

## **Gestione Iscrizioni Promtec**

**Titolo del progetto:** Gestione Iscrizioni Promtec  
**Alunno/a:** Alexandru Ciobanu  
**Classe:** Info 4BB  
**Anno scolastico:** 2024/2025  
**Docente responsabile:** Fabio Piccioni

1	Introduzione .....	4
1.1	Informazioni sul progetto .....	4
1.2	Abstract .....	4
1.3	Scopo .....	4
2	Analisi .....	5
2.1	Analisi del dominio .....	5
2.2	Analisi e specifica dei requisiti .....	5
2.2.1	Spiegazione elementi tabella dei requisiti: .....	8
2.3	Use case .....	9
2.4	Pianificazione .....	10
2.5	Analisi dei mezzi .....	11
2.5.1	Software .....	11
2.5.2	Hardware .....	11
3	Progettazione .....	12
3.1	Design dell'architettura del sistema .....	12
3.2	Design dei dati e database .....	13
3.2.1	Descrizione campi tabelle .....	15
3.3	Design delle interfacce .....	18
3.3.1	Pagina di login .....	18
3.3.2	Pagina gestione slot .....	18
3.3.3	Pagina dettagli slot .....	19
3.3.4	Pagina gestione utenti .....	19
3.3.5	Pagina iscrizione studenti .....	20
3.4	Design procedurale .....	21
3.4.1	Ruolo utente .....	21
3.4.2	Attivazione utente .....	21
3.4.3	Cancellazione utente .....	21
3.4.4	Approvazione utente .....	22
3.4.5	Azioni organizzatore SAMT (admin) .....	23
3.4.6	Azioni responsabile scuole medie (user) .....	24
4	Implementazione .....	25
4.1	Struttura cartelle .....	25
4.1.1	Struttura del Backend .....	25
4.1.2	Struttura del Frontend .....	26
4.1.3	Configurazione del Database .....	26
4.1.4	Configurazione NGINX .....	26
4.2	Docker .....	27
4.2.1	Avvio progetto .....	27
4.2.2	Variabili di ambiente .....	27
4.2.3	Container database .....	29
4.2.4	Container backend .....	30
4.2.5	Container frontend .....	30
4.2.6	Container Nginx .....	31
4.2.7	Rete container .....	33
4.2.8	Volumi container .....	33
4.2.9	Certificati SSL .....	33
4.2.10	Script di inizializzazione MySQL .....	33
4.3	Ambiente virtuale per lo sviluppo .....	34
4.4	Informazioni organizzatore .....	35
4.5	Scuole medie disponibili .....	36
4.6	Registrazione e accesso .....	36
4.6.1	Modello User .....	37
4.6.2	Autenticazione e token .....	37
4.6.3	Approvazione utenti .....	38
4.6.4	Auth storage .....	40
4.6.5	Sicurezza .....	41
4.7	Notifiche email .....	42
4.8	Reset password .....	43

4.9	Gestione utenti .....	44
4.10	Creazione slot.....	46
4.11	Iscrizione studenti da utente.....	48
4.11.1	Crittografia studenti.....	51
4.12	Dettagli slot.....	53
4.13	Conferma iscrizioni .....	55
4.14	Esportazione lettere.....	58
4.15	Protocollo di test .....	60
4.16	Risultati test .....	75
4.17	Mancanze/limitazioni conosciute .....	76
5	Consuntivo.....	77
6	Conclusioni .....	78
6.1	Sviluppi futuri.....	78
6.2	Considerazioni personali.....	79
7	Glossario .....	80
8	Indice delle figure .....	82
9	Bibliografia.....	83
9.1	Sitografia .....	83
9.2	Utilizzo IA .....	83
10	Allegati .....	83

## **1 Introduzione**

### **1.1 Informazioni sul progetto**

- Titolo Progetto: Gestione Iscrizioni Promtec
- Allievi: Alexandru Ciobanu I4BB,
- Docente responsabile: Fabio Piccioni
- Perito 1: Manuele Nolli
- Perito 2: Alberto Marletta
- Scuola: Arti e Mestieri Trevano, sezione Informatica
- Data di inizio e fine: 05.05.2025 – 16.05.2025
- Data di presentazione: 26.05.2025

### **1.2 Abstract**

*The growing need to simplify and digitize the registration process for promotional days at the Scuola Arti e Mestieri led to the development of the Promtec system. Previously, this process was carried out manually, requiring significant time and effort from administrative staff. The goal of the project was to create an intuitive and efficient web-based platform for school administrators and middle school coordinators, enabling the planning of promotional slots, student registrations, waiting list management, and the automated sending of notifications. The system includes mechanisms to protect the personal data of underage students through encryption. The implemented solution will significantly improve operational efficiency, reduce administrative workload, and minimize errors compared to the manual procedure.*

### **1.3 Scopo**

Lo scopo di questo progetto è realizzare una piattaforma dedicata alla gestione delle iscrizioni alle giornate promozionali informative organizzate dalla Scuola Arti e Mestieri, con l'obiettivo di semplificare e rendere più efficiente il lavoro degli amministratori scolastici e dei referenti delle scuole medie. Gli organizzatori della SAMT avranno la possibilità di pianificare gli eventi, distribuire gli studenti nei vari slot disponibili, gestire automaticamente le liste d'attesa e ricevere notifiche relative alle prenotazioni.

Le iscrizioni degli studenti saranno effettuate dai responsabili delle singole scuole medie, che potranno accedere facilmente alla piattaforma e inserire i dati richiesti.

Il sistema fornirà strumenti intuitivi e ben strutturati per configurare i dettagli delle giornate promozionali, monitorare in tempo reale l'andamento delle iscrizioni e garantire una comunicazione costante ed efficace tra tutti gli istituti coinvolti.

## 2 Analisi

### 2.1 Analisi del dominio

Attualmente, la gestione delle iscrizioni al Promtec avviene in modo manuale, tramite scambi di email tra la Scuola Arti e Mestieri e i referenti delle scuole medie, supportata da fogli Excel. Questo approccio risulta laborioso, soggetto ad errori e difficilmente sostenibile con l'aumento delle richieste. Il nuovo sistema si inserisce nel contesto delle attività promozionali rivolte agli studenti delle scuole medie, che prevedono l'organizzazione di giornate a tema presso la sede dell'istituto. Gli utenti coinvolti, ovvero organizzatori della SAMT ed i responsabili delle scuole medie, possiedono competenze digitali di base, ma necessitano di uno strumento semplice ed intuitivo, in grado di semplificare l'organizzazione e garantire una comunicazione chiara tra gli attori coinvolti.

### 2.2 Analisi e specifica dei requisiti

ID: REQ-001	
<b>Nome</b>	Login e registrazione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Un utente per poter utilizzare il sito deve effettuare il login
Sotto requisiti	
<b>001</b>	Approvazione registrazione utenti

ID: REQ-002	
<b>Nome</b>	Creazione slot di promozione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Da parte del gestore SAMT
Sotto requisiti	
<b>001</b>	Controllo date e sovrapposizioni, uno slot dura mezza giornata, non ci possono essere due slot dello stesso ambito nella stessa giornata
<b>002</b>	Slot specifici al genere
<b>003</b>	Numero partecipanti 24, ma modificabile

ID: REQ-003	
<b>Nome</b>	Visualizzazione iscrizioni agli slot di promozione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Lista con tutte le iscrizioni ad uno slot, indicante anche il nome della scuola ed in ordine di data decrescente

ID: REQ-004	
<b>Nome</b>	Modifica iscrizioni agli slot di promozione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Rimozione, modifica informazioni studenti da parte del gestore SAMT
Sotto requisiti	
<b>001</b>	Aggiungere eventuali studenti dalla lista d'attesa dei responsabili delle scuole medie
<b>002</b>	Confermare le iscrizioni

ID: REQ-005	
<b>Nome</b>	Visualizzazione slot di promozione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Da utenti: responsabili scuole medie Vede i periodi di promozione aperti

ID: REQ-006	
<b>Nome</b>	Iscrizione agli slot di promozione
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Da utenti: responsabili scuole medie  Inserisce i dati degli allievi: nome, cognome, sesso, via nap, luogo, telefono fisso, cellulare, scuole medie, classe
Sotto requisiti	
<b>001</b>	Modifica iscrizioni entro due settimane dall'evento
<b>002</b>	Iscrive massimo 10 studenti, ma il numero è parametrizzabile
<b>003</b>	Lista d'attesa per gli studenti in caso si liberino posti

ID: REQ-007	
<b>Nome</b>	Conferma via email delle iscrizioni
<b>Priorità</b>	2
<b>Versione</b>	1
<b>Note</b>	Quando un responsabile iscrive degli studenti, o l'organizzatore SAMT modifica la lista degli studenti, una email con tutti gli iscritti viene mandata ai vari responsabili delle medie coinvolti in uno slot

ID: REQ-008	
<b>Nome</b>	Esportazione iscrizioni per stampa
<b>Priorità</b>	2
<b>Versione</b>	1
<b>Note</b>	Esportare le iscrizioni creando una lettera per ogni iscritto

ID: REQ-009	
<b>Nome</b>	Gestione utenti
<b>Priorità</b>	1
<b>Versione</b>	1
<b>Note</b>	Aggiunta, visualizzazione, modifica, cancellazione
Sotto requisiti	
<b>001</b>	Se rimane solo un utente esso non si potrà cancellare
<b>002</b>	Ruolo utente (gestore/responsabile/iscritto) ed altre informazioni

ID: REQ-010	
<b>Nome</b>	Crittografia database
<b>Priorità</b>	2
<b>Versione</b>	1
<b>Note</b>	I dati del database devono essere criptati, così da MySQL Workbench o strumenti simili non si leggono, questo perché il database è popolato da informazioni sensibili di utenti minorenni

ID: REQ-011	
<b>Nome</b>	HTTPS
<b>Priorità</b>	2
<b>Versione</b>	1
<b>Note</b>	Connessione sicura

### 2.2.1 Spiegazione elementi tabella dei requisiti:

**ID:** identificativo univoco del requisito

**Nome:** breve descrizione del requisito

**Priorità:** indica l'importanza di un requisito nell'insieme del progetto, definita assieme al committente. Ad esempio, poter disporre di report con colonne di colori diversi ha priorità minore rispetto al fatto di avere un database con gli elementi al suo interno. Solitamente si definiscono al massimo di 2-3 livelli di priorità.

**Versione:** indica la versione del requisito. Ogni modifica del requisito avrà una versione aggiornata.

Sulla documentazione apparirà solamente l'ultima versione, mentre le vecchie dovranno essere inserite nei diari.

**Note:** eventuali osservazioni importanti o riferimenti ad altri requisiti.

**Sotto requisiti:** elementi che compongono il requisito.



## 2.3 Use case

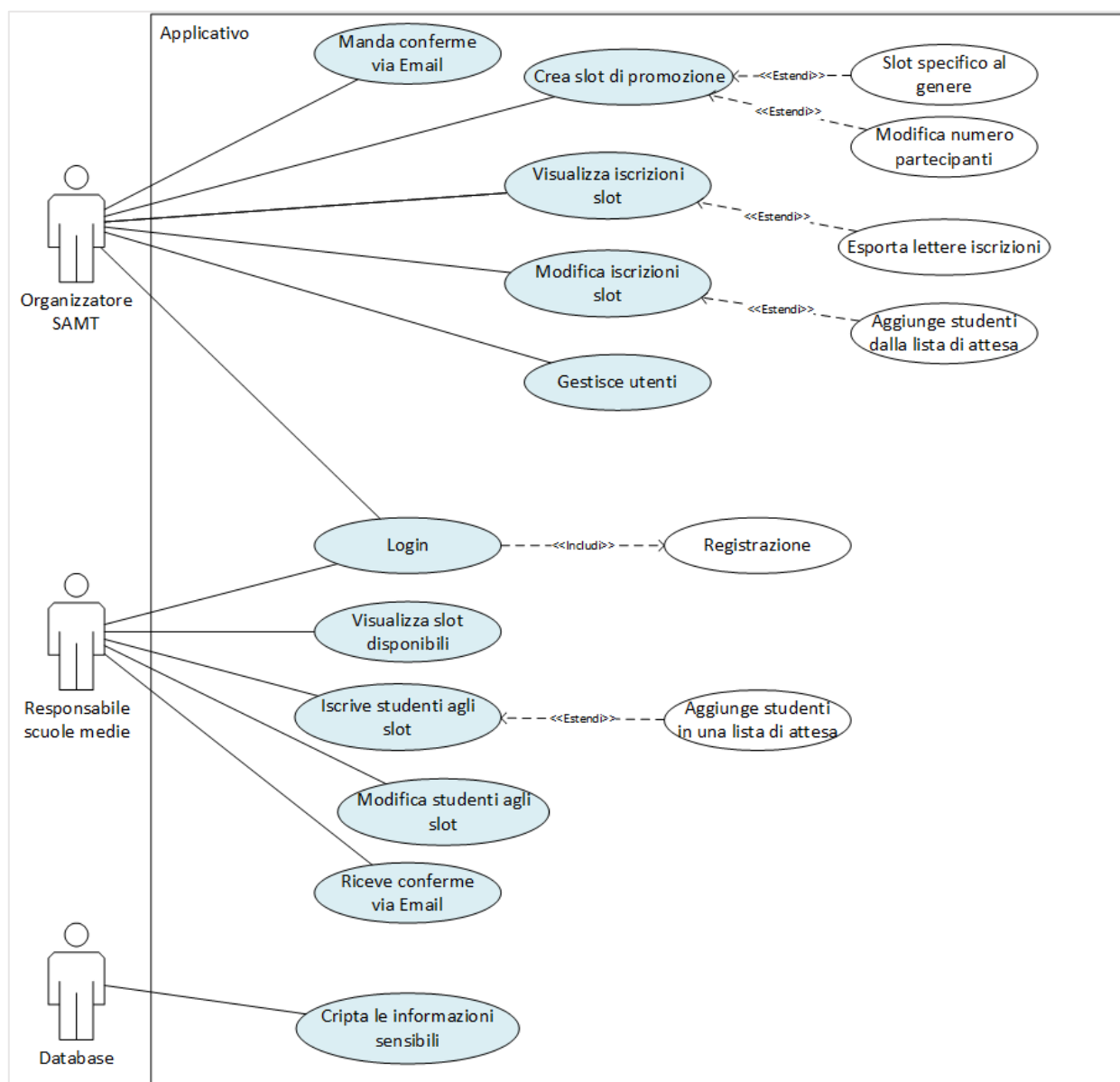


Figura 1 - Use case

Come si può osservare da questo schema, il sistema prevede due principali tipologie di utenti: l'Organizzatore SAMT e il Responsabile delle scuole medie, ciascuno con ruoli e funzionalità specifiche. L'Organizzatore dopo essersi registrato e autenticato tramite login, ha la possibilità di creare e gestire slot di promozione, anche specifici per genere, modificarne i parametri (come il numero massimo di partecipanti), visualizzare le iscrizioni, gestire gli utenti del sistema e inviare conferme via email. Può inoltre esportare lettere di iscrizione e aggiungere studenti dalla lista d'attesa allo slot.

Il Responsabile delle scuole medie, dopo essersi registrato e autenticato tramite login, può visualizzare gli slot disponibili, iscrivere o modificare l'assegnazione degli studenti agli slot, aggiungerli a una lista d'attesa se necessario, e ricevere conferme automatiche via email.

Infine, il Database ha il compito di criptare le informazioni sensibili, garantendo la protezione dei dati trattati dal sistema.

## 2.4 Pianificazione

Per quanto riguarda la pianificazione ho deciso di lavoro con il modello Waterfall perciò ho fatto un Gantt con fasi distinte analisi, progettazione, implementazione, test e poi consegna. Nonostante ciò ho deciso di utilizzare anche una Kanban board (Trello) per mettere più specificamente le attività da effettuare, inoltre lo uso anche per dare un'idea al mio docente responsabile di cosa sto facendo e siccome si possono avere delle problematiche all'interno del board posso mettere i commenti e il docente può darmi consigli.

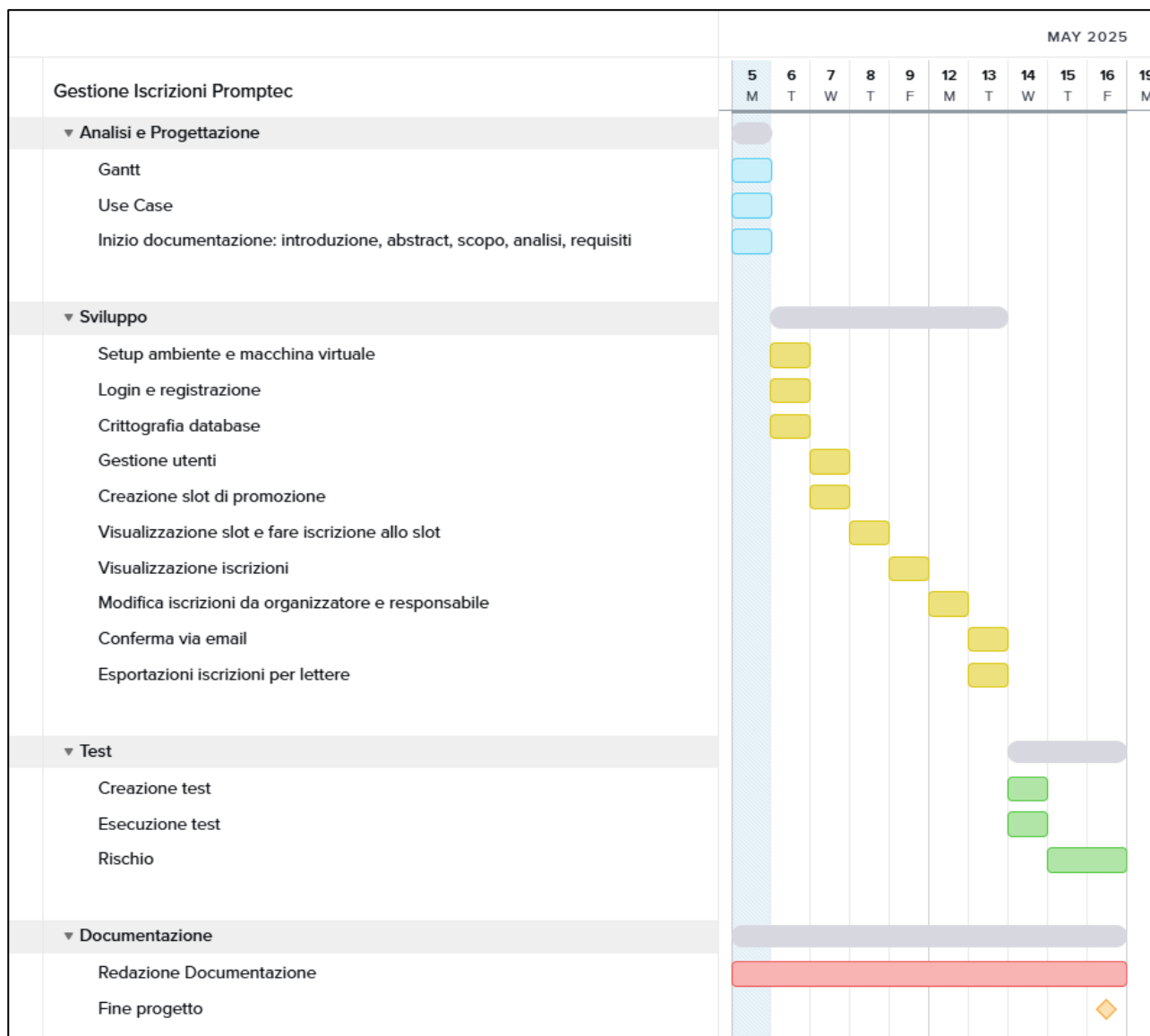


Figura 2 - Gantt

All'interno del Gantt è visibile l'utilizzo del modello waterfall. Il progetto inizia con la fase di analisi, seguita dalla progettazione, una consistente fase di implementazione e, infine, i test, senza lasciare margini di rischio nelle fasi conclusive.

Durante tutto il periodo del progetto, continuerò a documentare le attività svolte tramite un diario quotidiano, in cui indicherò cosa ho fatto durante la giornata, lo stato di avanzamento rispetto al Gantt, eventuali problematiche riscontrate e le soluzioni adottate. Infine, specificherò le attività previste per la giornata lavorativa successiva.

## **2.5 Analisi dei mezzi**

Per questo progetto si ha a disposizione un PC scolastico, su cui viene eseguita una macchina virtuale Linux Mint, su di essa girano dei container Docker.

La scelta di Linux Mint come sistema operativo della macchina virtuale deriva dalla sua stabilità, leggerezza e facilità d'uso.

### **2.5.1 Software**

- Cursor Editor 0.44.11, come editor di testo
- Mozilla Firefox 135.0.1
- Gitlab, come Repository online
- Microsoft Visio, per fare lo Use Case
- Microsoft Teams, per le comunicazioni
- Microsoft Word, per redigere la documentazione, il QdC e il diario
- Trello.com, per la gestione dei task
- Teamgantt.com, per la gestione dei task

### **2.5.2 Hardware**

Computer utilizzato come host delle macchine virtuali:

- Processore: 13th Gen Intel(R) Core(TM) i7-13700
- RAM: 32GB
- Scheda Video: NVIDIA T400 4GB
- SSD: 512GB
- 2x Display 1920x1080 60H

### 3 Progettazione

Questo capitolo descrive esaurientemente come deve essere realizzato il prodotto fin nei suoi dettagli. Una buona progettazione permette all'esecutore di evitare fraintendimenti e imprecisioni nell'implementazione del prodotto.

#### 3.1 Design dell'architettura del sistema

Gli utenti accedono al sito web tramite un browser, che stabilisce una connessione sicura tramite HTTPS verso un server reverse proxy Nginx. Questo server si occupa di terminare la connessione SSL (porta 443) e di inoltrare le richieste in HTTP verso i servizi interni, in base al tipo di richiesta.

Il frontend dell'applicazione, costituito da un server Nginx configurato per servire i file statici generati da Vue.js, è raggiungibile sulla porta 80. Il frontend sviluppato in Vue.js utilizza Typescript al posto di Javascript. Il reverse proxy reindirizza tutte le richieste generiche verso questo servizio.

Le richieste indirizzate al backend vengono invece inoltrate dal reverse proxy al servizio Flask (Python) in esecuzione sulla porta 5000. Questo backend gestisce la logica applicativa e comunica con un database MySQL, accessibile internamente sulla porta 3306.

Tutte le comunicazioni tra browser e server avvengono tramite HTTPS, mentre i collegamenti interni tra i container (reverse proxy, frontend, backend e database) utilizzano il protocollo HTTP, in quanto operano all'interno della stessa rete Docker.

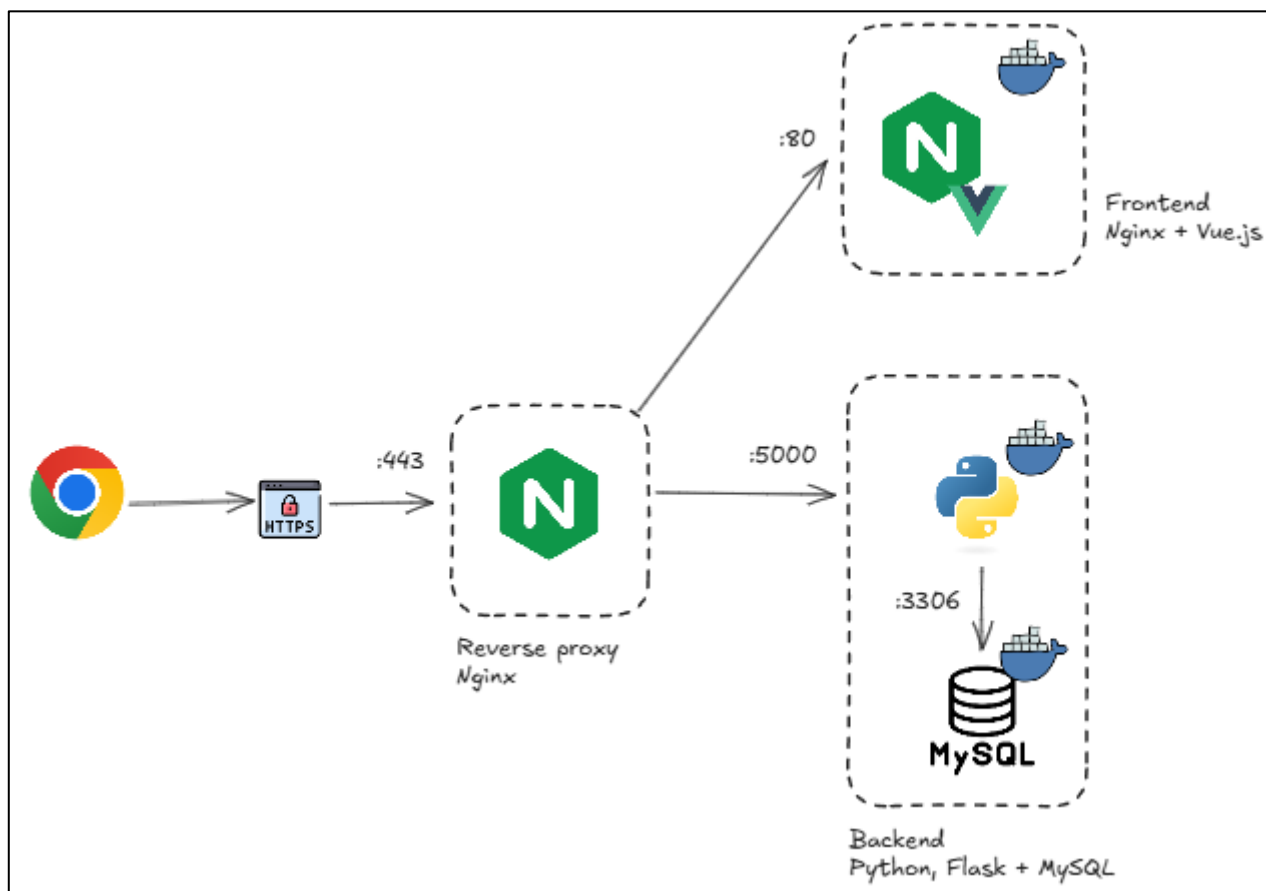


Figura 3 - Architettura

### 3.2 Design dei dati e database

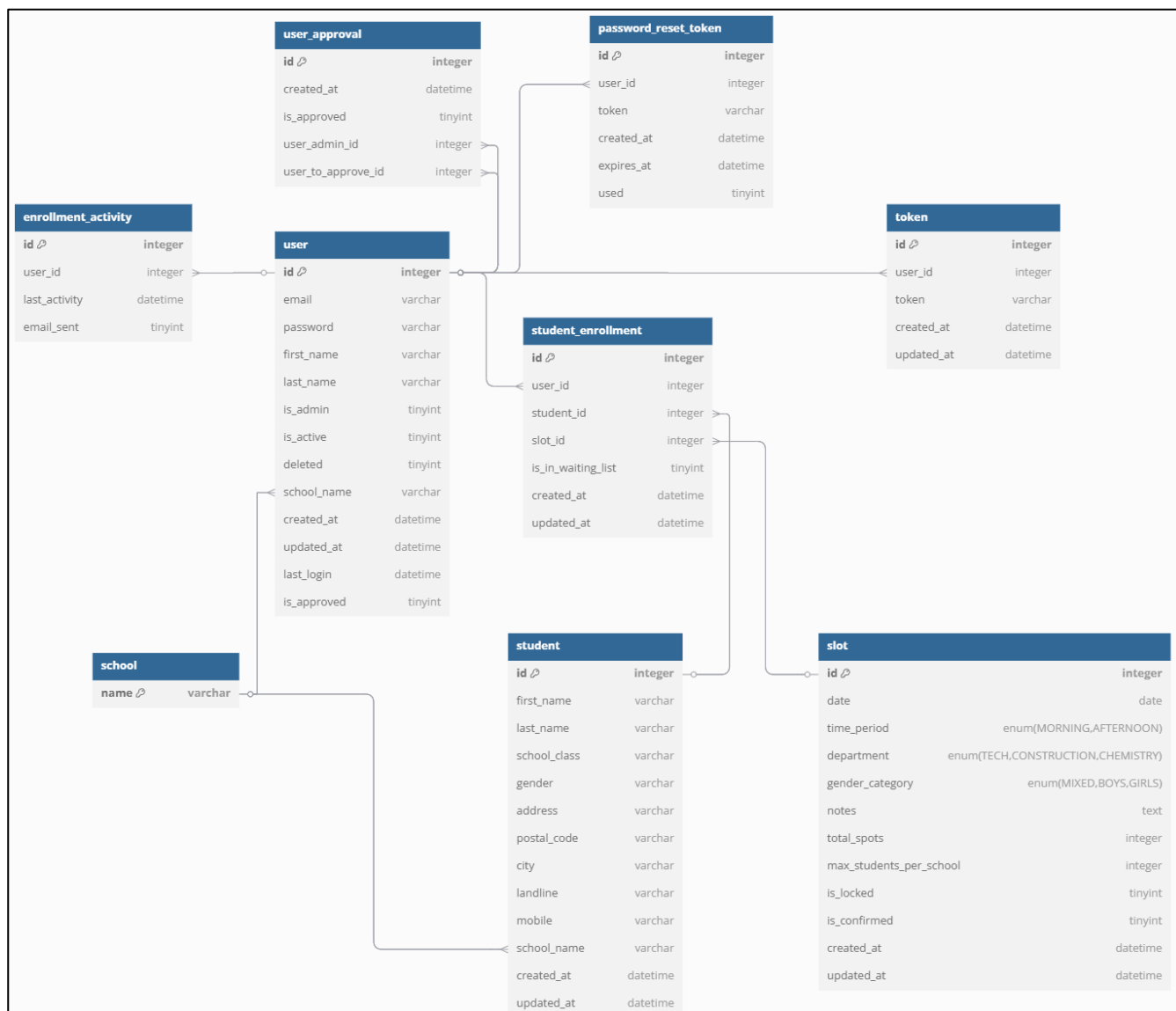


Figura 4 - Schema ER

Nello schema E/R possiamo osservare la struttura dati dell'applicativo.

Al centro troviamo l'entità utente, che può essere un responsabile delle scuole medie od un organizzatore SAMT (amministratore). Gli amministratori hanno la possibilità di approvare altri utenti, tramite la relazione `user_approval`, che tiene traccia anche di chi ha approvato chi e della data di approvazione.

Ogni utente è associato a una scuola e può creare slot, ovvero fasce orarie in cui gli studenti possono essere iscritti. Ogni slot è legato a un dipartimento, come disegno, chimica o tecnologie innovative, e definisce parametri come la data, il numero massimo di iscrizioni, il genere ammesso, e se lo slot è bloccato o confermato.

Gli studenti vengono iscritti agli slot tramite la relazione `student_enrollment`, che tiene traccia anche del responsabile (utente) che ha effettuato l'iscrizione e se lo studente è in lista d'attesa. Lo schema mostra quindi in modo chiaro tutte le entità fondamentali del sistema (utenti, scuole, studenti, slot) e le loro relazioni,

necessarie per organizzare al meglio le iscrizioni.

Oltre a queste entità principali, il sistema include anche alcune tabelle di supporto e sicurezza:

- **token**: usata per gestire token associati agli utenti, utile per sessioni o verifiche di sicurezza.
- **password\_reset\_token**: gestisce i token di recupero password, con scadenze e stato di utilizzo.
- **enrollment\_activity**: registra l'attività di iscrizione di un utente, la data dell'ultima attività e se è stata inviata un'email.

A supportare il database ci sono delle **classi Enum** presenti nei model, utilizzate per garantire coerenza e integrità dei dati nei campi che prevedono un insieme limitato e definito di valori.

Nel DB	Valore associato
MORNING	Mattina
AFTERNOON	Pomeriggio

**slot.time\_period**: enum che definisce il periodo della giornata in cui si svolge lo slot, *mattina* o *pomeriggio*.

Nel DB	Valore associato
TECH	Settore Tecnologie Innovative
CONSTRUCTION	Settore Costruzioni
CHEMISTRY	Settore Chimica

**slot.department**: enum che identifica la sezione SAMT della giornata, tecnologie innovative, costruzioni o chimica.

Nel DB	Valore associato
MIXED	Maschio
BOYS	Solo ragazzi
GIRLS	Solo ragazze

**slot.gender\_category**: enum che specifica se lo slot è aperto a *maschi*, *femmine*, o *entrambi*.

Nel DB	Valore associato
BOY	Maschio
GIRL	Femmina

**student.gender**: enum che specifica il sesso dello studente, maschio o femmina.

Queste classi Enum sono fondamentali per standardizzare i valori inseriti nel sistema, evitando errori e facilitando operazioni come filtri, ordinamenti e validazioni nei form di inserimento.

Le strategie di accesso al database sono descritte nel capitolo Docker.

Per l'accesso al database dall'applicativo viene creato un utente dedicato, con permessi specifici, tutto descritto nel capitolo Docker.

Il tipo di notazione usato nell'ER è "Crow's Foot".

### 3.2.1 Descrizione campi tabelle

In seguito le descrizioni di ogni tabella del database, insieme alle descrizioni dei campi.

#### 3.2.1.1 Tabella enrollment\_activity

Nome	Tipo	Descrizione
id	int	Identificativo univoco dell'attività
user_id	int	ID dell'utente associato
last_activity	datetime	Ultima attività registrata
email_sent	tinyint(1)	Indica se è stata inviata un'email (0 o 1)

#### 3.2.1.2 Tabella password\_reset\_token

Nome	Tipo	Descrizione
id	int	Identificativo univoco del token
user_id	int	ID dell'utente a cui è associato il token
token	varchar(100)	Token univoco per il reset della password
created_at	datetime	Data e ora di creazione del token
expires_at	datetime	Data e ora di scadenza del token
used	tinyint(1)	Flag per indicare se il token è stato utilizzato

#### 3.2.1.3 Tabella school

Nome	Tipo	Descrizione
name	varchar(50)	Nome univoco della scuola

#### 3.2.1.4 Tabella slot

Nome	Tipo	Descrizione
id	int	Identificativo univoco dello slot
date	date	Data dello slot
time_period	enum	Fascia oraria dello slot
department	enum	Dipartimento associato
gender_category	enum	Categoria di genere accettata
notes	text	Note aggiuntive
total_spots	int	Posti totali disponibili
max_students_per_school	int	Numero massimo di studenti per scuola

is_locked	tinyint(1)	Slot bloccato (1) o modificabile (0/null)
is_confirmed	tinyint(1)	Slot confermato (1) o no (0/null)
created_at	datetime	Data di creazione
updated_at	datetime	Data dell'ultima modifica

### 3.2.1.5 Tabella student

Nome	Tipo	Descrizione
id	int	Identificativo univoco dello studente
first_name	varchar(255)	Nome
last_name	varchar(255)	Cognome
school_class	varchar(255)	Classe frequentata
gender	varchar(255)	Genere
address	varchar(255)	Indirizzo
postal_code	varchar(255)	CAP
city	varchar(255)	Città
landline	varchar(255)	Numero fisso (opzionale)
mobile	varchar(255)	Cellulare
school_name	varchar(50)	Scuola di provenienza
created_at	datetime	Data di creazione
updated_at	datetime	Data dell'ultima modifica

### 3.2.1.6 Tabella student\_enrollment

Nome	Tipo	Descrizione
id	int	Identificativo univoco dell'iscrizione
user_id	int	ID dell'utente che ha effettuato l'iscrizione
student_id	int	ID dello studente
slot_id	int	ID dello slot selezionato
is_in_waiting_list	tinyint(1)	Lo studente è in lista d'attesa (1) o iscritto (0)
created_at	datetime	Data di creazione
updated_at	datetime	Data dell'ultima modifica



### 3.2.1.7 Tabella token

Nome	Tipo	Descrizione
id	int	Identificativo del token
user_id	int	ID dell'utente associato
token	varchar(255)	Token
created_at	datetime	Data di creazione
updated_at	datetime	Data dell'ultima modifica

### 3.2.1.8 Tabella user

Nome	Tipo	Descrizione
id	int	Identificativo dell'utente
email	varchar(150)	Email dell'utente (univoca)
password	varchar(200)	Password crittografata
first_name	varchar(50)	Nome
last_name	varchar(50)	Cognome
is_admin	tinyint(1)	L'utente è un amministratore (1) o no (0)
is_active	tinyint(1)	Account attivo (1) o disattivato (0)
deleted	tinyint(1)	L'utente è stato eliminato (1)
school_name	varchar(50)	Scuola associata (può essere NULL se admin)
created_at	datetime	Data di creazione
updated_at	datetime	Data dell'ultima modifica
last_login	datetime	Ultimo accesso
is_approved	tinyint(1)	L'account è stato approvato (1) o no (0)

### 3.2.1.9 Tabella user\_approval

Nome	Tipo	Descrizione
id	int	Identificativo univoco dell'approvazione
created_at	datetime	Data di creazione
is_approved	tinyint(1)	L'utente è stato approvato (1) o rifiutato (0)
user_admin_id	int	ID dell'amministratore che ha effettuato l'approvazione
user_to_approve_id	int	ID dell'utente da approvare

La tabella user\_approves serve a tenere internamente uno storico delle approvazioni.

### 3.3 Design delle interfacce

All'interno di questo capitolo mostrerò i mockup del progetto, quest'ultimi sono stati fatti con excalidraw.com, un sito per creare schemi e/o mockup.

#### 3.3.1 Pagina di login

The mockup shows a login interface within a rounded rectangle. Inside, there is a smaller rounded rectangle containing three input fields: 'Username', 'Password', and 'Login'. Below the 'Login' button is a link labeled 'Registrati'.

Figura 5 - Design login

Il mockup indica la pagina di login dell'applicativo, molto semplice, nell'applicativo è presente anche un link per resettare la password.

#### 3.3.2 Pagina gestione slot

The mockup shows a management interface. On the left is a sidebar with the text 'Gestione Promtec', 'slot' (underlined), and 'Gestione utenti'. The main area on the right contains a 'crea' button in the top right corner and three identical data cards below it. Each card contains the text: 'Data', 'Sezione SAM', 'Iscritti', 'Posti liberi', and 'ecc.'.

Figura 6 - Design gestione slot

Il mockup indica la pagina di visualizzazione degli slot da parte dell'utente admin. Sono presenti tutti gli slot, ogni slot ha i propri campi di base già visibili, si possono creare gli slot e andare a vedere i loro dettagli.

### 3.3.3 Pagina dettagli slot

Figura 7 - Design dettagli slot

Il mockup indica la pagina dei dettagli di uno slot, da parte dell'utente admin. Sono presenti le iscrizioni, le scuole medie, il numero di iscritti, la lista di attesa, la conferma delle iscrizioni e l'esportazione delle lettere.

### 3.3.4 Pagina gestione utenti

Figura 8 - Design gestione utenti

Il mockup indica la pagina di gestione utente, da parte dell'utente admin. Sono presenti due tab, una per l'approvazione degli utenti e l'altra per gli utenti approvati. Gli utenti sono filtrabili, modificabili, creabili.

### 3.3.5 Pagina iscrizione studenti

The mockup shows a web interface for 'Gestione Promtec'. At the top, there is a navigation bar with the title 'Gestione Promtec' on the left, and links for 'slots', 'contatti', and 'utente' (with a sub-link 'logout') on the right. The main content area contains four rounded rectangular buttons arranged vertically: 'Info slot', 'Form iscrizione studente', 'Studenti iscritti', and 'Studenti in lista di attesa'.

Figura 9 - Design iscrizione studenti da utente

Il mockup indica la pagina di dettagli slot, da parte di un utente normale.

Si possono vedere i campi di base dello slot, un form di iscrizione, la lista degli iscritti e la lista di attesa.

### 3.4 Design procedurale

#### 3.4.1 Ruolo utente

Is_admin	Ruolo
VERO	Organizzatore giornate SAMT
FALSO	Responsabile Scuole Medie

Il campo più significativo della tabella user è is\_admin: da questo attributo dipende il livello di accesso e le funzionalità disponibili all'interno dell'applicativo. Determinando così cosa ciascun utente può visualizzare e quali azioni può compiere nell'interfaccia

#### 3.4.2 Attivazione utente

Is_active	Azione
VERO	L'utente potrà accedere al sistema
FALSO	L'utente verrà bloccato dall'accesso

Il campo is\_active della tabella user gestisce la possibilità di accedere all'applicativo. Se l'utente admin disattiva un account, al tentativo di login l'utente vedrà un messaggio che informa della disattivazione.

#### 3.4.3 Cancellazione utente

Deleted	Azione
VERO	L'utente non sarà presente nella gestione utenti
FALSO	L'utente sarà presente nella gestione utenti

La cancellazione avviene tramite soft delete, quindi i dati dell'utente non vengono rimossi dal database ma solo nascosti all'interfaccia e resi inaccessibili all'utente.

Il campo deleted funziona in modo simile a is\_active, descritto in precedenza, ovvero sarà bloccato dall'accesso all'applicativo. La differenza principale è che un account eliminato scompare dall'interfaccia di gestione utenti dell'admin. Per riattivarlo, l'utente deve registrarsi nuovamente con la stessa email.

### 3.4.4 Approvazione utente

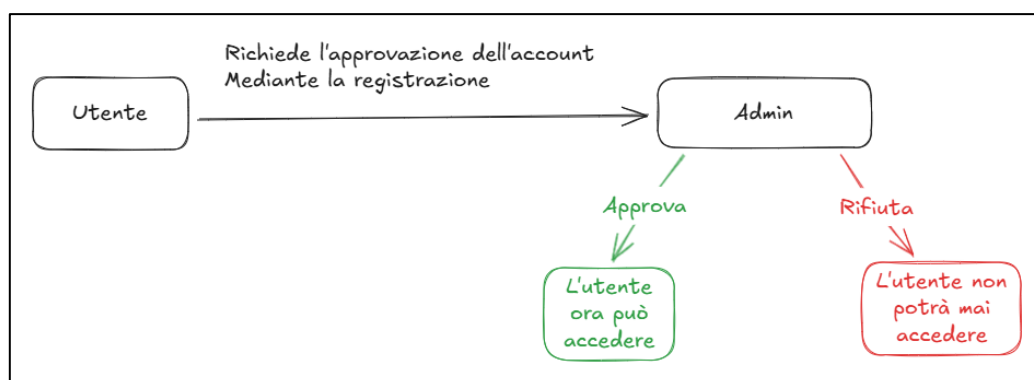


Figura 10 - Approvazione utente

L'approvazione utente viene effettuata quando un utente richiede di creare un account, subito dopo aver immesso i dati, l'utente amministratore riceve una richiesta di approvazione, dove decidere se accettare o rifiutare. Se rifiuta l'account non potrà registrarsi o accedere con la stessa email.

### 3.4.5 Azioni organizzatore SAMT (admin)

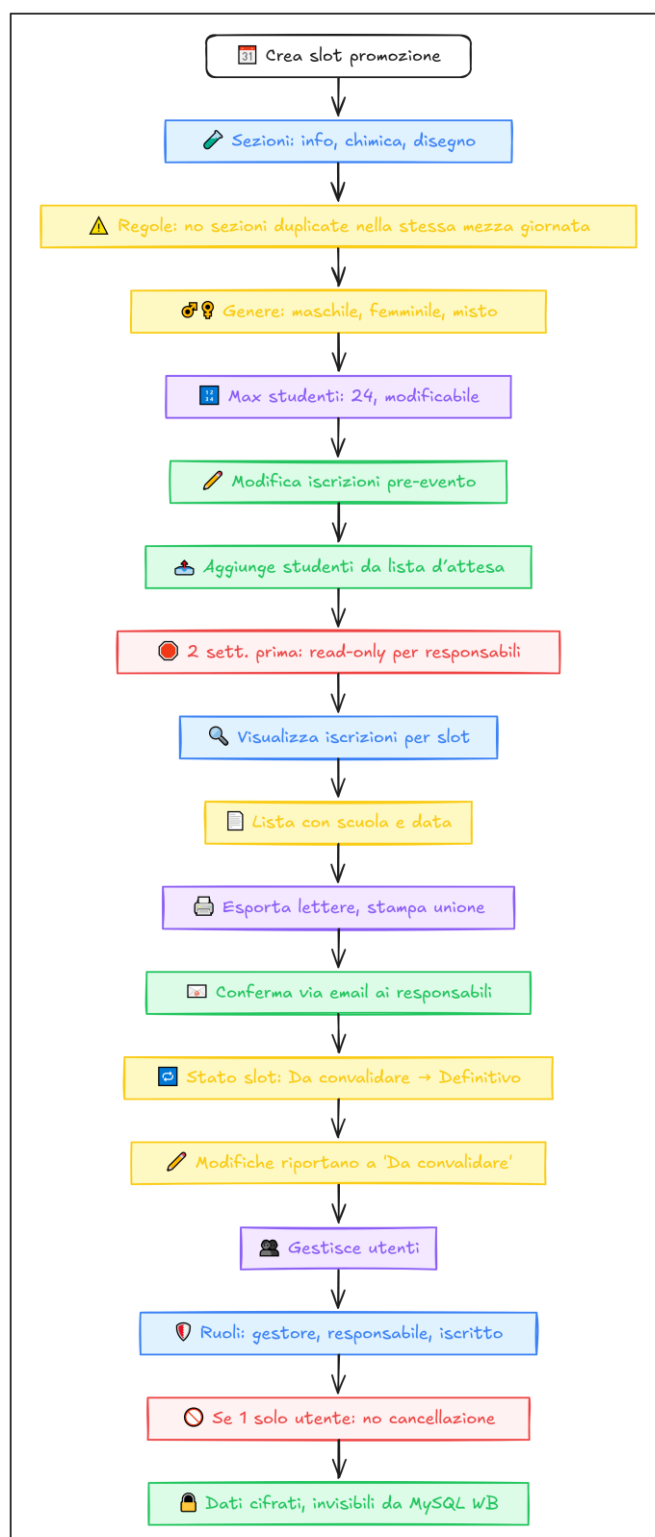


Figura 11 - Azioni admin

Questo diagramma è il riassunto schematicizzato del QDC, descrive le azioni dell'organizzatore SAMT, l'admin.

### 3.4.6 Azioni responsabile scuole medie (user)

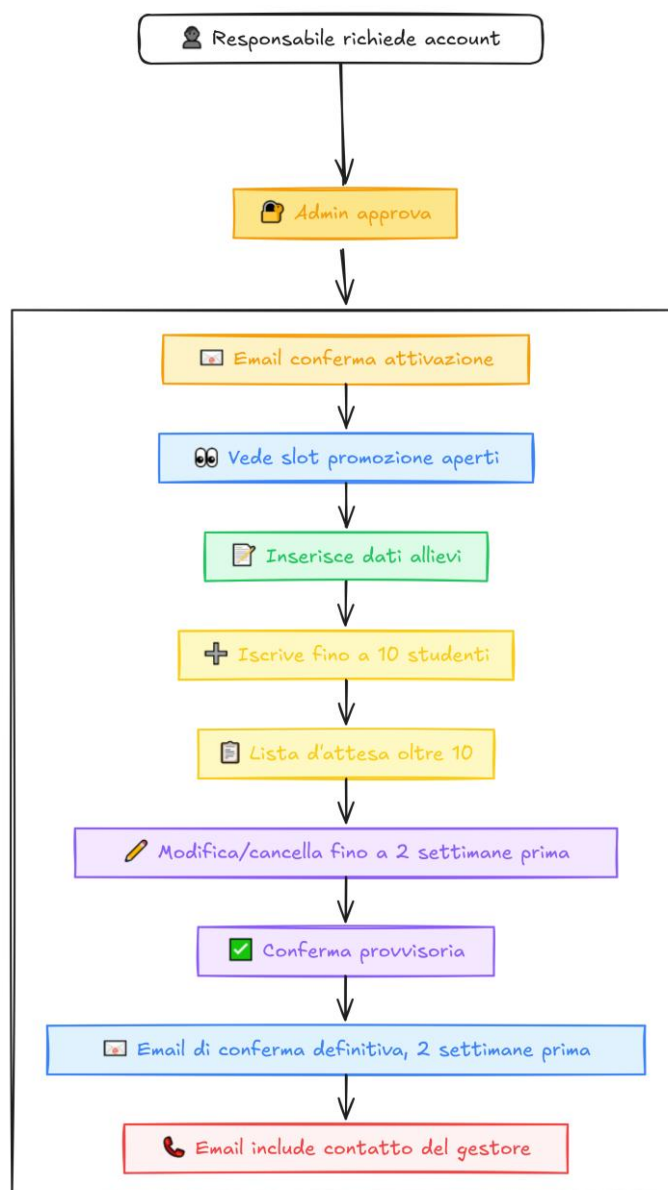


Figura 12 - Azioni utenti

Questo diagramma è il riassunto schematicizzato del QDC, descrive le azioni di un responsabile scuole medie, l'utente.



## 4 Implementazione

### 4.1 Struttura cartelle

Il progetto è strutturato su più cartelle, ognuna racchiude uno scopo.

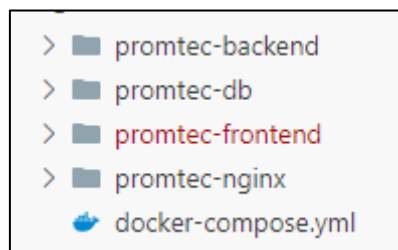


Figura 13 - Struttura cartelle

Nella root del progetto è presente il file di configurazione principale, docker-compose.yml.

Il file docker-compose.yml è utilizzato per definire e gestire i servizi del progetto tramite Docker, facilitando la creazione, l'esecuzione e la configurazione dei container necessari per l'ambiente di sviluppo e produzione.

La root contiene le seguenti directory principali:

- **promtec-backend/**: Il backend Flask dell'applicazione
- **promtec-frontend/**: Il frontend Vue.js dell'applicazione
- **promtec-db/**: Configurazione e script per il database MySQL
- **promtec-nginx/**: Configurazione del server NGINX che funge da reverse proxy

#### 4.1.1 Struttura del Backend

Il backend è sviluppato con Flask e organizzato in moduli funzionali per una migliore manutenibilità e separazione delle responsabilità.

- **/app/**: Contiene la logica core dell'applicazione.
- **/migrations/**: File di migrazione per la gestione delle modifiche al database
- **/scripts/**: Script di utilità per l'applicazione
- **/.venv/**: Ambiente virtuale Python
- **/requirements.txt**: Dipendenze Python dell'applicazione
- **/Dockerfile**: Definizione del container Docker per il backend
- **/.env**: File di configurazione delle variabili d'ambiente

Componenti principali in **/app/** :

- **/\_\_init\_\_.py**: Inizializzazione dell'applicazione Flask
- **/config.py**: Configurazione dell'applicazione
- **/extensions.py**: Estensioni Flask utilizzate nell'applicazione
- **/schools/**: Gestione delle scuole
- **/security/**: Moduli per la sicurezza e l'autenticazione degli utenti
- **/slots/**: Gestione degli slot di prenotazione
- **/user\_management/**: Gestione degli utenti
- **/utils/**: Funzioni di utilità e helper

#### 4.1.2 Struttura del Frontend

Il frontend è sviluppato con Vue.js 3 e TypeScript, organizzato in modo da separare i componenti riutilizzabili, la gestione dello stato e i servizi API.

Il codice è racchiuso in **/src/** :

- **/components/**: Componenti riutilizzabili sviluppati con Vue.js
- **/slots/**: Componenti relativi alla gestione degli slot
- **/views/**: Componenti per la gestione delle diverse pagine dell'applicazione
- **/user/**: Viste specifiche per gli utenti
- **/router/**: Definizioni delle rotte dell'applicazione
- **/stores/**: Gestione dello stato dell'applicazione con Pinia
- **/services/**: Servizi API per la comunicazione tra il frontend e il backend
- **/types/**: Definizioni dei tipi TypeScript
- **/utils/**: Funzioni di utilità
- **/assets/**: Risorse statiche come immagini, fogli di stile, e altri file non dinamici

#### 4.1.3 Configurazione del Database

- **/init.sql**: Script di inizializzazione del database che crea le tabelle e gli utenti con permessi limitati
- **/env**: Variabili d'ambiente per la configurazione del database

#### 4.1.4 Configurazione NGINX

- **/nginx.conf**: Configurazione principale di NGINX
- **/certs/**: Certificati SSL per HTTPS (nginx.key e nginx.crt)

## 4.2 Docker

Il progetto utilizza Docker Compose per orchestrare i seguenti servizi:

- **Backend** (Flask): API REST e logica applicativa
- **Frontend** (Vue.js): Interfaccia utente web costruita con Vue.js
- **Database** (MySQL): Database relazionale principale
- **Nginx**: Reverse proxy che gestisce il routing delle richieste HTTP/HTTPS e la terminazione SSL, fungendo da punto di ingresso sicuro per l'applicazione

### 4.2.1 Avvio progetto

Per avviare il progetto basta usare il seguente comando (Linux):

```
docker-compose down && docker-compose up --build -d
```

Il comando serve per aggiornare l'ambiente Docker. Prima ferma e rimuove i container e le risorse esistenti, poi ricostruisce le immagini e riavvia i servizi in background. È utile per assicurarsi che tutto sia aggiornato e pulito dopo modifiche al codice o alla configurazione.

### 4.2.2 Variabili di ambiente

Il progetto dispone di molte variabili di ambiente, ciò per garantire una configurazione flessibile e facilmente modificabile senza dover intervenire direttamente sul codice sorgente.

#### 4.2.2.1 Backend

Le variabili d'ambiente del backend sono raccolte nel file **promptec-backend/.env**:

```
DB_USER=  
DB_PASSWORD=  
DB_HOST=  
DB_PORT=  
DB_NAME=  
  
DEFAULT_ADMIN_EMAIL=  
DEFAULT_ADMIN_PASSWORD=  
DEFAULT_ADMIN_FIRST_NAME=  
DEFAULT_ADMIN_LAST_NAME=  
  
SMTP_SERVER=  
SMTP_PORT=  
SMTP_USER=  
SMTP_PASSWORD=  
  
FERNET_KEY=  
FLASK_SECRET_KEY=  
  
FRONTEND_URL=
```

Qua sono inoltre presenti le configurazioni per l'utente admin di default, che verrà creato sempre se manca nel DB, ad esempio se l'email dell'admin dovesse risultare inutilizzabile, basterebbe cambiare l'email di default nel .env, per creare un nuovo admin al riavvio del backend.

Per generare la FERNET\_KEY, essenziale per la crittografia dei dati degli studenti, bisogna eseguire:

```
python -c "from cryptography.fernet import Fernet; print(Fernet.generate_key().decode())"
```

#### 4.2.2.2 Frontend

Le variabili d'ambiente del frontend sono in **promtec-frontend/.env**:

```
VITE_BACKEND_URL=/api
```

L'unica variabile d'ambiente del frontend è VITE\_BACKEND\_URL, "/api" viene utilizzato dal reverse proxy per mandare le richieste al backend.

È importante che le variabili d'ambiente nel frontend abbiano "VITE\_" davanti al nome, perché VITE espone automaticamente solo quelle.

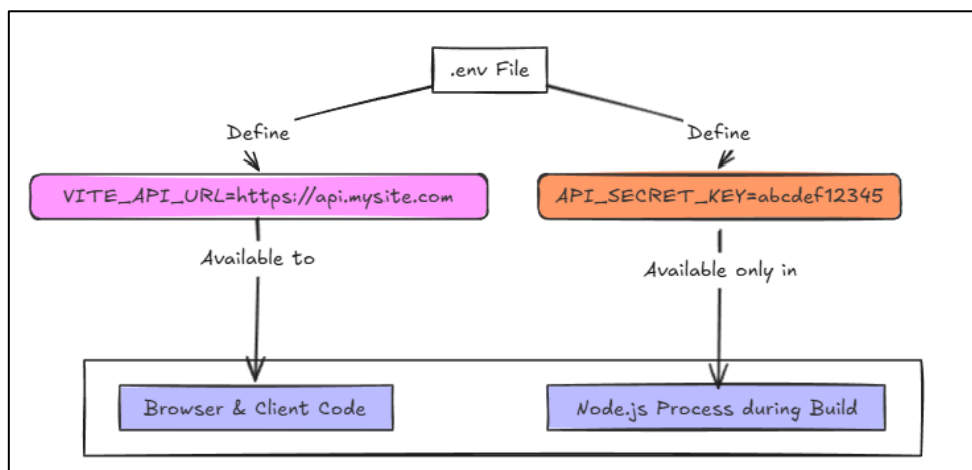


Figura 14 - Env frontend

Tutte le altre variabili d'ambiente non iniziate con "VITE\_" sono disponibili solo nel processo Node.js durante la build, ma non nel codice client.

#### 4.2.2.3 Database

Le variabili d'ambiente del database sono in **promtec-db/.env**:

```
MYSQL_ROOT_PASSWORD=
```

```
MYSQL_DATABASE=
```

```
MYSQL_USER=
```

```
MYSQL_PASSWORD=
```

```
# Additional Configuration
```

```
MYSQL_ALLOW_EMPTY_PASSWORD=no
```

```
MYSQL_RANDOM_ROOT_PASSWORD=no
```

Il database, l'utente e la password devono essere uguali al .env del backend.

È importante non avere la stessa password per l'utente root e l'utente MySQL di applicativo, importante anche che l'utente di applicativo MySQL non abbia la stessa password dell'admin di default dell'applicativo.

Viene disabilitata la generazione di password casuali per l'utente root, viene ad impedire anche che MySQL venga avviato con una password vuota per l'utente root.

#### 4.2.3 Container database

```
db:
  build:
    context: ./promtec-db
    dockerfile: Dockerfile
  ports:
    - "3307:3306"
  env_file:
    - ./promtec-db/.env
  volumes:
    - mysql_data:/var/lib/mysql
  networks:
    - api_bridge
  healthcheck:
    test: ["CMD", "mysqladmin", "ping", "-h", "localhost"]
    interval: 10s
    timeout: 5s
    retries: 5
    start_period: 15s
```

Il database viene costruito dalla directory ./promtec-db con il proprio Dockerfile. È esposto sulla porta 3307 e configurato tramite variabili d'ambiente definite nel file .env. I dati sono salvati in un volume persistente e il servizio è connesso alla rete api\_bridge. Un healthcheck verifica che il database sia pronto prima dell'avvio di altri servizi.

#### 4.2.4 Container backend

```
backend:
  build:
    context: ./promtec-backend
    dockerfile: Dockerfile
  ports:
    - 5001:5000 # Espone la porta 5000 del container sulla 5001 dell'host
  env_file:
    - ./promtec-backend/.env # Variabili d'ambiente per la configurazione
  depends_on:
    db:
      condition: service_healthy # Avvia solo quando il database è pronto
  networks:
    - api_bridge # Rete condivisa tra i servizi
  restart: always # Riavvio automatico in caso di errori
```

Il backend viene configurato per essere costruito dalla directory ./promtec-backend, utilizzando il proprio Dockerfile. Viene avviato solo quando il database è pronto (healthcheck) e configurato tramite variabili d'ambiente contenute nel file .env che sarà descritto nei prossimi capitoli.

#### 4.2.5 Container frontend

```
frontend:
  build:
    context: ./promtec-frontend
    dockerfile: Dockerfile
  ports:
    - 3007:80 # Accesso diretto al frontend sulla porta 3007
  env_file:
    - ./promtec-frontend/.env # Configurazione frontend
  networks:
    - api_bridge
```

Il frontend viene costruito dalla directory ./promtec-frontend, con l'applicazione Vue.js compilata per la produzione e servita tramite un server web interno. È accessibile direttamente sulla porta 3007, ma l'accesso principale avviene tramite NGINX sulla porta 80/443.

#### 4.2.6 Container Nginx

```
nginx:
  build:
    context: ./promtec-nginx
    dockerfile: Dockerfile
  restart: always
  ports:
    - 80:80 # HTTP (con redirectione a HTTPS)
    - 443:443 # HTTPS
  depends_on:
    - backend
    - frontend
  networks:
    - api_bridge
```

Nginx viene configurato con due server block:

1. HTTP (porta 80): Reindirizza automaticamente tutto il traffico verso HTTPS
2. HTTPS (porta 443): Gestisce le connessioni sicure con certificati SSL

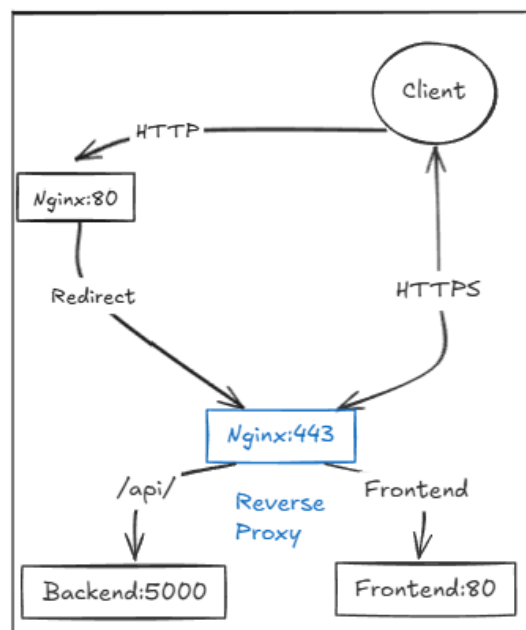


Figura 15 - Connessione tra container

Il diagramma mostra come un client interagisce con un server Nginx configurato per gestire sia traffico HTTP che HTTPS. Le richieste HTTP arrivate alla porta 80 di Nginx vengono automaticamente reindirizzate alla porta 443, dove Nginx gestisce le connessioni HTTPS sicure. Da qui, tramite un meccanismo di reverse proxy, Nginx inoltra le richieste con percorso /api/ al servizio backend sulla porta 5000, mentre tutte le altre richieste vengono inviate al servizio frontend sulla porta 80. Questo flusso garantisce sicurezza tramite HTTPS e una corretta separazione tra le chiamate API e l'interfaccia utente.

```
# Server HTTPS
server {
    # Attiva l'ascolto sulla porta 443 con SSL abilitato
    listen 443 ssl;

    # Accetta richieste per qualsiasi nome host
    server_name _;

    # Certificati SSL - Percorsi dei file di certificato autofirmati
    ssl_certificate /etc/nginx/certs/nginx.crt;      # Certificato pubblico
    ssl_certificate_key /etc/nginx/certs/nginx.key;  # Chiave privata

    # Sicurezza SSL
    ssl_protocols TLSv1.2 TLSv1.3;                  # Solo protocolli TLS moderni e sicuri
    ssl_prefer_server_ciphers on;                    # Preferisci cifrari del server (più sicuri)
    ssl_ciphers ECDHE-RSA-AES256-GCM-SHA512:DHE-RSA-AES256-GCM-SHA512; # Cifrari robusti

    # --- Reverse Proxy ---
    # Inoltra le richieste al servizio backend o frontend a seconda del percorso

    # Endpoint API del backend - Gestisce tutte le richieste API
    location /api/ {

        # Inoltra richieste al servizio backend sulla porta 5000
        proxy_pass http://backend:5000/api/;

        # Preserva informazioni sull'host originale
        proxy_set_header Host $host;

        # Passa l'indirizzo IP reale del client al backend per logging e sicurezza
        proxy_set_header X-Real-IP $remote_addr;
    }

    # Servizio frontend - Gestisce tutte le altre richieste (interfaccia utente)
    location / {
        # Inoltra al servizio frontend
        proxy_pass http://frontend:80;

        # Preserva informazioni sull'host originale
        proxy_set_header Host $host;

        # Passa l'indirizzo IP reale del client
        proxy_set_header X-Real-IP $remote_addr;
    }
}
```

Questa configurazione gestisce tre aspetti principali:

- **HTTPS:** Configurazione SSL sicura sulla porta 443
- **API Backend:** Reverse proxy per le chiamate API su /api/
- **Frontend:** Instradamento dell'interfaccia utente Vue.js



#### 4.2.7 Rete container

```
networks:
  api_bridge:
    driver: bridge
```

Tutti i servizi sono collegati alla stessa rete api\_bridge, garantendo che possano comunicare tra loro utilizzando i nomi dei servizi come hostname (es. backend, db).

#### 4.2.8 Volumi container

```
volumes:
  mysql_data: # Volume per la persistenza dei dati MySQL
```

Il volume mysql\_data garantisce che tutti i dati del database siano conservati anche in caso di aggiornamento, riavvio o ricostruzione dei container.

#### 4.2.9 Certificati SSL

La sicurezza è garantita da certificati SSL autofirmati gestiti nel container Nginx.

Implementazione:

- Certificati generati nella directory promtec-nginx/certs (nginx.key e nginx.crt)
- Montati nel container Nginx in /etc/nginx/certs/

I certificati utilizzati vengono generati tramite lo script generate-certs.sh nella cartella promtec-nginx.

Per usare lo script, generando nuovi certificati, bisogna scrivere nel terminale (Linux):

```
chmod +x generate-certs.sh
./generate-certs.sh
```

#### 4.2.10 Script di inizializzazione MySQL

Lo script init.sh viene eseguito automaticamente al primo avvio del container MySQL e configura i permessi di sicurezza per l'utente dell'applicazione, impedendo azioni pericolose come il drop del database.

```
mysql -u root -p"${MYSQL_ROOT_PASSWORD}" <<EOF
-- Grant specific permissions to the application user (everything except dangerous
operations)
GRANT SELECT, INSERT, UPDATE, DELETE, CREATE, INDEX, ALTER, CREATE TEMPORARY TABLES, LOCK
TABLES, EXECUTE, CREATE VIEW, SHOW VIEW, EVENT, TRIGGER ON \`${MYSQL_DATABASE}\`.* TO
'${MYSQL_USER}'@'%' IDENTIFIED BY '${MYSQL_PASSWORD}';

-- Apply changes
FLUSH PRIVILEGES;
EOF
```

I valori del nome del database e utente vengono dati dal file promtec-db/.env.

#### 4.2.10.1 Permessi non assegnati

Permesso MySQL	Descrizione Estesa
DROP	<b>Consente di eliminare oggetti</b> nel database: tabelle, viste, procedure, trigger, database interi. Revocandolo, si impedisce all'utente di <b>cancellare accidentalmente o intenzionalmente dati o strutture importanti</b> .
SHUTDOWN	<b>Permette di spegnere il server MySQL</b> . Solo gli amministratori di sistema dovrebbero avere questo privilegio. Senza questo permesso, l'utente non può <b>terminare il servizio MySQL</b> , evitando interruzioni del servizio.
RELOAD	Permette operazioni come FLUSH PRIVILEGES, FLUSH TABLES, FLUSH LOGS, cioè operazioni di <b>reset o ricaricamento della configurazione</b> interna del server. Revocarlo impedisce all'utente di <b>manipolare lo stato del server</b> senza controllo.
SUPER	Questo è un <b>permesso potentissimo</b> . Permette: Interrompere query di altri utenti (KILL) Modificare variabili globali con SET GLOBAL Ignorare limiti di connessione Forzare la replica Revocarlo garantisce che l'utente non possa <b>agire come amministratore del server</b> .
FILE	Permette all'utente di <b>leggere o scrivere file sul file system del server</b> , ad esempio tramite LOAD DATA INFILE o SELECT ... INTO OUTFILE. Revocandolo, si previene il <b>rischio di accesso non autorizzato al sistema operativo</b> .
PROCESS	Permette di vedere tutte le query attualmente in esecuzione sul server con SHOW PROCESSLIST. Revocandolo, l'utente non può <b>spiare o monitorare le query di altri utenti</b> , proteggendo la privacy e la sicurezza.
REPLICATION CLIENT	Permette l'accesso a informazioni sullo stato della replica (comandi come SHOW MASTER STATUS, SHOW SLAVE STATUS). È usato da tool che <b>monitorano la replica MySQL</b> , ma non è necessario per un normale utente applicativo.
REPLICATION SLAVE	Permette di <b>configurare il server come replica</b> (slave), eseguendo comandi come START SLAVE o STOP SLAVE. Revocarlo impedisce che un utente possa <b>alterare la topologia della replica</b> o connettere il server a un master non autorizzato.

### 4.3 Ambiente virtuale per lo sviluppo

Per creare l'ambiente virtuale ho dovuto eseguire il seguente comando (da Linux):

```
python3 -m venv .venv
```

Un ambiente virtuale è uno spazio isolato all'interno del sistema che permette di installare pacchetti e dipendenze di Python separatamente da quelli globali. Questo consente di lavorare su progetti diversi senza creare conflitti tra le versioni delle librerie utilizzate e il resto del sistema operativo.

Per poi entrarci ed installare i pacchetti PIP bisogna fare (da Linux):

```
source .venv/scripts/activate
```

Che attiva l'ambiente virtuale, i nomi dei pacchetti di cui il progetto necessita sono in requirements.txt, che include anche la versione, si installano con:

```
pip install -r requirements.txt
```

Gli ultimi due comandi sono utili solo in fase di sviluppo, il Dockerfile del backend lo fa in automatico.

#### 4.4 Informazioni organizzatore

Il progetto manda le informazioni dell'organizzatore varie volte, nelle email, nelle lettere esportate e nella pagina di contatti.

Per modificarle bisogna modificare questa classe in promtec-backend/slots/models.py

```
class OrganizationInfo(str, Enum):
    FIRST_NAME = "Mario"
    LAST_NAME = "Rossi"
    TELEPHONE = "+41 00 000 00 00"
    EMAIL = "decs-cpt.trevano.promtec@edu.ti.ch"
```

Inoltre bisogna anche modificare promtec-backend/utls/email\_utils.py

```
# Not using the one in slots/models.py to avoid circular imports
ORG_FIRST_NAME = "Mario"
ORG_LAST_NAME = "Rossi"
ORG_TELEPHONE = "+41 00 000 00 00"
ORG_EMAIL = "decs-cpt.trevano.promtec@edu.ti.ch"
```

Bisogna farlo in due posti per evitare importazioni circolari.

Una volta fatto le modifiche, le si vedranno subito, ad esempio nella pagina di contatto dell'utente o nelle email.

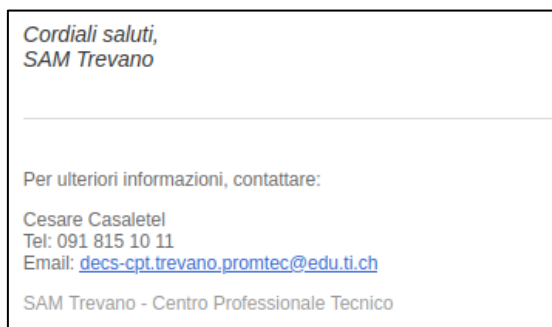


Figura 16 - Informazioni di contatto nella email

#### 4.5 Scuole medie disponibili

Attualmente le scuole medie vengono aggiunte all'inizio dell'applicativo, per modificare le scuole bisogna modificare il file **promtec-backend/app/schools/defaults.py**. Dentro al file è contenente una lista di scuole e una funzione per aggiungerle al database.

```
INITIAL_SCHOOLS = [
    "Scuola media di Acquarossa",
    "Scuola media di Agno",
    "Scuola media di Ambri",
```

#### 4.6 Registrazione e accesso

All'interno di questo capitolo mostrerò come funziona l'accesso e la registrazione all'applicativo. Per accedere si usa email e password, registrandosi si chiede nome, cognome, email, scuola media e password. Una volta registrato l'utente dovrà aspettare l'approvazione dell'account da parte dell'admin. Una volta approvato, riceverà una email di conferma.

Figura 17 - Registrazione utente

Nell'immagine si può vedere l'interfaccia di registrazione di un responsabile per le scuole medie, inoltre c'è anche un controllo della lunghezza e del contenuto della password.

#### 4.6.1 Modello User

Il modello User gestisce i dati degli utenti con attributi per email, password, informazioni personali e stati dell'account. Oltre ai campi base come id, email e password, il modello traccia anche lo stato di attivazione (is\_active), l'appartenenza a una scuola (school\_name) e il ruolo amministrativo (is\_admin).

Il modello implementa una property approval\_status che determina se l'utente è stato approvato:

```
@property
def approval_status(self):
    if self.is_admin:
        return True
    latest_approval = UserApproval.query.filter_by(
        user_to_approve_id=self.id
    ).order_by(UserApproval.created_at.desc()).first()

    is_approved = latest_approval.is_approved if latest_approval else False
```

Questa property verifica automaticamente se l'utente è un amministratore (approvato di default) o cerca l'ultima approvazione nel database. Il metodo aggiorna anche il campo is\_approved se è diverso dal valore calcolato, mantenendo la coerenza dello stato dell'utente.

#### 4.6.2 Autenticazione e token

L'autenticazione utilizza token generati con secrets.token\_hex. Durante il login, il sistema verifica le credenziali dell'utente e controlla anche che l'account sia attivo e approvato prima di generare un nuovo token di sessione. I token sono memorizzati nel database per ogni sessione attiva.

```
# Generazione token sicuro
token = secrets.token_hex(16)
new_token = Token(user_id=user.id, token=token)
db.session.add(new_token)
db.session.commit()
```

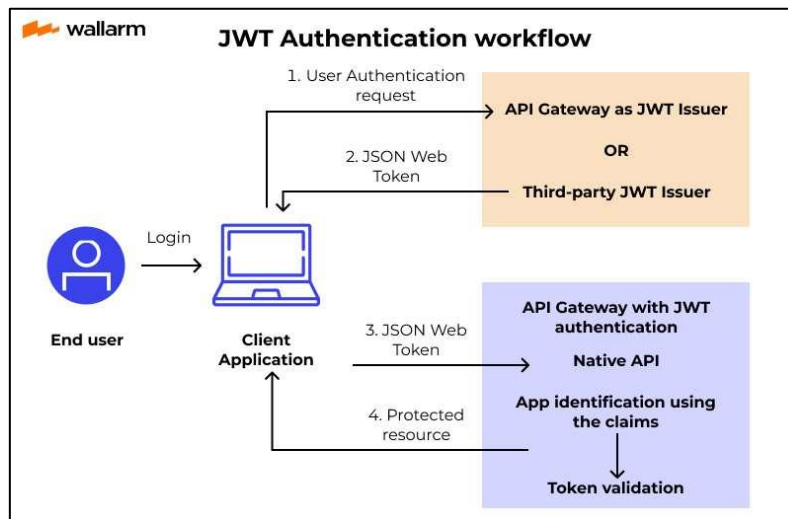


Figura 18 - Utilizzo JWT

Il JWT (JSON Web Token) è un sistema usato per riconoscere un utente in modo sicuro, spesso dopo il login. Quando un utente accede, il server crea un piccolo "biglietto" digitale che contiene alcune informazioni (come l'ID utente) e lo firma per evitare che venga modificato. Questo biglietto viene poi inviato al client (es. il browser) e usato in tutte le richieste future per dimostrare che l'utente è già stato riconosciuto. È composto da tre parti: chi è il token, cosa contiene e una firma di sicurezza.

#### 4.6.3 Approvazione utenti

Una volta registrato, l'utente dovrà aspettare l'approvazione del suo account.

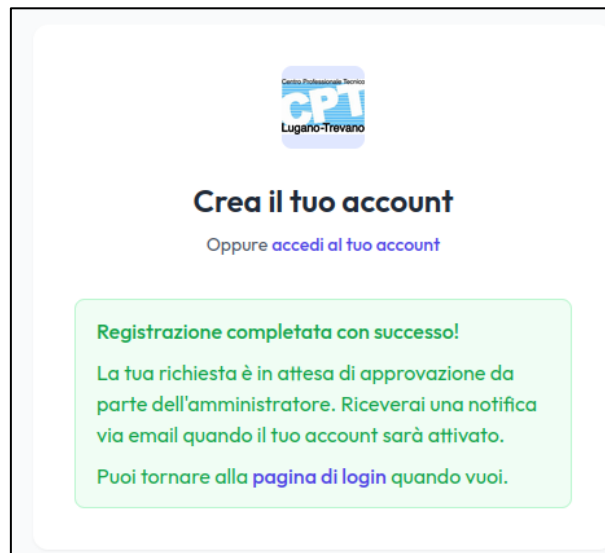


Figura 19 - Registrazione completata

Gli amministratori possono approvare o rifiutare le richieste di registrazione. Ogni azione di approvazione viene registrata nella tabella UserApproval, mantenendo uno storico completo delle decisioni di approvazione. Il sistema utilizza la relazione tra User e UserApproval per determinare l'ultimo stato di approvazione dell'utente. Una volta approvato l'utente riceve una email di notifica.

<div>Tutti gli utenti (3)</div> <div>Da approvare 1</div>				
<div>Cerca per nome, email o scuola...</div> <div>Filtri</div>				
NOME	EMAIL	SCUOLA	DATA RICHIESTA	AZIONI
Alexandru Ciobanu	alexandru.ciobanu@samtrevano.ch	Scuola media di Giubiasco	14/05/2025, 10:51	<div>✓</div> <div>✗</div>

Figura 20 - Utenti da approvare

Questa è l'interfaccia di approvazione degli utenti, dove l'admin rifiuta o accetta gli utenti, nel caso che li accetti appare un messaggio di successo.

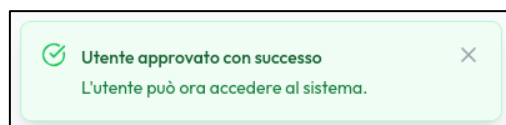


Figura 21 - Utente approvato

Una volta che l'utente è approvato riceve una email struttura in questo modo:

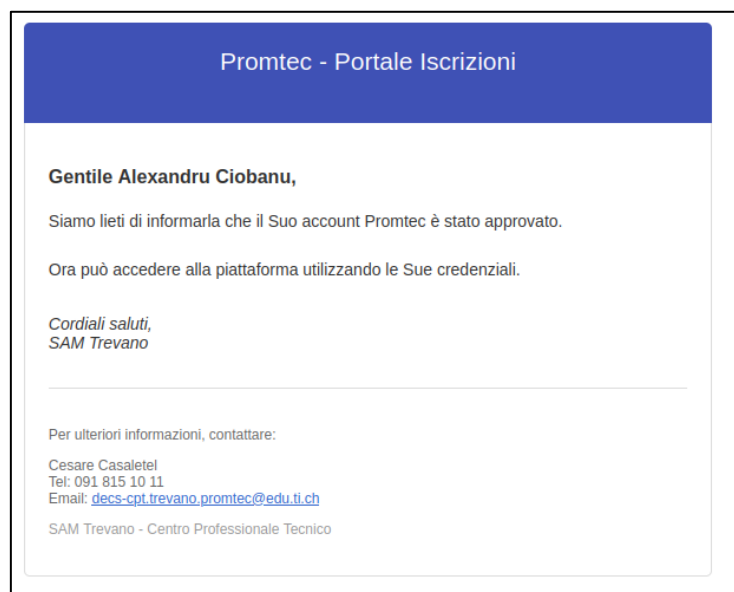


Figura 22 - Email approvazione account

#### 4.6.4 Auth storage

Al login, le informazioni come il token di accesso, il nome, cognome e se l'utente è admin, vengono salvate in un Auth storage, ovvero un sistema o un meccanismo utilizzato per memorizzare in modo sicuro le informazioni di autenticazione di un utente.

La gestione dello stato di autenticazione viene implementata tramite Pinia, un gestore di stato per Vue.js. Lo store auth.ts centralizza tutte le funzionalità relative all'autenticazione e gestisce la persistenza dei dati di sessione. I dati vengono salvati nel LocalStorage del browser.

Questo serve ad autenticare le chiamate API al backend, controlla già dal frontend quali pagine far visualizzare in base ai permessi, la home dell'admin è diversa da quella di un utente normale. In questo modo, ogni volta che un utente accede alla root del sito, verrà automaticamente indirizzato alla view corretta.

Un esempio è che nella gestione utenti, viene utilizzato per impedire la cancellazione dell'account che si sta usando.

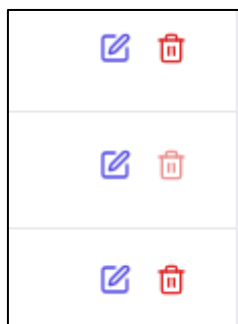


Figura 23 - Cancellazione utente stesso non permessa

Un altro esempio è la visualizzazione del nome dell'utente nel menu di navigazione.

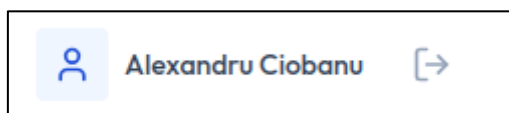


Figura 24 - Nome utente, menu navigazione



#### 4.6.5 Sicurezza

La sicurezza del sistema è garantita attraverso diversi meccanismi. Le password sono hashate con l'algoritmo pbkdf2:sha256, che offre una protezione robusta contro gli attacchi.

I token di autenticazione sono generati in modo sicuro tramite la funzione `secrets.token_hex`, che fornisce valori casuali difficilmente prevedibili. Il sistema implementa un controllo dei ruoli per l'accesso alle funzionalità amministrative.

Ad ogni login viene verificato che l'account sia attivo e approvato, impedendo l'accesso a utenti non autorizzati. Sono inoltre presenti meccanismi di protezione contro utenti eliminati o disattivati.

Per controllare in certe route che solo l'admin possa chiamarle è usato questo decorator:

```
def admin_required(f):
    @wraps(f)
    def decorated(*args, **kwargs):
        from .models import User, Token

        auth_header = request.headers.get('Authorization')
        if not auth_header or not auth_header.startswith('Bearer '):
            return jsonify({'error': 'Authentication required'}), 401

        token = auth_header.split(' ')[1]
        token_obj = Token.query.filter_by(token=token).first()

        if not token_obj:
            return jsonify({'error': 'Invalid or expired token'}), 401

        current_user = User.query.get(token_obj.user_id)

        if not current_user:
            return jsonify({'error': 'User not found'}), 401

        if not current_user.is_admin:
            return jsonify({'error': 'Admin privileges required'}), 403

        return f(*args, **kwargs)
    return decorated
```

Semplicemente ad ogni chiamata, estrae il token, controlla l'utente ed il suo ruolo.

#### 4.7 Notifiche email

Il sistema invia notifiche via email agli utenti, quando il loro account viene approvato o quando si gestiscono le iscrizioni agli slot. Le email vengono inviate utilizzando SMTP tramite un account dedicato “no-reply@samtrevano.ch”. Di seguito è riportata l'implementazione del sistema di notifica:

```
def send_email(
    to_email: str,
    subject: str,
    body: str,
) -> bool:
    """
    Send an email using SMTP
    Returns True if successful, False otherwise
    """

    smtp_server = os.getenv('SMTP_SERVER')
    smtp_port = int(os.getenv('SMTP_PORT'))
    smtp_user = os.getenv('SMTP_USER')
    smtp_password = os.getenv('SMTP_PASSWORD')

    msg = EmailMessage()
    msg["From"] = smtp_user
    msg["To"] = to_email
    msg["Subject"] = subject
    msg.set_content(body)

    try:
        with smtplib.SMTP_SSL(smtp_server, smtp_port) as server:
            server.login(smtp_user, smtp_password)
            server.send_message(msg)
        return True
    except Exception as e:
        print(f"Failed to send email: {e}")
        return False
```

La funzione send\_email, è fatta per essere riutilizzabile per tutte le email mandate dal sistema. Usa la libreria SMTP di Python, per mandare in maniera semplice le email.

Tutte le email sono formattate con HTML e CSS per fornire una grafica piacevole e non grezza.

## 4.8 Reset password

Il reset della password è stato pensato come requisito aggiuntivo, questo per migliorare l'esperienza dell'utente finale, per evitare che l'organizzatore SAMT ricevi email sul cambio delle password.

L'utente che dimentica la password, schiaccia il link sulla pagina di login, dove gli viene chiesta la propria email.

Figura 25 - Email per reset password

Nel backend viene creato un token, che serve a creare il link da mandare all'utente.

```
@staticmethod
def generate_token(user_id, expiration_hours=24):
    """Generate a new password reset token that expires after the specified hours"""
    token = secrets.token_urlsafe(32)
    expires_at = datetime.utcnow() + timedelta(hours=expiration_hours)

    reset_token = PasswordResetToken(
        user_id=user_id,
        token=token,
        expires_at=expires_at
    )
```

Il token è sicuro perché molto lungo ed ha una scadenza, la creazione è semplice.

Se l'email è valida l'utente riceve un link con dentro il token, il link assomiglia a questo:

<https://localhost/reset-password/kL5R3p4Xhpj5hb2hZWuuiQ6Uh9V-47r6gaYNomcLc0U>

Il dominio dell'applicativo viene definito nel file .env dentro a:

FRONTEND\_URL=

Una volta aperto il link l'utente ha la possibilità di modificare la password

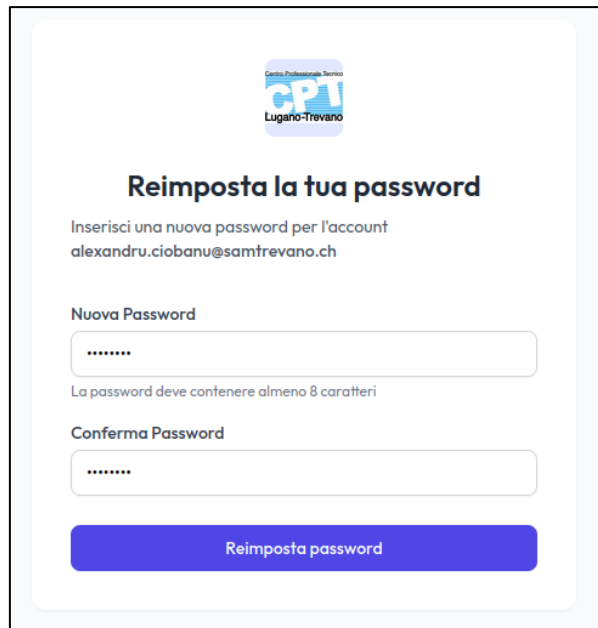


Figura 26 - Reimpostazione password

Se il token dovesse scadere ed il link dovesse essere aperto, l'utente riceve l'avviso che il token è scaduto non più valido perché già utilizzato.

## 4.9 Gestione utenti

La gestione utenti dell'admin oltre all'approvazione degli utenti, descritta precedentemente, comprende una lista di tutti gli utenti dell'applicativo, cancellabili e modificabili. Essi sono paginati, 10 utenti per pagina e filtrabili per tipo.








Gestione Utenti					Crea nuovo utente
<div>Tutti gli utenti (3)</div> <div>Da approvare</div>					
<div>Cerca per nome, email o scuola...</div> <div>Filtri</div>					
<div>Scuola</div> <div>Tutte le scuole</div>	<div>Ruolo</div> <div>Tutti i ruoli</div>	<div>Stato</div> <div>Tutti gli stati</div>			
NOME	EMAIL	SCUOLA	STATO	AZIONI	
Alexandru Ciobanu	alexandru.ciobanu@sam-trevano.ch	Scuola media di Giubiasco	Attivo	 	
Admin Admin 	admin@gmail.com	Scuola Arti e Mestieri di Trevano	Attivo	 	
sadsa 2121	alexandru.ciobanu2006+prova@gmail.com	Scuola media di Chiasso	Attivo	 	

Figura 27 - Gestione utenti

Come descritto precedentemente, l'utente admin che sta usando il progetto non può cancellarsi o modificare la propria email.

Mentre per gli altri utenti può modificare tutti i valori.

Figura 28 - Modifica utente

Per impedire la cancellazione del proprio utente, oltre ai controlli nell'API del backend, viene controllata l'email stessa dell'utente, presente nell'Auth storage, in confronto agli utenti nella tabella dell'interfaccia.

```
:class="{ 'opacity-50 cursor-not-allowed': isSelfDelete(user.email) }"
      :disabled="isSelfDelete(user.email)"
      :title="isSelfDelete(user.email) ? 'Non puoi eliminare il tuo account' :
'Elimina'"
    >
```

```
const isSelfDelete = (userEmail: string) => {
  if (!authStore.user) return false

  return authStore.user.email === userEmail
}
```

Se non è presente un utente autenticato, restituisce false. Altrimenti, restituisce true solo se l'email fornita corrisponde a quella dell'utente autenticato, indicando che si tratta di un tentativo di eliminazione del proprio account.

#### 4.10 Creazione slot

L'admin, organizzatore SAMT, crea, modifica gli slot, ovvero le giornate formative. Durano mezza giornata, al mattino o al pomeriggio, hanno 3 sezioni possibili, informatica (Tecnologie innovative), disegno (Costruzioni) e chimica.

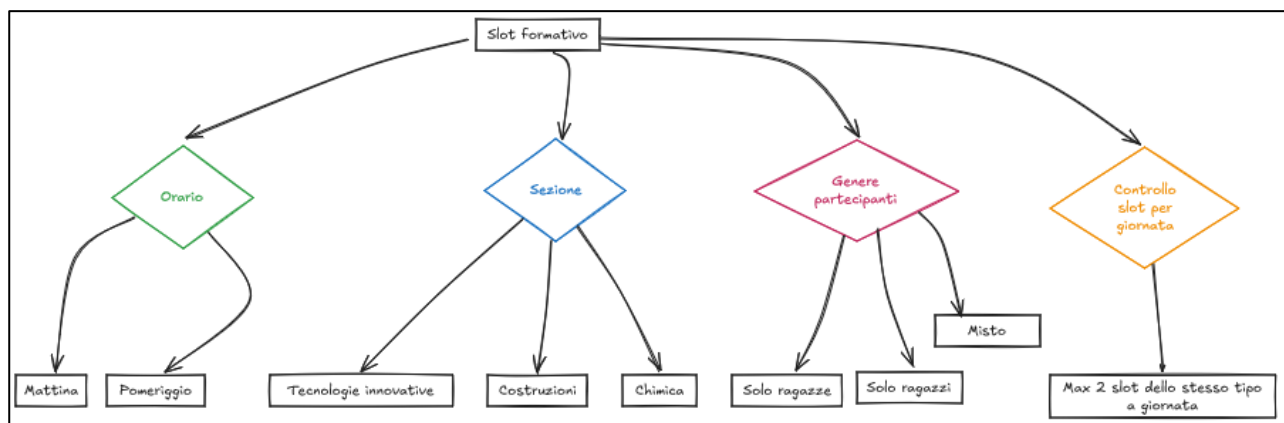


Figura 29 - Opzioni disponibili per lo slot

Non è possibile avere più di due slot dello stesso tipo nella stessa giornata, inoltre è possibile specificare se partecipano solo ragazze o ragazzi.

**Crea Nuovo Slot**

❌ Uno slot per Settore Costruzioni esiste già il 2025-05-31 durante Mattina

Data: 05 / 31 / 2025 Periodo: Mattina

Dipartimento: Settore Costruzioni Categoria: Misto

Posti Totali: 24 Max studenti per scuola (per responsabile): 10

Annulla Crea

Figura 30 - Errore creazione nuovo slot

Il numero massimo di partecipanti è 24, ma modificabile, le scuole (gli utenti) possono inserire di default un massimo di 10 studenti, ma anche questo numero è modificabile. Gli slot rimangono modificabili e cancellabili dopo la creazione, se si cancella uno slot vengono cancellati anche tutti gli iscritti.

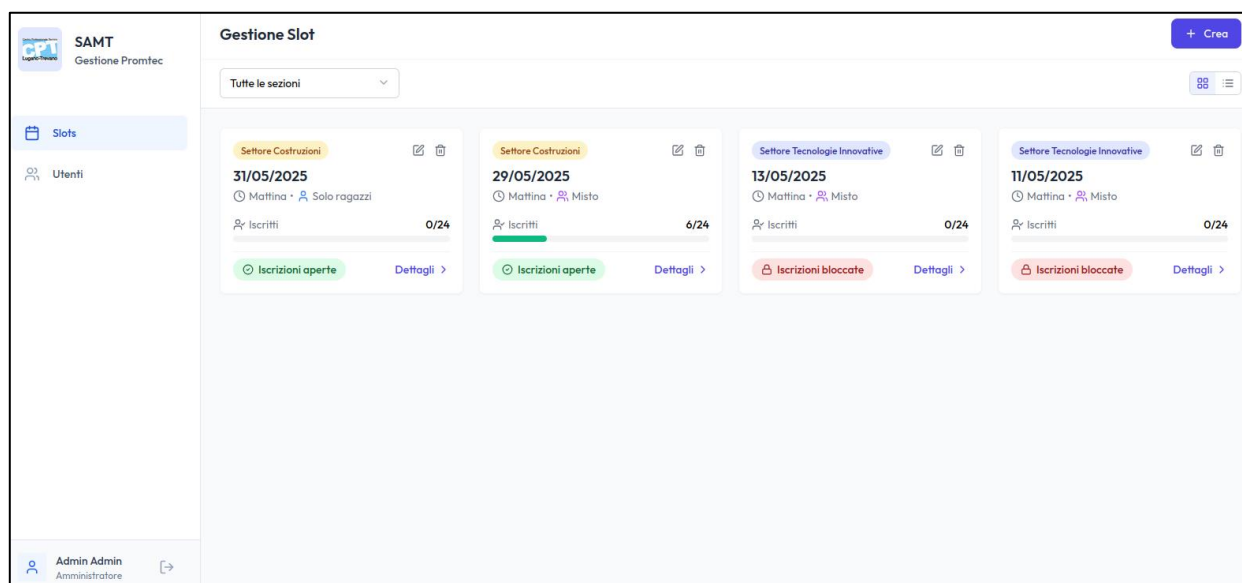


Figura 31 – Pagina slot

Vengono visualizzati dal più futuro al più passato, questa scelta è stata per evitare di dover creare un archivio, degli slot vecchi, inoltre sono paginati come gli utenti.

Due settimane prima della data dello slot, essi vengono bloccati in automatico, grazie ad una funzione direttamente nei slot/models.py.

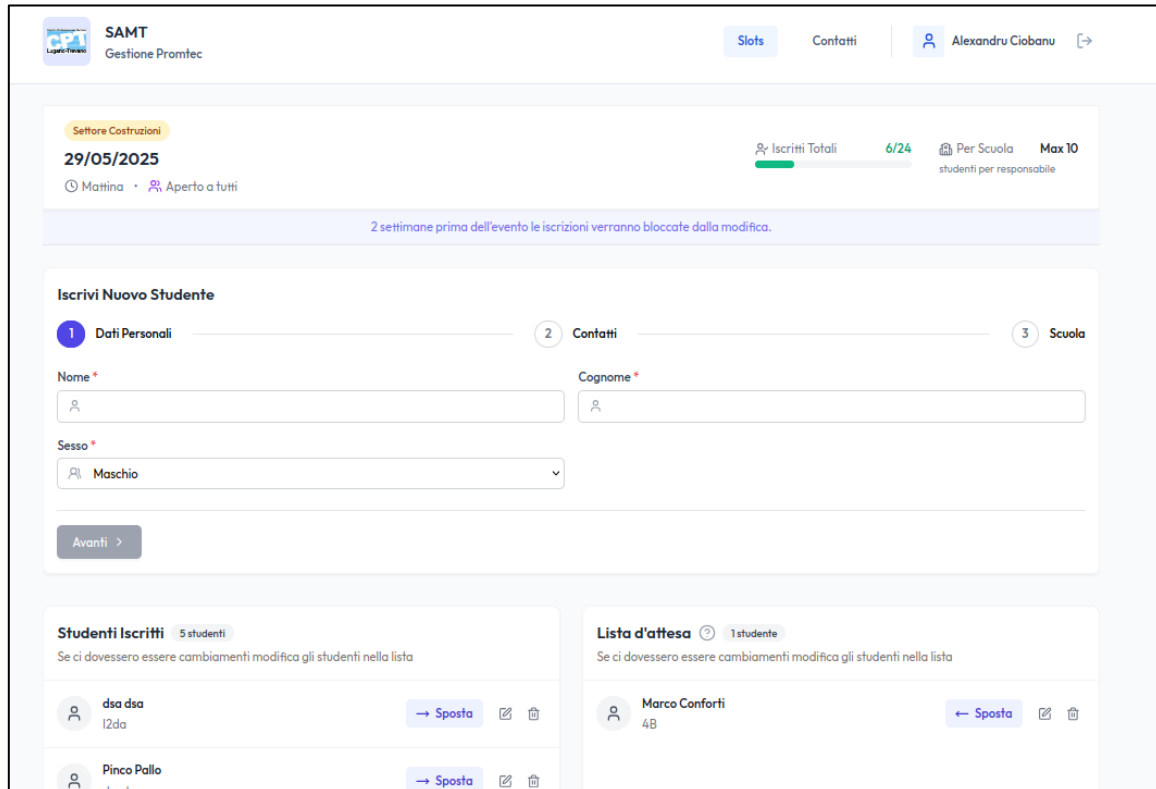
```
@hybrid_property
def should_be_locked(self):
    """Check if slot should be locked (2 weeks before date)"""
    today = datetime.now(timezone.utc).date()
    lock_date = self.date - timedelta(weeks=2)
    return today >= lock_date

@should_be_locked.expression
def should_be_locked(cls):
    today = datetime.now(timezone.utc).date()
    return cls.date - timedelta(weeks=2) <= today
```

La definizione doppia serve a definire la proprietà sia a livello di oggetto Python, sia a livello di query SQL. Controlla se mancano due settimane alla data dello slot.

#### 4.11 Iscrizione studenti da utente

Gli utenti iscrivono gli studenti per le giornate, immettendo nome, cognome, sesso, indirizzo, CAP, città, telefono (opzionale) e cellulare.



The screenshot displays the SAMT Gestione Promtec web interface. At the top, there's a header with the SAMT logo, navigation tabs for 'Slots' and 'Contatti', and a user profile for 'Alexandru Ciobanu'. The main content area is titled 'Settore Costruzioni' and shows the date '29/05/2025'. It includes a progress bar for 'Iscritti Totali' (6/24) and a 'Per Scuola' section indicating 'Max 10 studenti per responsabile'. A notification states: '2 settimane prima dell'evento le iscrizioni verranno bloccate dalla modifica.' Below this is the 'Iscrivi Nuovo Studente' section with a three-step process: 1. Dati Personali, 2. Contatti, and 3. Scuola. The 'Dati Personali' step is active, showing input fields for 'Nome', 'Cognome', and 'Sesso' (set to 'Maschio'). An 'Avanti >' button is at the bottom. Below the registration form are two panels: 'Studenti Iscritti' (5 studenti) and 'Lista d'attesa' (1 studente). The 'Studenti Iscritti' panel lists 'dsa dsa' and 'Pinco Pallo', each with a 'Sposta' button and a trash icon. The 'Lista d'attesa' panel lists 'Marco Conforti' with a 'Sposta' button and a trash icon.

Figura 32 - Iscrizione studenti

Una volta iscritto uno studente può essere spostato nella lista di attesa.

Se il responsabile prova ad iscrivere più studenti di quelli acconsentiti (minore di max\_per\_scuola e minore di posti disponibili), gli studenti vengono immessi direttamente nella lista di attesa, non potendo più essere spostati dall'altra parte.



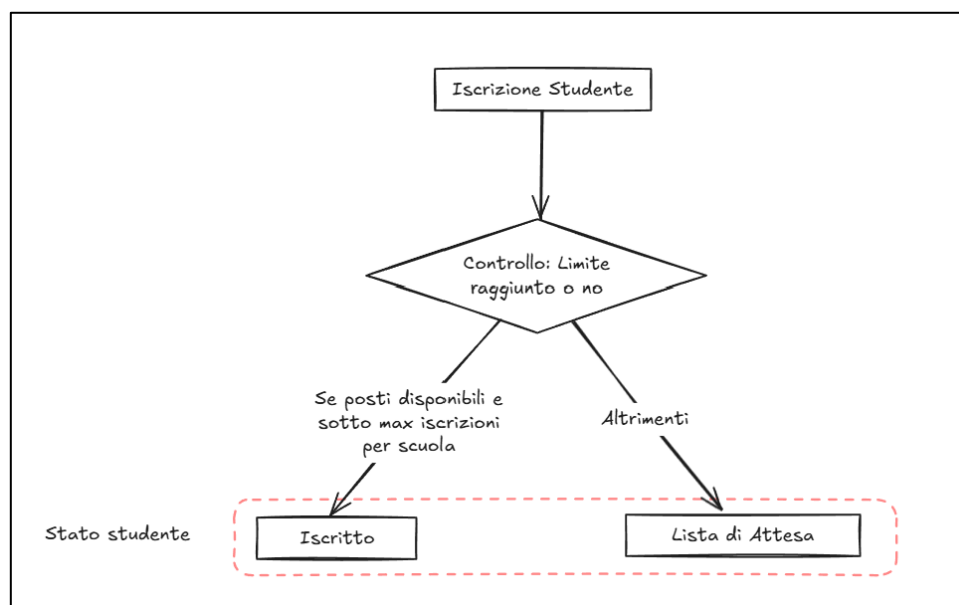


Figura 33 - Limite studenti

Questo diagramma spiega in maniera concisa il paragrafo precedente, lo studente verrà iscritto o messo in lista di attesa in base ai posti disponibili ed il limite per scuola impostato.

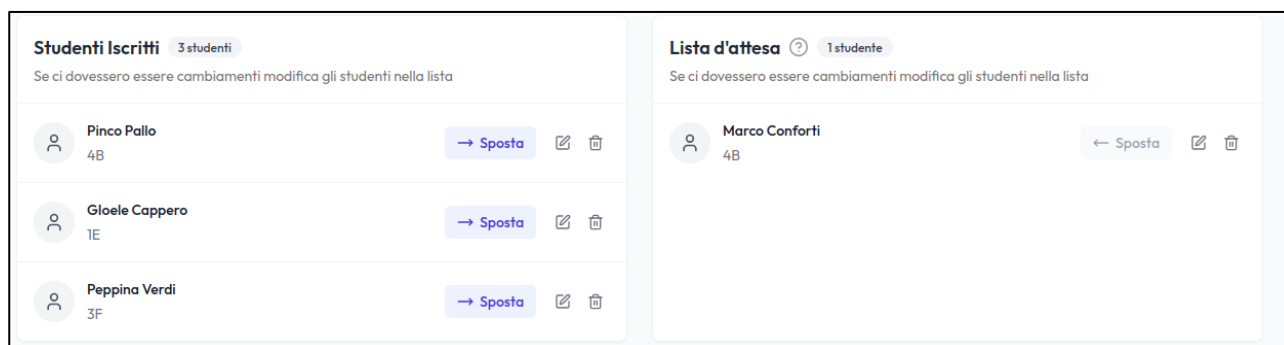


Figura 34 - Iscritti e lista d'attesa

Inoltre per i responsabili che potrebbero non capire l'utilizzo della lista d'attesa, è fornito un messaggio informativo che la spiega.

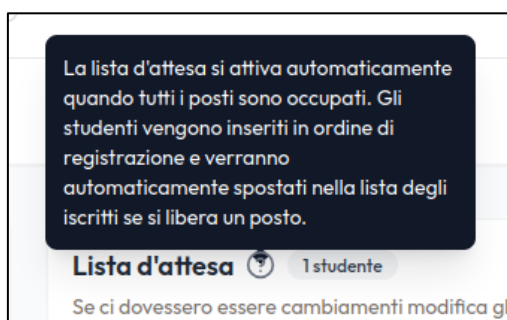


Figura 35 - Messaggio informativo lista d'attesa

Il codice per ricevere le informazioni è molto semplice:

```
@slots.route('/<int:slot_id>', methods=['GET'])
@auth.login_required
def get_slot(slot_id):
    slot = Slot.query.get_or_404(slot_id)
    return jsonify(format_slot(slot))
```

Se non trova lo slot richiesto manda un errore 404, ovvero non trovato.

Le informazioni dello slot vengono formattate in questo modo:

```
def format_slot(slot):
    return {
        'id': slot.id,
        'date': slot.date.isoformat() if slot.date else None,
        'time_period': slot.time_period.value if slot.time_period else None,
        'department': slot.department.value if slot.department else None,
        'gender_category': slot.gender_category.value if slot.gender_category else None,
        'notes': slot.notes,
        'total_spots': slot.total_spots,
        'max_students_per_school': slot.max_students_per_school,
        'is_locked': slot.is_locked,
        'is_confirmed': slot.is_confirmed,
        'created_at': slot.created_at.isoformat() if slot.created_at else None,
        'updated_at': slot.updated_at.isoformat() if slot.updated_at else None,
        'occupied_spots': slot.get_occupied_spots()
    }
```

Occupied\_spots ritorna il numero di iscrizioni presenti per lo slot, nella tabella student\_enrollment. Per calcolare questo numero si utilizza una funzione sul model.py dello slot.

```
def get_occupied_spots(self) -> int:
    """Get number of occupied spots in this slot"""
    return len([e for e in self.enrollments if not e.is_in_waiting_list])
```

Prende il numero degli elementi dati dal ciclo e lo ritorna.

Secondo consiglio del mio perito 1, l'utente dovrebbe avere la possibilità di modificare solo le informazioni di contatto e la classe, le informazioni anagrafiche e biologiche no, perché sarebbe come avere un'altra iscrizione.

Campo studente	Modificabile
Nome	NO
Cognome	NO
Sesso	NO
Scuola	NO
Classe	SI
Indirizzo	SI
CAP	SI
Città	SI
Telefono	SI
Cellulare	SI

Tutti i campi sovrastanti nel database sono cifrati.

C'è la possibilità come descritto nei capitoli precedenti, che lo slot sia bloccato sul genere, in tal caso l'utente visualizza questo obbligo così:

**Genere**

 Femmina

 Slot riservato alle ragazze

Figura 36 - Genere slot bloccato

#### 4.11.1 Crittografia studenti

Nel database i dati degli studenti sono tutti cifrati, ad eccezione di id, il nome della scuola, data di creazione e modifica. Questo per impedire alle persone che aprono il database da strumenti di visualizzazione, di vedere le informazioni sensibili degli studenti.

```
class Student(db.Model):
    id = db.Column(db.Integer, primary_key=True)

    _first_name = db.Column("first_name", db.String(255), nullable=False)
    _last_name = db.Column("last_name", db.String(255), nullable=False)
    _school_class = db.Column("school_class", db.String(255), nullable=False)
    _gender = db.Column("gender", db.String(255), nullable=False)
    _address = db.Column("address", db.String(255), nullable=False)
    ...
```

Per convenzione i valori da cifrare avranno “\_” davanti, in modo da renderli “privati”.

```
@hybrid_property
def first_name(self):
    return decrypt_value(self._first_name)

@first_name.setter
def first_name(self, value):
    self._first_name = encrypt_value(value)

@first_name.expression
def first_name(cls):
    return cls._first_name

...
```

Si tratta di una proprietà ibrida (hybrid property) di SQLAlchemy che gestisce automaticamente la crittografia e decrittografia dei dati.

@hybrid\_property in SQLAlchemy viene utilizzata per creare proprietà che funzionano sia a livello di istanza Python che a livello di query SQL.

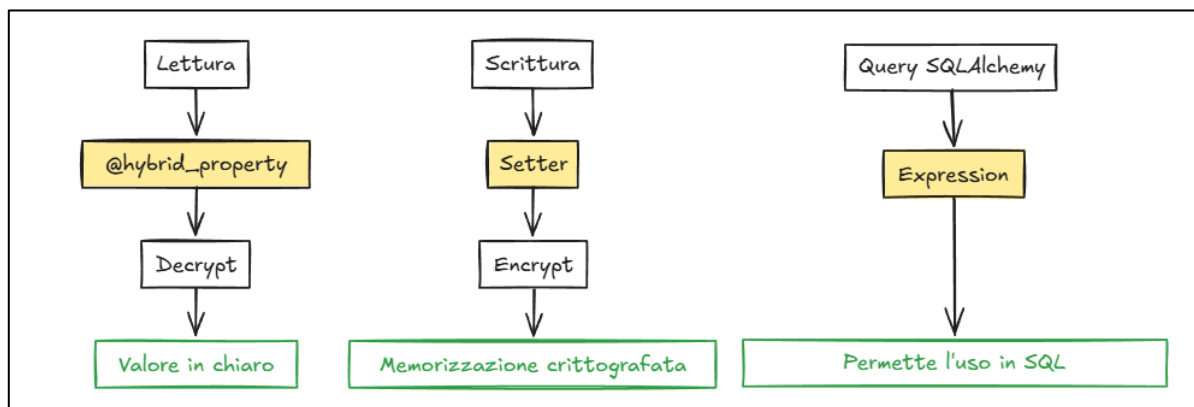


Figura 37 - Properties

@hybrid\_property definisce il comportamento quando si accede al nome dello studente, ad esempio quando si scrive **student.first\_name**. Questa funzione recupera il valore crittografato memorizzato nella colonna **\_first\_name** del database e lo decrittografa attraverso la funzione **decrypt\_value()** prima di restituirlo all'utente o all'applicazione.

@first\_name.setter viene chiamata quando si assegna un valore al nome dello studente, ad esempio **student.first\_name = "Mario"**. Cripta automaticamente il valore ricevuto utilizzando la funzione **encrypt\_value()** prima di memorizzarlo nella colonna **\_first\_name** del database.

```
.filter(User.first_name == "Mario")
```

@first\_name.expression definisce come la proprietà si comporta nelle query SQL. Restituisce direttamente il campo crittografato **\_first\_name** quando viene utilizzato in operazioni di query come filtri o ordinamenti. Questo è necessario perché SQLAlchemy deve sapere come trattare questa proprietà ibrida quando viene utilizzata nelle query. Serve quando vuoi usare first\_name in una query.

```
FERNET_KEY = os.environ.get('FERNET_KEY')
fernet = Fernet(FERNET_KEY)

def encrypt_value(value):
    if value is None:
        return None
    return fernet.encrypt(value.encode()).decode() # Cripta e converte a stringa

def decrypt_value(value):
    if value is None:
        return None
    return fernet.decrypt(value.encode()).decode() # Decripta e converte a stringa
```

Per la cifratura viene utilizzata la libreria Fernet, che utilizza l'algoritmo AES a 128 bit in modalità CBC con una chiave simmetrica, garantendo riservatezza e integrità dei dati.

La modalità CBC (Cipher Block Chaining) è un metodo di cifratura a blocchi in cui ogni blocco di testo in chiaro viene combinato con il blocco cifrato precedente prima di essere cifrato, rendendo la cifratura più sicura grazie alla dipendenza tra i blocchi.

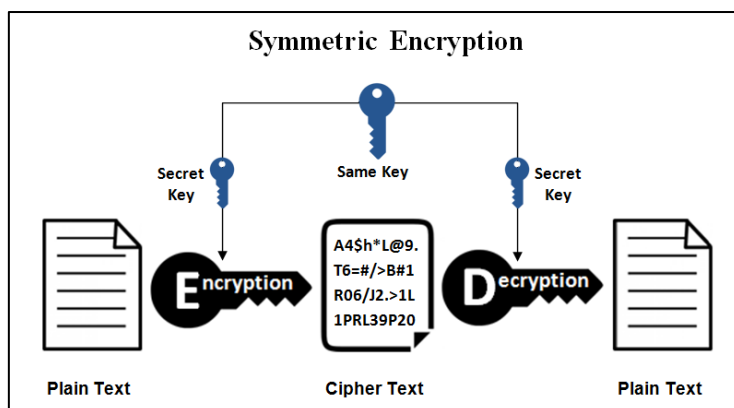


Figura 38 - Cifratura simmetrica

La cifratura con chiave simmetrica funziona utilizzando la stessa chiave sia per cifrare che per decifrare il testo.

#### 4.12 Dettagli slot

L'admin ha accesso a un elenco di iscritti, visualizzati in ordine di data di iscrizione e raggruppati per scuola. Può gestire gli iscritti spostandoli tra la lista di iscrizione e quella di attesa, nonché modificarli o aggiungerne di nuovi.

**Settore Costruzioni**  
**29/05/2025**  
 ☀ Mattina • 🧑 Aperto a tutti

Stato iscrizioni: ⏰ In attesa di conferma

✓ Conferma iscrizioni    📧 Esporta lettere

**Studenti Iscritti** (4 studenti) [Nuovo Studente](#)

Cerca studente... Tutte le scuole

STUDENTE	SCUOLA	CLASSE	DATA	AZIONI
Conforti Marco ♂ Maschio	Scuola media di Giubiasco	4B	13/05/2025	<a href="#">✎</a> <a href="#">➡</a> <a href="#">🗑</a>
Cappero Gioele ♂ Maschio	Scuola media di Giubiasco	3E	14/05/2025	<a href="#">✎</a> <a href="#">➡</a> <a href="#">🗑</a>
Verdi Peppina ♀ Femmina	Scuola media di Giubiasco	3F	14/05/2025	<a href="#">✎</a> <a href="#">➡</a> <a href="#">🗑</a>
Pero Matteo ♂ Maschio	Scuola media di Ambri	3G	15/05/2025	<a href="#">✎</a> <a href="#">➡</a> <a href="#">🗑</a>

**Lista di attesa** (2 studenti)

⚠ Come amministratore, puoi spostare studenti nella lista iscritti anche se il limite per scuola è stato raggiunto.

**Scuola media di Giubiasco** (1 studenti)

**Pallo Pinco**  
 ♀ Femmina - 4B  
 13/05/2025, 11:17

**Scuola media di Ambri** (1 studenti)

**Piccola Marchina**  
 ♀ Femmina - 4V  
 13/05/2025, 12:10

**Riepilogo Scuole** (2 scuole)

Figura 39 - Dettagli slot

È possibile visualizzare lo stato delle iscrizioni e consultare le informazioni relative a ciascuno slot. L'amministratore può superare il limite massimo di iscritti per singola scuola, spostando partecipanti dalla lista di attesa all'elenco degli iscritti. Tuttavia, non può superare il numero massimo complessivo di iscritti, se non in casi eccezionali. In tali circostanze, può modificare il limite massimo direttamente dalla pagina dell'elenco degli slot.

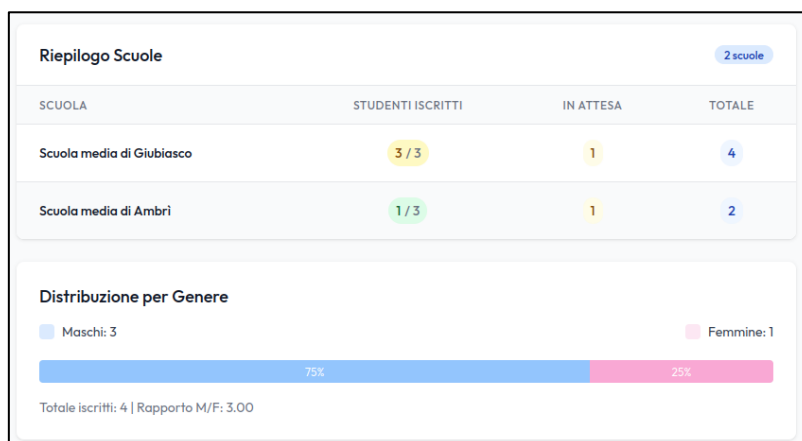


Figura 40 - Riepilogo scuole e distribuzione genere

In fondo alla pagina è possibile visualizzare il numero di iscritti per scuola e la distribuzione percentuale per genere. Queste informazioni aiutano a bilanciare le iscrizioni, nel caso si voglia intervenire per una distribuzione più equilibrata.

```
@slots.route('/<int:slot_id>/enrollments', methods=['GET'])
@auth.login_required
def get_slot_enrollments(slot_id):
    is_waiting_list = request.args.get('is_waiting_list', type=lambda x: x.lower() ==
    'true')

    query = StudentEnrollment.query.filter_by(slot_id=slot_id)

    if is_waiting_list is not None:
        query = query.filter_by(is_in_waiting_list=is_waiting_list)

    if not auth.current_user().is_admin:
        query = query.filter_by(user_id=auth.current_user().id)

    enrollments = query.all()

    return jsonify({
        'enrollments': [format_enrollment(enrollment) for enrollment in enrollments]
    })
```

Accetta un parametro facoltativo `is_waiting_list` per filtrare le iscrizioni in base alla lista d'attesa. Se l'utente autenticato non è un amministratore, la query viene limitata alle sole iscrizioni dell'utente stesso, garantendo così la riservatezza dei dati. Infine, le iscrizioni ottenute vengono convertite in formato JSON tramite la funzione `format_enrollment` e restituite come risposta.

#### 4.13 Conferma iscrizioni

Dalla pagina dei dettagli dello slot, l'amministratore può confermare le iscrizioni quando ritiene che siano complete.

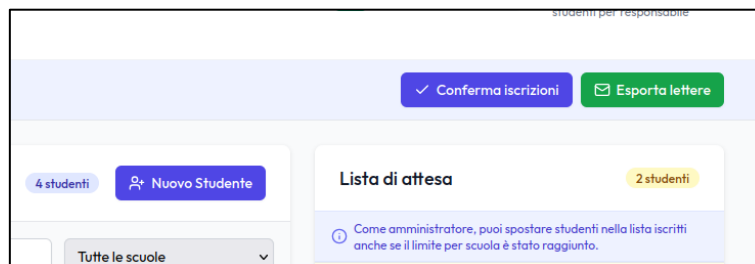


Figura 41 - Conferma slot

Per farlo deve schiacciare l'intuitivo "Conferma iscrizioni".

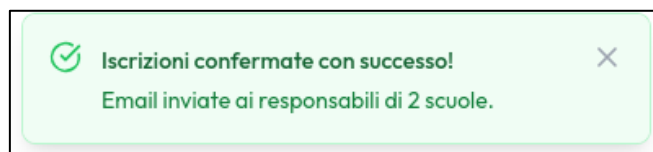


Figura 42 - Iscrizioni confermate con successo

Una volta schiacciato si riceverà un avviso “Sicuro di voler confermare tutte le iscrizioni? Questo invierà le email di conferma ai responsabili delle scuole.”, avvisando l'admin che manderà una email a tutti i responsabili delle scuole medie, una volta accettato verrà ritornato un messaggio di successo.

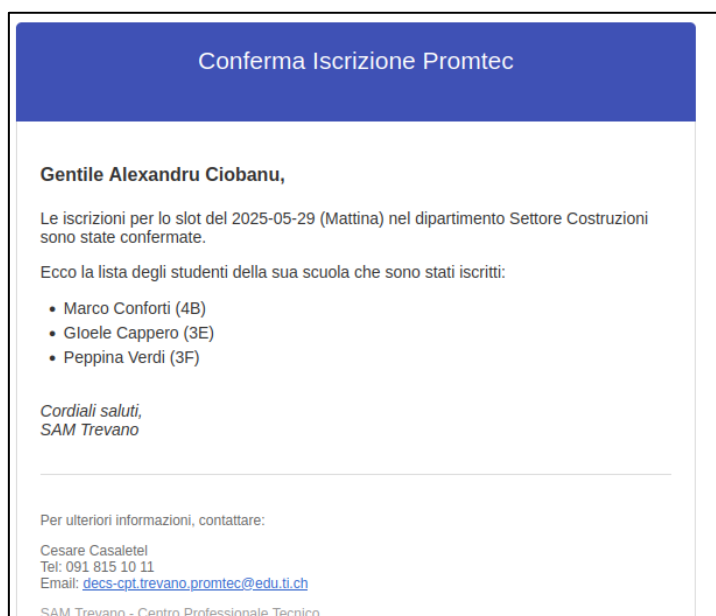


Figura 43 - Email conferma iscrizioni

I responsabili riceveranno un'email con le loro iscrizioni, una volta confermato le iscrizioni dello slot.

Se l'admin dovesse decidere di modificare alcune iscrizioni, lo stato dello slot cambierà in automatico a “Non confermato”.



```
@slots.route('/<int:slot_id>/confirm', methods=['POST'])
@auth.login_required
@admin_required
def confirm_slot(slot_id):
    slot = Slot.query.get_or_404(slot_id)

    try:
        slot.is_confirmed = True
        db.session.commit()

        enrollments = StudentEnrollment.query.filter_by(
            slot_id=slot_id,
            is_in_waiting_list=False
        ).all()

        school_data = {}
        for enrollment in enrollments:
            ...
            school_data[school_name]['students'].append(enrollment.student)

        for school_name, data in school_data.items():
            ...
            for user in data['users']:
                user_name = f"{user.first_name} {user.last_name}"

                send_slot_confirmation_email(
                    student_email=user.email,
                    user_full_name=user_name,
                    slot_info=slot_info
                )
            ...
```

Alla conferma dello slot, prima si modifica il campo `is_confirmed`, poi si prendono tutti gli utenti collegati allo slot e gli si manda una email a condizione che hanno iscritto degli studenti.

#### 4.14 Esportazione lettere

Accanto alla conferma delle iscrizioni si può far esportare le lettere, che verranno mandate via posta agli iscritti.

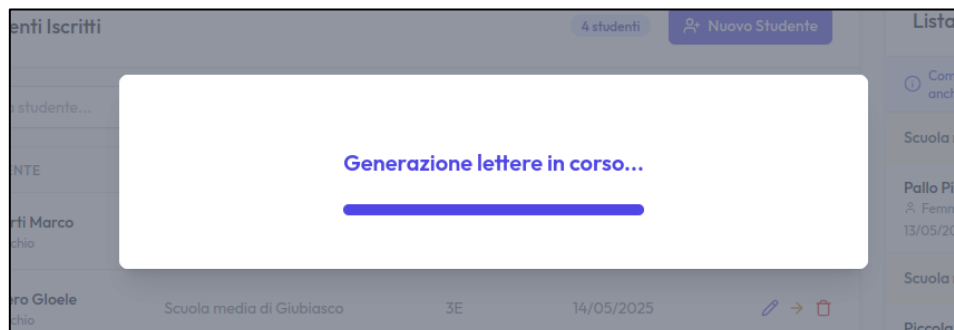


Figura 44 - Generazione lettere

Una volta iniziata la procedura di esportazione lettere, bisognerà aspettare qualche secondo. Una volta finita la generazione ed esportazione, si verrà mandati ad una nuova pagina con un file PDF contenente tutte le lettere. La scelta di mettere tutte le lettere nello stesso file è di facilitare la stampa.

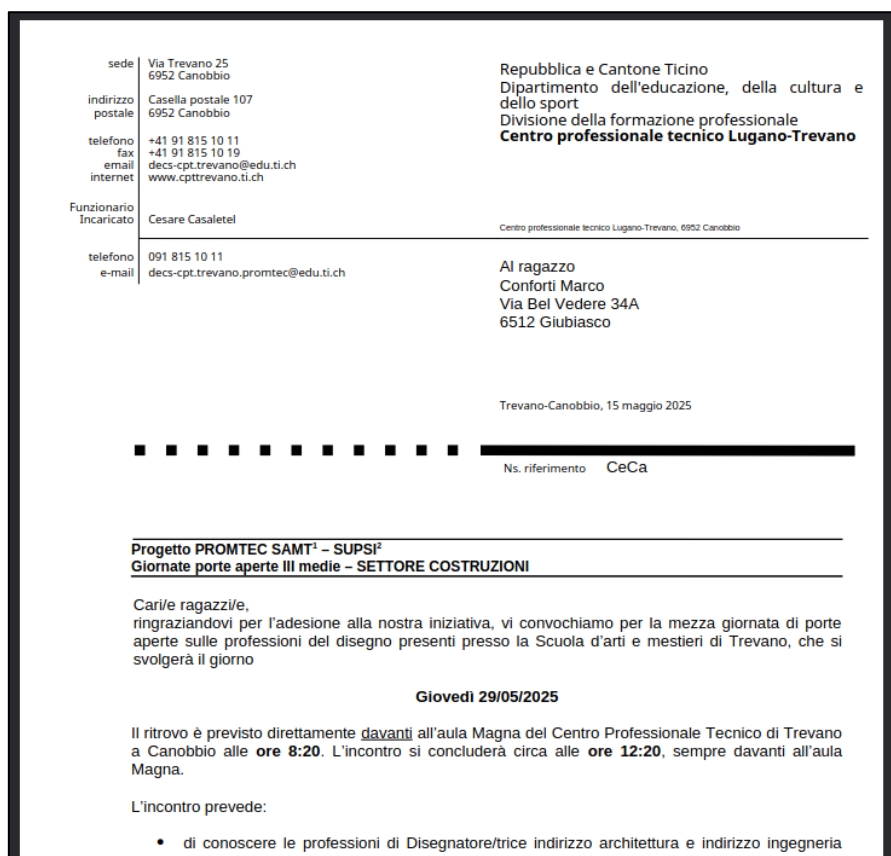


Figura 45 - Esportazione lettere

Le lettere vengono generate usando 3 template docx, in base al settore dello slot viene scelto in automatico il template da utilizzare. I template sono nella cartella **promtec-backend/app/utills/**.

```
department_template_map = {
    Department.TECH: 'template_conferma_TEC.docx',
    Department.CONSTRUCTION: 'template_conferma_DIS.docx',
    Department.CHEMISTRY: 'template_conferma_CHI.docx'
}

template_file = department_template_map[slot.department]
```

Funzionario Incaricato	{{ORGANIZZATORE}}		Centro professionale tecnico Lugano-Trevano, 6952 Canobbio
telefono	{{TEL}}		
e-mail	{{EMAIL}}	{{FORMA}} {{COGNOME}} {{NOME}} {{INDIRIZZO}} {{NAP}} {{LUOGO}}	
Trevano-Canobbio, {{DATA_ATTUALE}}			
Ns. riferimento <u>{{NoCo}}</u>			

**Progetto PROMTEC SAMT<sup>1</sup> – SUPSI<sup>2</sup>**  
**Giorate porte aperte III medie – {{SETTORE}}**

---

Cari/e ragazzi/e,  
 ringraziandovi per l'adesione alla nostra iniziativa, vi convochiamo per la mezza giornata di porte aperte sulle professioni delle tecnologie innovative presenti presso la Scuola d'arti e mestieri di Trevano, che si svolgerà il giorno

**{{GIORNO}} {{DATA}}**

Il ritrovo è previsto direttamente davanti all'aula Magna del Centro Professionale Tecnico di Trevano a Canobbio alle ore **{{ORA\_INIZIO}}**. L'incontro si concluderà circa alle ore **{{ORA\_FINE}}**, sempre davanti all'aula Magna.

L'incontro prevede:

- di conoscere le professioni di Elettronico/a multimediale e Informatico/a degli edifici (domotica),

Per ogni studente viene compilato il template del dipartimento appropriato, poi tutti i documenti docx, vengono convertiti in PDF per poi essere uniti e diventare un unico documento.

Ogni template viene riempito con:

- **Dati Studente**
  - Nome e cognome
  - Indirizzo completo, se “Via” manca, viene aggiunto
  - Forma (Al ragazzo / Alla ragazza)
  - Scuola di provenienza
- **Dati Evento**
  - Dipartimento slot
  - Giorno della settimana (tradotto in italiano)
  - Data (formato dd/mm/yyyy) (mese tradotto in italiano)
  - Orario di inizio e fine
- **Dati Organizzazione**
  - Nome e contatti dell'organizzatore
  - Data di generazione del documento

Oltre ad installare le librerie Python da requirements.txt è necessario fare:

```
sudo apt-get install unoconv libreoffice
```

#### 4.15 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.

<b>Test Case:</b>	TC-001	<b>Nome:</b>	Login con credenziali invalide
<b>Riferimento:</b>	REQ-001		
<b>Descrizione:</b>	Verificare che il sistema gestisca correttamente i tentativi di accesso con credenziali non valide.		
<b>Prerequisiti:</b>	Applicazione accessibile tramite browser.		
<b>Procedura:</b>	1. Accedere alla pagina di login dell'applicazione 2. Inserire un indirizzo email non registrato nel sistema 3. Inserire una password qualsiasi 4. Cliccare sul pulsante "Accedi" 5. In alternativa, ripetere con email valida ma password errata		
<b>Risultati attesi:</b>	Il sistema mostra un messaggio di errore.		

<b>Test Case:</b>	TC-002	<b>Nome:</b>	Login con credenziali admin
<b>Riferimento:</b>	REQ-001		
<b>Descrizione:</b>	Verificare l'accesso con credenziali di un utente amministratore.		
<b>Prerequisiti:</b>	Credenziali valide account admin di default.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla pagina di login dell'applicazione</li> <li>2. Inserire l'indirizzo email dell'amministratore</li> <li>3. Inserire la password corretta dell'amministratore</li> <li>4. Cliccare sul pulsante "Accedi"</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene reindirizzato alla dashboard amministrativa, nella dashboard sono visibili le opzioni di gestione (utenti, slot, ecc.), il nome utente dell'amministratore appare nell'interfaccia.		

<b>Test Case:</b>	TC-003	<b>Nome:</b>	Visualizzazione pagina slot vuota
<b>Riferimento:</b>	REQ-002		
<b>Descrizione:</b>	Verificare la corretta visualizzazione della pagina slot quando non ci sono slot configurati.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, nessuno slot presente nel sistema.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> </ol>		
<b>Risultati attesi:</b>	Viene mostrato un messaggio che indica l'assenza di slot, è presente un pulsante per la creazione di un nuovo slot, non sono presenti errori nella console del browser.		

<b>Test Case:</b>	TC-004	<b>Nome:</b>	Visualizzazione tabella utenti
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare la corretta visualizzazione della tabella utenti nella sezione amministrativa.		
<b>Prerequisiti:</b>	Utente amministratore autenticato.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> </ol>		
<b>Risultati attesi:</b>	La tabella degli utenti viene visualizzata correttamente, sono presenti le informazioni di base per ogni utente (nome, cognome, email, scuola, ruolo, stato), i controlli per la modifica e cancellazione sono accessibili		

<b>Test Case:</b>	TC-005	<b>Nome:</b>	Registrazione nuovo utente
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare il processo di registrazione di un nuovo utente.		
<b>Prerequisiti:</b>	Applicazione accessibile tramite browser.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla pagina principale dell'applicazione</li> <li>2. Cliccare sul link "Registrati" o navigare alla pagina di registrazione</li> <li>3. Compilare tutti i campi richiesti: <ul style="list-style-type: none"> <li>- Nome</li> <li>- Cognome</li> <li>- Email (non già registrata)</li> <li>- Password (conforme ai requisiti di sicurezza)</li> <li>- Selezionare una scuola dal menu a tendina</li> </ul> </li> <li>4. Cliccare sul pulsante "Registrati"</li> </ol>		
<b>Risultati attesi:</b>	La registrazione avviene con successo, viene mostrato un messaggio di conferma, l'utente non può ancora accedere finché non viene approvato.		

<b>Test Case:</b>	TC-006	<b>Nome:</b>	Approvazione utente
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare il processo di approvazione di un utente in attesa.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente in attesa di approvazione.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Selezionare la scheda "In attesa"</li> <li>4. Identificare l'utente da approvare</li> <li>5. Cliccare sul pulsante "Approva"</li> </ol>		
<b>Risultati attesi:</b>	Un messaggio di conferma viene mostrato, l'utente scompare dalla lista degli utenti in attesa, l'utente appare nella lista degli utenti attivi, una email di conferma viene inviata all'utente.		

<b>Test Case:</b>	TC-007	<b>Nome:</b>	Rifiuto utente
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare il processo di rifiuto di un utente in attesa.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente in attesa di approvazione.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Selezionare la scheda "In attesa"</li> <li>4. Identificare l'utente da rifiutare</li> <li>5. Cliccare sul pulsante "Rifiuta"</li> <li>6. Confermare l'azione nella finestra di dialogo</li> </ol>		
<b>Risultati attesi:</b>	un messaggio di conferma viene mostrato, l'utente scompare dalla lista degli utenti in attesa, l'utente non appare nella lista degli utenti attivi.		

<b>Test Case:</b>	TC-008	<b>Nome:</b>	Modifica utente da pannello admin
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare la capacità di modificare i dati di un utente dal pannello amministrativo.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente nel sistema.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Identificare l'utente da modificare</li> <li>4. Cliccare sul pulsante di modifica (icona a matita)</li> <li>5. Modificare alcuni campi (es. nome, cognome)</li> <li>6. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	I dati dell'utente vengono aggiornati correttamente, un messaggio di conferma viene mostrato, la tabella degli utenti viene aggiornata con i nuovi dati.		

<b>Test Case:</b>	TC-009	<b>Nome:</b>	Modifica parametro admin ad utente
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare la capacità di promuovere un utente normale ad amministratore.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente non amministratore attivo nel sistema.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Identificare l'utente normale da promuovere</li> <li>4. Cliccare sul pulsante di modifica (icona a matita)</li> <li>5. Attivare l'opzione "Amministratore"</li> <li>6. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene promosso ad amministratore, la tabella degli utenti viene aggiornata mostrando il nuovo ruolo, l'utente ha ora accesso alle funzionalità amministrative.		

<b>Test Case:</b>	TC-010	<b>Nome:</b>	Disattivazione utente
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare la capacità di disattivare un utente attivo.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente attivo nel sistema (diverso dall'amministratore corrente).		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Identificare l'utente da disattivare</li> <li>4. Cliccare sul pulsante di modifica (icona a matita)</li> <li>5. Disattivare l'opzione "Attivo"</li> <li>6. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene disattivato con successo, la tabella degli utenti viene aggiornata mostrando lo stato inattivo, l'utente disattivato non può più accedere al sistema.		



<b>Test Case:</b>	TC-011	<b>Nome:</b>	Cancellazione utente e prova login
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare la capacità di eliminare un utente e confermare che non possa più accedere.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno un utente attivo nel sistema (diverso dall'amministratore corrente).		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Identificare l'utente da eliminare</li> <li>4. Cliccare sul pulsante di eliminazione (icona cestino)</li> <li>5. Confermare l'azione nella finestra di dialogo</li> <li>6. Disconnettersi come amministratore</li> <li>7. Tentare di accedere con le credenziali dell'utente eliminato</li> </ol>		
<b>Risultati attesi:</b>	L'utente viene eliminato con successo, l'utente scompare dalla lista degli utenti, il tentativo di login con le credenziali dell'utente eliminato fallisce.		

<b>Test Case:</b>	TC-012	<b>Nome:</b>	Filtri pannello admin utenti
<b>Riferimento:</b>	REQ-009		
<b>Descrizione:</b>	Verificare il funzionamento dei filtri nella gestione utenti.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, più utenti registrati con caratteristiche diverse (scuole, ruoli, stati).		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Utenti"</li> <li>3. Utilizzare i filtri disponibili: <ul style="list-style-type: none"> <li>- Filtrare per nome/cognome</li> <li>- Filtrare per scuola</li> <li>- Filtrare per ruolo (admin/user)</li> <li>- Filtrare per stato (attivo/inattivo)</li> </ul> </li> <li>4. Verificare i risultati</li> <li>5. Resettare i filtri</li> </ol>		
<b>Risultati attesi:</b>	I filtri restringono correttamente la lista degli utenti, combinando più filtri, i risultati sono l'intersezione dei criteri, il reset dei filtri mostra nuovamente tutti gli utenti.		

<b>Test Case:</b>	TC-013	<b>Nome:</b>	Creazione slot tra 1 settimana
<b>Riferimento:</b>	REQ-002		
<b>Descrizione:</b>	Verificare la capacità di creare un nuovo slot programmato per la prossima settimana.		
<b>Prerequisiti:</b>	Utente amministratore autenticato.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Cliccare sul pulsante "Crea"</li> <li>4. Compilare il form di creazione: <ul style="list-style-type: none"> <li>- Selezionare un dipartimento (es. TEC)</li> <li>- Impostare la data a una settimana nel futuro</li> <li>- Selezionare il periodo (mattina/pomeriggio)</li> <li>- Impostare il numero di posti disponibili</li> <li>- Impostare il numero massimo di studenti per scuola</li> </ul> </li> <li>5. Cliccare su "Crea"</li> </ol>		
<b>Risultati attesi:</b>	Lo slot viene creato con successo, il nuovo slot appare nella lista degli slot, i dettagli dello slot sono corretti, lo slot ha le iscrizioni bloccate.		

<b>Test Case:</b>	TC-014	<b>Nome:</b>	Modifica parametri slot
<b>Riferimento:</b>	REQ-002		
<b>Descrizione:</b>	Verificare la capacità di modificare i parametri di uno slot esistente.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot esistente		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Identificare lo slot da modificare</li> <li>4. Cliccare sul pulsante di modifica (icona a matita)</li> <li>5. Modificare alcuni parametri (es. numero posti, max per scuola)</li> <li>6. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	I parametri dello slot vengono aggiornati correttamente, la lista degli slot viene aggiornata con i nuovi dati.		

<b>Test Case:</b>	TC-015	<b>Nome:</b>	Aggiunta studente da dettagli slot admin
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare la capacità di aggiungere un nuovo studente a uno slot dalla vista amministrativa.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot esistente con posti disponibili.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare uno slot per visualizzarne i dettagli</li> <li>4. Cliccare sul pulsante "Aggiungi Studente"</li> <li>5. Compilare tutti i campi del form di registrazione: <ul style="list-style-type: none"> <li>- Nome</li> <li>- Cognome</li> <li>- Sesso</li> <li>- Classe</li> <li>- Indirizzo</li> <li>- CAP</li> <li>- Città</li> <li>- Cellulare</li> <li>- Selezionare la scuola</li> </ul> </li> <li>6. Cliccare su "Aggiungi"</li> </ol>		
<b>Risultati attesi:</b>	Lo studente viene aggiunto con successo, lo studente appare nella lista degli studenti iscritti allo slot, il contatore degli iscritti viene incrementato.		

<b>Test Case:</b>	TC-016	<b>Nome:</b>	Aggiunta studente da dettagli slot utente
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare la capacità di aggiungere un nuovo studente a uno slot dalla vista utente.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot esistente con posti disponibili.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare uno slot per visualizzarne i dettagli</li> <li>4. Cliccare sul pulsante "Iscrivi Studente"</li> <li>5. Compilare tutti i campi del form di registrazione: <ul style="list-style-type: none"> <li>- Nome</li> <li>- Cognome</li> <li>- Sesso</li> <li>- Classe</li> <li>- Indirizzo</li> <li>- CAP</li> <li>- Città</li> <li>- Cellulare</li> </ul> </li> <li>6. Cliccare su "Iscrivi"</li> </ol>		
<b>Risultati attesi:</b>	Lo studente viene aggiunto con successo, lo studente appare nella lista degli studenti iscritti allo slot, il contatore degli iscritti viene incrementato.		

<b>Test Case:</b>	TC-017	<b>Nome:</b>	Aggiunta num max_per_scuola di studenti da utente,
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che il sistema rispetti il limite massimo di studenti per scuola in uno slot.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot esistente con valore max_per_scuola definito, l'utente ha già iscritto (max_per_scuola - 1) studenti allo slot.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Verificare il numero di studenti già iscritti dal proprio account</li> <li>5. Cliccare sul pulsante "Iscrivi Studente"</li> <li>6. Compilare tutti i campi del form di registrazione per un nuovo studente</li> <li>7. Cliccare su "Iscrivi"</li> <li>8. Tentare di iscrivere un altro studente allo stesso slot</li> </ol>		
<b>Risultati attesi:</b>	Il primo studente viene iscritto con successo, il secondo studente viene aggiunto alla lista di attesa.		

<b>Test Case:</b>	TC-018	<b>Nome:</b>	Aggiunta più studenti dei posti liberi da utente
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che il sistema gestisca correttamente la lista d'attesa quando uno slot è pieno.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot esistente con un solo posto disponibile.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot (quasi pieno) per visualizzarne i dettagli</li> <li>4. Verificare il numero di posti disponibili (dovrebbe essere 1)</li> <li>5. Cliccare sul pulsante "Iscrivi Studente"</li> <li>6. Compilare tutti i campi e scrivere un primo studente</li> <li>7. Tentare di iscrivere un secondo studente allo stesso slot</li> </ol>		
<b>Risultati attesi:</b>	Il primo studente viene iscritto con successo, il secondo studente viene aggiunto alla lista di attesa.		

<b>Test Case:</b>	TC-019	<b>Nome:</b>	Spostamento studenti iscritti - lista di attesa
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare la funzionalità di spostamento di uno studente dalla lista iscritti alla lista d'attesa.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot con uno studente iscritto dall'utente corrente.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Individuare uno studente iscritto dall'utente corrente</li> <li>5. Cliccare sull'opzione per spostare lo studente in lista d'attesa</li> </ol>		
<b>Risultati attesi:</b>	Lo studente viene spostato dalla lista iscritti alla lista d'attesa.		

<b>Test Case:</b>	TC-020	<b>Nome:</b>	Spostamento da lista di attesa a iscritti non possibile
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che non sia possibile spostare manualmente uno studente dalla lista d'attesa agli iscritti se non ci sono posti disponibili.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot con zero posti disponibili, almeno uno studente in lista d'attesa inserito dall'utente corrente.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Verificare che non ci siano posti disponibili</li> <li>5. Verificare che l'interfaccia non offra la possibilità di spostare uno studente dalla lista d'attesa agli iscritti</li> </ol>		
<b>Risultati attesi:</b>	Non è presente l'opzione per spostare studenti dalla lista d'attesa agli iscritti.		

<b>Test Case:</b>	TC-021	<b>Nome:</b>	Cancellazione iscrizione studente da utente
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che un utente possa cancellare l'iscrizione di uno studente che ha iscritto.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot con uno studente iscritto dall'utente corrente.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Individuare lo studente da cancellare</li> <li>5. Cliccare sull'opzione di cancellazione (icona cestino)</li> <li>6. Confermare l'operazione</li> </ol>		
<b>Risultati attesi:</b>	L'iscrizione dello studente viene cancellata, lo studente scompare dalla lista degli iscritti, il contatore degli iscritti viene decrementato se lo studente non era in lista d'attesa.		

<b>Test Case:</b>	TC-022	<b>Nome:</b>	Modifica iscrizione studente da utente
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che un utente possa modificare i dati di uno studente che ha iscritto.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot con uno studente iscritto dall'utente corrente.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare lo slot target per visualizzarne i dettagli</li> <li>4. Individuare lo studente da modificare</li> <li>5. Cliccare sull'opzione di modifica (icona matita)</li> <li>6. Modificare alcuni dati (telefono)</li> <li>7. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	I dati dello studente vengono aggiornati, la lista degli studenti mostra le informazioni aggiornate.		

<b>Test Case:</b>	TC-023	<b>Nome:</b>	Spostamento studenti iscritti - lista di attesa, da pannello admin
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare la funzionalità di spostamento di uno studente dalla lista iscritti alla lista d'attesa dal pannello amministrativo.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot con studenti iscritti, almeno uno studente in lista d'attesa per lo stesso slot.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Individuare uno studente iscritto</li> <li>5. Cliccare sull'opzione per spostare lo studente in lista d'attesa</li> </ol>		
<b>Risultati attesi:</b>	Lo studente viene spostato dalla lista iscritti alla lista d'attesa, il contatore degli iscritti decrementa.		

<b>Test Case:</b>	TC-024	<b>Nome:</b>	Spostamento da lista di attesa a iscritti non possibile, da pannello admin
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che non sia possibile spostare manualmente uno studente dalla lista d'attesa agli iscritti se non ci sono posti disponibili, anche dal pannello amministrativo.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot con zero posti disponibili, almeno uno studente in lista d'attesa.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot per visualizzarne i dettagli</li> <li>4. Verificare che non ci siano posti disponibili</li> <li>5. Verificare che l'interfaccia non offra la possibilità di spostare uno studente dalla lista d'attesa agli iscritti</li> </ol>		
<b>Risultati attesi:</b>	Non è presente alcun controllo per spostare manualmente studenti dalla lista d'attesa agli iscritti quando lo slot è pieno.		

<b>Test Case:</b>	TC-025	<b>Nome:</b>	Cancellazione iscrizione studente, da pannello admin
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che un amministratore possa cancellare l'iscrizione di qualsiasi studente.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot con uno studente iscritto.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot target per visualizzarne i dettagli</li> <li>4. Individuare lo studente da cancellare</li> <li>5. Cliccare sull'opzione di cancellazione (icona cestino)</li> <li>6. Confermare l'operazione</li> </ol>		
<b>Risultati attesi:</b>	L'iscrizione dello studente viene cancellata, lo studente scompare dalla lista degli iscritti, il contatore degli iscritti viene decrementato.		

<b>Test Case:</b>	TC-026	<b>Nome:</b>	Modifica iscrizione studente, da pannello admin
<b>Riferimento:</b>	REQ-006		
<b>Descrizione:</b>	Verificare che un amministratore possa modificare i dati di qualsiasi studente iscritto.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot con uno studente iscritto.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot target per visualizzarne i dettagli</li> <li>4. Individuare lo studente da modificare</li> <li>5. Cliccare sull'opzione di modifica (icona matita)</li> <li>6. Modificare alcuni dati (es. indirizzo)</li> <li>7. Cliccare su "Salva"</li> </ol>		
<b>Risultati attesi:</b>	I dati dello studente vengono aggiornati.		

<b>Test Case:</b>	TC-027	<b>Nome:</b>	Conferma slot
<b>Riferimento:</b>	REQ-007		
<b>Descrizione:</b>	Verificare la funzionalità di conferma di uno slot.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot non confermato con studenti iscritti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot target per visualizzarne i dettagli</li> <li>4. Cliccare sul pulsante "Conferma Slot"</li> <li>5. Confermare l'operazione</li> </ol>		
<b>Risultati attesi:</b>	Lo slot viene confermato con successo, lo stato dello slot cambia in "Confermato", le email di conferma vengono inviate ai responsabili delle scuole medie.		



<b>Test Case:</b>	TC-028	<b>Nome:</b>	Modifica iscrizioni dopo conferma slot
<b>Riferimento:</b>	REQ-007		
<b>Descrizione:</b>	Verificare che non sia possibile modificare le iscrizioni dopo che uno slot è stato confermato (utente normale).		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot confermato con studenti iscritti dall'utente.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot confermati</li> <li>3. Selezionare lo slot confermato per visualizzarne i dettagli</li> <li>4. Verificare la presenza o meno di controlli per modificare o cancellare le iscrizioni</li> </ol>		
<b>Risultati attesi:</b>	È presente un'indicazione che lo slot è stato confermato, le iscrizioni allo slot non è modificabile.		

<b>Test Case:</b>	TC-029	<b>Nome:</b>	Riconferma slot
<b>Riferimento:</b>	REQ-007		
<b>Descrizione:</b>	Verificare che un amministratore possa modificare anche da slot confermato, per riconfermarlo dopo.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot già confermato.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot confermato per visualizzarne i dettagli</li> <li>4. Spostare uno studente in lista di attesa e rispostarlo negli iscritti.</li> <li>5. Verificare la presenza del pulsante "Conferma slot"</li> <li>6. Cliccare sul pulsante " Conferma slot"</li> <li>7. Confermare l'operazione</li> </ol>		
<b>Risultati attesi:</b>	Lo slot viene riconfermato con successo, vengono mandate le email ai responsabili.		

<b>Test Case:</b>	TC-030	<b>Nome:</b>	Esportazione lettere
<b>Riferimento:</b>	REQ-008		
<b>Descrizione:</b>	Verificare la funzionalità di esportazione delle lettere di conferma per uno slot.		
<b>Prerequisiti:</b>	Utente amministratore autenticato, almeno uno slot confermato con studenti iscritti.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard amministrativa</li> <li>2. Navigare alla sezione "Gestione Slot"</li> <li>3. Selezionare lo slot confermato per visualizzarne i dettagli</li> <li>4. Cliccare sul pulsante "Esporta Lettere"</li> </ol>		
<b>Risultati attesi:</b>	Il sistema genera un file PDF contenente tutte le lettere di conferma, il download del file inizia automaticamente, il file contiene una lettera per ogni studente iscritto, ogni lettera contiene i dati corretti dello studente e dello slot.		

<b>Test Case:</b>	TC-031	<b>Nome:</b>	Email conferma iscrizioni provvisorie
<b>Riferimento:</b>	REQ-007		
<b>Descrizione:</b>	Verificare che il sistema invii correttamente le email di conferma per le iscrizioni provvisorie.		
<b>Prerequisiti:</b>	Utente normale (non amministratore) autenticato, almeno uno slot con posti disponibili.		
<b>Procedura:</b>	<ol style="list-style-type: none"> <li>1. Accedere alla dashboard utente</li> <li>2. Visualizzare gli slot disponibili</li> <li>3. Selezionare uno slot per visualizzarne i dettagli</li> <li>4. Iscrivere un nuovo studente</li> <li>5. Attendere l'invio dell'email al responsabile</li> <li>6. Controllare la casella di posta dell'indirizzo specificato</li> </ol>		
<b>Risultati attesi:</b>	Un'email di conferma dell'iscrizione provvisoria viene inviata, l'email contiene le informazioni corrette dello studente e dello slot		

#### 4.16 Risultati test

In seguito una tabella riassuntiva in cui sono presenti i risultati dei test. Negli allegati, nel file “risultati\_test.docx” sono presenti i risultati effettivi con le immagini.

Test	Requisito	Risultato
TC-001	REQ-001	Passato
TC-002	REQ-001	Passato
TC-003	REQ-002	Passato
TC-004	REQ-009	Passato
TC-005	REQ-009	Passato
TC-006	REQ-009	Passato
TC-007	REQ-009	Passato
TC-008	REQ-009	Passato
TC-009	REQ-009	Passato
TC-010	REQ-009	Passato
TC-011	REQ-009	Passato
TC-012	REQ-009	Passato
TC-013	REQ-002	Passato
TC-014	REQ-002	Passato
TC-015	REQ-006	Passato
TC-016	REQ-006	Passato
TC-017	REQ-006	Passato
TC-018	REQ-006	Passato
TC-019	REQ-006	Passato
TC-020	REQ-006	Passato
TC-021	REQ-006	Passato
TC-022	REQ-006	Passato
TC-023	REQ-006	Passato
TC-024	REQ-006	Passato
TC-025	REQ-006	Passato
TC-026	REQ-006	Passato
TC-027	REQ-007	Passato
TC-028	REQ-007	Passato
TC-029	REQ-007	Passato
TC-030	REQ-008	Passato
TC-031	REQ-007	Passato

#### **4.17 Mancanze/limitazioni conosciute**

Attualmente, la disattivazione di un utente non comporta la sua immediata espulsione dal sistema, infatti, per applicare effettivamente i cambiamenti, l'utente deve eseguire un nuovo login. Questo può generare situazioni in cui l'utente disattivato mantiene temporaneamente l'accesso fino a un nuovo accesso al sistema.

Inoltre, il sistema gestisce in modo limitato gli account scuola: la visibilità delle iscrizioni è circoscritta esclusivamente all'utente collegato, senza una panoramica completa che mostri tutte le iscrizioni presenti nella scuola. Questo limita la gestione centralizzata e la supervisione da parte degli amministratori scolastici.

La funzionalità di gestione della scuola, in particolare per quanto riguarda l'aggiunta e la rimozione di utenti e iscrizioni, risulta manuale, con la modifica del codice, comportando possibili inefficienze operative e margini di errore.

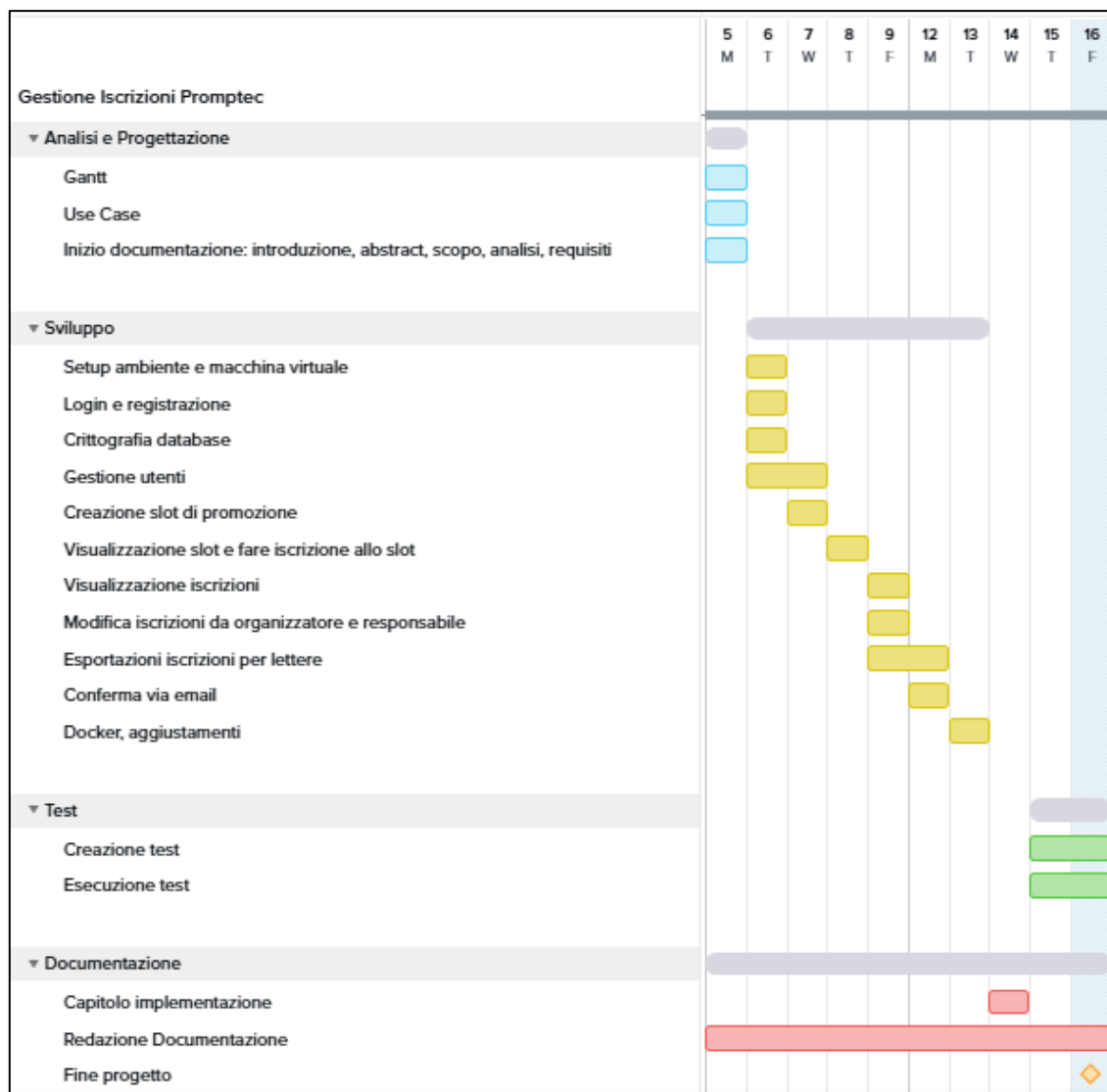
Un'altra importante limitazione riguarda l'assenza di un sistema completo di test automatizzati che coprano tutte le funzionalità principali dell'applicativo.

Dal punto di vista delle notifiche, le email inviate al momento della riconferma degli slot sono attualmente recapitate a tutti i responsabili della scuola, indipendentemente dal fatto che abbiano o meno avuto iscrizioni modificate, generando un eccesso di comunicazioni.

Infine, la tabella delle iscrizioni degli studenti non supporta un ordinamento su più parametri, limitando così la capacità di analisi e consultazione delle informazioni.

A livello di usabilità, l'applicativo non è completamente responsive con certe interfacce che non si adattano correttamente a dispositivi con schermi di dimensioni diverse.

## 5 Consuntivo



La suddivisione delle attività di sviluppo è stata pianificata in modo chiaro e strutturato. Alcune task si sono rivelati meno complessi del previsto e sono stati completati rapidamente. Tuttavia, la pianificazione iniziale non ha considerato del tutto i tempi necessari per le modifiche finali, come piccoli miglioramenti all'interfaccia, all'usabilità o l'aggiunta di funzionalità non previste. Questo ha portato a posticipare l'attività di testing verso la fine del progetto. Inoltre, non avevo stimato correttamente il tempo richiesto per la redazione del capitolo sull'implementazione, che ha occupato un'intera giornata.

Nonostante ciò, i giorni di margine previsti per eventuali imprevisti si sono rivelati utili ed hanno permesso di gestire la situazione in modo efficace e senza ritardi.

## 6 Conclusioni

La soluzione sviluppata, pur senza introdurre innovazioni rivoluzionarie, rappresenta un passo importante verso una gestione più moderna, centralizzata ed efficiente delle attività promozionali della SAMT. Attraverso la piattaforma web, l'organizzazione delle giornate informative dedicate agli studenti delle scuole medie risulta più chiara, automatizzata e accessibile.

I referenti degli istituti scolastici potranno iscrivere direttamente gli studenti ai vari slot disponibili, evitando l'uso di fogli Excel o comunicazioni frammentarie. L'interfaccia consente un'inserzione guidata e controllata dei dati, riducendo gli errori e velocizzando l'intero processo.

Dall'altra parte, l'amministratore SAMT avrà una panoramica completa e aggiornata in tempo reale delle iscrizioni, guadagnando tempo, potrà gestire gli slot, monitorare la disponibilità dei posti, controllare le liste d'attesa e tener conto sviluppi. Rispetto alla soluzione manuale precedente, il nuovo sistema offre una gestione più solida, tracciabile e adatta a rispondere alle esigenze operative di un ambiente scolastico.

### 6.1 Sviluppi futuri

All'interno dell'applicativo ci sono diversi sviluppi futuri che si potrebbero implementare. Il login tramite Network Microsoft edu.ti.ch sarebbe un'aggiunta fondamentale, in quanto permetterebbe una gestione centralizzata degli accessi, migliorando l'integrazione con l'infrastruttura scolastica e rendendo l'applicativo più adatto all'ambiente educativo cantonale.

Un altro sviluppo utile sarebbe la possibilità di riutilizzare gli studenti da un anno all'altro, così da poter dare la precedenza a chi non ha ancora partecipato a una determinata giornata, evitando doppie partecipazioni e garantendo più equità.

La creazione di un'interfaccia dedicata allo storico delle approvazioni utenti semplificherebbe il lavoro degli amministratori, che potrebbero così controllare in modo più chiaro chi è stato approvato, quando e da chi.

Anche l'aggiunta di un archivio per gli slot rappresenterebbe un miglioramento importante, per avere meno slot nelle pagine principali.

Le statistiche sulle iscrizioni degli slot sarebbero un'aggiunta molto utile per analizzare l'andamento delle iscrizioni, capire quali attività sono più richieste e prendere decisioni basate su dati reali.

Un altro sviluppo è l'importazione degli studenti iscritti da file, per supportare il vecchio sistema di iscrizioni.

Oltre a questi sviluppi già pianificati, ci sono anche delle migliorie che potrebbero risolvere le attuali mancanze.

Ad esempio, la disattivazione di un utente dovrebbe comportare l'espulsione immediata dal sistema, senza dover aspettare un nuovo login.

Anche la gestione degli account scuola andrebbe migliorata, perché attualmente ogni utente vede solo le proprie iscrizioni e manca una visione generale per tutta la scuola, che sarebbe invece molto utile per gli amministratori scolastici.

La gestione degli utenti e delle iscrizioni, che ora richiede modifiche manuali al codice, dovrebbe essere completamente integrata nell'interfaccia amministrativa per evitare errori e semplificare le operazioni.

Sarebbe anche importante introdurre un sistema di test automatizzati che copra tutte le funzionalità principali, così da garantire maggiore stabilità e affidabilità dell'applicativo.

Dal punto di vista delle notifiche, bisognerebbe evitare che le email di riconferma degli slot vengano inviate a tutti i responsabili della scuola, anche a quelli che non hanno subito modifiche, per ridurre il numero di comunicazioni inutili.

Infine, la tabella delle iscrizioni dovrebbe supportare un ordinamento su più parametri per facilitare l'analisi dei dati, e l'interfaccia andrebbe resa completamente responsive, in modo da funzionare correttamente su qualsiasi tipo di dispositivo, anche con schermi piccoli.

## 6.2 Considerazioni personali

Sono soddisfatto del lavoro svolto e, soprattutto, del fatto che ciò che ho realizzato verrà effettivamente utilizzato. Questo conferisce un valore concreto al progetto. Fin dall'inizio, la progettazione è stata pensata in modo realistico.

Le scelte iniziali sono state, nel complesso, ordinate e consapevoli, ma durante lo sviluppo ho commesso alcuni errori dovuti alla fretta o a una gestione meno rigorosa di alcune fasi. In particolare, alcune decisioni prese velocemente per avanzare con lo sviluppo si sono rivelate poco sostenibili e hanno richiesto successive revisioni. Questo mi ha insegnato l'importanza di mantenere coerenza e attenzione anche ai dettagli apparentemente secondari. Ad esempio, alcuni file soprattutto nel frontend sono diventati troppo lunghi. Inoltre, in alcuni casi mancano messaggi di conferma: le operazioni vengono eseguite, ma è necessario verificarne manualmente l'esito.

Sono però molto soddisfatto di essere riuscito a utilizzare Docker in modo efficace, consolidando competenze che fino a questo progetto non avevo praticato in modo completo. Inoltre, ho affrontato tematiche che in passato avevo volutamente evitato, come la gestione degli utenti e la funzionalità di reset della password. Questo mi ha permesso di uscire dalla mia comfort zone e affrontare in modo più completo lo sviluppo di un'applicazione reale.

La parte che mi ha dato più soddisfazione è quella relativa all'esportazione delle lettere. Anche se il sistema precedente era in parte automatizzato, so quanto lavoro manuale ci sia spesso dietro attività simili. Mi piace automatizzare i processi, soprattutto quando si tratta di attività utili, concrete e che verranno effettivamente utilizzate.

Inoltre, ho potuto lavorare con tecnologie che mi piacciono e che so gestire bene, come Python per il backend e Vue per il frontend. Questo mi ha permesso di concentrarmi meglio sulla logica applicativa e sulla qualità del codice, e allo stesso tempo di consolidare le mie competenze su strumenti che trovo intuitivi e potenti.

Nonostante gli inevitabili margini di miglioramento, sono contento del risultato finale. Ho imparato molto, soprattutto dal punto di vista metodologico, e ho compreso quanto una buona pianificazione e un approccio ordinato possano fare la differenza nello sviluppo di un progetto concreto. Spero che gli utilizzatori ne saranno soddisfatti quanto me, o anche di più.

## 7 Glossario

In questo capitolo è presente il glossario, con le definizioni dei termini tecnici.

Termine	Definizione
<b>Admin</b>	Utente con privilegi elevati per gestire un sistema o un'app.
<b>AES</b>	Algoritmo di cifratura simmetrica usato per proteggere i dati.
<b>API</b>	Interfaccia che permette la comunicazione tra software diversi.
<b>Auth</b>	Processo di autenticazione di un utente.
<b>Backend</b>	Parte di un'app che gestisce logica, dati e server.
<b>Browser</b>	Programma per navigare su Internet (es. Chrome, Firefox).
<b>CBC</b>	Modalità di cifratura che usa blocchi concatenati.
<b>Certificato HTTPS</b>	Certificato digitale che abilita la comunicazione sicura via HTTPS.
<b>CSS</b>	Linguaggio per lo stile grafico delle pagine web.
<b>Crow's Foot</b>	Notazione grafica per rappresentare relazioni nei database.
<b>Database</b>	Insieme organizzato di dati memorizzati e gestiti elettronicamente.
<b>DB</b>	Abbreviazione di "database".
<b>Docker</b>	Piattaforma per creare, distribuire e gestire container.
<b>Dockerfile</b>	File di testo con istruzioni per creare un'immagine Docker.
<b>DOCX</b>	Formato di documento Microsoft Word.
<b>Enum</b>	Tipo di dato che contiene un insieme fisso di valori.
<b>ENV</b>	File che contiene le variabili d'ambiente usate da un'applicazione.
<b>Fernet</b>	Sistema di cifratura simmetrica con scadenza e autenticazione integrata.
<b>Flask</b>	Framework leggero per creare applicazioni web in Python.
<b>Frontend</b>	Parte visibile di un sito o app con cui interagisce l'utente.
<b>Gantt</b>	Diagramma per pianificare attività su una scala temporale.
<b>Hostname</b>	Nome assegnato a un dispositivo in una rete.
<b>HTML</b>	Linguaggio per strutturare le pagine web.
<b>IP</b>	Indirizzo che identifica un dispositivo in rete.
<b>JWT</b>	Metodo per trasmettere in modo sicuro informazioni tra parti come token.
<b>Kanban</b>	Metodo di gestione del lavoro visuale con schede e colonne.
<b>Linux Mint</b>	Distribuzione di Linux orientata alla semplicità d'uso.
<b>Local storage</b>	Memoria del browser per salvare dati localmente.
<b>Mockup</b>	Rappresentazione visiva di un'interfaccia o prodotto.
<b>MySQL</b>	Sistema di gestione di database relazionali.
<b>Nginx</b>	Web server usato per servire contenuti web e fare da proxy.



<b>Node.js</b>	Ambiente di esecuzione JavaScript lato server.
<b>PDF</b>	Formato di documento portatile, leggibile su ogni dispositivo.
<b>Pinia</b>	Libreria per gestire lo stato in applicazioni Vue.js.
<b>PIP</b>	Gestore di pacchetti per Python.
<b>Proxy</b>	Server che funge da intermediario tra client e server.
<b>Query</b>	Richiesta per ottenere o manipolare dati da un database.
<b>QDC</b>	Qualità, Disponibilità e Coerenza di un dato o sistema.
<b>REST</b>	Stile architetturale per progettare API basate su HTTP.
<b>Reverse proxy</b>	Proxy che riceve richieste dal client e le inoltra al server corretto.
<b>rete bridge</b>	Rete virtuale che collega container come se fossero sulla stessa rete fisica.
<b>Root</b>	Utente o cartella principale in un sistema.
<b>SAMT</b>	Scuola Arti e Mestieri di Trevano
<b>Schema ER</b>	Diagramma che mostra entità e relazioni in un database.
<b>SMTP</b>	Protocollo per inviare email tra server.
<b>SQL</b>	Linguaggio per gestire e interrogare database relazionali.
<b>SQLalchemy</b>	Libreria Python per lavorare con database SQL in modo più semplice.
<b>SSL</b>	Protocollo per proteggere le comunicazioni in rete.
<b>TLS</b>	Evoluzione di SSL per garantire connessioni sicure.
<b>Token</b>	Codice usato per autenticare o autorizzare un utente o una sessione.
<b>Trello</b>	Strumento online per la gestione di progetti con schede.
<b>Typescript</b>	Linguaggio basato su JavaScript con tipizzazione statica.
<b>URL</b>	Indirizzo usato per accedere a una risorsa su Internet.
<b>User</b>	Persona che utilizza un sistema o applicazione.
<b>VITE</b>	Strumento di build rapido per progetti frontend moderni.
<b>Vue.js</b>	Framework JavaScript per costruire interfacce web interattive.
<b>Waterfall</b>	Modello di sviluppo software lineare a fasi sequenziali.
<b>Web server</b>	Software che consegna pagine web ai browser.
<b>Variabili d'ambiente</b>	Valori definiti nel sistema operativo usati da applicazioni per configurarsi.

## 8 Indice delle figure

In seguito l'indice delle figure, del documento:

Figura 1 - Use case .....	9
Figura 2 - Gantt.....	10
Figura 3 - Architettura .....	12
Figura 4 - Schema ER .....	13
Figura 5 - Design login.....	18
Figura 6 - Design gestione slot.....	18
Figura 7 - Design dettagli slot.....	19
Figura 8 - Design gestione utenti.....	19
Figura 9 - Design iscrizione studenti da utente .....	20
Figura 10 - Approvazione utente .....	22
Figura 11 - Azioni admin.....	23
Figura 12 - Azioni utenti.....	24
Figura 13 - Struttura cartelle.....	25
Figura 14 - Env frontend .....	28
Figura 15 - Connessione tra container .....	31
Figura 16 - Informazioni di contatto nella email.....	35
Figura 17 - Registrazione utente .....	36
Figura 18 - Utilizzo JWT .....	38
Figura 19 - Registrazione completata.....	38
Figura 20 - Utenti da approvare.....	39
Figura 21 - Utente approvato.....	39
Figura 22 - Email approvazione account.....	39
Figura 23 - Cancellazione utente stesso non permessa .....	40
Figura 24 - Nome utente, menu navigazione .....	40
Figura 25 - Email per reset password.....	43
Figura 26 - Reimpostazione password .....	44
Figura 27 - Gestione utenti .....	44
Figura 28 - Modifica utente .....	45
Figura 29 - Opzioni disponibili per lo slot.....	46
Figura 30 - Errore creazione nuovo slot .....	46
Figura 31 - Pagina slot.....	47
Figura 32 - Iscrizione studenti .....	48
Figura 33 - Limite studenti .....	49
Figura 34 - Iscritti e lista d'attesa .....	49
Figura 35 - Messaggio informativo lista d'attesa .....	49
Figura 36 - Genere slot bloccato .....	51
Figura 37 - Properties .....	52
Figura 38 - Cifratura simmetrica .....	53
Figura 39 - Dettagli slot.....	54
Figura 40 - Riepilogo scuole e distribuzione genere .....	54
Figura 41 - Conferma slot.....	55
Figura 42 - Iscrizioni confermate con successo .....	56
Figura 43 - Email conferma iscrizioni .....	56
Figura 44 - Generazione lettere.....	58
Figura 45 - Esportazione lettere .....	58
Figura 46 - Template lettera Tecnologie Innovative .....	59

## 9 Bibliografia

### 9.1 Sitografia

Nella lista i siti web consultati per l'acquisizione del sapere, questi siti sono in concomitanza all'utilizzo dell'IA:

- <https://www4.ti.ch/decs/ds/portale-scuole/scuole/medie?start=2&cHash=2f63ca99ff0ec42dd436ce9a1cb227fd>, *Scuole Medie del Ticino*, 05-05-2025
- <https://stackoverflow.com/questions/60311657/vuejs-route-loading-based-on-user-type>, *Vue.js route loading based on user type*, 06-05-2025
- <https://realpython.com/python-send-email/>, *Sending Emails With Python*, 06-05-2025
- <https://pinia.vuejs.org/>, *Pinia Documentation*, 07-05-2025
- <https://www.geeksforgeeks.org/fernet-symmetric-encryption-using-cryptography-module-in-python/>, *Fernet using Cryptography module in Python*, 08-05-2025
- [https://www.youtube.com/watch?v=RrhLImE0Zf8&ab\\_channel=PythonicAccountant](https://www.youtube.com/watch?v=RrhLImE0Zf8&ab_channel=PythonicAccountant), *Automatizza i documenti Word con Python*, 09-05-2025
- <https://github.com/daddydrac/vuejs-flask-docker/blob/master/services/frontend/ui/vue.config.js>, *vuejs-flask-docker*, 13-05-2025
- <https://stackoverflow.com/questions/48299505/docker-set-up-mysql-db-environment>, *Docker Set Up mysql db environment*, 15-05-2025.

### 9.2 Utilizzo IA

L'uso dell'intelligenza artificiale è documentato dentro ai diari in allegato.

## 10 Allegati

Elenco degli allegati, esempio:

- Diari di lavoro
- Codici sorgente/documentazione macchine virtuali
- Mandato e/o QdC
- Prodotto
- Risultati test
- Use case
- Gantt pianificazione
- Gantt consuntivo