dentro i file .js, la cartella style con all'interno i file .css e la cartella img. Ho creato tutte le pagine a livello strutturale e ho iniziato a creare il codice lato client per i controlli e inviare i dati da una pagina all'altra

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Leggermente in ritardo

Programma di massima per la prossima giornata di lavoro

Continuare la convalida dati e terminarla

Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-10

Lavori svolti

Oggi ho continuato la struttura e il design delle pagine. Ho iniziato a fare i controlli lato client dei campi, e l'invio dei dati tramite metodo get (dati non sensibili) dalla pagina di registrazione alla pagina di riassunto dati. Se tutti i campi obbligatori non sono compilati non si può accedere alla pagina di riassunto.

```
//metodo che controlla che tutti i campi principali (obbligatori) sono stati compilati
function checkMainInputs(){
    var mainInputs = getMainInputs();
    for (var i = 0; i <= mainInputs.length-1; i++) {
        if(mainInputs[i].value == null || mainInputs[i].value == ""){
            return false;
        }
    }
    return true;
}
```

Dalla pagina di riassunto si possono modificare i dati tornando alla pagina di registrazione con il bottone modifica. I dati vengono mantenuti e non cancellati.

```
//metodo che abilita abilita il funzionamento del bottone avanti se tutti i campi obbligatori sono compilati e invia i valori alla pagina di riassunto
function enableNextButton(){
    if(checkMainInputs()){
        if(confirm("Sicuro di voler procedere?")){
            var queryString = "?name=" + inputList[0].value
            + "&surname=" + inputList[1].value
            + "&date=" + inputList[2].value
            + "&address=" + inputList[3].value
            + "&civichNumber=" + inputList[4].value
            + "&city=" + inputList[5].value
            + "&nap=" + inputList[6].value
            + "&phoneNumber=" + inputList[7].value
            + "&email=" + inputList[8].value
            + "&gener=" + inputList[9].value
            + "&hobby=" + inputList[10].value
            + "&profession=" + inputList[11].value;
            window.location.href = 'riassuntoDati.php' + queryString;
        }
    }else{
        alert("Devi compilare tutti i campi obbligatori")
    }
}
```

Ho terminato la parte grafica di tutte le pagine.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-leggermente in ritardo.

Programma di massima per la prossima giornata di lavoro

Continuare i controlli lato client ed iniziare i controlli lato server

Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-12

Lavori svolti

Ho creato la funzione che cancella tutti i dati degli input. Questa funzione viene richiamata quando si preme il tasto cancella.

```
//metodo che cancella tutti i dati inseriti
function deleteData(){
    if(confirm("Sicuro di voler cancellare tutti i dati?")){
        //numero di input da cancellare meno i 2 button finali
        var inputLength = inputList.length-2;
        for (var i = 0; i < inputLength; i++) {
            inputList[i].value = "";
            inputList[i].style.borderBottom = "0.2vw solid white";
        }
    }
}
```

Metodo che controlla che la data sia corretta dove controllo che la data inserita non sia nulla o futura.

```
//metodo che controlla che sia stata inserita una data corretta
function checkDate(id){
    //salvo i millisecondi della data odierna
    var date = new Date();
    var userDate = new Date(inputList[id].value);

    //controllo che la data non sia del futuro e che il valore non sia nullo
    if(date.getTime()-userDate.getTime()>0 && !isNaN(userDate)) {
        inputList[id].style.borderBottom = "0.2vw solid green";
    }else{
        inputList[id].style.borderBottom = "0.2vw solid red";
    }

}
```

Fonte: https://www.w3schools.com/js/js_date_methods.asp

Metodo che controlla che il nap sia corretto. Deve essere un numero di 4 o 5 cifre.

```
//metodo che controlla che sia stata inserita una data corretta
function checkDate(id){
    //salvo i millisecondi della data odierna
    var date = new Date();
    var userDate = new Date(inputList[id].value);

    //controllo che la data non sia del futuro e che il valore non sia nullo
    if(date.getTime()-userDate.getTime()>0 && !isNaN(userDate)) {
        inputList[id].style.borderBottom = "0.2vw solid green";
    }else{
        inputList[id].style.borderBottom = "0.2vw solid red";
    }

}
```

Ho aggiunto i metodi che controllano i campi non obbligatori professione e hobby.

```
//metodo che controllo che il numero di caratteri del campo hobby non sia superiore a 500 caratteri
function checkHobby(id){
    var hobby = inputList[id].value;
    if(hobby.length <= 500){
        inputList[id].style.borderBottom = "0.2vw solid green";
    }else{
        inputList[id].style.borderBottom = "0.2vw solid red";
    }
}

//metodo che controllo che il numero di caratteri del campo profession non sia superiore a 500 caratteri
function checkProfession(id){
    var profession = inputList[id].value;
    if(profession.length <= 500){
        inputList[id].style.borderBottom = "0.2vw solid green";
    }else{
        inputList[id].style.borderBottom = "0.2vw solid red";
    }
}
```

Ho modificato il metodo che controlla il numero civico. Il numero civico deve essere composto

```
//metodo che controlla se il numero civico è valido
function setCivicNumber(id){
    var value = inputList[id].value;
    var regex = /^[a-zA-Z0-9]+$/;
    var civicNumber = regex.test(value);

    if(civicNumber && value.length <= 4 && value.length > 0 && value != "0"){
        inputList[id].style.borderBottom = "0.2vw solid green";
    }else{
        inputList[id].style.borderBottom = "0.2vw solid red";
    }
}
```

Controlli eseguito e testate fino ad adesso:

- Data di nascita
- No. Civico
- Nap
- Hobby
- Professione

Controlli da svolgere:

- nome
- cognome
- indirizzo
- citta
- email

Aggiunta dell'icona info a lato degli input.

Fonte: https://www.w3schools.com/css/tryit.asp?filename=trycss_tooltip

Problemi riscontrati e soluzioni adottate

-

Terminare controllo campi

Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-17

Lavori svolti

Ho finito di creare i metodi che convalidano i dati lato client.

Ho aggiunto a tutti i metodi un return (true o false) a dipendenza del valore inserito.

Controllo in un metodo che tutti i campi siano corretti. Se sono corretti posso passare alla pagina successiva dopo una richiesta di conferma.

```
//metodo che controlla che tutti i campi principali (obbligatori) sono stati compilati
function checkMainInputs(){
    var mainInputs = getMainInputs();
    /* Controllo tutti che tutti gli input abbiano un valore valido.
       Il valore del campo genere non viene controllato perché non può mai essere non valido
    */
    if(checkText(0) && checkText(1)){
        if(checkDate(2) && checkText(3)){
            if(checkCivicNumber(4) && checkText(5)){
                if(checkNap(6) && checkPhoneNumber(7)){
                    if(checkEmail(8)){
                        return true;
                    }
                }
            }
        }
    }
    return false;
}
```

Problemi riscontrati e soluzioni adottate

-

Prossima lezione

Fare controlli lato server e scrivere nel file csv

Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-19

Lavori svolti

Oggi ho concluso la parte lato client di tutti i file. Ho creato il codice lato server per scrivere i dati nei due file csv e la pagina contente tutti i dati. Nella pagina con tutti i giornalieri.

Problemi riscontrati e soluzioni adottate

-Non riuscivo a leggere il valore degli input dal codice php. Questo perché ho disabilitato gli input invece di impostarli a read only.

Prossima lezione

Devo controllare se il file odierno è già stato creato. Devo leggere i dati dal file giornaliero. Quindi fare il controllo su quale sia il file giornaliero.

Diario di lavoro

Luogo	SAM Trevano
Data	2018-10-24

Lavori svolti

Ho continuato i controlli lato server ed inserito il campo id nei file csv.

Ho fatto una parte di documentazione.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

Terminare i controlli lato server e continuare la documentazione

Diario di lavoro

Luogo	SAM Trevano
Data	2018-11-07

Lavori svolti

Oggi ho iniziato e terminato la stesura del capitolo **implementazione**. Ho terminato i test case. E ho iniziato la presentazione. Inoltre ho caricato sul infomaniak l'intero progetto e si può raggiungere al link www.samtinfo.ch/i16befbry/web/progetto. Ho aggiunto alla pagina iniziale un messaggio di avviso, se l'utente disattiva js, questo appare.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Sono a buon punto, in leggero anticipo

Programma di massima per la prossima giornata di lavoro

Terminare la presentazione e consegnarla

Diario di lavoro

Luogo	SAM Trevano
Data	2018-11-09

Lavori svolti

Oggi ho rilegato tutta la documentazione e creato il documento powerpoint per la presentazione. Inoltre ho dovuto ricaricare il progetto sull'FTP. Ho consegnato il progetto.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

Programma di massima per la prossima giornata di lavoro

Diario di lavoro

Luogo	SAM Trevano
Data	2018-09-05

Lavori svolti

Ci è stato introdotto il modulo 306. Successivamente i docenti hanno parlato di come va gestito un progetto e documentato. Dopo ci è stato anticipato il nostro primo progetto, che sarà individuale e principalmente incentrato su php. Il progetto consiste nel creare un piccolo forum dove potremo controllare gli accessi o eventualmente registrarci.

Abbiamo avuto tempo per porre le nostre domande a riguardo.

Problemi riscontrati e soluzioni adottate

-

Punto della situazione rispetto alla pianificazione

-

Programma di massima per la prossima giornata di lavoro

Presentare un diagramma di Gantt per la prossima lezione e successivamente discuterne in classe.

Pensare ad eventuali domande da porre inerenti al progetto.