



erasmus activity after the covid-19



Relazione progetto d'esame per
tecniche di sviluppo software

Realizzato da
FRANCESCA ROSSI
LEANDRO CORSANO

SOMMARIO

I. INTRODUZIONE

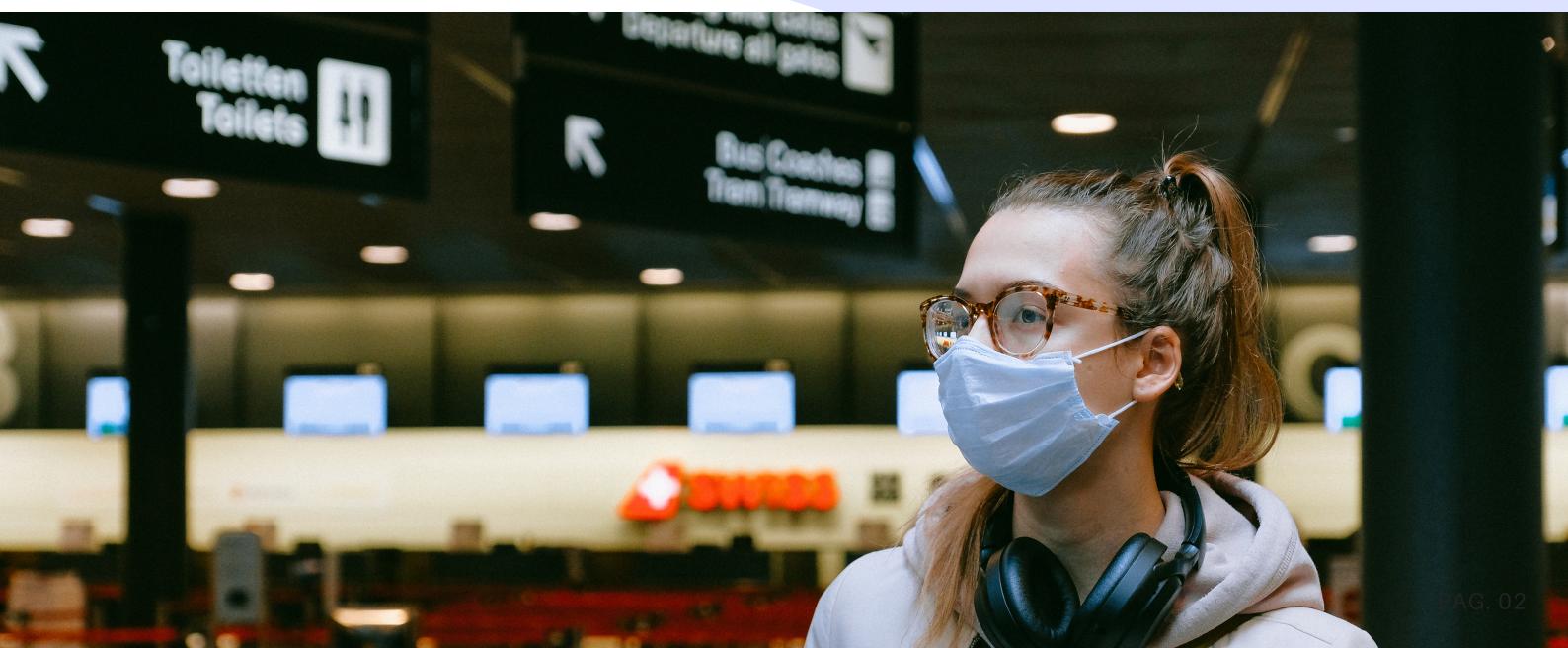
<i>Analisi del problema</i>	3
<i>Breve descrizione</i>	3
<i>Scopo del progetto</i>	3
<i>Descrizione dettagliata</i>	4

II. IMPLEMENTAZIONE

<i>Modello relazionale</i>	5
<i>Modello logico</i>	6
<i>Linguaggio, software e librerie</i>	6
<i>Principali funzionalità</i>	7
<i>Archittetura del software</i>	7
<i>Analisi dei servizi WCF</i>	8
<i>Principali interfacce grafiche</i>	8

III. CONCLUSIONI

9



Analisi del problema

Nei primi mesi del 2020 una grossa pandemia ha colpito il mondo. Questa pandemia ha colpito e stravolto diversi settori, in particolare quello dell'entertainment. Nei prossimi mesi saranno limitate molte attività culturali, di svago e associative e purtroppo, a rimetterci, saranno anche molti studenti Erasmus che non potranno prenderne parte, rovinando un'esperienza unica della loro vita.

Breve descrizione del progetto

La soluzione a questo problema è la piattaforma ErasEasyLife. Una piattaforma che ha l'intento di aiutare l'organizzazione di tutte le attività extra-scolastiche dello studente Erasmus seguendo le norme governative per la partecipazione ai vari eventi. ErasEasyLife è pensata, non solo per gli studenti, ma anche per le associazioni aiutandole nell'organizzazione dei suddetti eventi.

Obiettivo del progetto

Lo scopo di questa piattaforma è la creazione da parte dell'associazioni di eventi, l'iscrizione agli eventi da parte degli studenti e dei volontari, per ottimizzare al meglio i tempi di organizzazione e ridurre il rischio di contagi, assicurando divertimento e relax a tutti gli studenti.

Descrizione dettagliata

La piattaforma include 3 aree personali: l'**associazione**, lo **studente** e il **volontario**.

Un'associazione organizza diversi **eventi**, anche di più giorni (gita) per gli studenti Erasmus. I **volontari**, che fanno parte di una determinata associazione, si occupano della gestione di uno o più eventi. Ogni **associazione** può organizzare più eventi, in luoghi diversi, ma anche nello stesso momento. Ogni **studente** può partecipare a più eventi e ad ogni **evento** possono essere presenti più **volontari**.

Alla piattaforma ogni tipologia di utente si registra in modo autonomo e può accedervi tramite un login(email e password).

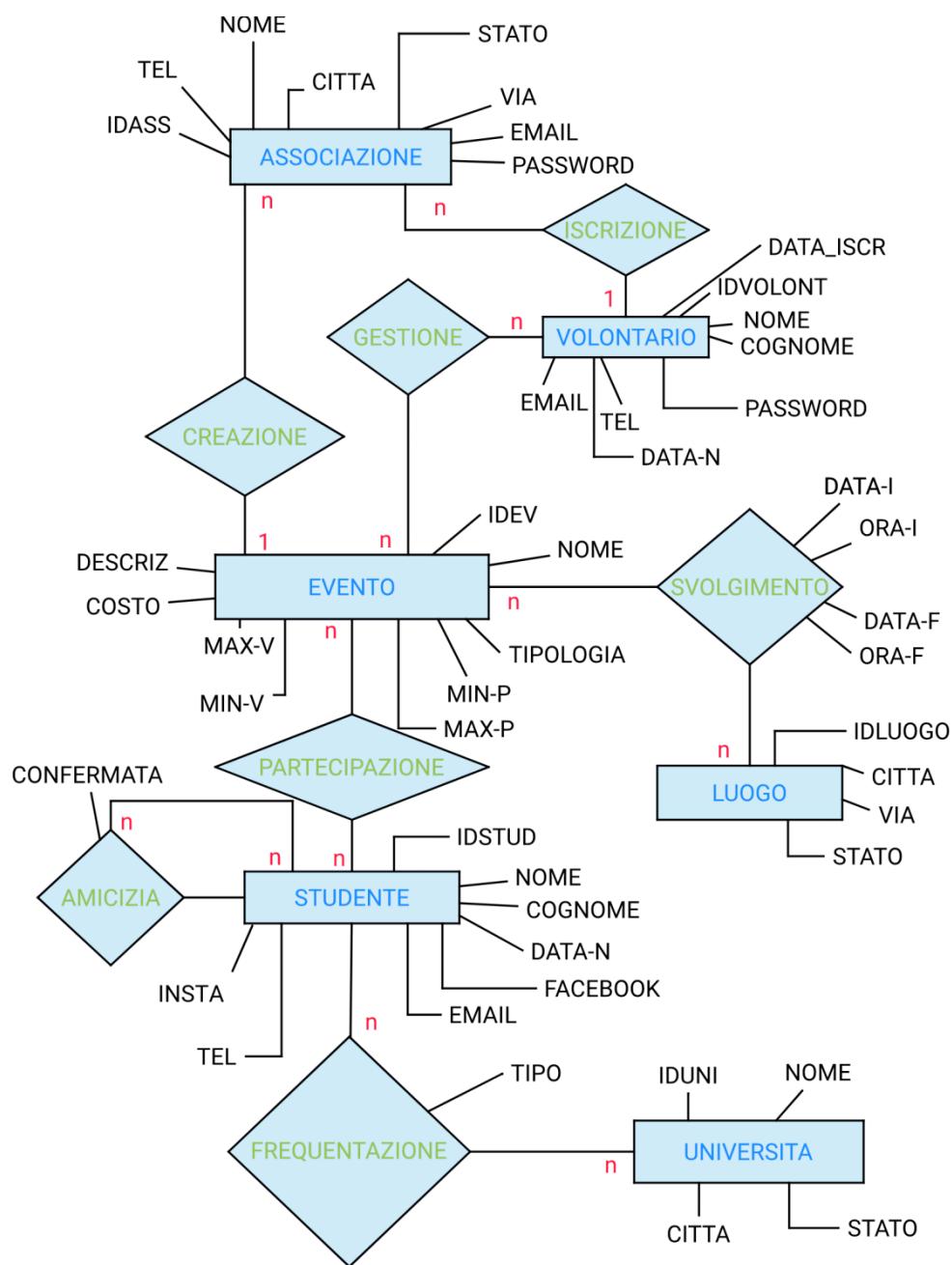
Durante la registrazione per **l'associazione** è obbligatorio indicare **ragione sociale**, **sede in cui opera** (città, via, stato), **numero di telefono**, **email**; per gli **studenti**: **nome**, **cognome** , **università di partenza**, **università di arrivo**, **numero di telefono**, **email**, **data di nascita**, **nazionalità**; per i **volontari**: **nome**, **cognome**, **data di nascita**, **email**, **numero di telefono**, **data d'iscrizione all'associazione**;

L'associazione ha il compito di confermare i suoi volontari attribuendogli un **ruolo**. Di conseguenza un **volontario** non confermato può entrare, ma non può accedere ai servizi della piattaforma. È inoltre possibile, per uno **studente**, vedere la *community* degli altri studenti Erasmus. In questa community è possibile chiedere *l'amicizia*, e successivamente contattare l'amico.

Per ogni **evento** il sistema memorizza: **il nome**, **la data d'inizio**, **la data di fine**, **l'ora d'inizio**, **l'ora di fine**, **la tipologia di evento**, **il luogo dell'evento**, **il minimo e il massimo numero di studenti partecipanti**, **il minimo e il massimo numero di volontari**, **il costo** e **la descrizione dell'evento**.

Un **evento** può essere modificato o annullato dall'**associazione**. In questo caso il sistema provvederà ad inviare un'apposita *email* ai partecipanti. Infine le **associazioni** possono inserire degli "**eventi speciali**" chiamati *riunioni* visibili solo ai **volontari** dell'associazione.

Modello Relazionale



Modello logico

STUDENTE(idstud, nome, cognome, email, cellulare, data_n, citta, stato, nazionalita, password)

UNIVERSITA(iduni, nome, citta, stato)

ASSOCIAZIONE(idass, nome, citta, stato, via, tel, email, password)

EVENTO(idev, nome, tipologia, min_p, min_v, max_p, max_v, costo, descrizione, idass)

FK: EVENTO.idass=ASSOCIAZIONE.idass

VOLONTARIO(idvolont, nome, cognome, data_n, email, telefono, data_iscrizione, password, idass)

LUOGO (idluogo, citta , via, stato)

FREQUENTAZIONE(tipo, idstud, iduni) N.B. tipo="sending", "receiving"

FK: FREQUENTAZIONE.idstud=STUDENTE.idstud

FREQUENTAZIONE.iduni=UNIVERSITA.iduni

AMICIZIA(confermata, idstud1, idstud2) N.B. confermata="richiesta", "conferma", "annulla"

FK: AMICIZIA.idstud1=STUDENTE.idstud

AMICIZIA.idstud2=STUDENTE.idstud

PARTECIPAZIONE(idstud, idev)

FK: PARTECIPAZIONE.idstud=STUDENTE.idstud

PARTECIPAZIONE.idev=EVENTO.idev

GESTIONE(idvolont, idev)

FK: GESTIONE.idvolont=VOLONTARIO.idvolont

GESTIONE.idev=EVENTO.idev

SVOLGIMENTO(idev, idluogo, ora_i, ora_f, data_i, data_f)

FK: SVOLGIMENTO.idev=EVENTO.idev

SVOLGIMENTO.idluogo=EVENTO.idev

Linguaggio, software e librerie

Linguaggio di sviluppo : c# 8

Piattaforma di sviluppo: microsoft .NET framework

DBMS: MySQL (<https://www.freysqlatabase.com/>)

Altre librerie:

- Grafica: Admin LTE , template di grafica in bootstrap
- Icône generali: fontawesome (<https://fontawesome.com/>)
- Icône bandiere: countryflags (<https://www.countryflags.io>)

Principali funzionalità

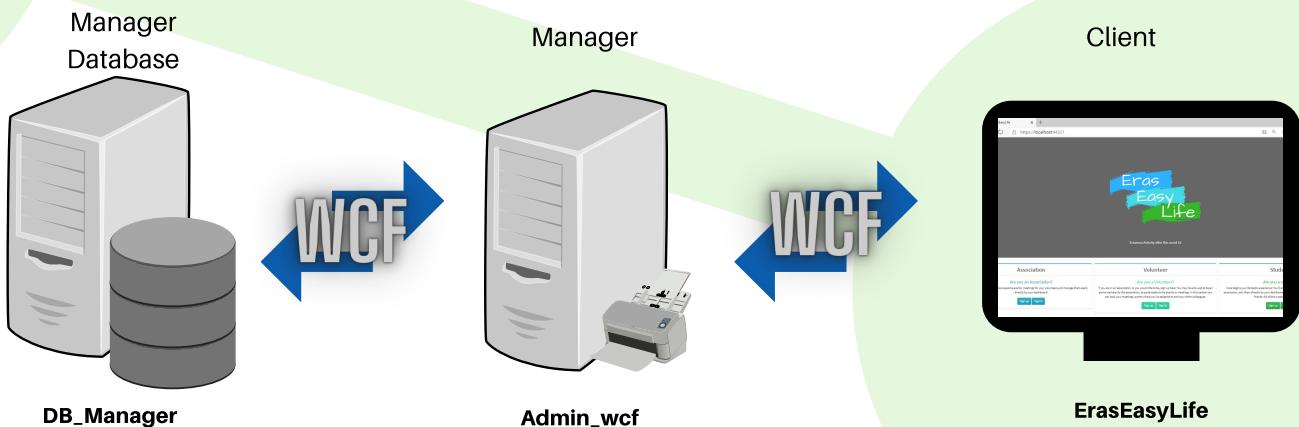
```

6     List<ErasEasyLife.Event.Studente> studenti = ViewData["studenti"] as List<Eras
7     int posti_rimasti = s.evento.max_p - studenti.Count();
8     ErasEasyLife.Student.Studente studente_entrato = (ErasEasyLife.Student.Studente)
9     DateTime Date1 = DateTime.Parse(s.data_i);
10    DateTime Date2 = DateTime.Parse(s.data_f);
11    bool prenotato = false; //mi dice se ho già prenotato quest'evento
12    foreach (var stud in studenti)
13    {
14      if (stud.IdStud == studente_entrato.IdStud)

```

- Registrazione e autenticazione dello studente, dell'associazione e del volontario
- Creazione di un area riservata per ogni tipologia di utente
- Visualizzazione e modifica dati personali di ogni utente
- Creazione, modifica, eliminazione degli eventi e delle riunioni da parte dell'Associazione
- Visualizzazione degli eventi/riunioni per i volontari e gli studenti
- Possibilità di partecipazione agli eventi da parte del volontario/studente, possibilità di disdire la partecipazione e possibilità di partecipazione alle riunioni da parte del volontario.
- Abilitazione del volontario, attribuendogli un ruolo da parte dell'associazione
- Gestione delle amicizie nella community dello studente
- Registrazione delle proprie università da parte dello studente
- Avviso, tramite mail, di modifica o annullamento di un evento o una riunione

Architettura del software



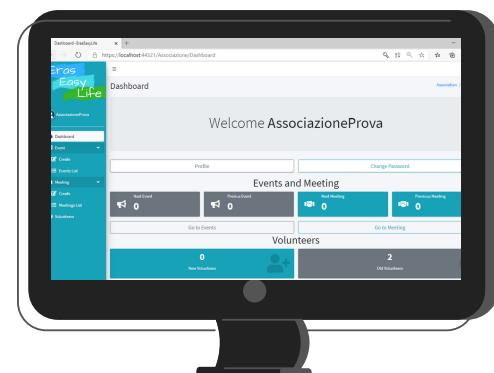
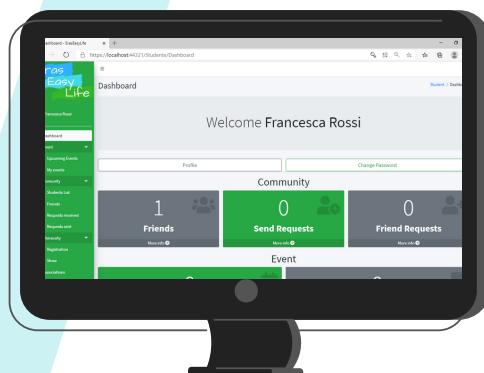
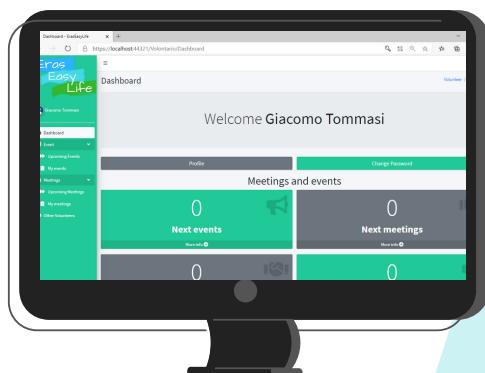
Analisi dei servizi WCF

Nel Database Manager troviamo il servizio DBManager, che implementa la connessione con il database e funzioni generiche di creazione automatica delle query.

All'interno del Manager invece, troviamo un servizio per ogni tipologia di utente, e un servizio per la gestione degli eventi. Inoltre troviamo tutte le classi che ci permettono di creare degli oggetti per far interagire i vari servizi tra di loro e con il client.

Principali interfacce grafiche

Per realizzare la grafica della piattaforma è stato usato un template Bootstrap, ADMIN LTE. Questo template è stato riadattato per le esigenze di questa piattaforma.



L'idea:

ErasEasyLife è pensata da ex studenti Erasmus per i prossimi studenti Erasmus.

Punti di forza

100% user friendly:

ErasEasyLife cerca di avvicinarsi all'utente con interfacce semplici ed intuitive.



100% Resposive:

ErasEasyLife è creata per qualsiasi dispositivo.



Cosa abbiamo imparato da questo progetto?

ErasEasyLife ci ha insegnato a collaborare, utilizzando strumenti di collaborazione online (Github, Canva, One Drive), ma anche attraverso le incomprensioni, il dialogo, e la suddivisione dei compiti.

Inoltre abbiamo imparato ad utilizzare un importante strumento di sviluppo software quale Visual Studio. Con questa piattaforma ci siamo immersi (e certe volte persi) nel mondo Microsoft, imparando a programmare con C# e .NET Framework.

Concludendo possiamo dire che grazie a questo progetto siamo maturati e migliorati nell'affrontare i problemi di programmazione.