

# **Volontariato - DoIT**

The Singleton Squad

**Daniele Buser**

Matricola: 894514

Email: d.buser@campus.unimib.it

**Gabriele Groppo**

Matricola: 902238

Email: g.groppo@campus.unimib.it

**Andrea Cozzi**

Matricola: 899627

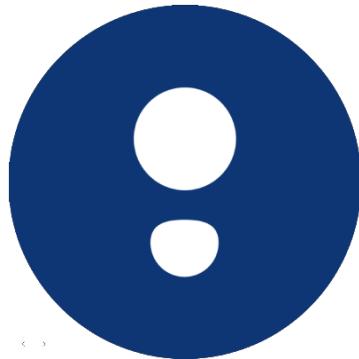
Email: a.cozzi25@campus.unimib.it

**Matteo Cervini**

Matricola: 902225

Email: m.cervini1@campus.unimib.it

26 Gennaio, 2025



# Contents

<b>Introduzione</b>	<b>6</b>
<b>Compilazione</b>	<b>7</b>
<b>Iterazione 1</b>	<b>7</b>
<b>1 Diagramma di Gantt</b>	<b>7</b>
<b>2 Requisiti</b>	<b>13</b>
1 Casi d'uso . . . . .	13
1.1 Caso d'uso UC1: RegistraVolontario . . . . .	13
1.2 Caso d'uso UC2: RegistraOrganizzazione . . . . .	17
1.3 Caso d'uso UC3: GestioneRichiesteVolontariato . . . . .	21
2 Diagramma dei Casi d'Uso . . . . .	25
3 Diagramma di Sequenza di Sistema . . . . .	25
3.1 Diagramma di Sequenza di Sistema SSD1: RegistraVolontario . . . . .	25
3.2 Diagramma di Sequenza di Sistema SSD2: RegistraOrganizzazione . . . . .	27
3.3 Diagramma di Sequenza di Sistema SSD3: GestioneRichiesteVolon- tariato . . . . .	29
3.4 Diagramma di Sequenza di Sistema SSD3.2: ControlloCorrettez- zaCampi . . . . .	31
4 Contratti . . . . .	32
4.1 Contratto CO01: RegistraOrganizzazione . . . . .	32
4.2 Contratto CO02: GestioneRichiesteVolontariato . . . . .	33
4.3 Contratto CO02.2: ControlloCorrettezzaCampi . . . . .	34
5 Specifica Supplementare . . . . .	35
5.1 Nota Introduttiva . . . . .	35
5.2 Requisiti Funzionali . . . . .	35
5.3 Requisiti Non Funzionali . . . . .	38
5.4 Requisiti Hardware . . . . .	39
5.5 Requisiti Software . . . . .	40
5.6 Specifiche di Interfaccia . . . . .	41
5.7 Vincoli di Progetto . . . . .	42
5.8 Gestione del Rischio . . . . .	42
6 Glossario . . . . .	43
6.1 Nota Introduttiva . . . . .	43
6.2 Glossario Tecnico . . . . .	43

7	Visione . . . . .	47
7.1	Nota Introduttiva . . . . .	47
<b>3</b>	<b>Modellazione di Business</b>	<b>50</b>
1	Modello delle Classi Concettuali . . . . .	50
<b>4</b>	<b>Progettazione</b>	<b>51</b>
1	Diagrammi d'Interazione . . . . .	51
1.1	Diagramma di Sequenza SD1: RegisterOrganization . . . . .	51
1.2	Diagramma di Sequenza SD2: SubmitRequestData . . . . .	52
1.3	Diagramma di Sequenza SD3: GetVolunteerRequestAsList . . . . .	53
2	Diagrammi delle Attività . . . . .	54
2.1	Diagramma delle Attività DA1: SubmitRequest . . . . .	54
3	Diagrammi delle Macchine a Stati . . . . .	57
3.1	Diagramma delle Macchine a Stati DS1: RegisterVolunteer . . . . .	57
3.2	Diagramma delle Macchine a Stati DS2: VolunteerOffers . . . . .	59
4	Diagrammi delle Classi Software di Progetto . . . . .	61
4.1	Diagramma delle Classi Software di Progetto DCSP1: Dominio . . . . .	61
4.2	Diagramma delle Classi Software di Progetto DCSP2: Frontend . . . . .	62
4.3	Diagramma delle Classi Software di Progetto DCSP3: Backend/-Controller . . . . .	63
4.4	Diagramma delle Classi Software di Progetto DCSP4: Backend/-Domain . . . . .	64
4.5	Diagramma delle Classi Software di Progetto DCSP5: Backend/DTO . . . . .	65
4.6	Diagramma delle Classi Software di Progetto DCSP6: Backend/Map-pers . . . . .	66
4.7	Diagramma delle Classi Software di Progetto DCSP7: Backend/Ser-vice . . . . .	67
4.8	Diagramma delle Classi Software di Progetto DCSP8: Backend/Repos-itory . . . . .	68
5	Diagramma dell'Architettura Logica . . . . .	69
5.1	Diagramma dell'Architettura Logica . . . . .	69
6	Pattern . . . . .	71
6.1	Design Patterns . . . . .	71
<b>Iterazione 2</b>		<b>72</b>
<b>1</b>	<b>Diagramma di Gantt</b>	<b>72</b>

<b>2 Requisiti</b>	<b>77</b>
1 Casi d'uso . . . . .	77
1.1 Caso d'uso UC4: GestioneOfferteVolontariato . . . . .	77
1.2 Caso d'uso UC5: RicercaAbbinata . . . . .	81
2 Diagramma dei Casi d'Uso . . . . .	84
3 Diagramma di Sequenza di Sistema . . . . .	86
3.1 Diagramma di Sequenza di Sistema SSD4: GestioneOfferteVolontariato . . . . .	86
3.2 Diagramma di Sequenza di Sistema SSD5: RicercaAbbinata . . . . .	87
4 Contratti . . . . .	89
4.1 Contratto CO03: RegistratiAdUnaRichiestaDiVolontariato . . . . .	89
4.2 Contratto CO04: OrdinaRichiesteSuPreferenzeUtente . . . . .	91
5 Specifica Supplementare . . . . .	91
5.1 Nota Introduttiva . . . . .	92
5.2 Requisiti Funzionali . . . . .	92
6 Glossario . . . . .	93
6.1 Nota Introduttiva . . . . .	93
6.2 Glossario Tecnico . . . . .	93
<b>3 Modellazione di Business</b>	<b>94</b>
1 Modello delle Classi Concettuali . . . . .	94
<b>4 Progettazione</b>	<b>95</b>
1 Diagrammi delle Classi Software di Progetto . . . . .	95
1.1 Diagramma delle Classi Software di Progetto DCSP9: Dominio . . . . .	95
1.2 Diagramma delle Classi Software di Progetto DCSP10: Frontend . . . . .	97
1.3 Diagramma delle Classi Software di Progetto DCSP11: Backend/Controller . . . . .	98
1.4 Diagramma delle Classi Software di Progetto DCSP12: Backend/Domain . . . . .	98
1.5 Diagramma delle Classi Software di Progetto DCSP13: Backend/DTO . . . . .	100
1.6 Diagramma delle Classi Software di Progetto DCSP14: Backend/Mappers . . . . .	101
1.7 Diagramma delle Classi Software di Progetto DCSP15: Backend/Service . . . . .	101
1.8 Diagramma delle Classi Software di Progetto DCSP16: Backend/Repository . . . . .	102

2	Diagramma dell'Architettura Logica . . . . .	103
2.1	Diagramma dell'Architettura Logica . . . . .	103
3	Pattern . . . . .	105
3.1	Design Patterns . . . . .	105
3.2	Pattern Architetturali . . . . .	106
3.3	Buone Pratiche . . . . .	107
3.4	Design Principles . . . . .	107
	<b>Artefatti Supplementari nella loro versione finale</b>	<b>108</b>
1	Specifiche Supplementare . . . . .	108
1.1	Nota Introduttiva . . . . .	108
1.2	Requisiti Funzionali . . . . .	108
1.3	Requisiti Non Funzionali . . . . .	110
1.4	Requisiti Hardware . . . . .	112
1.5	Requisiti Software . . . . .	112
1.6	Specifiche di Interfaccia . . . . .	113
1.7	Vincoli di Progetto . . . . .	114
1.8	Gestione del Rischio . . . . .	115
2	Glossario . . . . .	115
2.1	Nota Introduttiva . . . . .	116
2.2	Glossario Tecnico . . . . .	116
3	Visione . . . . .	119
3.1	Nota Introduttiva . . . . .	119
	<b>Understand</b>	<b>123</b>
1	Analisi Generale . . . . .	123
2	Metrics Treemaps . . . . .	123
3	Metriche Generali . . . . .	125
4	Metriche del file: VolunteerRequestMatchingService . . . . .	125
5	Conclusione . . . . .	126

# Introduzione

Il progetto DoIT rappresenta una piattaforma innovativa dedicata alla gestione delle richieste e offerte di volontariato. L'obiettivo principale è creare un ponte digitale tra volontari e organizzazioni, semplificando il processo di incontro tra domanda e offerta di supporto. Questa iniziativa si pone l'ambizione di migliorare la partecipazione al volontariato e di ottimizzare l'uso delle risorse, offrendo un ambiente digitale sicuro, intuitivo e ricco di funzionalità avanzate. Con un focus iniziale sulla città di Milano e un'espansione futura alla Lombardia, DoIT si ispira a soluzioni già esistenti, ma introduce nuove caratteristiche tecniche e un approccio adattato alle esigenze locali. Il documento esplora i vari aspetti della piattaforma, tra cui **Requisiti Funzionali** e **Non Funzionali**, **Casi d'Uso Testuali**, **Diagrammi** di vari tipo tra i quali:

- **Diagrammi dei Casi d'Uso**
- **Modello delle Classi Concettuali**
- **Diagrammi di Sequenza di Sistema**
- **Diagrammi di Interazione, Attività e di Macchina a Stati**
- **Diagrammi delle Classi di Progetto**
- **Diagramma dell'Architettura Logica**

e **Contratti**. Inoltre sono presenti i documenti supplementari della disciplina dei **Requisiti** che offrono informazioni più dettagliate, sono quindi presenti la **Specifica Supplementare**, la **Visione** e il **Glossario**. Questo approccio sistematico garantisce un'**Analisi** e una **Progettazione** chiara e orientata all'utente, con particolare attenzione a sicurezza, accessibilità e usabilità. Ogni documento presenta la seguente nota introduttiva:

- **Assigned** Nome Cognome
- **Blocked by** Disciplina, Task oppure insieme di Task
- **Disciplina** Disciplina
- **Due** giorno Mese, anno ora AM/PM (GMT+1) → giorno Mese, anno ora AM/PM (GMT+1)
- **Is blocking** Disciplina, Task oppure insieme di Task

- **Parent-task** Task
- **Priority** Alta/Media oppure Bassa
- **Projects** L'iterazione dell'artefatto, ad esempio DoIT-1 oppure DoIT-2

Assigned negli **Artefatti** è il nome e cognome della persona a cui è assegnato l'**Artefatto**, al contrario nei **Diagrammi delle Classi Software di Progetto** Assigned è il nome e cognome delle persone che hanno lavorato al codice di quella parte del progetto. Tutti i **Diagrammi delle Classi** sono stati realizzati con il sito **Lucidcharts** e i diagrammi di: **Sequenza di Sistema, d'Interazione, d'Attività e di Macchina a Stati** sono stati realizzati con il linguaggio di **MarkDown Mermeid**. Ogni iterazione è divisa in **Diagramma di Gantt**, documenti della disciplina dei **Requisiti**, documenti della disciplina della **Progettazione**, i **Pattern** utilizzati e infine una piccola analisi con **Understand**.

## Compilazione

Per compilare il progetto sono necessari i seguenti passaggi:

1. **E' necessario installare npm, nextJS e react:** per farlo è necessario eseguire il comando **npm install -g next react**.
2. **npm install:** per installare le dipendenze necessarie creare la build del progetto.
3. **npm run dev:** per avviare il server Frontend, è necessario entrare nella cartella: `/frontend/doit/`.
4. **'mvn clean install':** per avviare il server backend, è necessario entrare nella cartella: `/backend/doit/`.
5. **'mvn spring-boot:run':** per avviare il server backend, è necessario entrare nella cartella: `/backend/doit/`.

## Iterazione 1

### 1 Diagramma di Gantt

Questi sono i diagrammi di Gantt per l'iterazione 2.

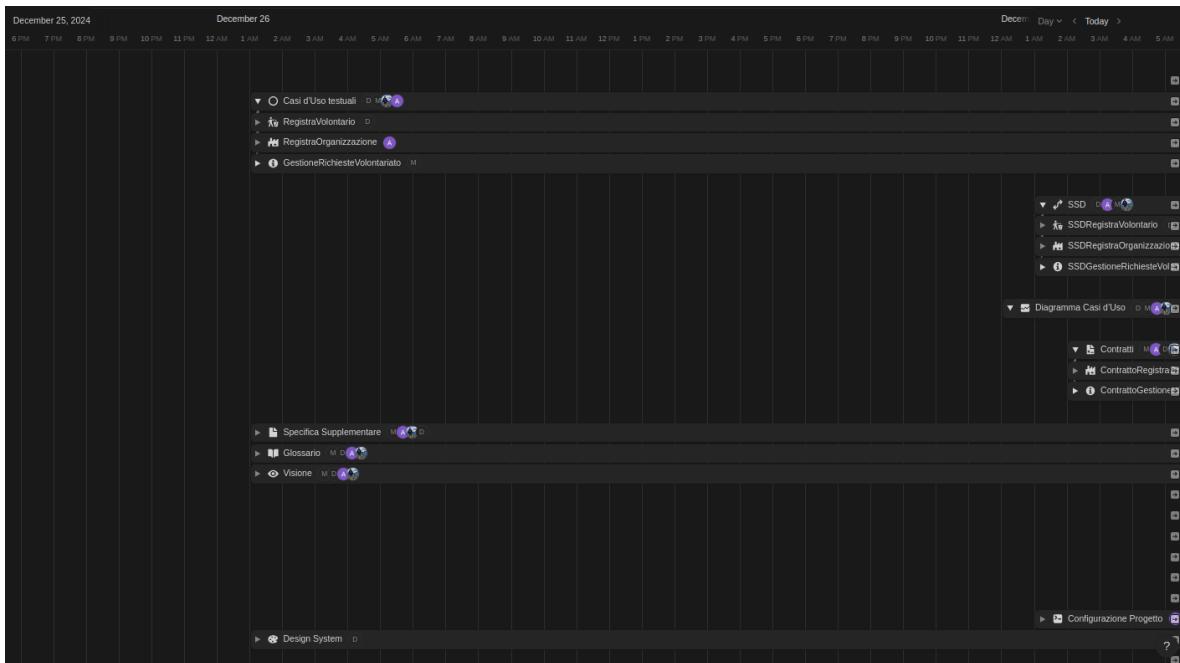


Figure 1: Diagramma di Gantt 1

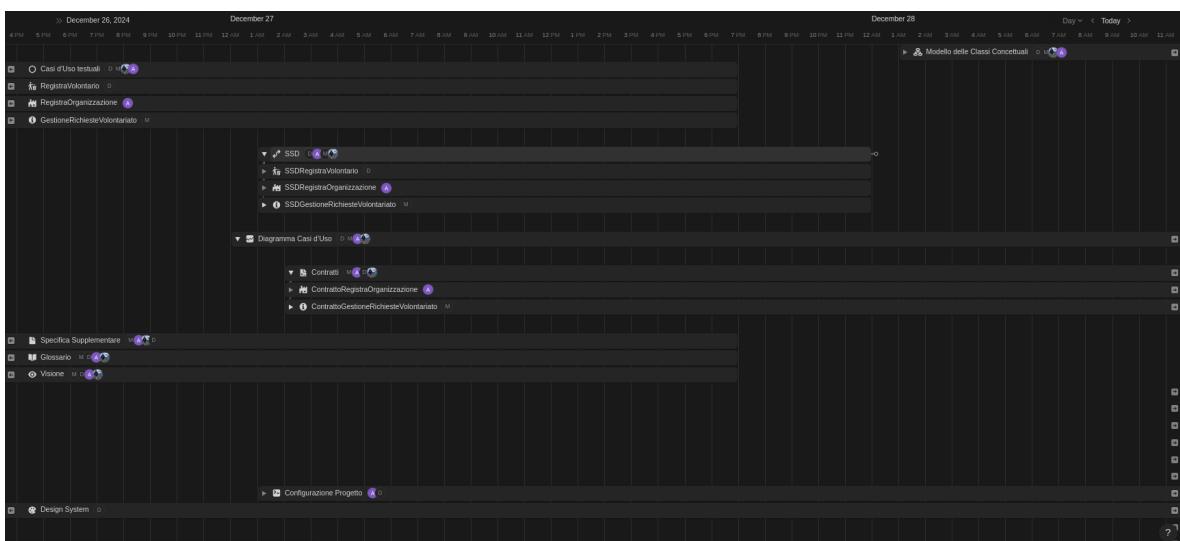


Figure 2: Diagramma di Gantt 2

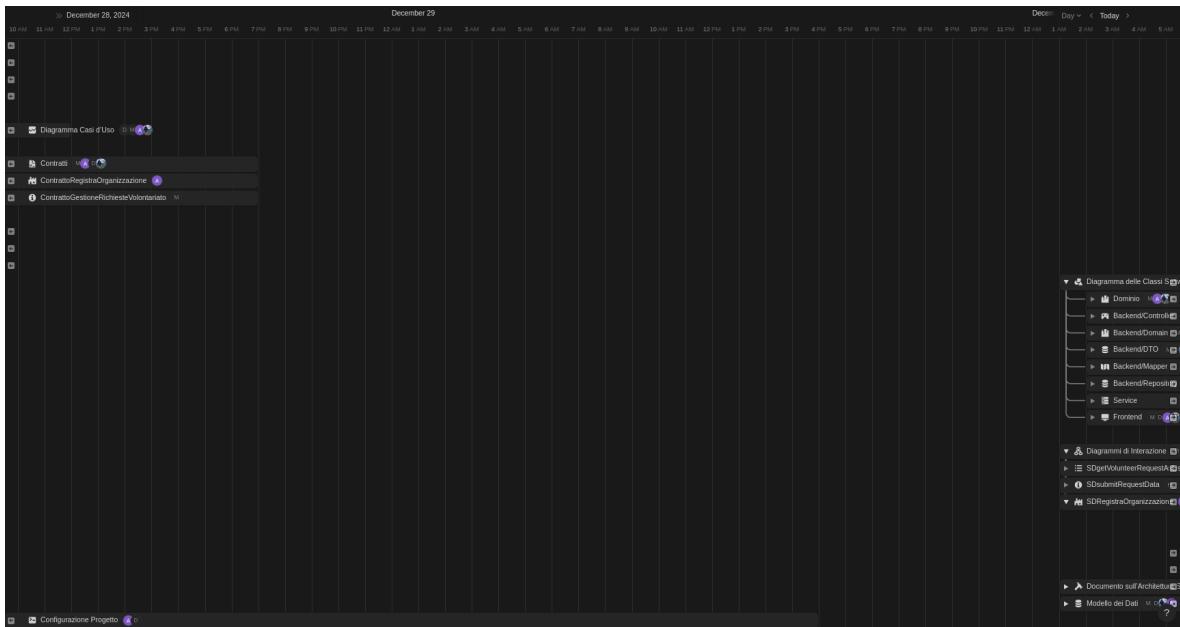


Figure 3: Diagramma di Gantt 3

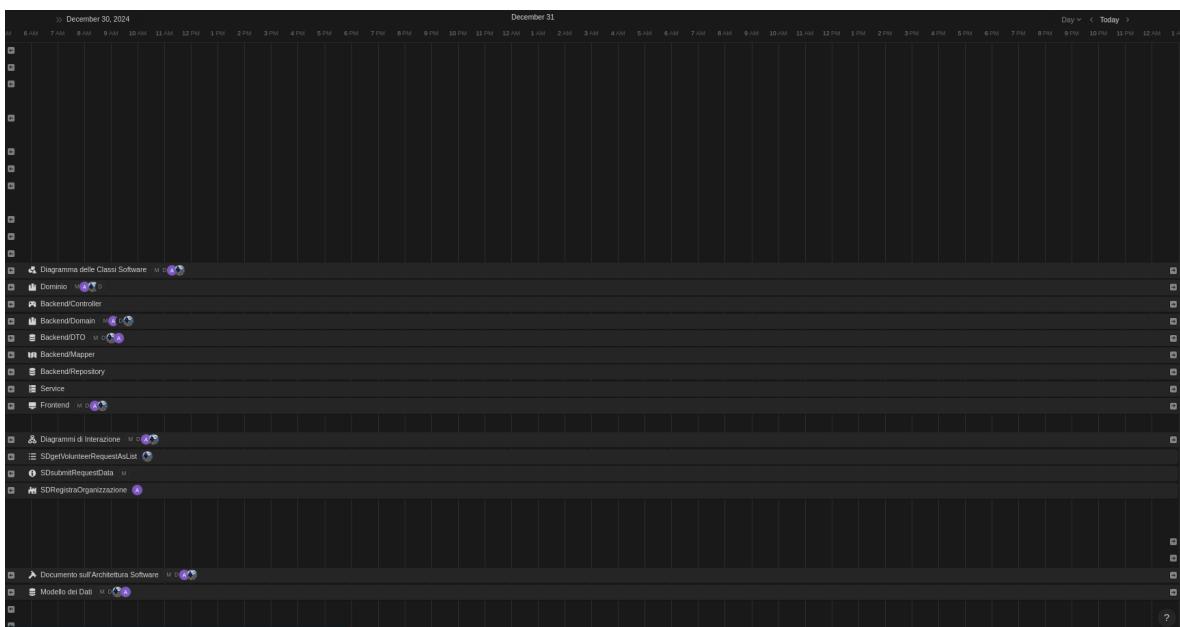


Figure 4: Diagramma di Gantt 4

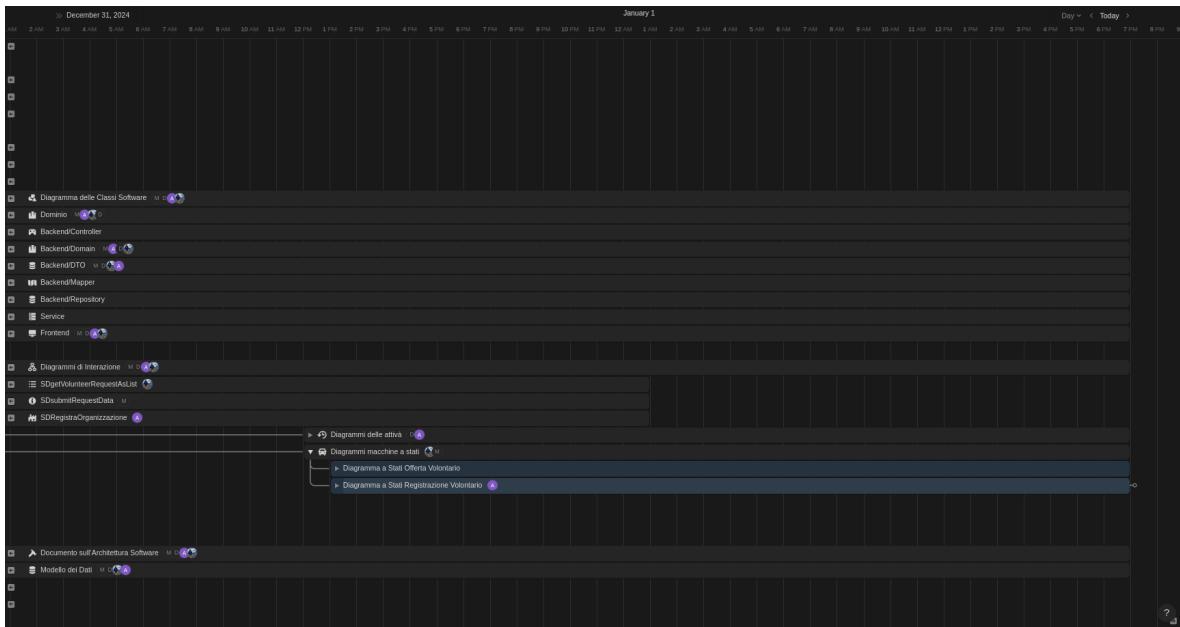


Figure 5: Diagramma di Gantt 5

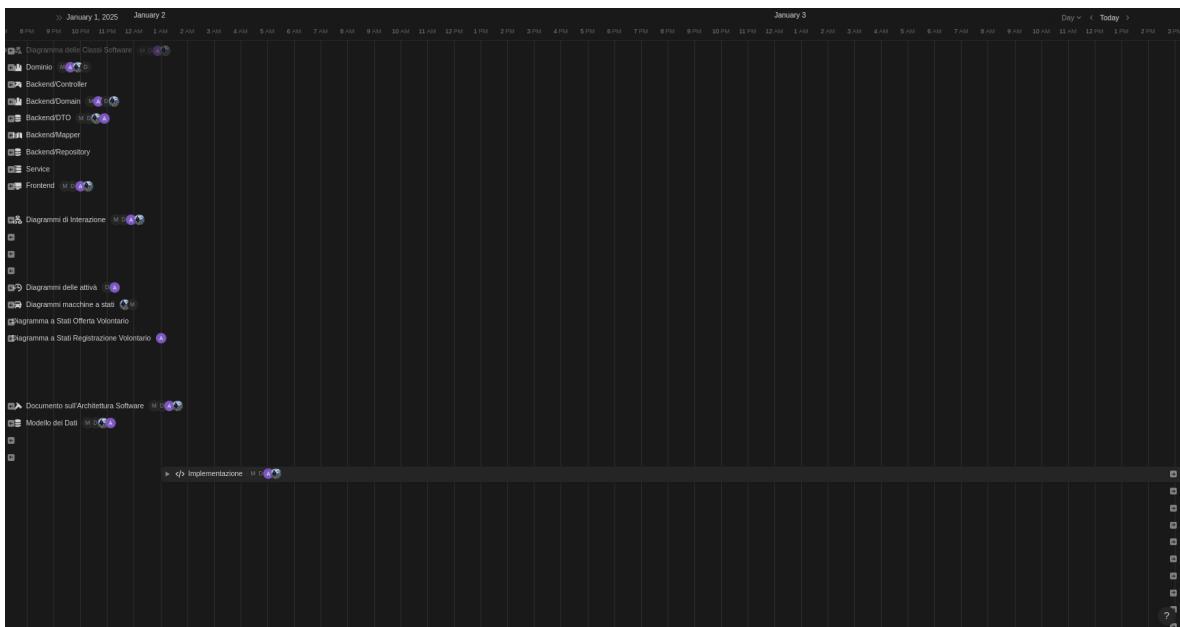


Figure 6: Diagramma di Gantt 6

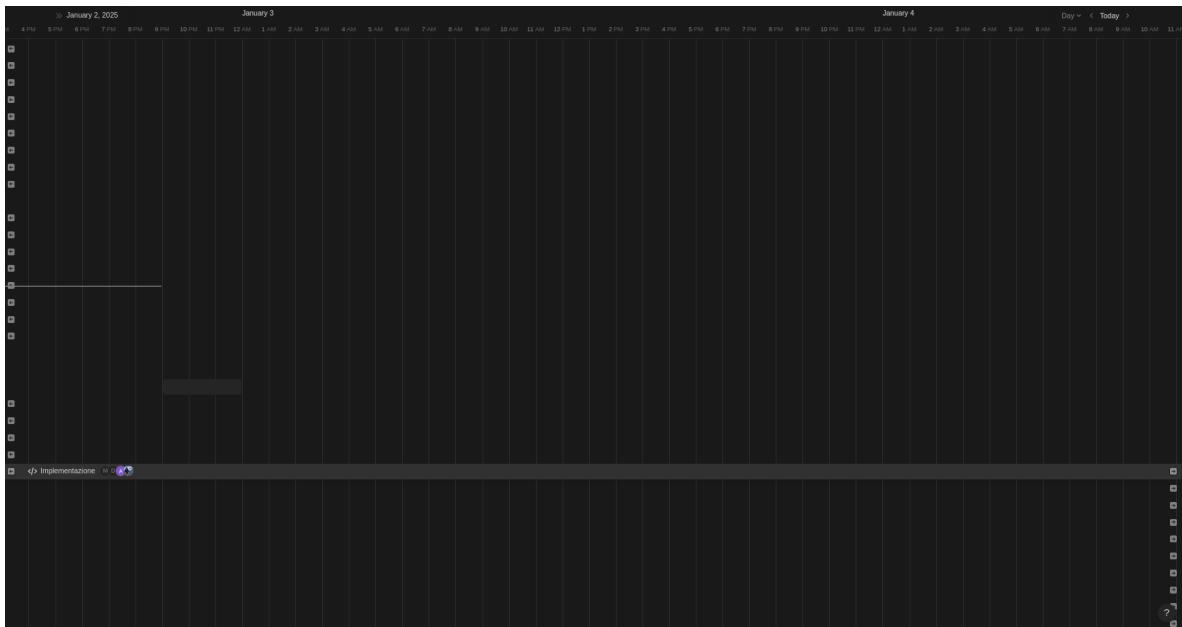


Figure 7: Diagramma di Gantt 7

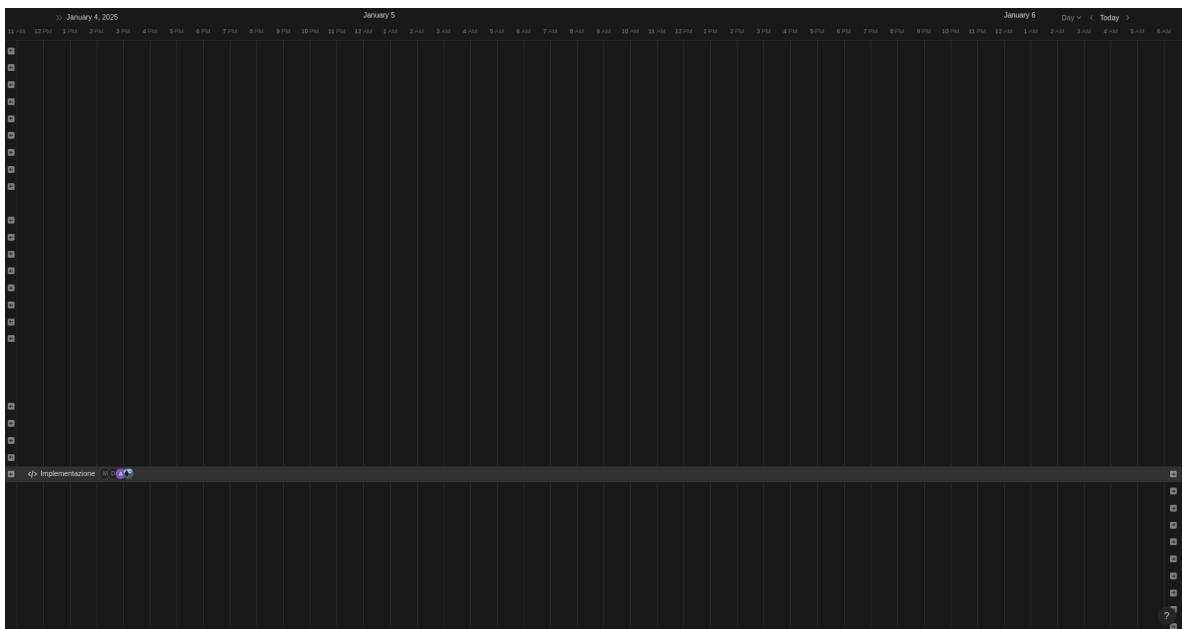


Figure 8: Diagramma di Gantt 8

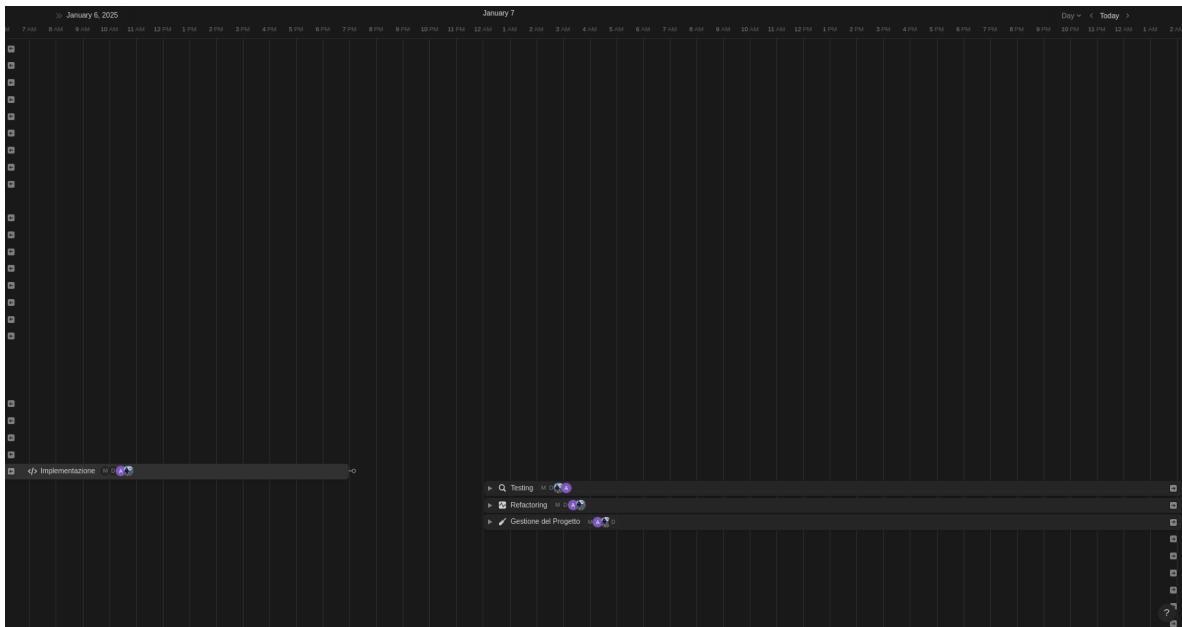


Figure 9: Diagramma di Gantt 9

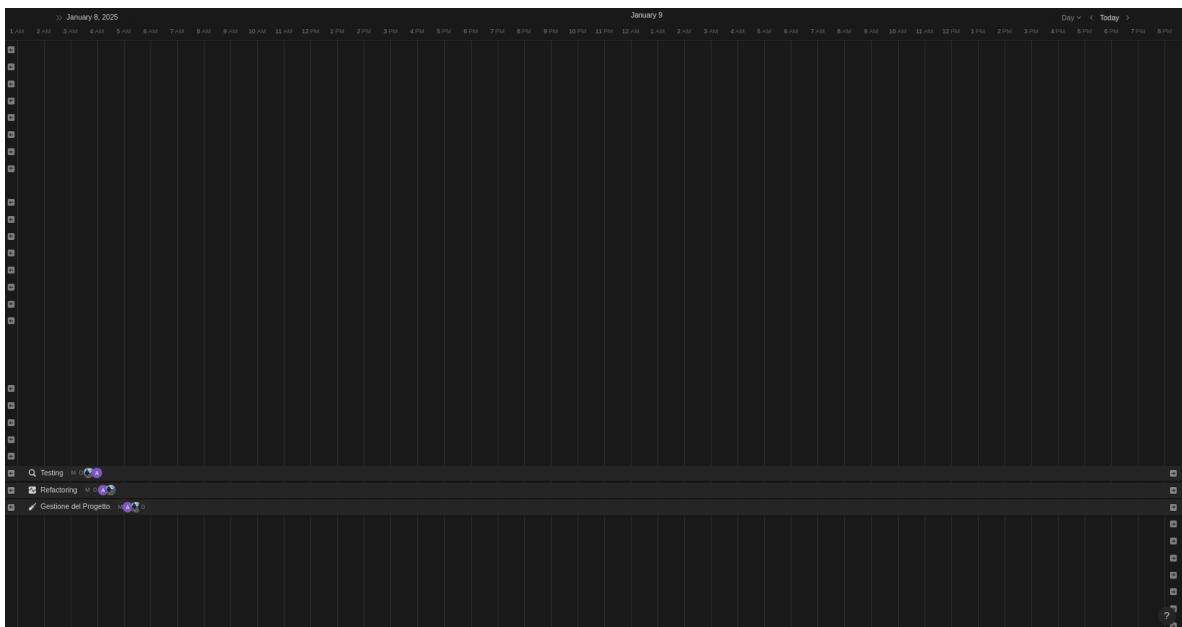


Figure 10: Diagramma di Gantt 10

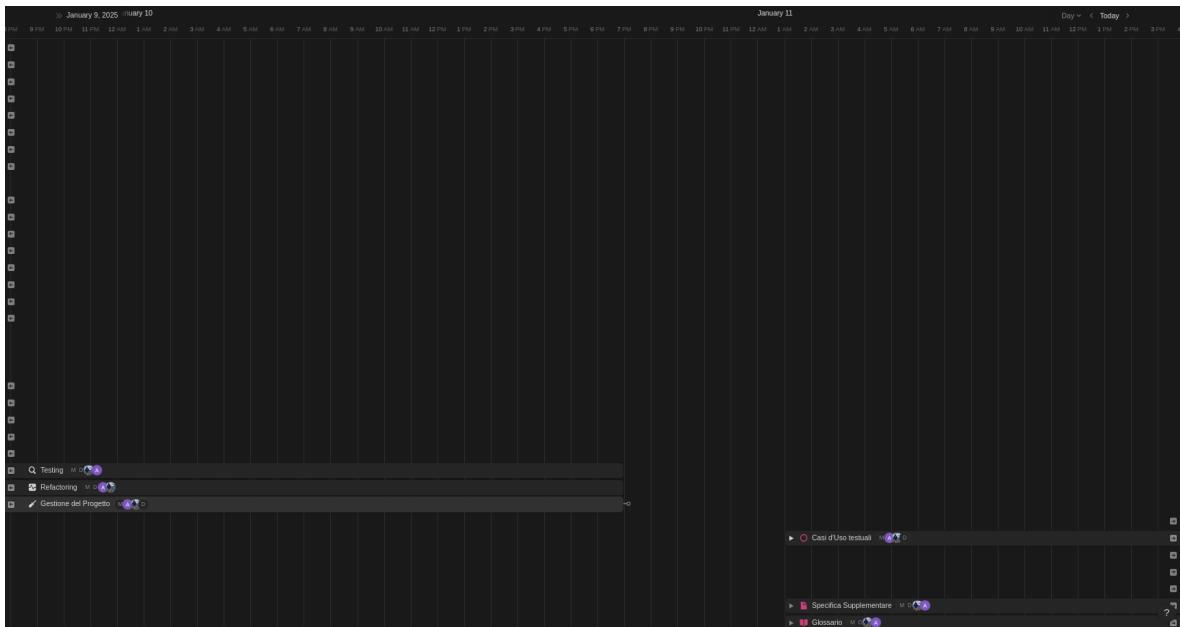


Figure 11: Diagramma di Gantt 11

## 2 Requisiti

### 1 Casi d'uso

#### 1.1 Caso d'uso UC1: RegistraVolontario

RegistraVolontario Daniele Buser Scadenza: 27 Dicembre 2024

#### Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Is blocking:** RicercaAbbinata, SSDRegistraVolontario
- **Parent-task:** Casi d'Uso testuali
- **Priority:** Alta

- **Projects:** DoIT-1

Il caso d'uso **"Registrazione Nuovo Volontario"** (UC-001) descrive il processo attraverso il quale un nuovo utente può registrarsi come volontario nella piattaforma. Questo processo è considerato di alta priorità essendo una funzionalità core del sistema, con una frequenza prevista di oltre 100 registrazioni mensili.

L'utente inizia il processo dalla schermata iniziale dell'applicazione, dove deve scegliere se accedere come volontario o come organizzazione. Dopo aver selezionato "Volontario", il sistema reindirizza l'utente all'autenticazione Google, un requisito fondamentale per la registrazione.

Una volta fornite le credenziali Google, il sistema verifica automaticamente se esiste già un account associato a quell'utente nel database. Se l'utente è già registrato, viene effettuato automaticamente il login e mostrata la dashboard personale. Se invece l'utente non è presente nel sistema, viene mostrato un form di registrazione specifico per i volontari.

Il sistema importa automaticamente le informazioni base come nome, email e foto profilo dall'account Google e le precompila nel form. L'utente deve quindi fornire informazioni aggiuntive come numero di telefono, data di nascita, città e aree di interesse. Durante la compilazione, il sistema effettua validazioni in tempo reale. Se ci sono campi obbligatori mancanti o compilati in modo scorretto, il sistema impedisce il completamento della registrazione, mostrando messaggi informativi specifici.

Una volta che tutti i dati sono stati inseriti correttamente, l'utente può confermare la registrazione. Il sistema crea l'account e effettua automaticamente il login, reindirizzando l'utente alla dashboard personale dove trova opportunità consigliate e un riepilogo del proprio profilo.

Il processo è stato progettato considerando requisiti non funzionali importanti: la registrazione deve essere completabile in massimo 5 minuti e tutti i dati sensibili devono essere crittografati. Dal punto di vista tecnico, il sistema implementa l'integrazione OAuth con Google, validazione real-time dei dati inseriti e garantisce uno storage sicuro dei dati personali. Per monitorare il corretto funzionamento, il sistema mantiene un logging dettagliato degli errori.

Al termine del processo, il sistema garantisce tre post-condizioni essenziali: la creazione di un nuovo account volontario, l'autenticazione dell'utente nel sistema e la verifica della completezza del profilo.

#### **Informazioni Generali ID Caso d'Uso: UC-001**

**Nome:** Registrazione Nuovo Volontario

**Attore Principale:** Utente non registrato

**Precondizioni:**

- Utente non registrato nel sistema come volontario
- Utente possiede un account Google
- Dispositivo con connessione internet

**Post-condizioni:**

- Nuovo account volontario creato
- Utente autenticato nel sistema
- Profilo completo e verificato

**Priorità:** Alta

### Scenario Principale

**Obiettivo** Permettere a un nuovo utente di registrarsi come volontario nella piattaforma utilizzando il proprio account Google e fornendo le informazioni necessarie.

#### Flusso Base

1. Utente apre l'applicazione
  - Sistema: Mostra schermata iniziale
2. Utente seleziona "Volontario"
  - Sistema: Reindirizza all'autenticazione Google
3. Utente esegue autenticazione con Google
  - Sistema: Verifica esistenza account nel database
4. Se l'utente non esiste:
  - Sistema: Mostra form registrazione volontario con:
    - Intervallo di date per la disponibilità
    - Città
    - Aree di interesse
    - Descrizione volontario

- Sistema: Precompila i dati importati da Google (nome, email, foto)
5. Utente compila e invia form
    - Sistema: Salva nuovo account volontario
    - Sistema: Effettua login automatico
  6. Sistema mostra dashboard personale
    - Opportunità consigliate
    - Riepilogo profilo

## Scenari Alternativi

### Scenario: Utente già registrato

1. Dopo l'autenticazione Google, sistema rileva account esistente
2. Sistema effettua login automatico
3. Sistema mostra dashboard personale

## Eccezioni e Gestione Errori

### Errore: Form incompleto

- **Condizione:** Campi obbligatori mancanti
- **Azione:** Impossibilità a completare registrazione
- **Risultato:** Messaggio informativo su campi mancanti o scorretti

### Errore: Autenticazione Google fallita

- **Condizione:** Problema con l'autenticazione Google
- **Azione:** Interruzione del processo
- **Risultato:** Messaggio di errore e possibilità di riprovare

## Requisiti Speciali

### **Requisiti Non Funzionali**

- Performance: Registrazione completabile in max 5 minuti
- Sicurezza: Crittografia dati sensibili

### **Vincoli Tecnici**

- Integrazione OAuth Google
- Validazione real-time
- Storage sicuro dati personali

### **Note Aggiuntive**

- **Frequenza:** Alta (previsione 100+ registrazioni/mese)
- **Criticità:** Alta (funzionalità core del sistema)
- **Note implementative:**
  - Logging dettagliato errori
  - Verifica automatica formato dati inseriti
  - Backup immediato dei dati di registrazione

## **1.2 Caso d'uso UC2: RegistraOrganizzazione**

RegistraOrganizzazione Andrea Cozzi Scadenza: 27 Dicembre 2024

### **Nota Introduttiva**

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Is blocking:** RicercaAbbinata, SSDRegistraOrganizzazione
- **Parent-task:** Casi d'Uso testuali
- **Priority:** Alta

- **Projects:** DoIT-1

Il caso d'uso **"Registrazione Nuova Organizzazione"** (UC-002) descrive il processo attraverso il quale una nuova organizzazione può registrarsi nella piattaforma. Questo processo è considerato di alta priorità essendo una funzionalità core del sistema, con una frequenza prevista di 20-30 registrazioni mensili.

L'organizzazione inizia il processo dalla schermata iniziale dell'applicazione, dove deve scegliere se accedere come volontario o come organizzazione. Dopo aver selezionato "Organizzazione", il sistema reindirizza all'autenticazione Google, un requisito fondamentale per la registrazione.

Una volta fornite le credenziali Google, il sistema verifica automaticamente se esiste già un account associato nel database. Se l'organizzazione è già registrata, viene effettuato automaticamente il login e mostrata la dashboard organizzazione. Se invece l'organizzazione non è presente nel sistema, viene mostrato un form di registrazione specifico.

Il sistema importa automaticamente le informazioni base come nome, email e foto profilo dall'account Google e le precompila nel form. L'organizzazione deve quindi fornire informazioni aggiuntive come nome organizzazione, indirizzo, partita IVA, numero di telefono, descrizione, tipo di organizzazione e sito web (opzionale). Durante la compilazione, il sistema effettua validazioni in tempo reale. Se ci sono campi obbligatori mancanti o compilati in modo scorretto, il sistema impedisce il completamento della registrazione, mostrando messaggi informativi specifici.

#### **Informazioni Generali ID Caso d'Uso:** UC-002

**Nome:** Registrazione Nuova Organizzazione

**Attore Principale:** Organizzazione

**Precondizioni:**

- Organizzazione non registrata nel sistema
- Possesso di un account Google
- Dispositivo con connessione internet

**Post-condizioni:**

- Nuovo account organizzazione creato
- Organizzazione autenticata nel sistema
- Profilo completo e verificato

**Priorità:** Alta

## Scenario Principale

**Obiettivo** Permettere a una nuova organizzazione di registrarsi nella piattaforma utilizzando un account Google e fornendo le informazioni necessarie.

### Flusso Base

1. Organizzazione apre l'applicazione
  - Sistema: Mostra schermata iniziale
2. Organizzazione seleziona "Organizzazione"
  - Sistema: Reindirizza all'autenticazione Google
3. Organizzazione esegue autenticazione con Google
  - Sistema: Verifica esistenza account nel database
4. Se l'organizzazione non esiste:
  - Sistema: Mostra form registrazione organizzazione con:
    - Nome organizzazione
    - Indirizzo
    - Tipo di organizzazione
    - Descrizione
    - Partita IVA (opzionale)
    - Sito web (opzionale)
  - Sistema: Precompila i dati importati da Google (email, foto)
5. Organizzazione compila e invia form
  - Sistema: Salva nuovo account organizzazione
  - Sistema: Effettua login automatico
6. Sistema mostra dashboard organizzazione
  - Gestione offerte di volontariato
  - Riepilogo profilo

## Scenari Alternativi

### Scenario: Organizzazione già registrata

1. Dopo l'autenticazione Google, sistema rileva account esistente
2. Sistema effettua login automatico
3. Sistema mostra dashboard organizzazione

## Eccezioni e Gestione Errori

### Errore: Form incompleto

- **Condizione:** Campi obbligatori mancanti
- **Azione:** Impossibilità a completare registrazione
- **Risultato:** Messaggio informativo su campi mancanti o scorretti

### Errore: Autenticazione Google fallita

- **Condizione:** Problema con l'autenticazione Google
- **Azione:** Interruzione del processo
- **Risultato:** Messaggio di errore e possibilità di riprovare

## Requisiti Speciali

### Requisiti Non Funzionali

- Performance: Registrazione completabile in max 5 minuti
- Sicurezza: Crittografia dati sensibili
- Disponibilità: Sistema accessibile 99.9% del tempo

## Vincoli Tecnici

- Integrazione OAuth Google
- Validazione real-time
- Storage sicuro dati organizzazione
- Gestione upload documenti

## Note Aggiuntive

- **Frequenza:** Media (20-30 registrazioni/mese)
- **Criticità:** Alta (funzionalità core del sistema)
- **Note implementative:**
  - Logging dettagliato errori
  - Sistema di notifica per amministratori
  - Backup automatico dati organizzazione
  - Verifica automatica formato dati inseriti

### 1.3 Caso d'uso UC3: GestioneRichiesteVolontariato

GestioneRichiesteVolontariato Matteo Cervini Due: 27 Dicembre 2024

#### Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Is blocking:** SSDGestioneRichiesteVolontariato, SSDControlloCorrettezzaCampi
- **Parent-task:** Casi d'Uso testuali
- **Priority:** Alta
- **Projects:** DoIT-1

Il caso d'uso **”GestioneRichiesteVolontariato”** (**UC-003**) descrive l'interazione che c'è tra l'Organizzazione e il Sistema quando questa vuole inserire un'inserzione di una richiesta di volontariato.

L'Incaricato inizia il processo entrando nella schermata di autenticazione a meno che questa non sia già stata fatta.

L'incaricato una volta effettuato l'autenticazione deve poter inserire nel Sistema la proposta di volontariato e deve comunicare la data, l'orario, il luogo, la categoria, il tipo d'aiuto e infine una descrizione dettagliata di cosa sia richiesto al Volontario.

Una volta inseriti tutti i campi l'Incaricato può procedere all'invio della richiesta. Il Caso d'Uso **“ControlloCorrettezzaCampi”** comunica come viene gestito il flusso alternativo nel caso in cui uno o più campi non siano validi una volta che è stata confermato l'invio della richiesta verso il Sistema.

La richiesta una volta inviata non viene istantaneamente pubblicata sul Sistema, viene momentaneamente bloccata la sua pubblicazione e viene inviata tramite ticket al Comitato di Validazione e Controllo che si deve occupare di accettare o no la richiesta di Volontariato.

Quando la richiesta è stata accettata allora verrà confermata la pubblicazione pubblica sul Sistema.

**Informazioni Generali** **ID Caso d'Uso:** UC-003 **Nome:** GestioneRichiesteVolontario **Attore Principale:** Organizzazione

### **Pre-condizioni**

1. L'Organizzazione che intende effettuare l'inserzione possiede i prerequisiti per poter pubblicare un'inserzione
2. L'Organizzazione è registrata nel Sistema

### **Post-condizioni**

1. L'inserzione ha tutti i campi correttamente riempiti
2. L'inserzione viene memorizzata nel Sistema
3. L'inserzione viene inoltrata al comitato di supervisione
4. L'inserzione è accettata e viene pubblicata pubblicamente nel Sistema

### **Scenario Principale**

**Obiettivo** Un'Organizzazione deve poter pubblicare un'inserzione.

### **Flusso Base**

1. L'Organizzazione effettua l'autenticazione
2. L'Organizzazione va nell'apposita schermata di inserimento richieste
3. Il Sistema mostra l'apposita schermata d'inserimento richieste
4. In tale pagina l'organizzazione riempie tutti gli opportuni campi del form:
  - (a) Data
  - (b) Orario
  - (c) Luogo
  - (d) Categoria
  - (e) Tipologia di attività
  - (f) Descrizione dettagliata di cosa è richiesto ai volontari
  - (g) Capacità massima dei Volontari che si possono candidare
5. L'Organizzazione conferma l'inserimento della richiesta
6. Il Sistema memorizza correttamente la richiesta
7. Il Sistema inoltra un ticket al Comitato di Controllo e Validazione per validare la richiesta appena memorizzata

*L'incaricato ripete i punti 2-6 fino a che non indica di aver terminato*

In un secondo momento, non appena il Comitato di Controllo e Validazione ha confermato la richiesta:

1. Questa viene ufficialmente salvata nel Sistema
2. Viene mostrata nell'apposita schermata tramite annuncio o tramite mappa
3. Un qualsiasi volontario che rispetta i pre-requisiti può inoltrare un'offerta di aiuto

### **Scenari Alternativi**

### **Scenario: Connessione non disponibile**

1. Al punto 1 chi ha necessità di inserire un'inserzione deve effettuare l'accesso:
  - (a) L'Organizzazione è già esistente: Ci si riporta al Caso d'Uso: "RegistraVolontario"
  - (b) L'Organizzazione non è già esistente:
    - i. Il Sistema non permette l'inserimento di inserzione senza autenticazione
    - ii. Un opportuno bottone per uscire dalla pagina di autenticazione è reso disponibile
    - iii. L'incaricato preme il bottone
    - iv. Il Caso d'Uso termina
2. Al punto 5 è presente qualche inesattezza nei form: si rimanda al Caso d'Uso: "ControlloCorrettezzaCampi"
3. Al punto 8 in caso di rifiuto la gestione è rimandata all'opportuno Caso d'Uso non ancora modellato

### **Eccezioni e Gestione Errori**

**Errore: Form incompleto Condizione:** il form riporta irregolarità **Azione:** si attiva il Caso d'Uso: "ControlloCorrettezzaCampi" **Risultato:** il Sistema si aspetta che l'utilizzatore sistemi le relative irregolarità, il Sistema quindi non procederà col Caso d'Uso finché queste non saranno sistematiche

### **Requisiti Speciali**

#### **Requisiti Non Funzionali**

- **Usabilità:** L'interfaccia deve essere intuitiva e accessibile
- **Privacy:** I dati personali del volontario sono condivisi con l'Organizzazione solo dopo l'accettazione dell'offerta

### **Note Aggiuntive**

- **Frequenza:** in base alle necessità, potenzialmente ininterrotto
- **Criticità:** mancanza di connessione

## 2 Diagramma dei Casi d'Uso

Diagramma dei Casi d'Uso Matteo Cervini Scadenza: 29 Dicembre 2024

- **Disciplina:** Requisiti
- **Scadenza:** 27 Dicembre 2024 12:00 AM (GMT+1) → 28 Dicembre 2024 12:00 PM (GMT+1)
- **Priorità:** Bassa
- **Progetto:** DoIT-1

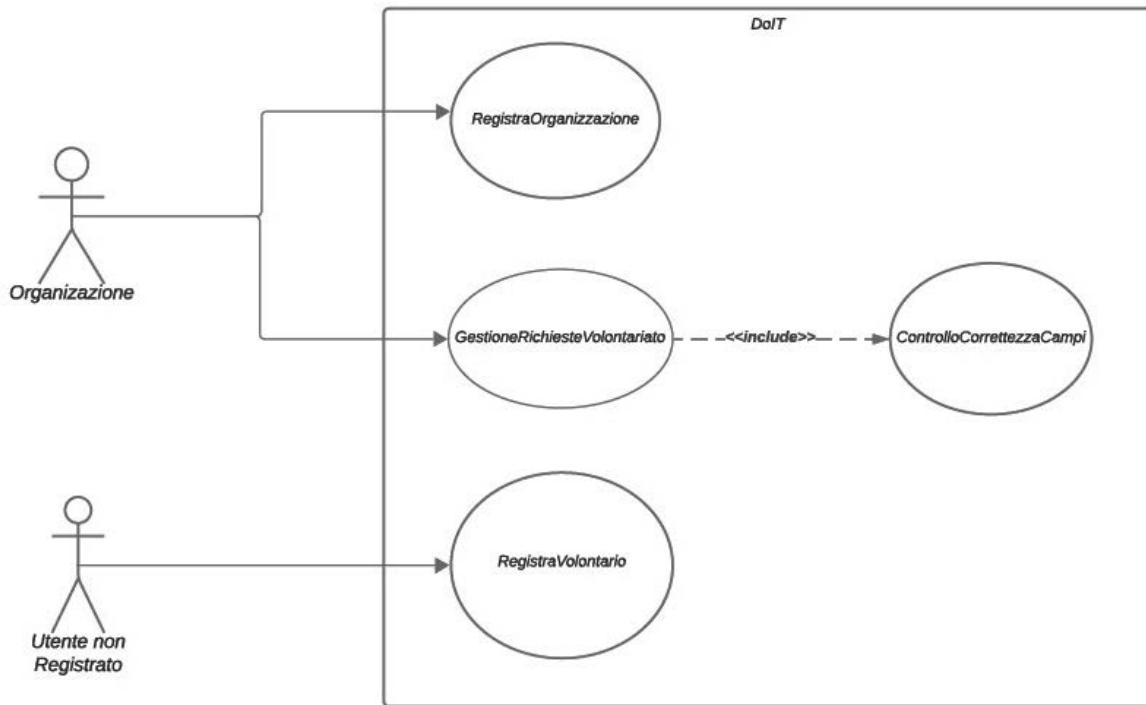


Figure 12: Diagramma dei Casi d'Uso

## 3 Diagramma di Sequenza di Sistema

### 3.1 Diagramma di Sequenza di Sistema SSD1: RegistraVolontario

SSDRegistraVolontario Daniele Buser Scadenza: 28 Dicembre 2024

- **Blocked by:** RegistraVolontario
- **Disciplina:** Requisiti
- **Scadenza:** 27 Dicembre 2024 1:00 AM (GMT+1) → 28 Dicembre 2024 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Is Blocking:** SSDRicercaAbbinata
- **Priorità:** Alta
- **Progetto:** DoIT-1

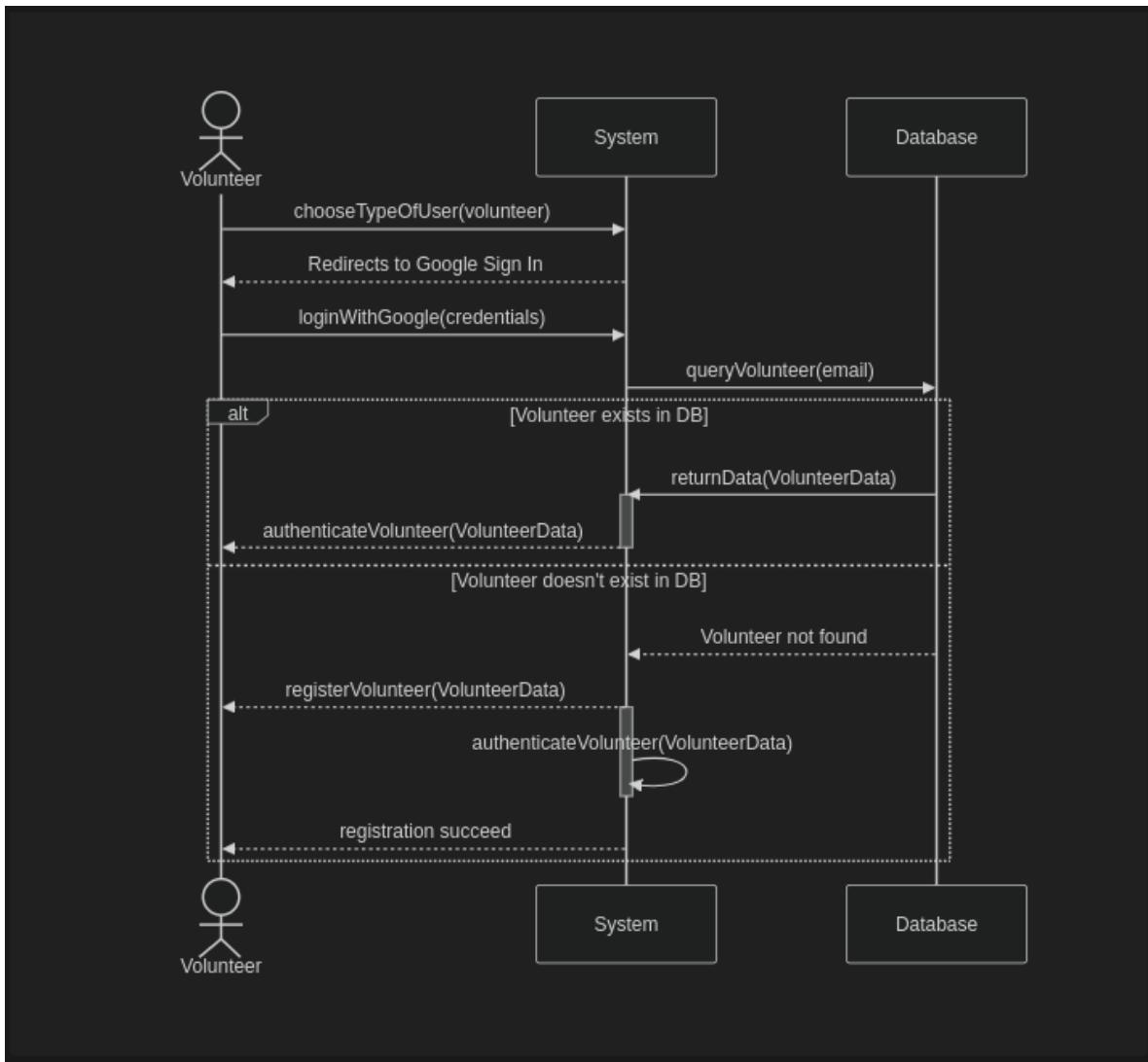


Figure 13: Diagramma di Sequenza di Sistema

### 3.2 Diagramma di Sequenza di Sistema SSD2: RegistraOrganizzazione

SSDRigaOrganizzazione Andrea Cozzi Scadenza: 28 Dicembre 2024

- **Blocked by:** RegistraOrganizzazione
- **Disciplina:** Requisiti

- **Scadenza:** 27 Dicembre 2024 1:00 AM (GMT+1) → 28 Dicembre 2024 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Is Blocking:** SSDRicercaAbbinata
- **Priorità:** Media
- **Progetto:** DoIT-1

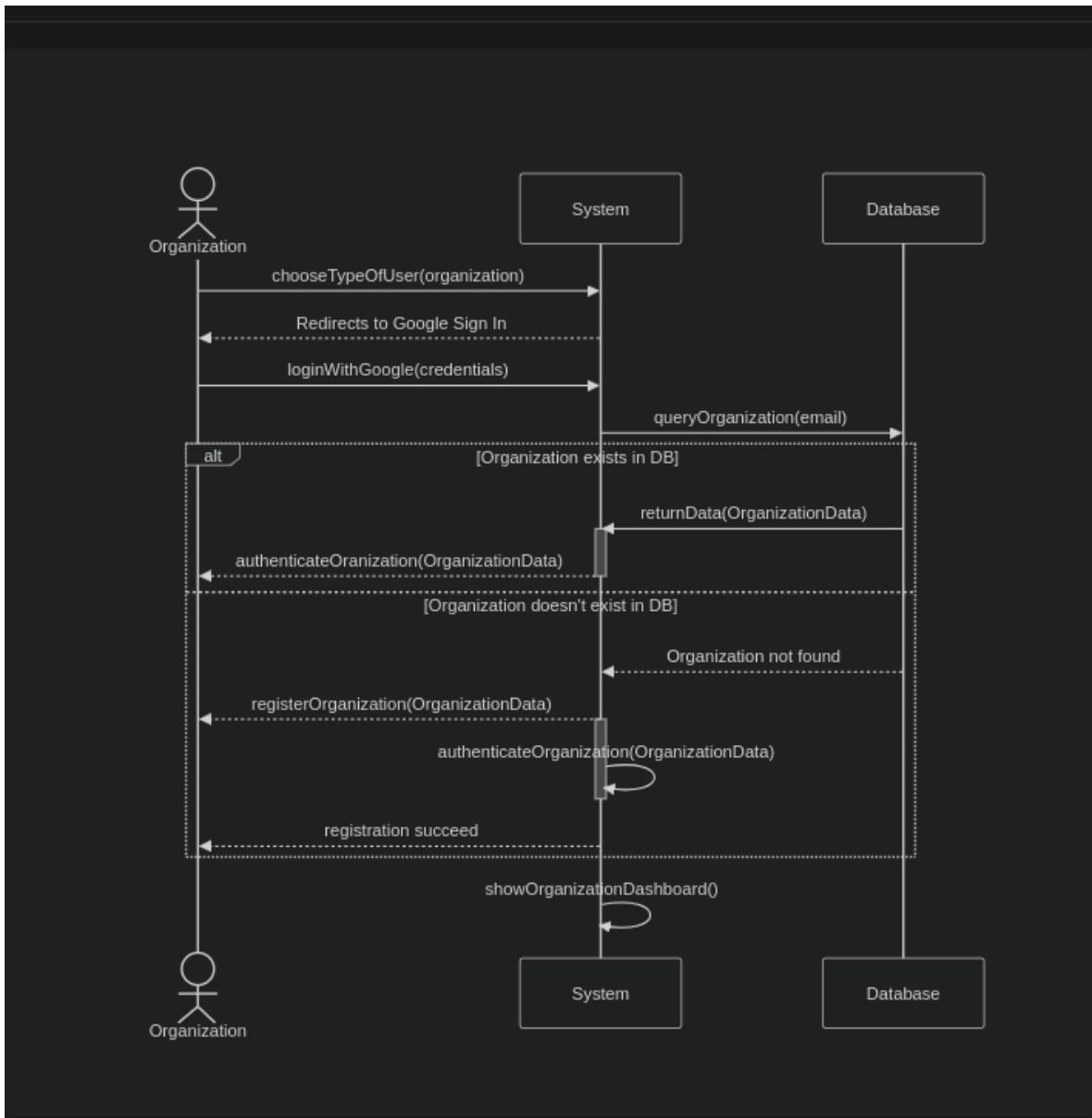


Figure 14: Diagramma di Sequenza di Sistema

### 3.3 Diagramma di Sequenza di Sistema SSD3: GestioneRichiesteVolontariato

SDDGestioneRichiesteVolontariato Matteo Cervini Scadenza: 28 Dicembre 2024

- **Blocked by:** GestioneRichiesteVolontariato ControlloCorrettezzaCampi
- **Disciplina:** Requisiti
- **Scadenza:** 27 Dicembre 2024 1:00 AM (GMT+1) → 28 Dicembre 2024 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Is Blocking:** SSDControlloCorrettezzaCampi ContrattoGestioneRichiesteVolontariato
- **Priorità:** Media
- **Progetto:** DoIT-1

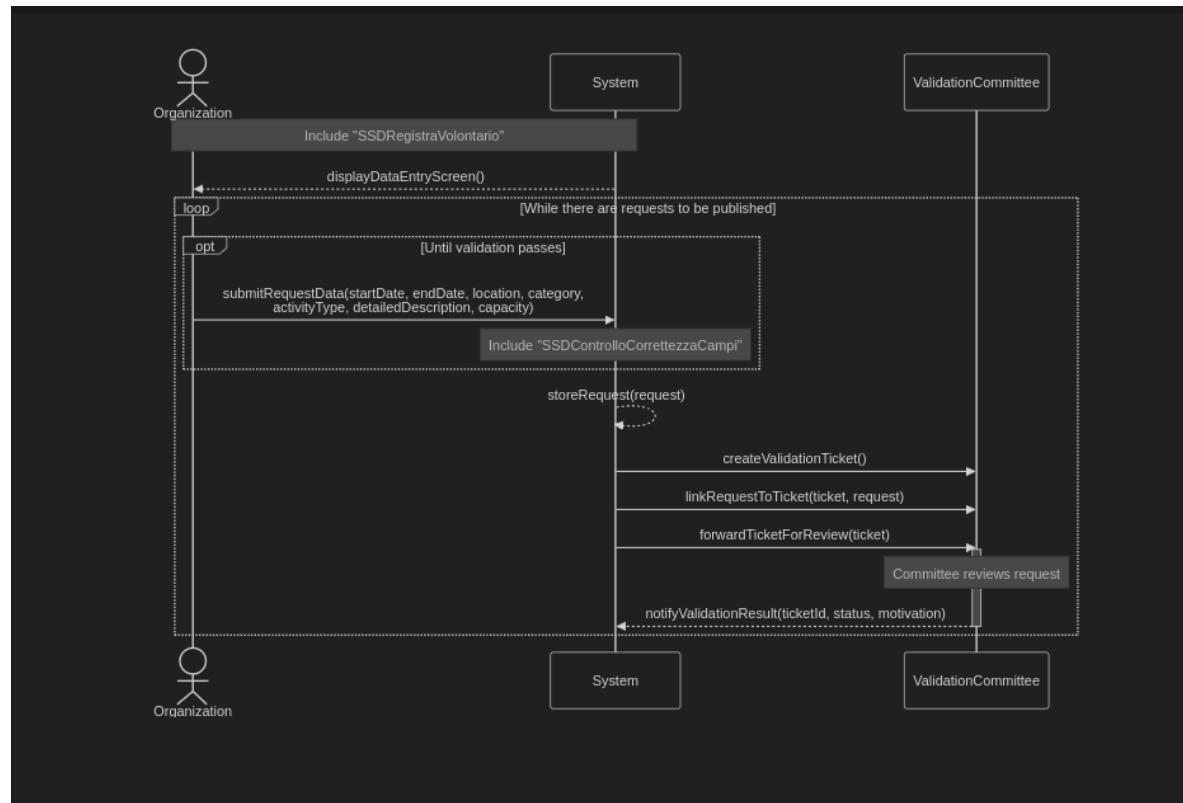


Figure 15: Diagramma di Sequenza di Sistema

### 3.4 Diagramma di Sequenza di Sistema SSD3.2: ControlloCorrettezzaCampi

SSDControlloCorrettezzaCampi Matteo Cervini Scadenza: 28 Dicembre 2024

- **Blocked by:** GestioneRichiesteVolontariato ControlloCorrettezzaCampi
- **Disciplina:** Requisiti
- **Scadenza:** 27 Dicembre 2024 1:00 AM (GMT+1) → 28 Dicembre 2024 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Is Blocking:** SSDRicercaAbbinata ContrattoControlloCorrettezzaCampi
- **Priorità:** Media
- **Progetto:** DoIT-1

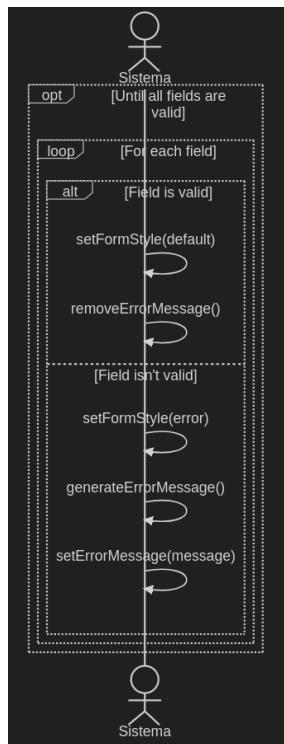


Figure 16: Diagramma di Sequenza di Sistema

## 4 Contratti

### 4.1 Contratto CO01: RegistraOrganizzazione

ContrattoRegistraOrganizzazione Andrea Cozzi Scadenza: 28 Dicembre 2024

#### Nota Introduttiva

- **Assigned:** Andrea Cozzi
- **Disciplina:** Requisiti
- **Due:** 27 Dicembre, 2024 2:00 AM (GMT+1) → 28 Dicembre, 2024 7:00 PM (GMT+1)
- **Parent task:** Contratti
- **Priority:** Bassa
- **Projects:** DoIT-1

**Contratto CO01:** RegistraOrganizzazione

**NomeOperazione**

RegistraOrganizzazione

**RiferimentoCasoD'uso**

RegistrazioneOrganizzazione

**Pre-Condizioni**

1. L'organizzazione deve avere un account Google
2. L'organizzazione non deve essere già registrata nel sistema
3. E' in corso la creazione di una **Organizzazione** *o*

**Post-Condizioni**

1. Viene creata un'istanza *o* di **Organization**
2. Gli attributi di *o* sono stati inizializzati
3. L'istanza *o* è stata autenticata e associata al SuD

## 4.2 Contratto CO02: GestioneRichiesteVolontariato

ContrattoGestioneRichiesteVolontariato Matteo Cervini Scadenza: 28 Dicembre 2024

- **Assigned:** Matteo Cervini
- **Blocked by:** SSDGestioneRichiesteVolontariato
- **Disciplina:** Requisiti
- **Due:** 27 Dicembre, 2024 2:00 AM (GMT+1) → 28 Dicembre, 2024 7:00 PM (GMT+1)
- **Parent-task:** Contratti
- **Priority:** Bassa
- **Projects:** DoIT-1
- **Sub-tasks:** ContrattoControlloCorrettezzaCampo

### Contratto CO002: CreaRichiestaVolontariato

#### Operazione

```
submitRequestData(date: Date, time: TimeStamp, location: Location,  
category: Category, activityType: activityType,  
detailedDescription: detailedRequestDescription)
```

#### Riferimenti

Casi d'uso: GestioneRichiesteVolontariato

#### Pre-Condizioni

- Tutti i campi sono validi
- E' in corso la creazione di una **Request**  $r$

#### Post-Condizioni

- E' stata creata una nuova istanza  $r$  di **Request**
- Gli attributi di  $r$  sono stati inizializzati
- $r$  è stata associata al SuD

### 4.3 Contratto CO02.2: ControlloCorrettezzaCampi

ContrattoControlloCorrettezzaCampi Matteo Cervini Scadenza: 28 Dicembre 2024

- **Assigned** Matteo Cervini
- **Blocked by** SSDControlloCorrettezzaCampi(att1: tipo, att2; tipo)
- **Disciplina** Requisiti
- **Due** 27 Dicembre, 2024 2:00 AM (GMT+1) → 28 Dicembre, 2024 7:00 PM (GMT+1)
- **Parent-task** ContrattoGestioneRichiesteVolontariato
- **Priority** Bassa
- **Projects** DoIT-1

## Contratto CO02.2: ControllaCampi

### Operazione

checkFields()

### Riferimenti

Casi d'uso: ControlloCorrettezzaCampi

### Pre-Condizioni

- Almeno un campo non è valido

### Post-Condizioni

- Postcondizione
- Tutti i campi sono validi
- Eventuali campi con stile **errore** sono stati ripristinati con lo stile **default**
- Eventuali messaggi di errori sui campi sono stati rimossi

## 5 Specifica Supplementare

Specifiche Supplementari: Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele  
Buser Scadenza: 27 Dicembre, 2024

### 5.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-1

### 5.2 Requisiti Funzionali

Registrazione e Gestione degli Utenti

1. Gli utenti devono poter creare uno o entrambi gli account scegliendo tra i seguenti profili:
  - Volontario singolo
  - Associazione o organizzazione di volontariato o beneficiario singolo
2. Devono essere richieste informazioni obbligatorie per la registrazione, quali:
  - Nome e cognome (o ragione sociale per aziende/enti)
  - Email e numero di telefono
  - Password sicura con requisiti minimi di complessità
3. Ogni utente deve avere la possibilità di completare un profilo personalizzato, comprensivo di:
  - Disponibilità orarie
  - Competenze specifiche
  - Località

### **Gestione delle Richieste di Volontariato**

1. I beneficiari devono poter creare richieste di supporto specificando:
  - Categoria di aiuto
  - Descrizione dettagliata della necessità
  - Data e orario preferiti
  - Localizzazione geografica tramite mappa interattiva
2. Le richieste devono poter essere modificate o cancellate fino a quando non vengono accettate da un volontario.

### **Gestione delle Offerte di Volontariato**

1. I volontari devono poter indicare la propria disponibilità attraverso:
  - Calendario settimanale con fasce orarie selezionabili
  - Tipologia di attività offerte
  - Zone geografiche preferite
2. Devono poter modificare o aggiornare le proprie disponibilità in qualsiasi momento.

## **Sistema di Abbinamento (Matching)**

1. La piattaforma deve includere un algoritmo di matching che:
  - Confronti le richieste dei beneficiari con le disponibilità dei volontari
  - Utilizzi parametri quali categoria, localizzazione e orario per proporre abbinamenti
2. Gli utenti devono ricevere notifiche per ogni proposta di abbinamento rilevante.

## **Sistema di Messaggistica**

1. Deve essere integrato un sistema di chat sicuro per consentire la comunicazione tra volontari e beneficiari.
2. Le funzionalità principali della chat includono:
  - Invio di messaggi testuali
  - Notifiche in tempo reale
  - Cronologia dei messaggi

## **Sistema di Valutazione e Feedback**

1. Dopo il completamento di un'attività, entrambe le parti devono poter lasciare una valutazione.
2. Le recensioni devono includere:
  - Punteggio da 1 a 5 stelle
  - Commento opzionale
3. I punteggi devono essere visibili sui profili pubblici degli utenti.

## **Comitato di Controllo e Validazione**

1. La piattaforma deve consentire a un team dedicato di:
  - Verificare le richieste e le offerte inserite dagli utenti
  - Segnalare e rimuovere contenuti inappropriati
2. Deve essere previsto un sistema per ricevere e gestire segnalazioni dagli utenti.

## **Integrazione con Social Media**

1. Gli utenti devono poter condividere richieste e offerte di volontariato sui principali social network.

### **5.3 Requisiti Non Funzionali**

#### **Prestazioni**

1. Il sistema deve garantire tempi di risposta inferiori a 2 secondi per il caricamento delle principali pagine e operazioni.
2. L'algoritmo di matching deve essere in grado di elaborare richieste e offerte in meno di 5 secondi.
3. Aggiornamento in tempo reale delle notifiche di chat

#### **Scalabilità**

1. Deve essere possibile aumentare la capacità del sistema con l'aumento del numero di utenti.

#### **Sicurezza**

1. I dati degli utenti devono essere protetti attraverso crittografia sia a riposo che in transito.

#### **Compatibilità**

1. L'app deve essere compatibile con:
  - iOS
  - Android
2. Se previsto un portale web, deve essere ottimizzato per i principali browser.

#### **Usabilità**

1. L'interfaccia deve essere intuitiva e accessibile
2. Deve essere garantita un'esperienza coerente su tutti i dispositivi
3. Il processo di offerta deve essere completabile in pochi passaggi intuitivi

## **Manutenibilità**

1. Il codice deve essere scritto seguendo standard di programmazione modulari
2. Deve essere garantito il supporto per aggiornamenti regolari della piattaforma

## **Disponibilità e Affidabilità**

1. La piattaforma deve essere operativa almeno il 99,5% del tempo su base mensile.

## **Privacy**

1. I dati personali del volontario devono essere condivisi con l'organizzazione solo dopo l'accettazione dell'offerta
2. Le conversazioni nella chat devono essere protette e accessibili solo alle parti coinvolte

## **5.4 Requisiti Hardware**

**Server e Servizi di Hosting** Il progetto DoIT utilizzerà una soluzione di hosting cloud per garantire scalabilità e affidabilità:

- **Cloud Provider:** Firebase sarà utilizzato come piattaforma principale
- **Database:**
  - PostgreSQL per dati strutturati
  - Firestore per dati non strutturati

**Esigenze di Storage** Per la gestione dei dati, il sistema utilizzerà:

- PostgreSQL per dati strutturati
- Firebase Firestore per messaggistica
- Firebase Storage per file utente

## 5.5 Requisiti Software

### Tecnologie Preferite per lo Sviluppo

- **Frontend:**

- HTML, CSS e TailwindCSS
- TypeScript e Next.js

- **Backend:**

- Java con Spring Boot
- Spring Security
- Spring Data JPA
- OpenAPI

- **Database:**

- PostgreSQL
- Flyway

- **Tools/Utilities:**

- Git
- Maven
- npm
- SonarQube
- Understand
- Mermaid Live e Lucidchart

### Integrazioni con Altre Piattaforme o API Esterne

- Geolocalizzazione (Google Maps o Mapbox)
- Social Media (Facebook, Twitter, LinkedIn)

## **5.6 Specifiche di Interfaccia**

### **Requisiti per il Design dell'Interfaccia Utente**

#### **1. Design Responsivo:**

- Interfaccia completamente responsive
- Utilizzo di TailwindCSS

#### **2. Estetica Pulita e Semplice:**

- Design minimale
- Colori chiari e contrastanti
- Interfaccia coerente

#### **3. Navigazione Intuitiva:**

- Barra di navigazione chiara
- Funzionalità di ricerca avanzata
- Sistema di notifiche visibile

#### **4. Iconografia Chiara:**

- Icône comprensibili
- Stile uniforme

#### **5. Accessibilità:**

- Conformità WCAG 2.1
- Supporto screen reader
- Font leggibili

### **Linee Guida per l'Esperienza Utente (UX)**

#### **1. Semplicità e Chiarezza**

#### **2. Flusso Intuitivo**

#### **3. Feedback Visivo e Interattivo**

#### **4. Personalizzazione**

## 5. Chiarezza nelle Interazioni

## 6. Performance e Velocità

### 5.7 Vincoli di Progetto

#### Budget disponibile per sviluppo e manutenzione

- Progetto universitario senza budget specifico
- Sviluppo da parte di team accademico

#### Tempi di rilascio per MVP e versioni successive

- **MVP:** Completamento entro 1 mese
- **Versioni successive:** Cicli mensili di aggiornamento

### 5.8 Gestione del Rischio

#### Rischi Aggiuntivi

1. Mancanza di partecipazione da parte degli utenti
2. Problemi di sicurezza e privacy dei dati
3. Difficoltà tecniche nell'integrazione
4. Scalabilità limitata nell'MVP
5. Ritardi nello sviluppo

#### Strategie per Mitigare i Rischi

##### 1. Strategia per l'integrazione tecnica:

- Seguire guide ufficiali
- Creare prototipi

##### 2. Strategia per la scalabilità:

- Architettura modulare
- Load testing

### 3. Strategia per i ritardi:

- Pianificazione agile
- Valutazione risorse aggiuntive

## 6 Glossario

Specifica Supplementare Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser Scadenza: 27 Dicembre, 2024

### 6.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-1

### 6.2 Glossario Tecnico

- **Account Google**

Sistema di autenticazione richiesto per la registrazione nella piattaforma. Fornisce le informazioni base dell'utente come nome, email e foto profilo attraverso il protocollo OAuth.

- **Autenticazione**

Processo di verifica dell'identità dell'utente effettuato tramite OAuth Google. È un passaggio obbligatorio durante la registrazione che garantisce la sicurezza e l'unicità dell'account.

- **Crittografia**

Tecnica di sicurezza utilizzata per proteggere i dati sensibili degli utenti nel sistema. È un requisito non funzionale che garantisce la protezione delle informazioni personali memorizzate.

- **Dashboard Personale**

Interfaccia utente principale mostrata dopo il login, contenente:

- Opportunità di volontariato consigliate
- Riepilogo del profilo utente
- Altre funzionalità personalizzate

- **Form Aggiuntivo**

Modulo di registrazione che raccoglie informazioni specifiche del volontario non disponibili tramite l'account Google. Include:

- Campi obbligatori da compilare
- Validazione in tempo reale

- **Login Automatico**

Processo che avviene dopo la registrazione completata con successo, permettendo l'accesso immediato alla dashboard personale senza necessità di inserire nuovamente le credenziali.

- **OAuth**

Protocollo di autenticazione utilizzato per l'integrazione con Google, che permette:

- Accesso sicuro ai dati dell'account Google
- Importazione automatica delle informazioni base
- Verifica dell'identità dell'utente

- **Opportunità**

Attività di volontariato disponibili sulla piattaforma che vengono:

- Consigliate ai volontari sulla dashboard
- Filtrate in base al profilo dell'utente e area geografica
- Presentate dopo la registrazione
- Pubblicate solo da Organizzazioni verificate
- Non sovrapponibili temporalmente per lo stesso volontario

- **Organizzazione**

Tipologia di account alternativa a quella del volontario, con funzionalità e processo di registrazione specifici.

- **Profilo**

Insieme delle informazioni che caratterizzano il volontario, composto da:

- Dati importati da Google (nome, email, foto)
- Informazioni aggiuntive fornite durante la registrazione
- Stato di completezza e verifica

- **Registrazione**

Processo di creazione di un nuovo account volontario che include:

- Verifica dei prerequisiti (account Google)
- Raccolta dati tramite form
- Validazione in tempo reale
- Tempo massimo di completamento: 5 minuti

- **Storage Sicuro**

Sistema di memorizzazione dei dati personali che implementa:

- Crittografia delle informazioni sensibili
- Protezione dei dati personali
- Conformità con i requisiti di sicurezza

- **Validazione Real-time**

Sistema di controllo immediato dei dati inseriti che:

- Verifica la correttezza delle informazioni
- Mostra messaggi di errore specifici
- Impedisce la sottomissione di dati non validi

- **Volontario**

Utente registrato nella piattaforma che:

- Possiede un account Google verificato
- Ha completato il processo di registrazione
- Può accedere alle opportunità di volontariato
- Può offrire disponibilità per attività non sovrapposte temporalmente

- Deve specificare competenze e disponibilità temporali

- **Area Geografica**

Zona territoriale in cui il volontario è disponibile ad operare, specificata durante l'offerta di disponibilità per un'attività di volontariato.

- **Chat Integrata**

Sistema di messaggistica interno alla piattaforma che permette la comunicazione diretta tra Volontari e Organizzazioni dopo l'offerta di disponibilità.

- **Competenze**

Capacità e abilità specifiche richieste dalle Organizzazioni e selezionabili dai Volontari durante l'offerta di disponibilità per un'attività.

- **Mappa Interattiva**

Visualizzazione geografica delle richieste di volontariato disponibili che permette ai volontari di:

- Visualizzare la distribuzione delle opportunità sul territorio
- Interagire direttamente con le richieste
- Filtrare le attività per zona

- **Organizzazione Verificata**

Status speciale assegnato alle organizzazioni che hanno completato un processo di verifica nel sistema, requisito necessario per pubblicare richieste di volontariato.

- **Controllo Duplicato Account**

Sistema che verifica l'esistenza di un account già registrato associato alla stessa organizzazione, impedendo la duplicazione degli account e fornendo all'utente le alternative per risolvere eventuali conflitti.

- **Errore di Registrazione**

Messaggio che il sistema mostra quando il processo di registrazione non può essere completato, generalmente dovuto a dati mancanti, errati o duplicati. L'errore fornisce solitamente dettagli specifici per correggere il problema.

- **Conferma di Registrazione**

La fase finale del processo di registrazione in cui l'organizzazione approva i dati inseriti e conferma la creazione del proprio account, permettendo al sistema di procedere con il login automatico e il reindirizzamento alla dashboard.

- **Piattaforma**

Il sistema complessivo che include sia le funzionalità per i volontari che quelle per le organizzazioni, come la gestione dei profili, l'accesso alle opportunità di volontariato e le interazioni tra le diverse entità.

## 7 Visione

Specifiche Supplementari Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser Scadenza: 27 Dicembre, 2024

### 7.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 26 Dicembre, 2024 1:00 AM (GMT+1) → 27 Dicembre, 2024 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-1

**Scopo del Documento** Il presente documento descrive la visione per il progetto **DoIT**, una piattaforma innovativa dedicata alla gestione delle richieste e delle offerte di volontariato. L'obiettivo principale è facilitare la connessione tra chi offre il proprio tempo e chi necessita di supporto, con un focus iniziale sulla città di Milano e, successivamente, sull'intera Lombardia. Il progetto è ispirato all'app “Attivati”, già operativa nel Trentino, ma mira a proporre un'evoluzione adattata al contesto lombardo, introducendo nuove funzionalità tecniche avanzate e rispondendo alle specifiche necessità locali.

### Descrizione del Problema

- **Contesto attuale:** La gestione del volontariato soffre spesso di una mancata connessione tra chi offre aiuto e chi lo richiede. Le opportunità di volontariato

non sempre vengono adeguatamente pubblicizzate o rese accessibili, creando un disallineamento tra domanda e offerta.

- **Necessità principali:** Vi è il bisogno di una piattaforma centralizzata e user-friendly che consenta di gestire in modo efficace le richieste e le offerte di volontariato.
- **Benefici attesi:** Un aumento della partecipazione al volontariato, una migliore distribuzione delle risorse e una maggiore trasparenza e sicurezza per tutti i partecipanti.

## Stakeholder

- **Cittadini:** Persone disposte a offrire il proprio tempo per il volontariato.
- **Associazioni di volontariato:** Organizzazioni che coordinano e gestiscono attività di volontariato.
- **Aziende ed enti:** Realtà interessate a promuovere il volontariato aziendale.
- **Beneficiari:** Persone o gruppi che necessitano di supporto.
- **Comitato di controllo:** Team responsabile di verificare la veridicità delle informazioni e garantire la sicurezza della piattaforma.

## Obiettivi del Sistema

1. Creare un ambiente digitale sicuro e intuitivo per la gestione del volontariato.
2. Facilitare il matching tra richieste e offerte di volontariato.
3. Garantire la sicurezza e la qualità delle interazioni tra gli utenti.
4. Promuovere il volontariato come pratica sociale attraverso l'uso di tecnologie avanzate.

## Funzionalità Principali

### 1. Registrazione degli utenti:

- Creazione di profili per volontari, associazioni, aziende ed enti.
- Inserimento di informazioni su disponibilità, competenze e preferenze.

**2. Gestione delle richieste di volontariato:**

- Inserimento di richieste specifiche (es. supporto per anziani, studenti, ecc.).
- Dettaglio di durata, orari e posizione geografica.
- Descrizione dettagliata delle necessità.

**3. Gestione delle offerte di volontariato:**

- Inserimento di disponibilità di tempo e competenze.
- Localizzazione della disponibilità e delle aree di intervento.

**4. Sistema di abbinamento e ricerca:**

- Algoritmi di matching basati su categorie, orari e posizione geografica.
- Filtri per ricerca avanzata.

**5. Comitato di controllo e validazione:**

- Verifica delle richieste e delle offerte.
- Segnalazione di contenuti inappropriati.

**6. Sistema di messaggistica integrato:**

- Chat per comunicazioni dirette tra volontari e richiedenti.
- Notifiche in tempo reale.

**7. Sistema di valutazione e feedback:**

- Recensioni degli utenti per garantire qualità e affidabilità.

**8. Integrazione con social media:**

- Condivisione delle attività e delle disponibilità sui social network.

**Ambito del Sistema**

**• Incluso:**

- Funzionalità principali elencate sopra.
- Focus iniziale su Milano con espansione successiva alla Lombardia.

- **Escluso:**
  - Gestione di richieste fuori dal territorio lombardo (nella fase iniziale).
  - Supporto per lingue diverse dall’italiano.

## Vincoli

- **Tecnologici:** L’app deve essere compatibile con i principali sistemi operativi (iOS e Android).
- **Economici:** Rispetto del budget assegnato per lo sviluppo e la manutenzione ossia 0.
- **Temporali:** Lancio della prima versione entro 1 mese dall’inizio dello sviluppo.

## Rischi e Assunzioni

- **Rischi:**
  - Mancata partecipazione degli utenti.
  - Problemi tecnici durante lo sviluppo.
  - Difficoltà nel garantire la sicurezza delle interazioni.
- **Assunzioni:**
  - Gli utenti disporranno di smartphone per accedere all’app.
  - Esiste una domanda significativa di volontariato nella regione.

**Conclusione** Il progetto **DoIT** si propone come uno strumento essenziale per favorire il volontariato nella regione Lombardia, con particolare attenzione alla città di Milano. Grazie all’integrazione di tecnologie avanzate e a un approccio centrato sugli utenti, l’app rappresenta un passo avanti verso una comunità più solidale e connessa.

## 3 Modellazione di Business

### 1 Modello delle Classi Concettuali

Modello delle Classi Concettuali Matteo Cervini Andrea Cozzi Gabriele Groppo  
Daniele Buser Scadenza: 29 Dicembre 2024

- **Disciplina:** Modellazione del Business
- **Scadenza:** 28 Dicembre, 2024 1:00 AM (GMT+1) → 29 Dicembre, 2024 7:00 AM (GMT+1)
- **Priorità:** Alta
- **Progetto:** DoIT-1

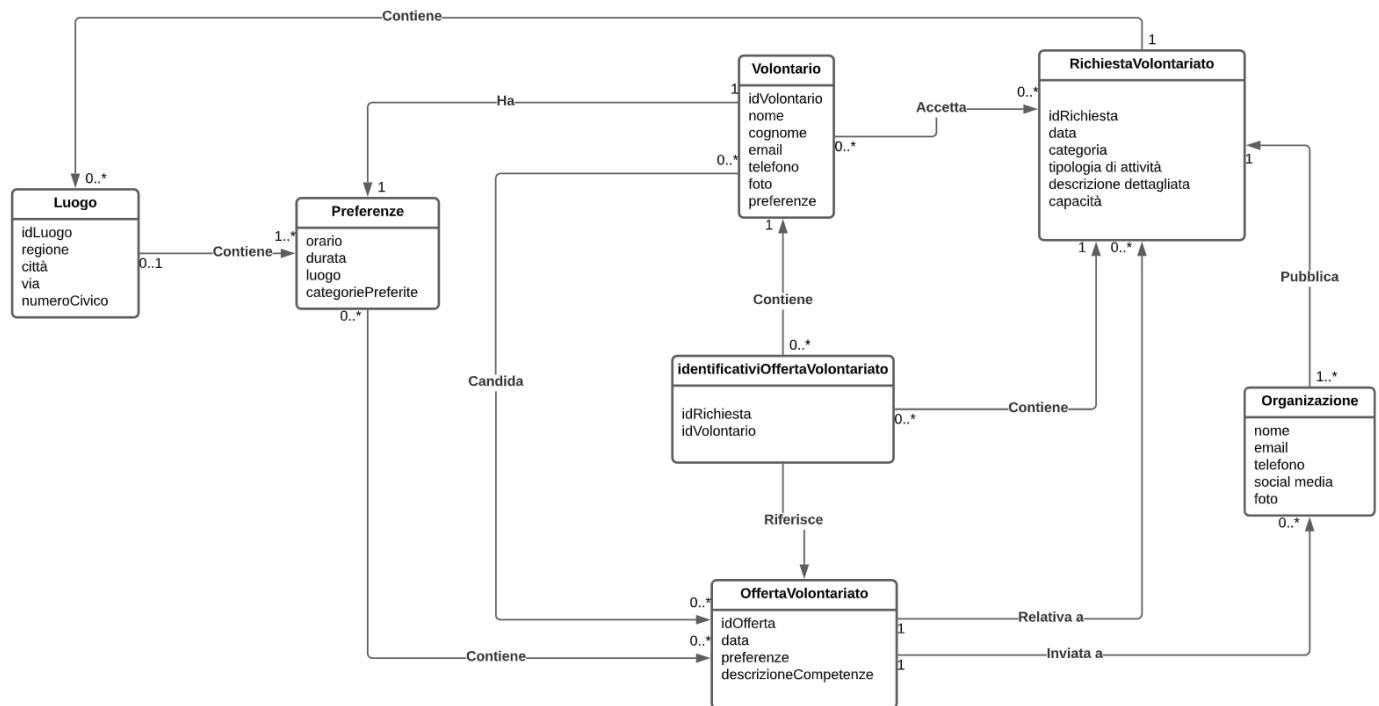


Figure 17: Modello delle Classi Concettuali

## 4 Progettazione

### 1 Diagrammi d'Interazione

#### 1.1 Diagramma di Sequenza SD1: RegisterOrganization

SDRegisterOrganization Andrea Cozzi Scadenza: 1 Gennaio 2025

- **Assigned** Andrea Cozzi

- **Blocked by SSD, Contratti**
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2024 1:00 AM (GMT+1) → 1 Gennaio, 2025 1:00 AM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi di Interazione
- **Priority** Alta
- **Projects** DoIT-1

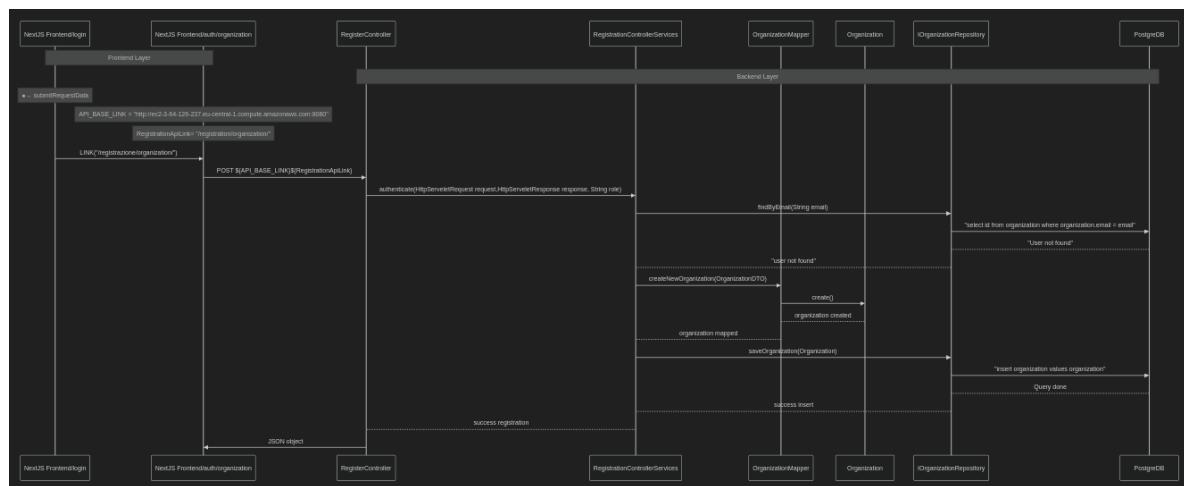


Figure 18: Diagramma di Interazione

## 1.2 Diagramma di Sequenza SD2: SubmitRequestData

SDSubmitRequestData Matteo Cervini Scadenza: 1 Gennaio 2025

- **Assigned** Matteo Cervini
- **Blocked by SSD, Contratti**
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2024 1:00 AM (GMT+1) → 1 Gennaio, 2025 1:00 AM (GMT+1)

- **Is blocking** Implementazione
- **Parent-task** Diagrammi di Interazione
- **Priority** Alta
- **Projects** DoIT-1

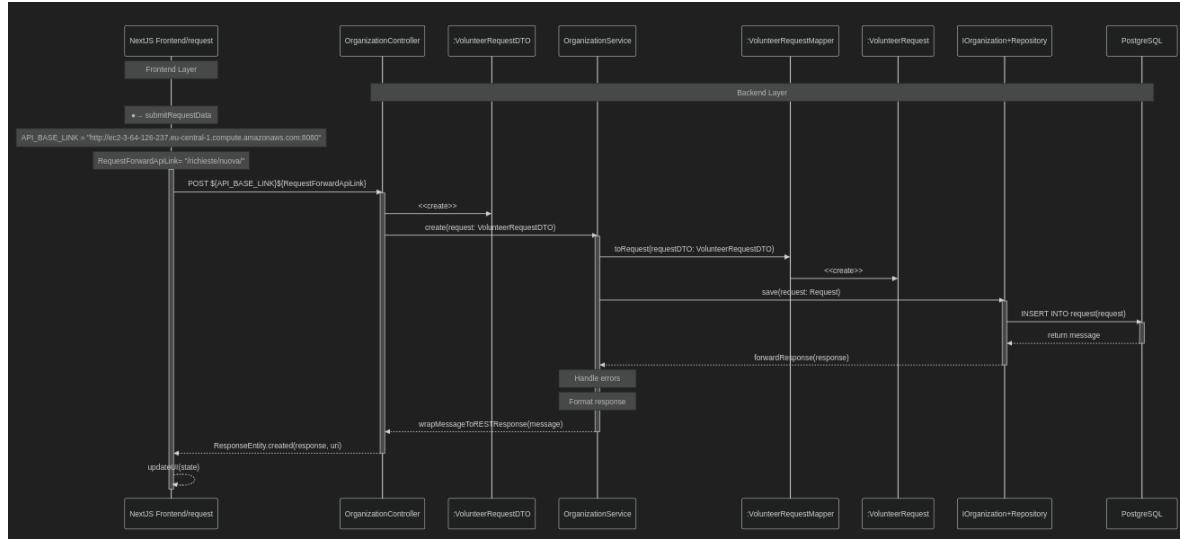


Figure 19: Diagramma di Interazione

### 1.3 Diagramma di Sequenza SD3: GetVolunteerRequestAsList

SDGetVolunteerRequestAsList Gabriele Scadenza: 1 Gennaio 2025

- **Assigned** Gabriele Groppo
- **Blocked by** SSD, Contratti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2024 1:00 AM (GMT+1) → 1 Gennaio, 2025 1:00 AM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi di Interazione

- Priority Alta
- Projects DoIT-1

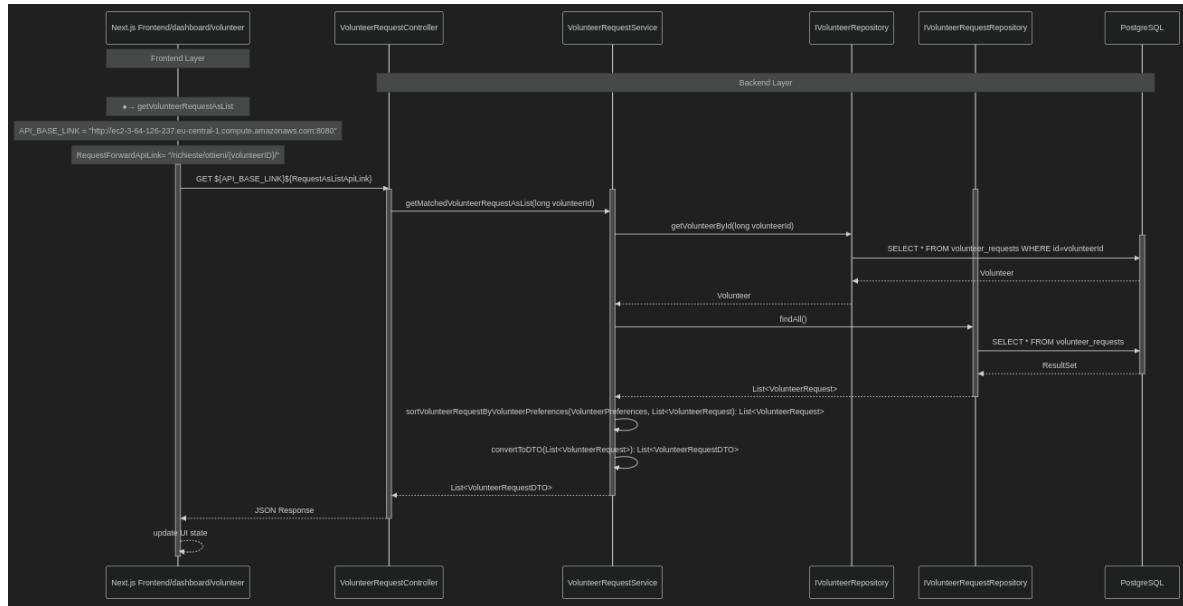


Figure 20: Diagramma di Interazione

## 2 Diagrammi delle Attività

### 2.1 Diagramma delle Attività DA1: SubmitRequest

DASubmitRequestData Daniele Buser Andrea Cozzi Scadenza: 2 Gennaio 2025

- **Assigned** Daniele Buser Andrea Cozzi
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 1 Gennaio, 2025 12:00 PM (GMT+1) → 2 Gennaio, 2025 7:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Attività

- Priority Alta
- Projects DoIT-1

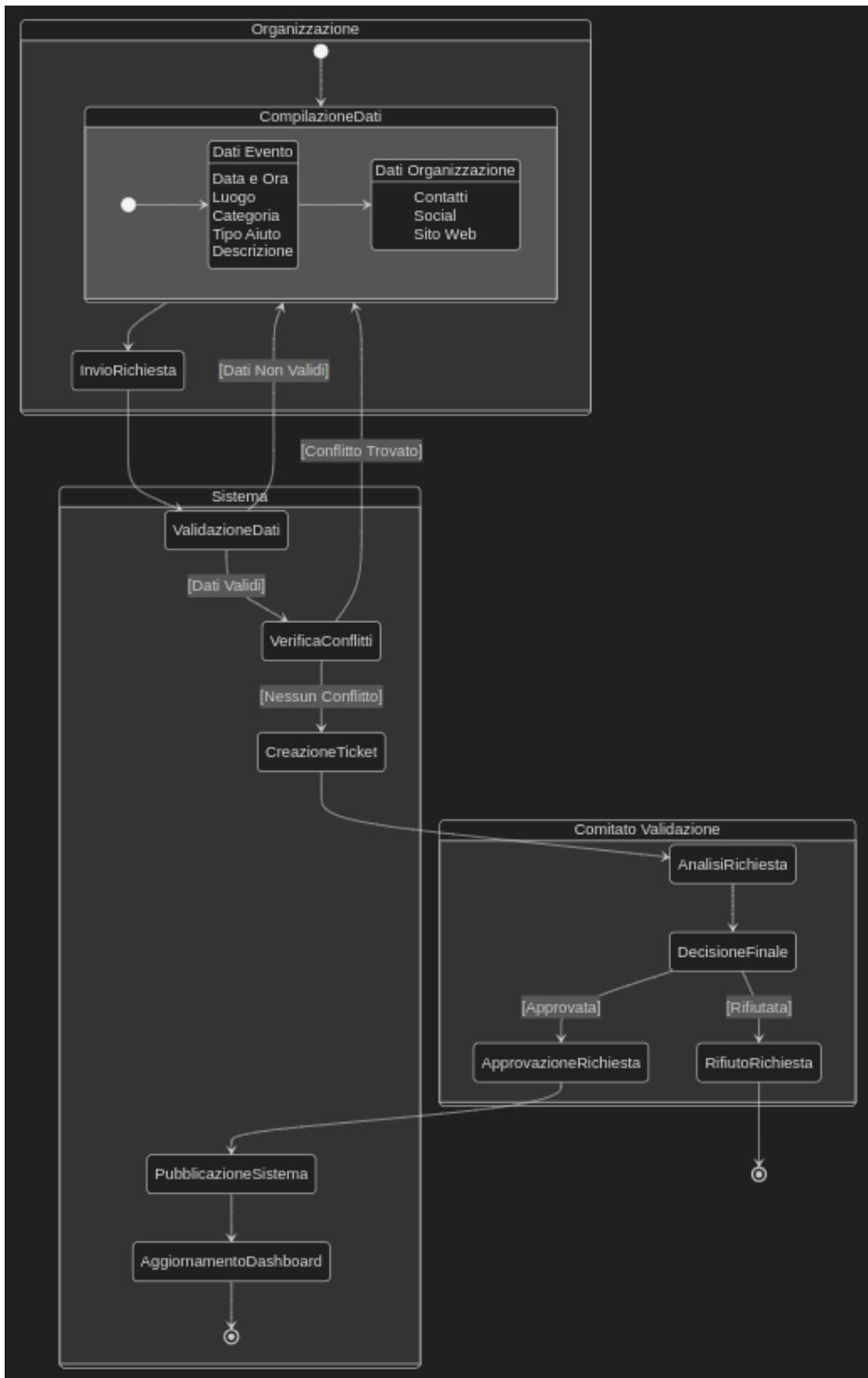


Figure 21: Diagramma delle Attività  
56

### 3 Diagrammi delle Macchine a Stati

#### 3.1 Diagramma delle Macchine a Stati DS1: RegisterVolunteer

DSRegisterVolunteer Andrea Cozzi Scadenza: 2 Gennaio 2025

- **Assigned** Andrea Cozzi
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 1 Gennaio, 2025 12:00 PM (GMT+1) → 2 Gennaio, 2025 7:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Macchine a Stati
- **Priority** Alta
- **Projects** DoIT-1

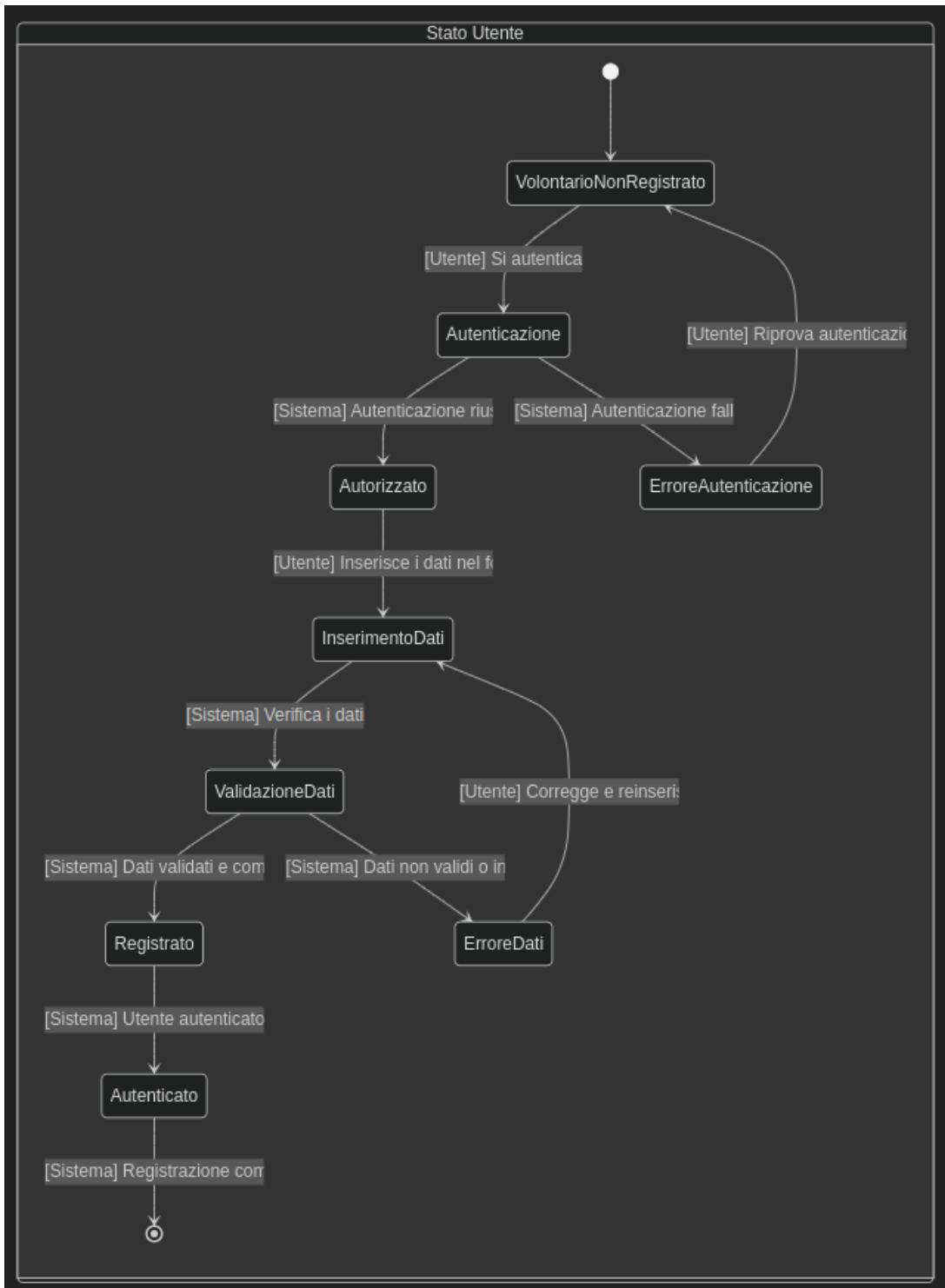


Figure 22: Diagramma delle Attività  
58

### 3.2 Diagramma delle Macchine a Stati DS2: VolunteerOffers

DSVolunteerOffers Daniele Buser Scadenza: 2 Gennaio 2025

- **Assigned** Daniele Buser
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 1 Gennaio, 2025 12:00 PM (GMT+1) → 2 Gennaio, 2025 7:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Macchine a Stati
- **Priority** Alta
- **Projects** DoIT-1

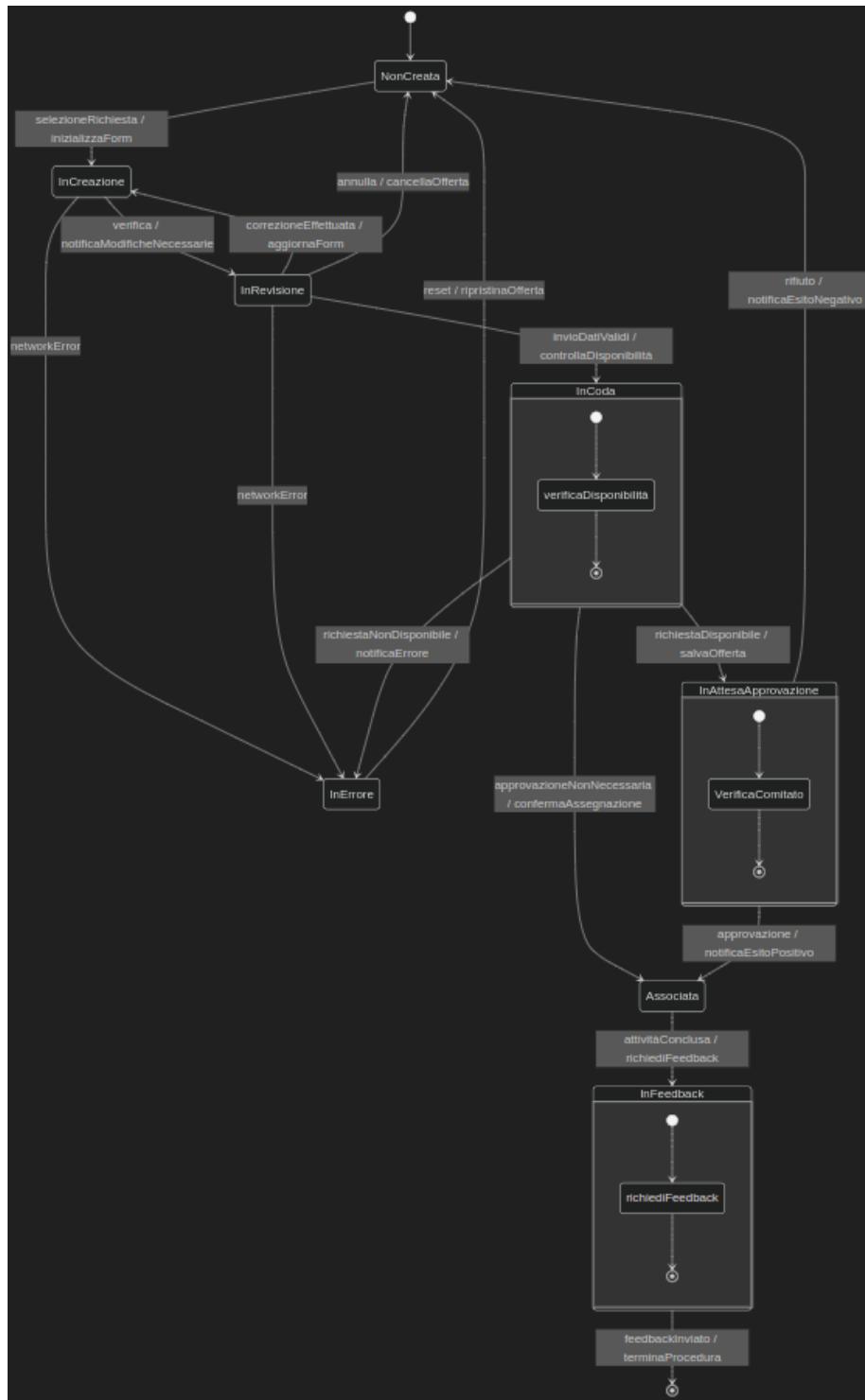


Figure 23: Diagramma delle Attività  
60

## 4 Diagrammi delle Classi Software di Progetto

---

### 4.1 Diagramma delle Classi Software di Progetto DCSP1: Dominio

DCSPDominio Matteo Cervini Gabriele Groppo Andrea Cozzi Daniele Buser Scadenza: 5 Gennaio 2025

- **Assigned** Matteo Cervini Gabriele Groppo Andrea Cozzi
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

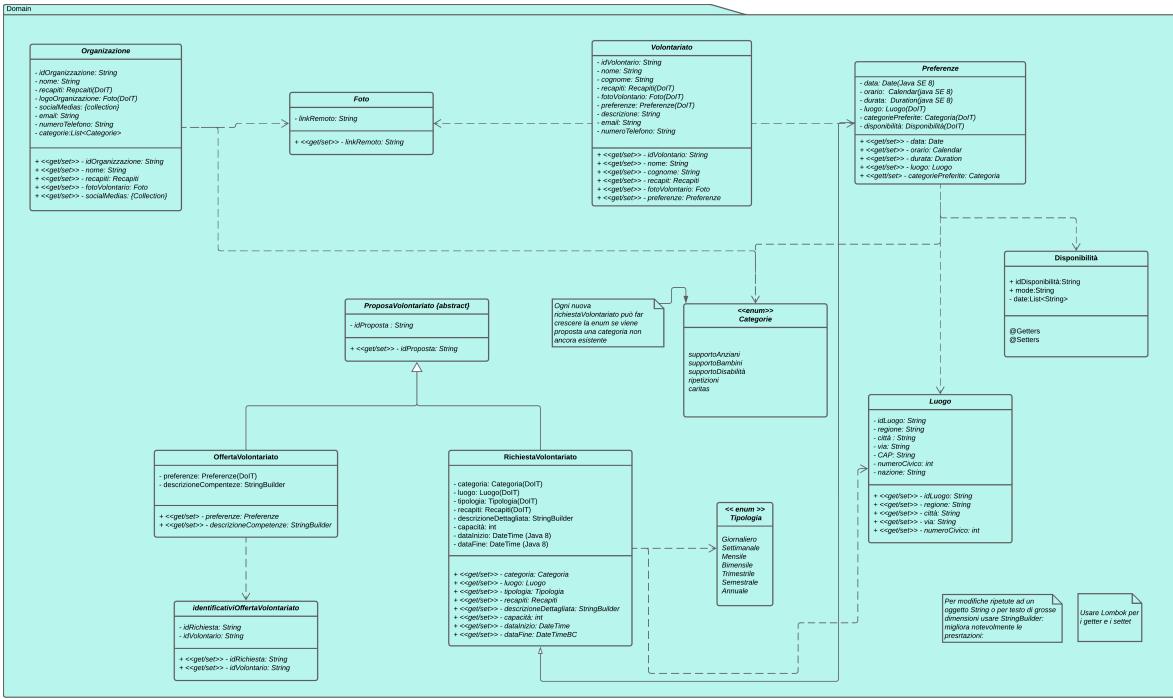


Figure 24: Diagramma delle Classi Software di Progetto

## 4.2 Diagramma delle Classi Software di Progetto DCSP2: Frontend

DCSPFrontend Matteo Cervini Daniele Buser Scadenza: 5 Gennaio 2025

- **Assigned** Matteo Cervini Daniele Buser
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

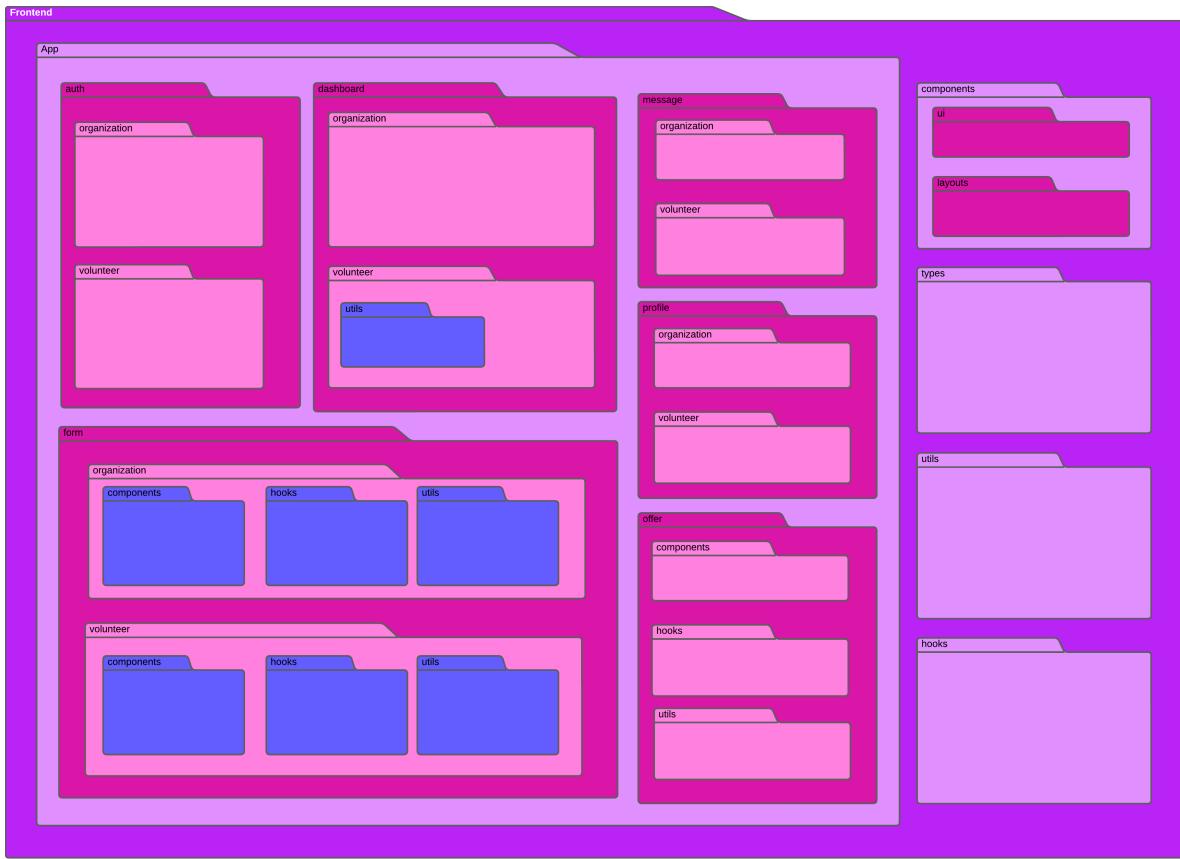


Figure 25: Diagramma delle Classi Software di Progetto

#### 4.3 Diagramma delle Classi Software di Progetto DCSP3: Backend/Controller

DCSPController Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione

- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

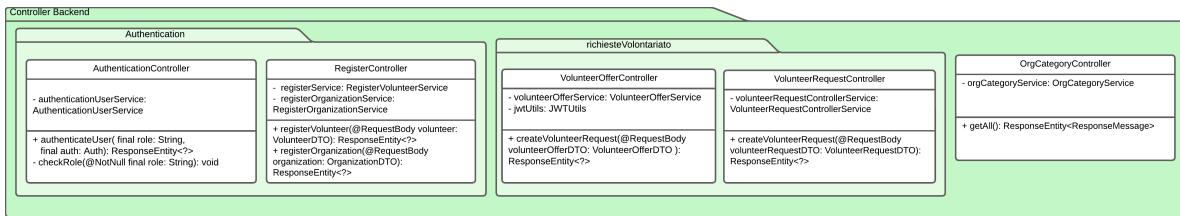


Figure 26: Diagramma delle Classi Software di Progetto

#### 4.4 Diagramma delle Classi Software di Progetto DCSP4: Backend/Domain

DCSPDomain Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

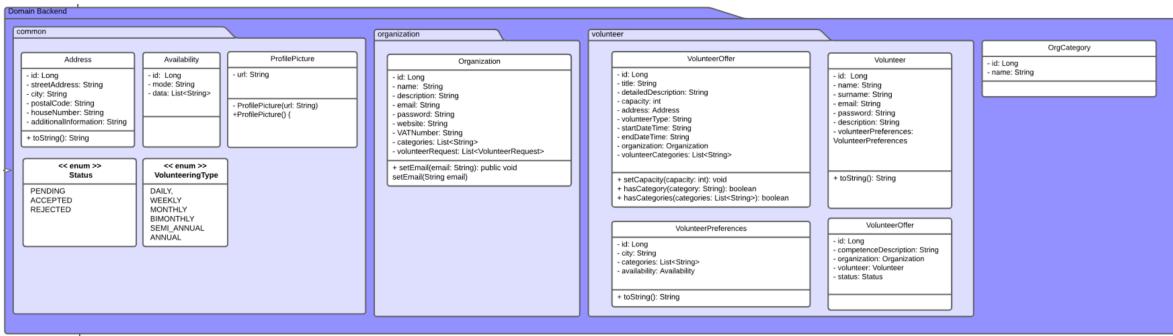


Figure 27: Diagramma delle Classi Software di Progetto

#### 4.5 Diagramma delle Classi Software di Progetto DCSP5: Backend/DTO

DCSPDTO Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

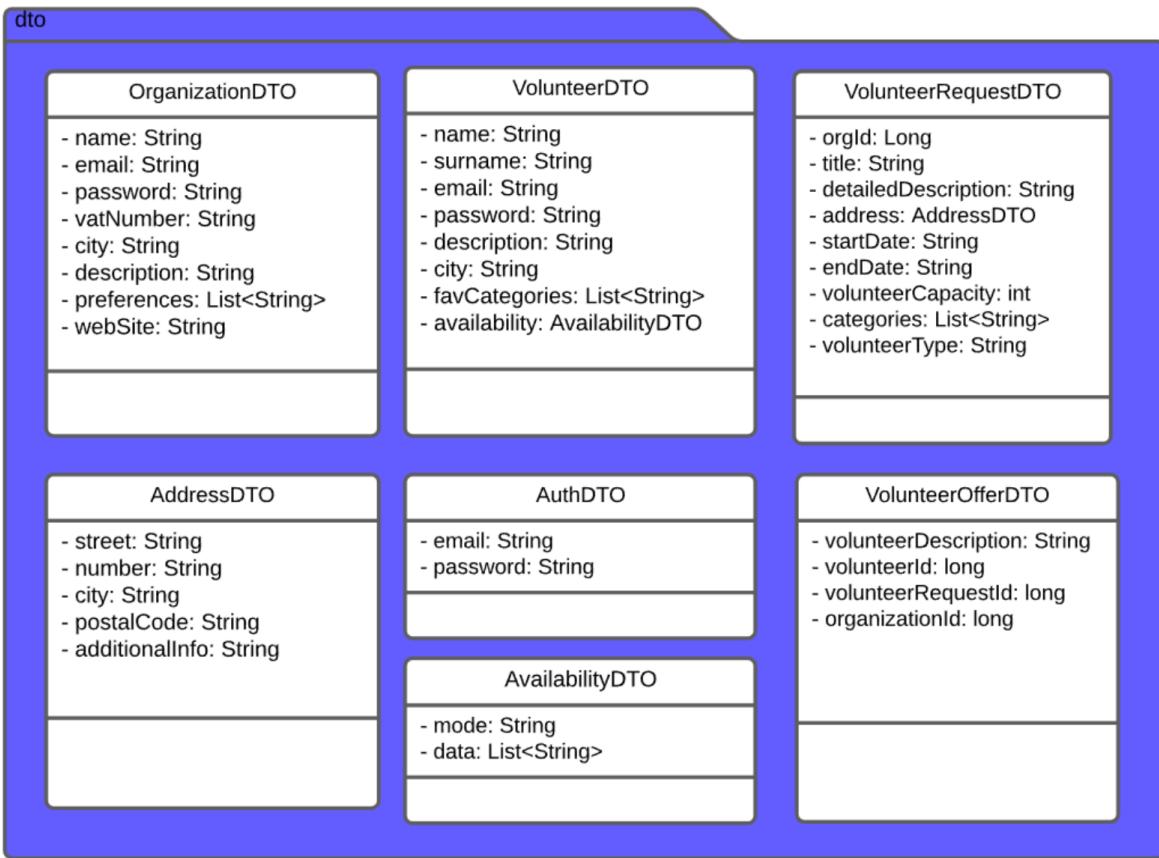


Figure 28: Diagramma delle Classi Software di Progetto

#### 4.6 Diagramma delle Classi Software di Progetto DCSP6: Backend/Mappers

DCSPMappers Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione

- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

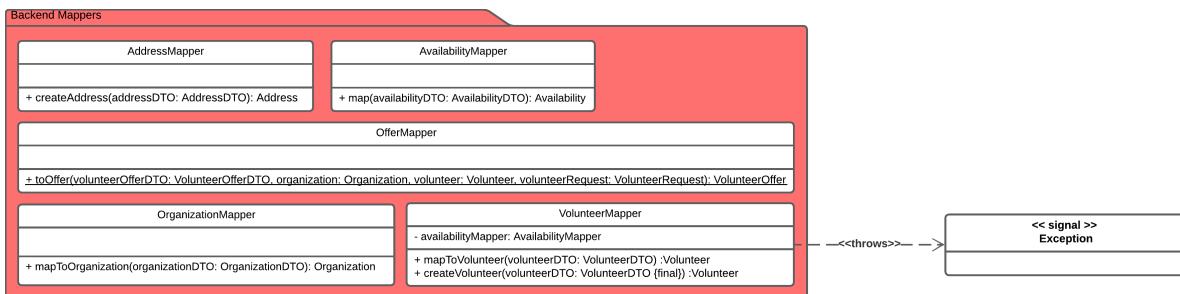


Figure 29: Diagramma delle Classi Software di Progetto

#### 4.7 Diagramma delle Classi Software di Progetto DCSP7: Backend/Service

DCSPService Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-1

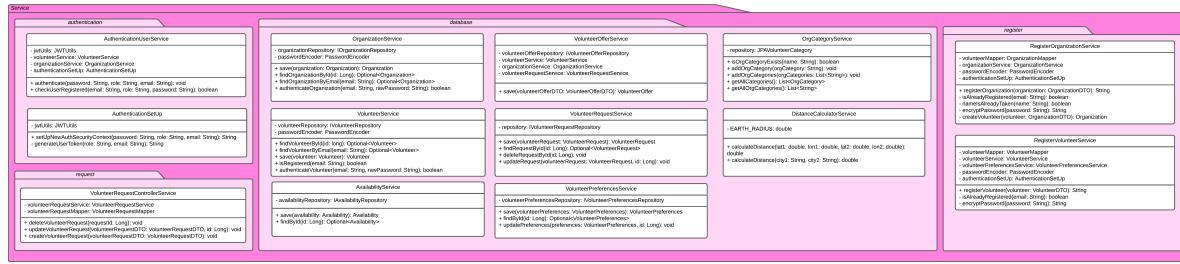


Figure 30: Diagramma delle Classi Software di Progetto

#### 4.8 Diagramma delle Classi Software di Progetto DCSP8: Backend/Repository

DCSPRepository Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Gropo
  - **Blocked by** Requisiti
  - **Disciplina** Progettazione
  - **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
  - **Is blocking** Implementazione
  - **Parent-task** Diagrammi delle Classi Software di Progetto
  - **Priority** Alta
  - **Projects** DoIT-1

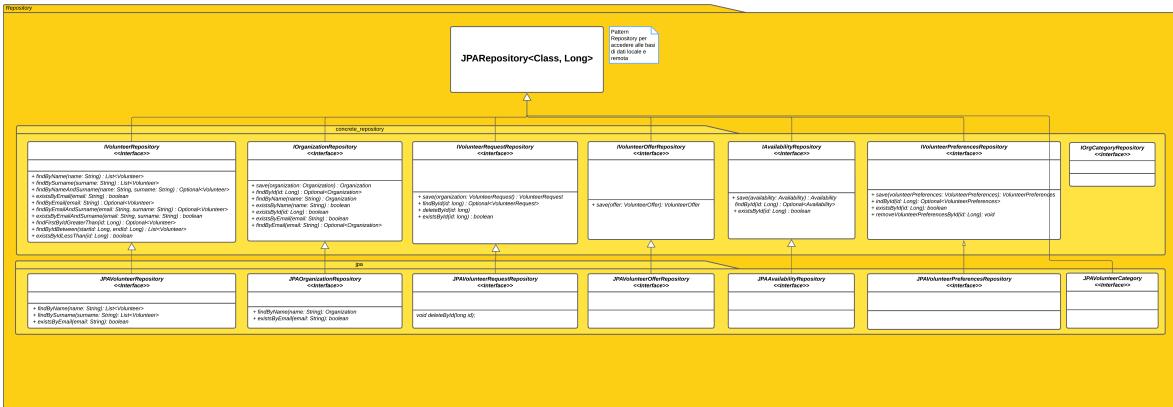


Figure 31: Diagramma delle Classi Software di Progetto

## 5 Diagramma dell'Architettura Logica

### 5.1 Diagramma dell'Architettura Logica

Architettura Logica Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser  
Scadenza: 5 Gennaio, 2025

- **Assigned** Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser
- **Blocked by Requisiti**
- **Disciplina** Progettazione
- **Due** 30 Dicembre, 2024 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Priority** Alta
- **Projects** DoIT-1

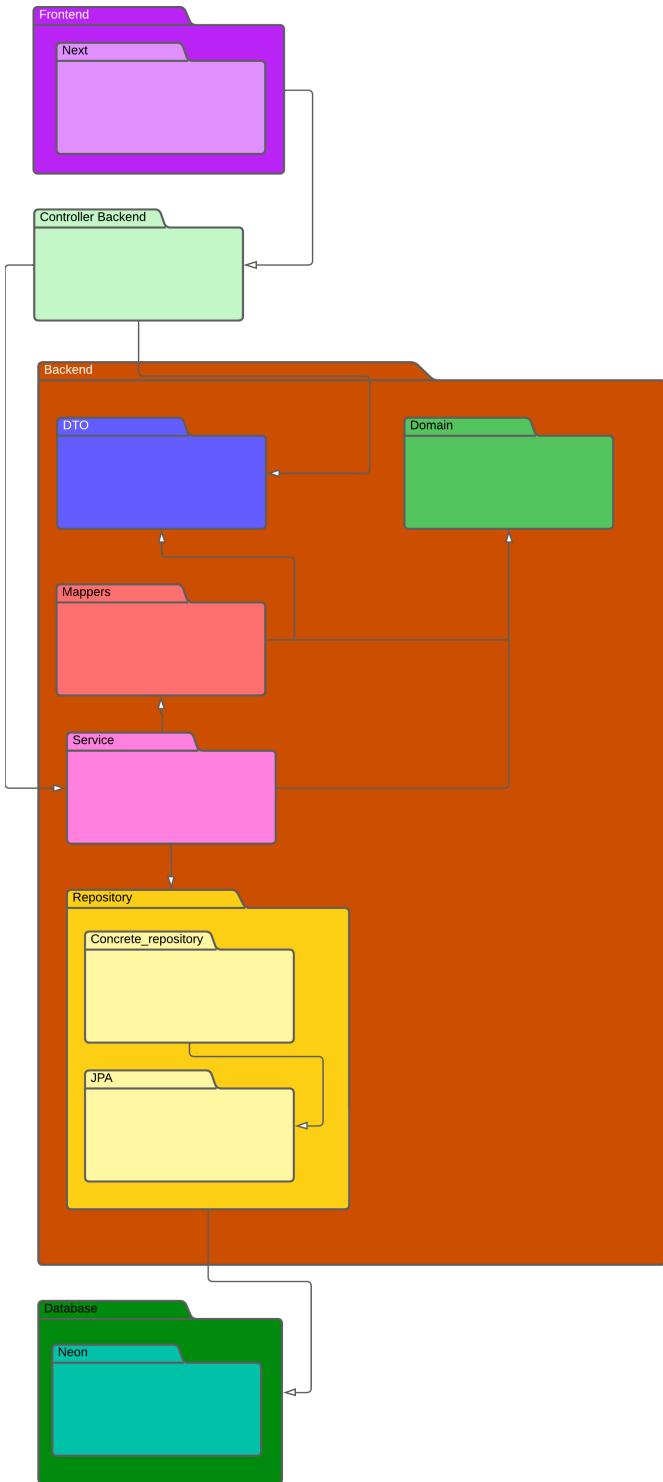


Figure 32: Diagramma dell'Architettura Logica  
70

## 6 Pattern

Pattern Matteo Cervini Daniele Buser Andrea Cozzi Gabriele Groppo Scadenza: 5 Gennaio 2025

- **Assigned** Matteo Cervini Daniele Buser Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Due** 30 Dicembre, 2025 1:00 AM (GMT+1) → 5 Gennaio, 2025 11:00 PM (GMT+1)
- **Priority** Alta
- **Projects** DoIT-1

### 6.1 Design Patterns

Frontend:

- MVVM

Backend:

- Singleton
- Facade Controller
- Facade
- Repository
- Proxy
- Service
- Template (authRedirectService)
- DTO

# Iterazione 2

## 1 Diagramma di Gantt

Questi sono i diagrammi di Gantt per l'iterazione 2.

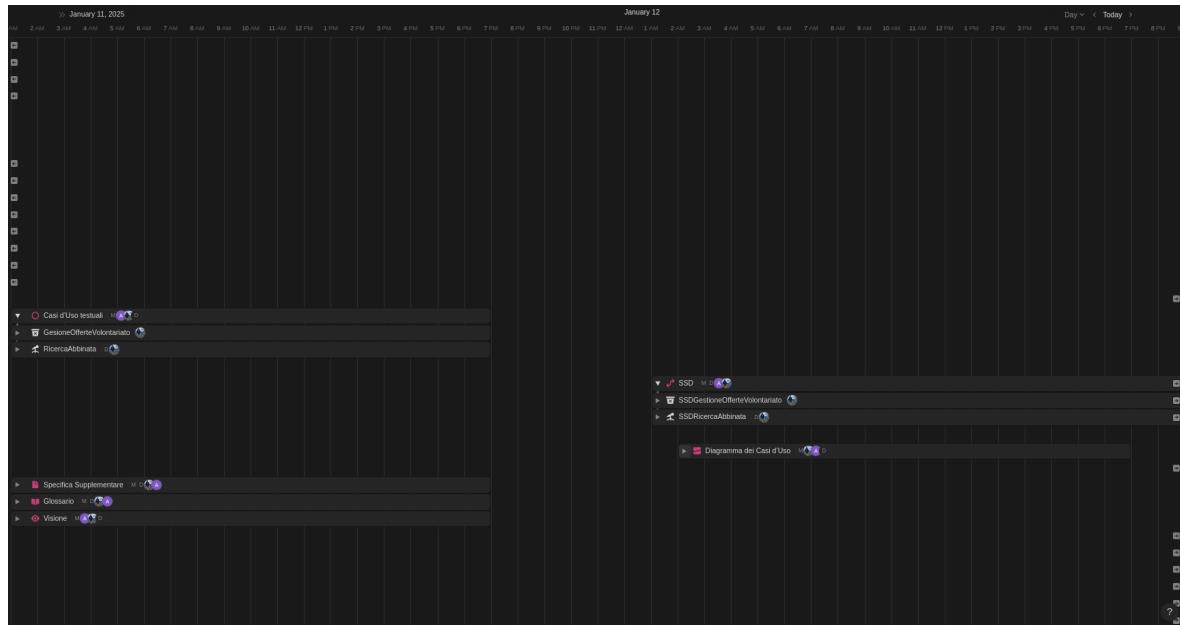


Figure 33: Diagramma di Gantt 12

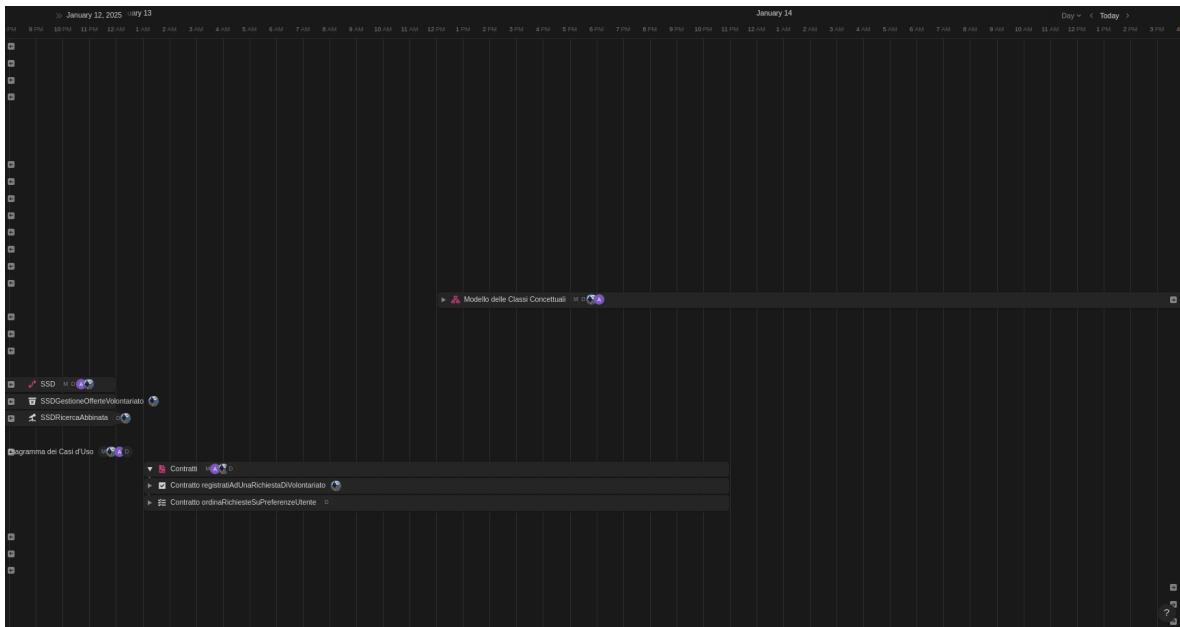


Figure 34: Diagramma di Gantt 13

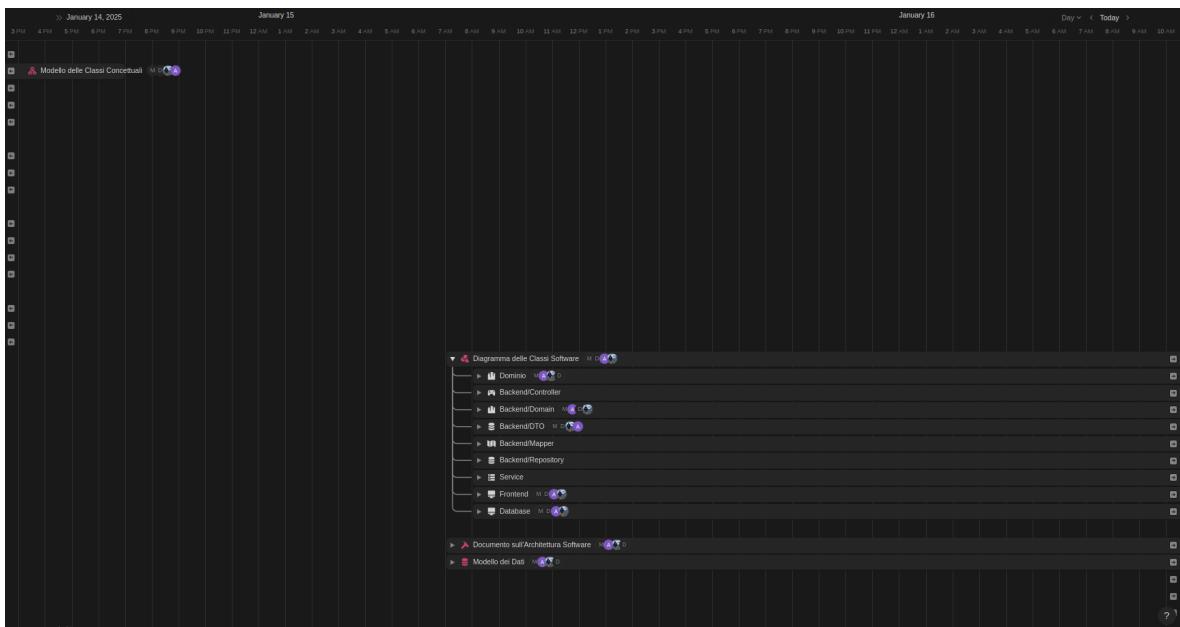


Figure 35: Diagramma di Gantt 13

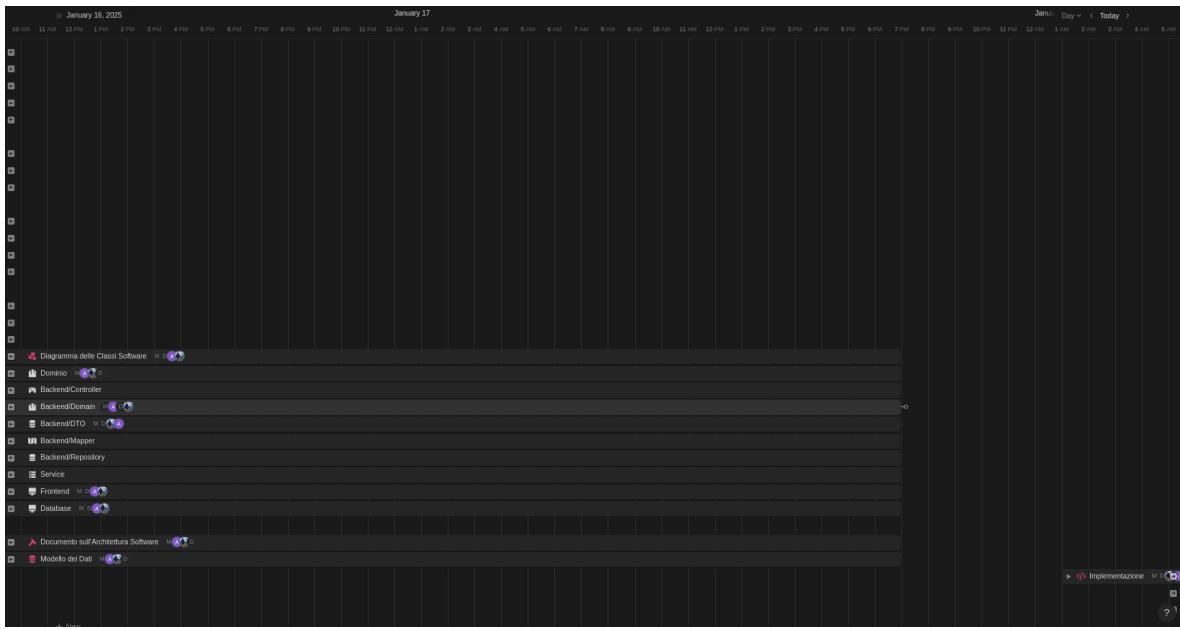


Figure 36: Diagramma di Gantt 15

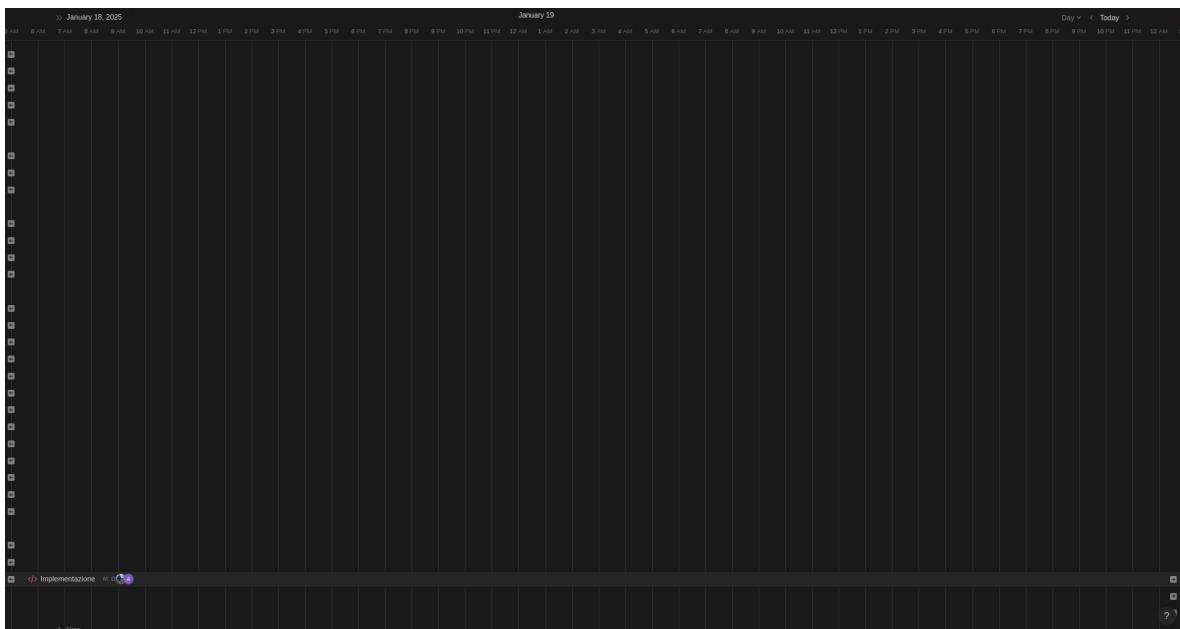


Figure 37: Diagramma di Gantt 16

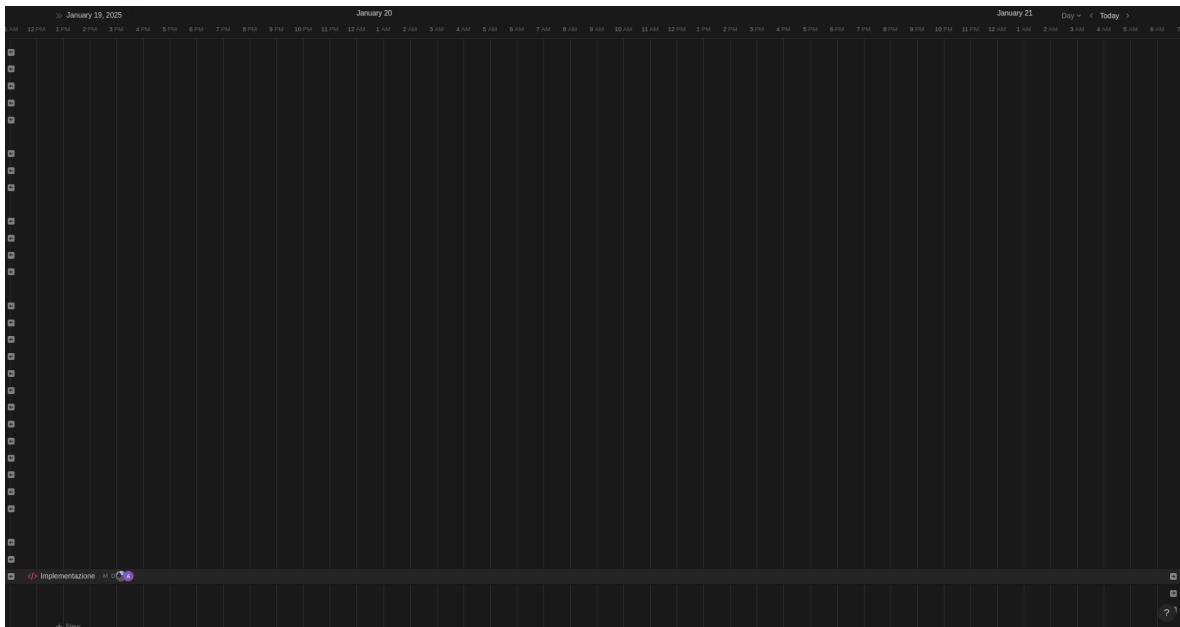


Figure 38: Diagramma di Gantt 17



Figure 39: Diagramma di Gantt 18

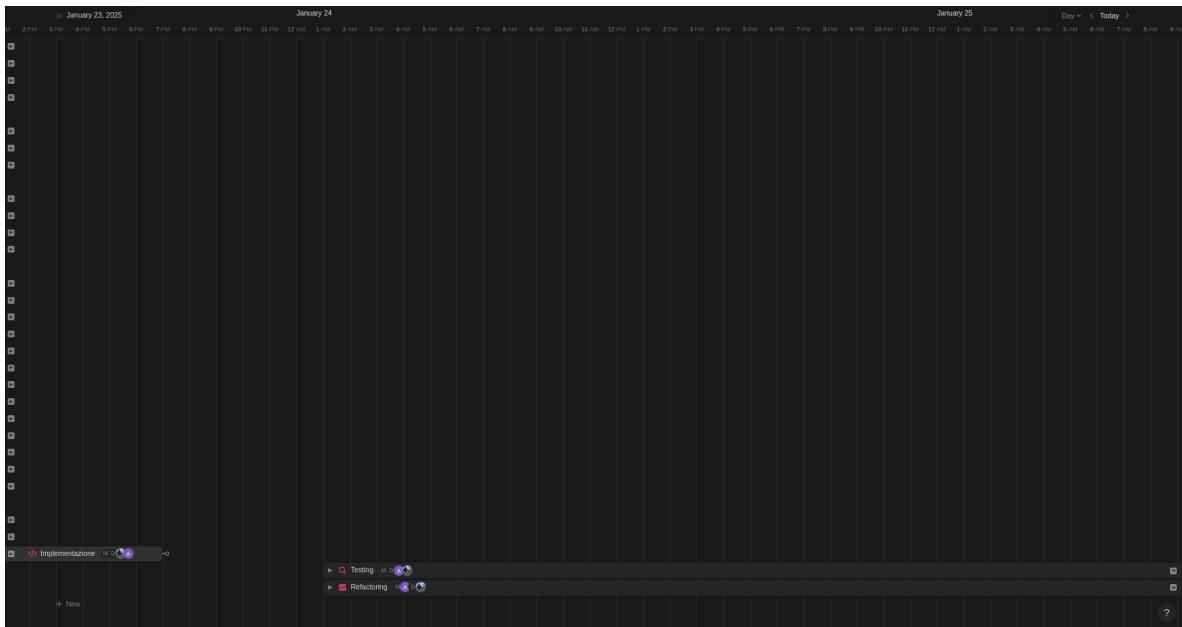


Figure 40: Diagramma di Gantt 19

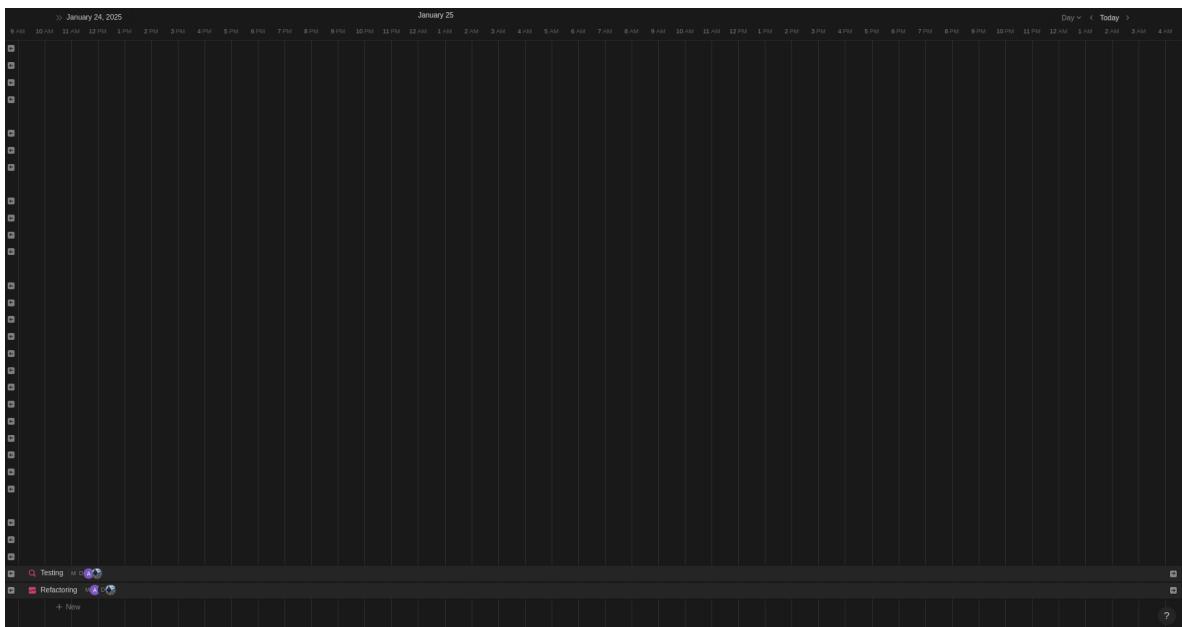


Figure 41: Diagramma di Gantt 20



Figure 42: Diagramma di Gantt 21

## 2 Requisiti

### 1 Casi d'uso

#### 1.1 Caso d'uso UC4: GestioneOfferteVolontariato

Gestione Offerte Volontariato Gabriele Groppo Scadenza: 11 Gennaio 2025

#### Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 1:00 PM (GMT+1)
- **Is blocking:** RicercaAbbinata, GestioneOfferteVolontariato
- **Parent-task:** Casi d'Uso testuali
- **Priority:** Alta

- **Projects:** DoIT-2

Il caso d'uso **"GestioneOfferte Volontariato"** (**UC-004**) descrive l'interazione che c'è tra un Utente Volontario e il Sistema quando questo vuole offrire la propria disponibilità per un'attività di volontariato.

L'Utente Volontario una volta effettuato l'autenticazione deve poter visualizzare nel Sistema le richieste di volontariato disponibili e deve comunicare le sue disponibilità in termini di tempo, competenze, area geografica di intervento.

L'Utente Volontario può offrire la propria disponibilità solo per le attività pubblicate da Organizzazioni verificate nel sistema. Non è possibile offrirsi privatamente per attività non registrate attraverso un'Organizzazione.

L'Utente Volontario deve selezionare una o più competenze tra quelle proposte dall'organizzazione e disponibilità, includendo: fasce orarie precise, e area geografica di azione.

L'Utente Volontario non può offrire la propria disponibilità per più attività che si sovrappongono temporalmente. È però possibile offrirsi per più attività diverse in orari diversi nella stessa giornata.

Il Sistema mostra una mappa interattiva con tutte le richieste di volontariato disponibili, permettendo all'Utente Volontario di interagire direttamente con le Organizzazioni attraverso una chat integrata nella piattaforma.

#### **Informazioni Generali ID Caso d'Uso:** UC-004

**Nome:** Gestisci Offerte Volontariato **Attore Principale:** Utente Volontario  
**Stakeholder:**

1. L'Organizzazione ha interesse nel ricevere offerte di volontariato pertinenti alle proprie richieste.
2. Amministratori del sistema hanno interesse nel matching tra offerte e richieste di volontariato sintomo di un corretto funzionamento dell'sistema stesso.

#### **Precondizioni:**

1. L'Utente Volontario ha effettuato l'accesso al sistema.
2. Esistono richieste di volontariato attive nel sistema.
3. L'Utente Volontario ha completato il proprio profilo con informazioni di base.

#### **Post-condizioni:**

1. L'offerta di volontariato è registrata nel sistema.
2. L'Organizzazione riceve una notifica della nuova partecipazione.
3. Viene creato un canale di chat tra Utente Volontario e Organizzazione.

**Priorità:** Alta

Scenario Principale

**Obiettivo** L'obiettivo del caso d'uso UC-004 "Gestione Offerte Volontariato" è permettere ai volontari di offrire la propria disponibilità per le attività di volontariato pubblicate dalle organizzazioni nel sistema, assicurando un matching efficace tra le competenze/disponibilità offerte e le necessità richieste.

### **Flusso Base**

1. Il Sistema mostra all'Utente Volontario una lista con le richieste di volontariato disponibili sulla base delle sue preferenze (posizione, tipologia di volontariato, ...).
2. L'Utente Volontario visualizza l'elenco delle richieste di volontariato filtrabili per:
  - Tipo di supporto richiesto
  - Zona geografica
  - Data e fascia oraria
  - Competenze richieste
3. L'Utente Volontario seleziona una richiesta di interesse.
4. Il Sistema mostra i dettagli completi della richiesta selezionata.
5. L'Utente Volontario indica la propria disponibilità per la richiesta selezionata specificando tra le proposte dall'Organizzazione:
  - Le fasce orarie
  - Il tipo specifico di supporto che intende offrire
  - L'area geografica in cui è disponibile ad operare
6. Il Sistema verifica la compatibilità tra disponibilità offerta e richiesta.

7. Il Sistema notifica l'Organizzazione della nuova offerta di volontariato.
8. L'Utente Volontario può iniziare una chat con l'Organizzazione per eventuali chiarimenti.

## section Scenari Alternativi

### **Scenario: L'utente consulta la mappa**

1. L'utente decide di consultare le richieste di volontariato attraverso la mappa.
  - Punto 2 scenario principale.

## Eccezioni e Gestione Errori

### **Errore: Connessione assente**

- **Condizione:** Il dispositivo dell'utente perde la connessione durante l'interazione col sistema.
- **Azione:** Il sistema mostra un messaggio di errore all'utente.
- **Risultato:** L'inserimento dell'offerta di volontariato viene annullato.

### **Errore: Dati incompleti**

- **Condizione:** Il volontario tenta di inviare l'offerta senza tutti i dati obbligatori.
- **Azione:** Il sistema verifica la completezza dei campi richiesti.
- **Risultato:** Vengono evidenziati i campi mancanti con richiesta di compilazione.

### **Eccezione: Richiesta già presa**

- **Condizione:** Il volontario tenta di inviare l'offerta ma il server risponde che quella richiesta di volontariato è già stata soddisfatta.
- **Azione:** Il sistema mostra un messaggio di errore.
- **Risultato:** L'inserimento dell'offerta di volontariato viene annullato.

## Requisiti Speciali

### Requisiti Non Funzionali

- **Usabilità:** L'interfaccia deve essere intuitiva e accessibile.
- **Privacy:** I dati personali del volontario sono condivisi con l'Organizzazione solo dopo l'accettazione dell'offerta.

### Vincoli Tecnici

- **Performance:** Il sistema deve mostrare la mappa e le richieste entro 3 secondi.

## Note Aggiuntive

- **Frequenza:** Il caso d'uso viene eseguito ogni volta che un utente si offre per effettuare volontariato.
- **Criticità:** Alta, in quanto è una delle attività di business principali del sistema software.
- **Note implementative:** Cercare di automatizzare il più possibile le operazioni di matching tra richieste e offerte in modo da non lasciare spazio ad ambiguità.

## 1.2 Caso d'uso UC5: RicercaAbbinata

RicercaAbbinata Daniele Buser, Gabriele Groppo Scadenza: 11 Gennaio 2025

### Informazioni di Progetto

- **Blocked by:** RegistraVolontario, GestioneRichiesteVolontariato, RegistraOrganizzazione, GestioneOfferteVolontariato, ControlloCorrettezzaCampi
- **Disciplina:** Requisiti
- **Scadenza:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
- **Parent-task:** Casi d'Uso testuali
- **Priorità:** Alta

- **Progetto:** DoIT-2

**Nota:** Il caso d'uso **"RicercaAbbinata"** (**UC-005**) descrive il processo attraverso il quale il sistema suggerisce automaticamente opportunità di volontariato agli utenti in base al loro profilo, preferenze e disponibilità. [...] Il processo garantisce la privacy degli utenti e ottimizza le suggestioni per massimizzare la probabilità di match positivi.

## 1. Informazioni Generali

- **ID Caso d'Uso:** UC-005
- **Nome:** RicercaAbbinata
- **Attore Principale:** Sistema

### Stakeholder:

- Volontari: ricevono suggerimenti pertinenti
- Organizzazioni: ottengono visibilità mirata

### Precondizioni:

- Profilo volontario completo con preferenze
- Presenza di richieste attive nel sistema

### Post-condizioni:

- Match potenziali identificati
- Match potenziali mostrati in cima alla lista del volontario

## 2. Scenario Principale

**Obiettivo** Facilitare l'incontro tra domanda e offerta di volontariato attraverso un sistema automatico di matching basato su profili e preferenze

## **Flusso Base**

1. Sistema avvia analisi nel momento in cui il volontario accede alla sua dashboard o modifica filtri
  - Recupera attività volontariato
2. Il sistema per ogni volontario estrae i dati utili al matching
  - Estrae preferenze e vincoli
  - Identifica richieste compatibili
3. Sistema applica algoritmo matching
  - Calcola score compatibilità
  - Ordina risultati per rilevanza
4. Sistema genera suggerimenti
  - Aggiorna dashboard utente

### **3. Scenari Alternativi**

#### **Scenario: Nessun Match Trovato**

1. Viene mostrata una lista non ordinata

### **4. Eccezioni e Gestione Errori**

#### **Errore:**

- Condizione:
- Azione:
- Risultato:

### **5. Requisiti Speciali**

### **Requisiti Non Funzionali**

- Performance: Analisi completata in max 1 minuto
- Accuratezza: Match pertinenti >80%

### **Vincoli Tecnici**

- Rispettare algoritmo di matching

## **6. Note Aggiuntive**

- **Frequenza:** Per ogni accesso utente
- **Criticità:** Alta
- **Note implementative:**
  - Ottimizzare query database
  - Logging risultati matching

## **2 Diagramma dei Casi d’Uso**

Diagramma dei Casi d’Uso Matteo Cervini Scadenza: 12 Gennaio 2025

- **Disciplina:** Requisiti
- **Scadenza:** 12 Gennaio, 2025 2:00 AM (GMT+1) → 12 Gennaio, 2025 7:00 PM (GMT+1)
- **Priorità:** Bassa
- **Progetto:** DoIT-1

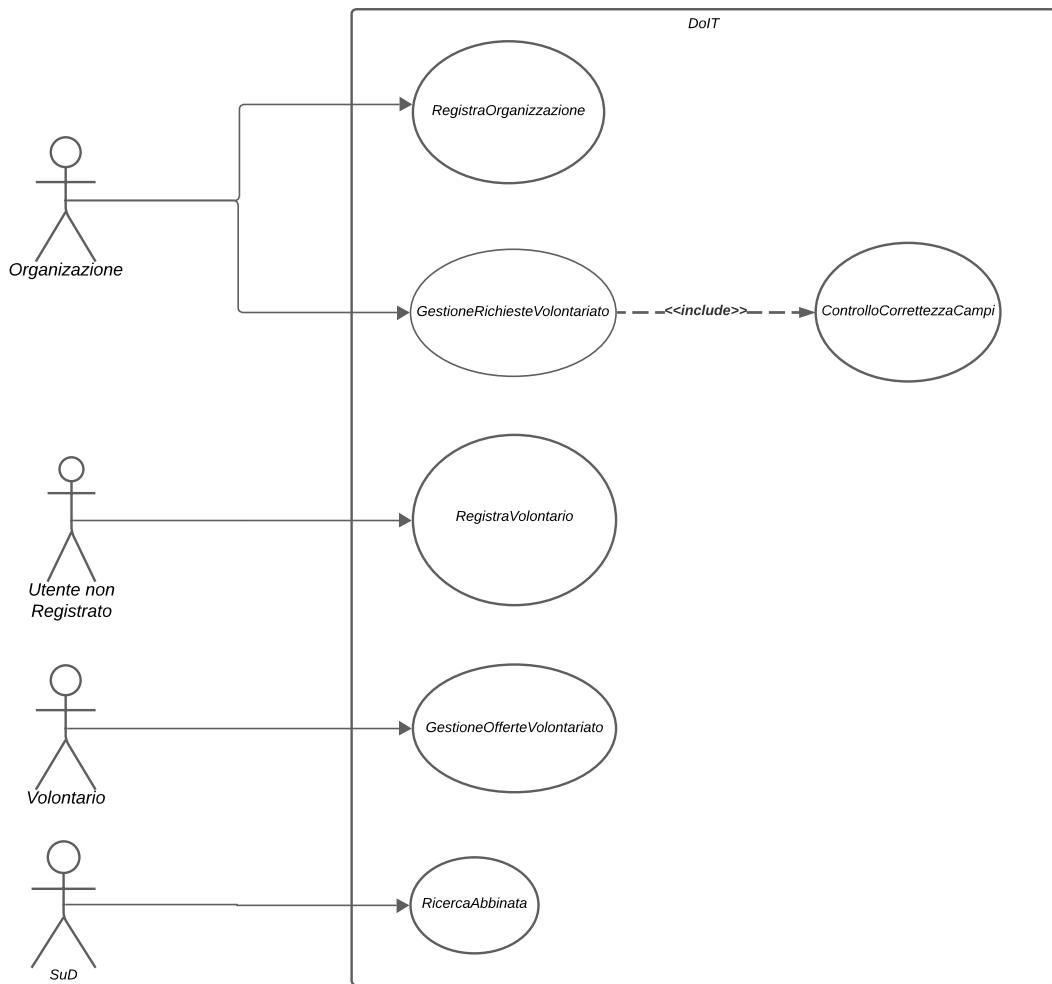


Figure 43: Diagramma dei Casi d’Uso a seguito dell’iterazione 2

### 3 Diagramma di Sequenza di Sistema

#### 3.1 Diagramma di Sequenza di Sistema SSD4: GestioneOfferteVolontariato

SSDGestioneOfferteVolontariato Gabriele Groppo Scadenza: 13 Gennaio 2025

- **Blocked by:** GestioneRichiesteVolontariato ControlloCorrettezzaCampi
- **Disciplina:** Requisiti
- **Scadenza:** 12 Gennaio, 2025 1:00 AM (GMT+1) → 13 Gennaio, 2025 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Is Blocking:** SSDControlloCorrettezzaCampi SSDRicercaAbbinata Contratto-GestioneRichiesteVolontariato
- **Priorità:** Media
- **Progetto:** DoIT-2

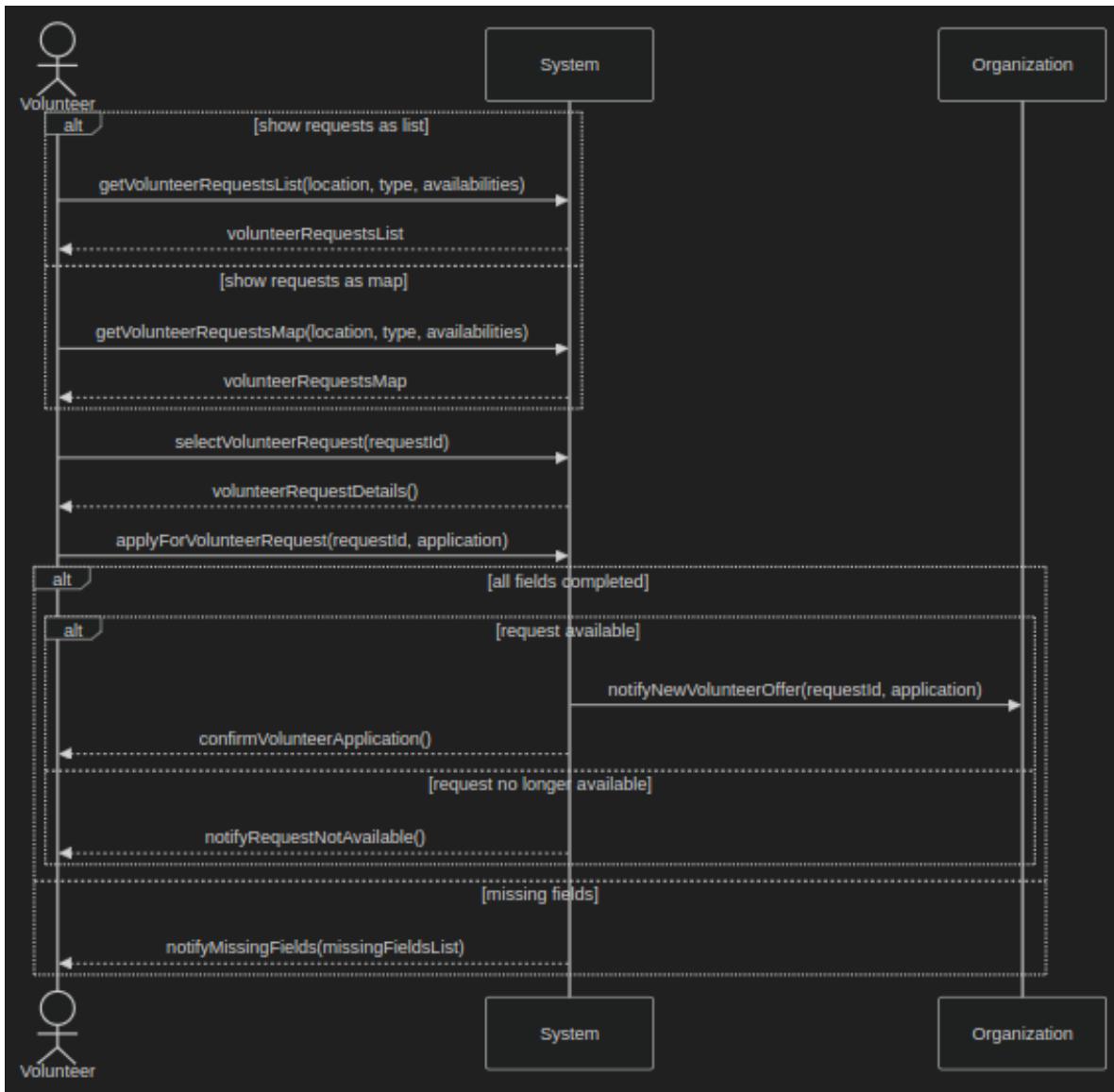


Figure 44: Diagramma di Sequenza di Sistema

### 3.2 Diagramma di Sequenza di Sistema SSD5: RicercaAbbinata

SSDRicercaAbbinata Gabriele Groppo Daniele Buser Scadenza: 13 Gennaio 2025

- **Blocked by:** SSDRegistraVolontario SSDRegistraOrganizzazione SSDGestioneRichiesteVolontariato SSDGestioneOfferteVolontariato SSDControlloCorrettezzaCampi

- **Disciplina:** Requisiti
- **Scadenza:** 12 Gennaio, 2025 1:00 AM (GMT+1) → 13 Gennaio, 2025 12:00 AM (GMT+1)
- **Parent-task:** Diagramma di Sequenza di Sistema
- **Priorità:** Media
- **Progetto:** DoIT-2

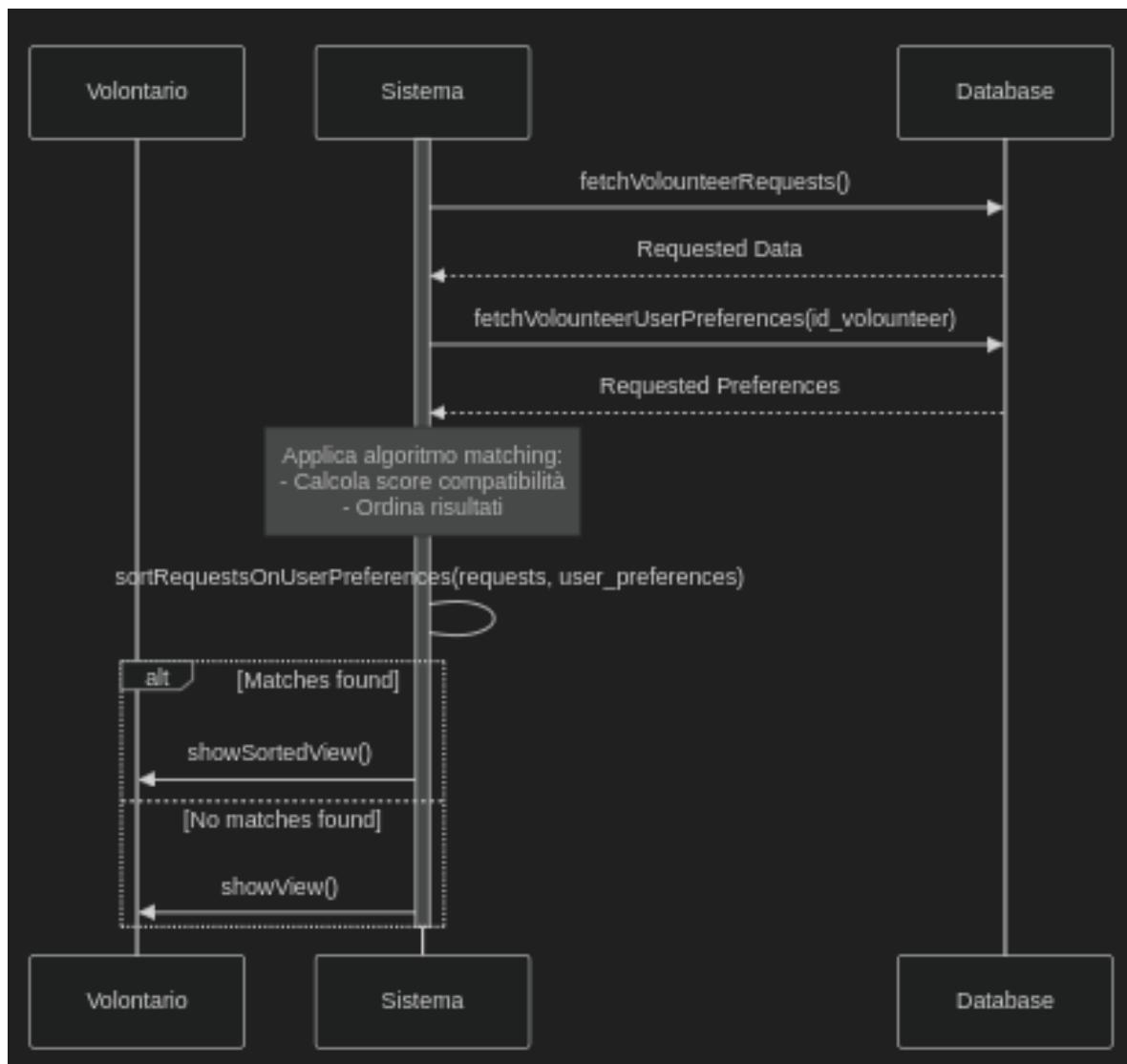


Figure 45: Diagramma di Sequenza di Sistema

## 4 Contratti

### 4.1 Contratto CO03: RegistratiAdUnaRichiestaDiVolontariato

ContrattoRegistratiAdUnaRichiestaDiVolontariato Gabriel Gropo Scadenza: 13 Gennaio, 2025

**Assigned:** Gabriele Gropo

**Disciplina:** Requisiti

**Due:** 13 Gennaio, 2025 1:00 AM (GMT+1) → 13 Gennaio, 2025 11:00 PM (GMT+1)

**Parent-task:** Contratti

**Priority:** Bassa

**Projects:** DoIT-2

## Contratto CO03: applyForVolunteerRequest

**Operazione:** applyForVolunteerRequest(requestId: String, application JSON)

**Riferimenti:** casi d'uso: GestioneOfferteVolontariato

### Pre-Condizioni:

- La richiesta di volontariato identificata da **requestId** deve esistere ed essere attiva nel sistema.
- I dati dell'applicazione (**application**) devono essere completi e conformi ai vincoli definiti.
- Il volontario identificato in **application.volunteerId** deve essere registrato e attivo nel sistema.
- E' in corso l'inserimento di una **Candidatura c**

### Post-Condizioni:

- E' stata creata un istanza *c* di **Candidatura**
- Gli attributi di *c* sono stati inizializzati
- L'attributo di *c*: candidaturaID è stato collegato ad: **applicationId**
- L'istanza *c* è memorizzata nel SuD
- E' stata creata un istanza *n* di **Notifica**
- L'istanza *n* è stata inviata correttamente all'**Organizzazione o**
- L'istanza *v* di **Volontario** ha ricevuto una callback di notifica della sua offerta

## 4.2 Contratto CO04: OrdinaRichiesteSuPreferenzeUtente

ContrattoOrdinaRichiesteSuPreferenzeUtente Daniele Buser Scadenza: 13 Gennaio, 2025

**Assigned:** Daniele Buser

**Disciplina:** Requisiti

**Due:** 13 Gennaio, 2025 1:00 AM (GMT+1) → 13 Gennaio, 2025 11:00 PM (GMT+1)

**Parent-task:** Contratti

**Priority:** Bassa

**Projects:** DoIT-2

## Contratto CO04: sortRequestsOnUserPreferences

**Operazione:** sortRequestsOnUserPreferences(requests, user\_preferences)

**Riferimenti:** casi d'uso: RicercaAbbinata

### Pre-Condizioni:

- Il sistema deve essere attivo.
- Il database deve contenere richieste di volontariato disponibili.
- Le preferenze utente devono essere registrate nel database.

### Post-Condizioni:

- Viene restituita una lista di richieste ordinate in base alla compatibilità con le preferenze utente.
- Se non ci sono corrispondenze, il sistema mostra una vista generica.

## 5 Specifica Supplementare

Sono presenti solo i cambiamenti effettuati durante l'iterazione 2, i contenuti comuni non sono riscritti. Ciò che è stato rimosso presenta una cancellatura, ciò che è nuovo è sottolineato

Specifiche Supplementari Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser  
Scadenza: 11 Gennaio, 2025

### 5.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-2

### 5.2 Requisiti Funzionali

#### Registrazione e Gestione degli Utenti

1. Devono essere richieste informazioni obbligatorie per la registrazione, quali:
  - Nome e cognome (o ragione sociale per aziende/enti)
  - Email e numero di telefono
  - Password sicura con requisiti minimi di complessità

**Server e Servizi di Hosting** Il progetto DoIT utilizzerà una soluzione di hosting cloud per garantire scalabilità e affidabilità:

- **Cloud Provider:**
  - Firebase sarà utilizzato come piattaforma principale.
  - Render è usato per hostare il server
- **Database:**
  - Neon Database hosta il database remoto
  - PostgreSQL per dati strutturati
  - Firestore per dati non strutturati

**Esigenze di Storage** Per la gestione dei dati, il sistema utilizzerà:

- PostgreSQL per dati strutturati
- Firebase Firestore Neon per messaggistica
- Firebase Storage Neon per file utente

## 6 Glossario

Sono presenti solo i cambiamenti effettuati durante l'iterazione 2, i contenuti comuni non sono riscritti. Ciò che è stato rimosso presenta una cancellatura, ciò che è nuovo è sottolineato

Specifica Supplementare Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser  
Scadenza: 11 Gennaio, 2025

### 6.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-2

### 6.2 Glossario Tecnico

- **Account Google**

Sistema di autenticazione richiesto per la registrazione nella piattaforma. Fornisce le informazioni base dell'utente come nome, email e foto profilo attraverso il protocollo OAuth.

- **Autenticazione**

Processo di verifica dell'identità dell'utente effettuato tramite OAuth Google email e password. È un passaggio obbligatorio durante la registrazione che garantisce la sicurezza e l'unicità dell'account.

- **Form Aggiuntivo di Registrazione**

Modulo di registrazione che raccoglie informazioni specifiche del volontario ~~non disponibili tramite l'account Google~~. Include:

- Campi obbligatori da compilare
- Validazione in tempo reale

- **OAuth**

Protocollo di autenticazione utilizzato per l'integrazione con Google, che permette:

- Accesso sicuro ai dati dell'account Google
- Importazione automatica delle informazioni base
- Verifica dell'identità dell'utente

- **Profilo**

Insieme delle informazioni che caratterizzano il volontario, composto da:

- Dati importati da Google (nome, email, foto)
- Informazioni aggiuntive fornite durante la registrazione
- Stato di completezza e verifica

- **Volontario**

Utente registrato nella piattaforma che:

- Possiede un account Google verificato
- Ha completato il processo di registrazione
- Può accedere alle opportunità di volontariato
- Può offrire disponibilità per attività non sovrapposte temporalmente
- Deve specificare competenze e disponibilità temporali

## 3 Modellazione di Business

### 1 Modello delle Classi Concettuali

Modello delle Classi Concettuali Matteo Cervini Andrea Cozzi Gabriele Groppo  
Daniele Buser Scadenza: 14 Gennaio, 2025

- **Disciplina:** Modellazione del Business
- **Scadenza:** 13 Gennaio, 2025 1:00 AM (GMT+1) → 14 Gennaio, 2025 7:00 AM (GMT+1)
- **Priorità:** Alta
- **Progetto:** DoIT-2

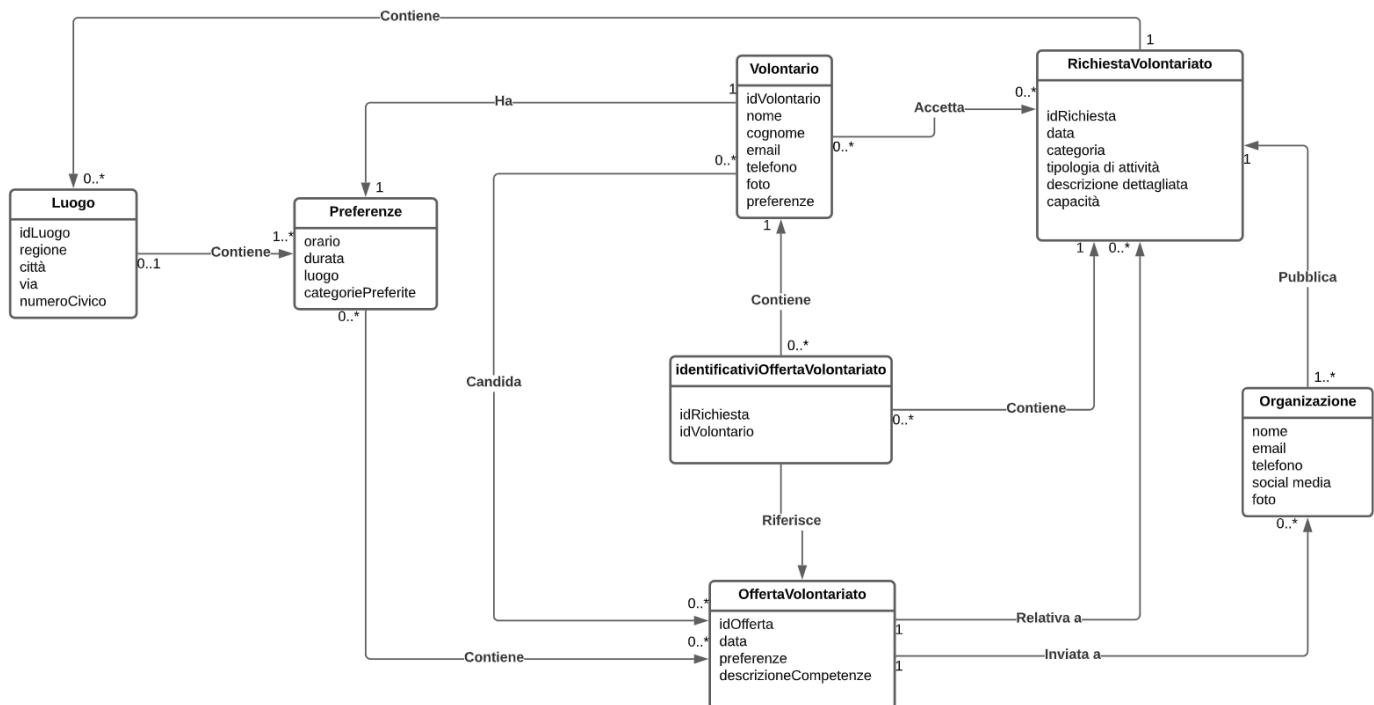


Figure 46: Modello delle Classi Concettuali

## 4 Progettazione

### 1 Diagrammi delle Classi Software di Progetto

#### 1.1 Diagramma delle Classi Software di Progetto DCSP9: Dominio

DCSPDominio Matteo Cervini Gabriele Groppo Andrea Cozzi Daniele Buser Scadenza: 17 Gennaio 2025

- **Assigned** Matteo Cervini Gabriele Groppo Andrea Cozzi
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-2

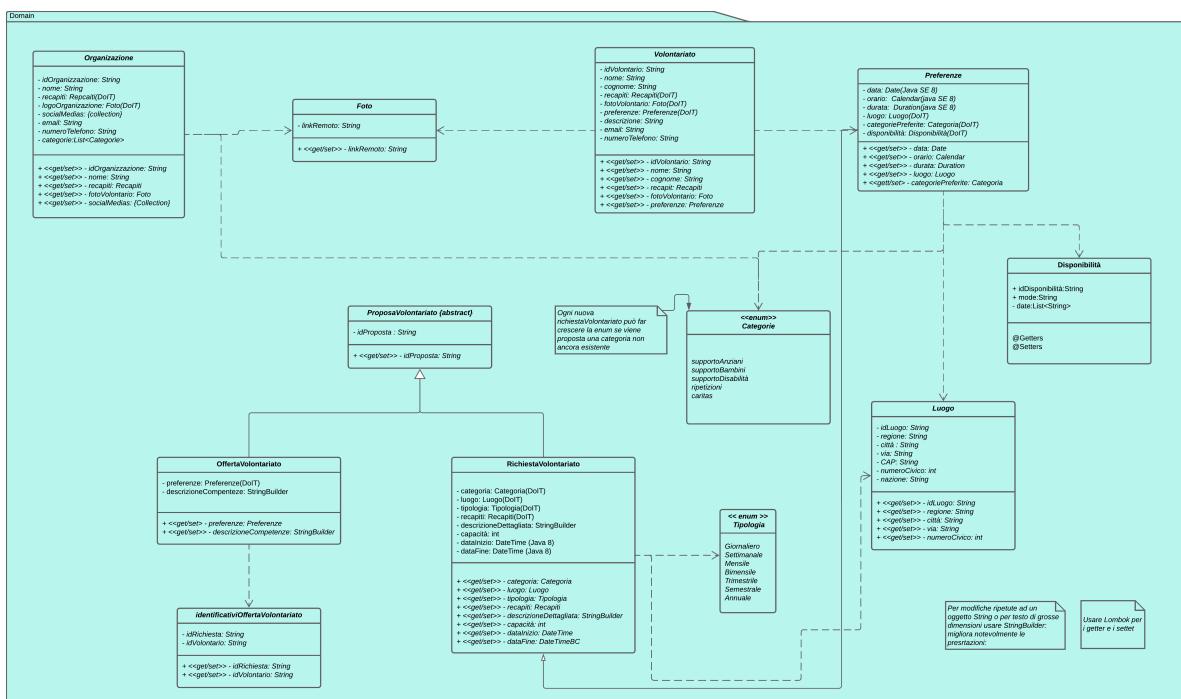


Figure 47: Diagramma delle Classi Software di Progetto

## 1.2 Diagramma delle Classi Software di Progetto DCSP10: Frontend

DCSPFrontend Matteo Cervini Daniele Buser Scadenza: 17 Gennaio 2025

- **Assigned** Matteo Cervini Daniele Buser
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-2



Figure 48: Diagramma delle Classi Software di Progetto

### 1.3 Diagramma delle Classi Software di Progetto DCSP11: Backend/Controller

DCSPController Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Gropo
  - **Blocked by** Requisiti
  - **Disciplina** Progettazione
  - **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
  - **Is blocking** Implementazione
  - **Parent-task** Diagrammi delle Classi Software di Progetto
  - **Priority** Alta
  - **Projects** DoIT-2

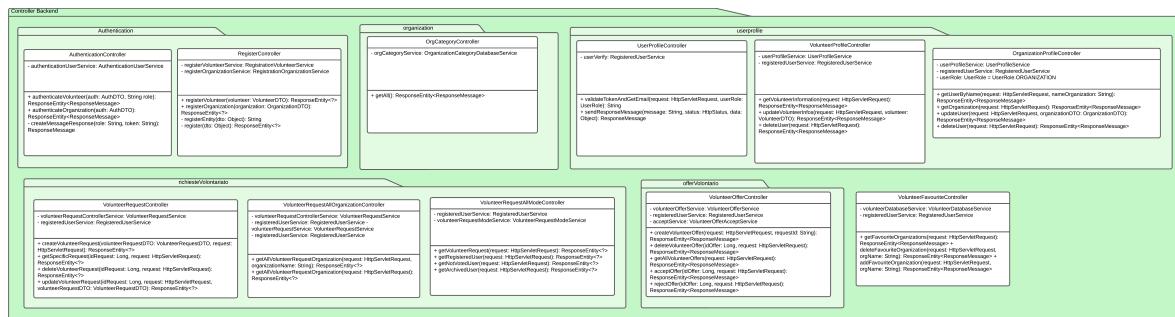


Figure 49: Diagramma delle Classi Software di Progetto

## 1.4 Diagramma delle Classi Software di Progetto DCSP12: Backend/Domain

DCSPDomain Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
  - **Blocked by** Requisiti

- **Disciplina** Progettazione
- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-2

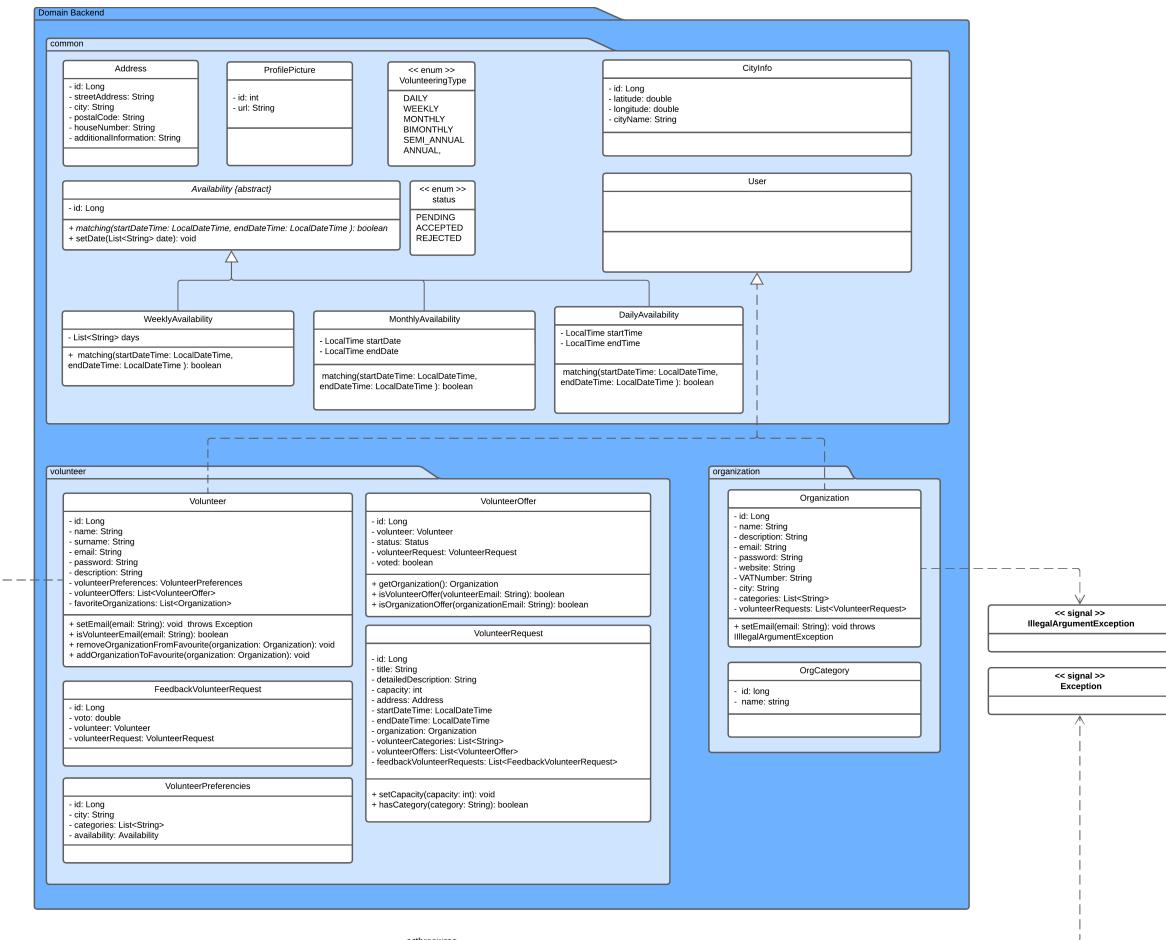


Figure 50: Diagramma delle Classi Software di Progetto

## 1.5 Diagramma delle Classi Software di Progetto DCSP13: Backend/DTO

DCSPDTO Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-2

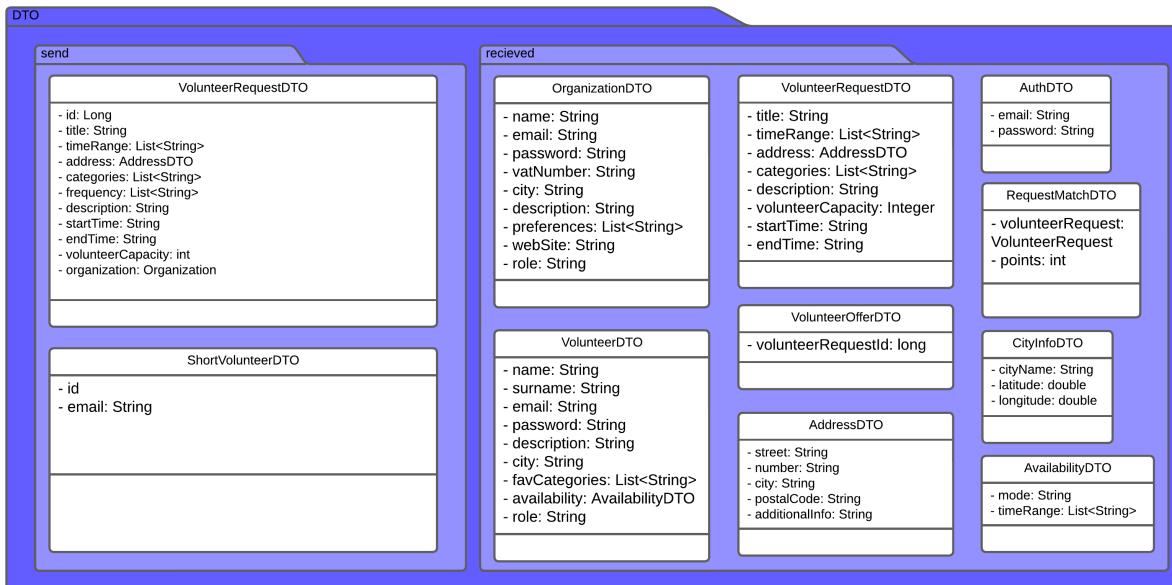


Figure 51: Diagramma delle Classi Software di Progetto

## 1.6 Diagramma delle Classi Software di Progetto DCSP14: Backend/Mappers

DCSPMappers Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione
- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
- **Is blocking** Implementazione
- **Parent-task** Diagrammi delle Classi Software di Progetto
- **Priority** Alta
- **Projects** DoIT-2

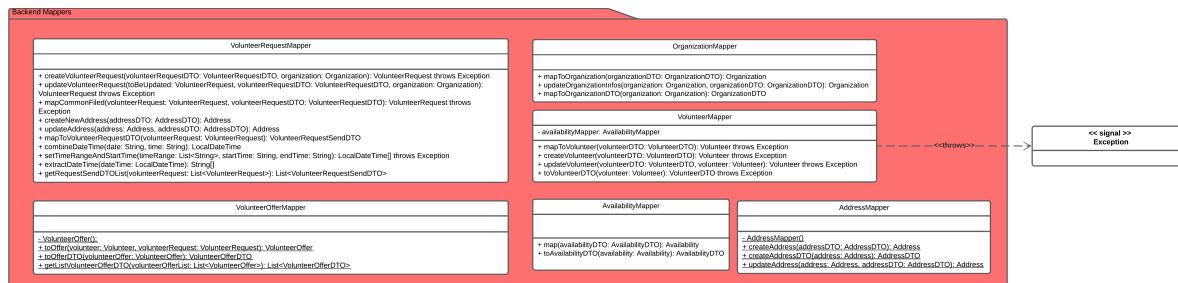


Figure 52: Diagramma delle Classi Software di Progetto

## 1.7 Diagramma delle Classi Software di Progetto DCSP15: Backend/Service

DCSPService Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Disciplina** Progettazione

- **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
  - **Is blocking** Implementazione
  - **Parent-task** Diagrammi delle Classi Software di Progetto
  - **Priority** Alta
  - **Projects** DoIT-2

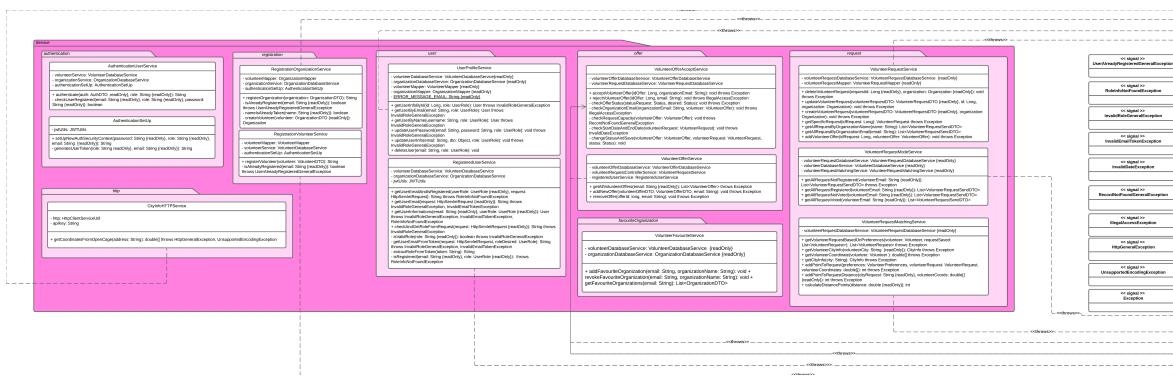


Figure 53: Diagramma delle Classi Software di Progetto

## 1.8 Diagramma delle Classi Software di Progetto DCSP16: Backend/Repository

DCSPRepository Andrea Cozzi Gabriele Groppo Scadenza: 17 Gennaio 2025

- **Assigned** Andrea Cozzi Gabriele Groppo
  - **Blocked by** Requisiti
  - **Disciplina** Progettazione
  - **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
  - **Is blocking** Implementazione
  - **Parent-task** Diagrammi delle Classi Software di Progetto

- Priority Alta
  - Projects DoIT-2

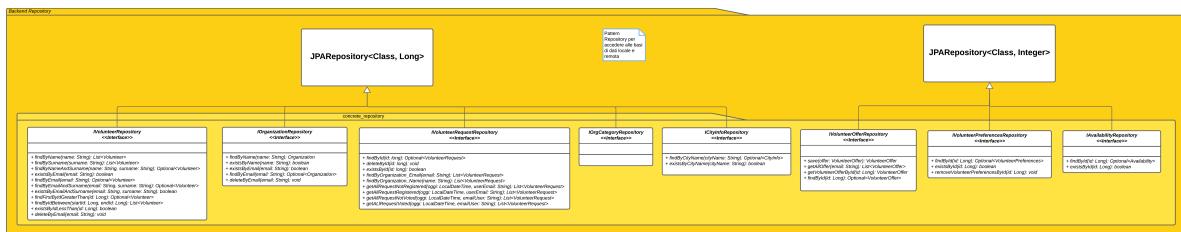


Figure 54: Diagramma delle Classi Software di Progetto

## 2 Diagramma dell'Architettura Logica

## 2.1 Diagramma dell'Architettura Logica

Architettura Logica Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser  
Scadenza: 5 Gennaio, 2025

- **Assigned** Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser
  - **Blocked by** Requisiti
  - **Disciplina** Progettazione
  - **Due** 15 Gennaio, 2025 1:00 AM (GMT+1) → 17 Gennaio, 2025 11:00 PM (GMT+1)
  - **Is blocking** Implementazione
  - **Priority** Alta
  - **Projects** DoIT-2

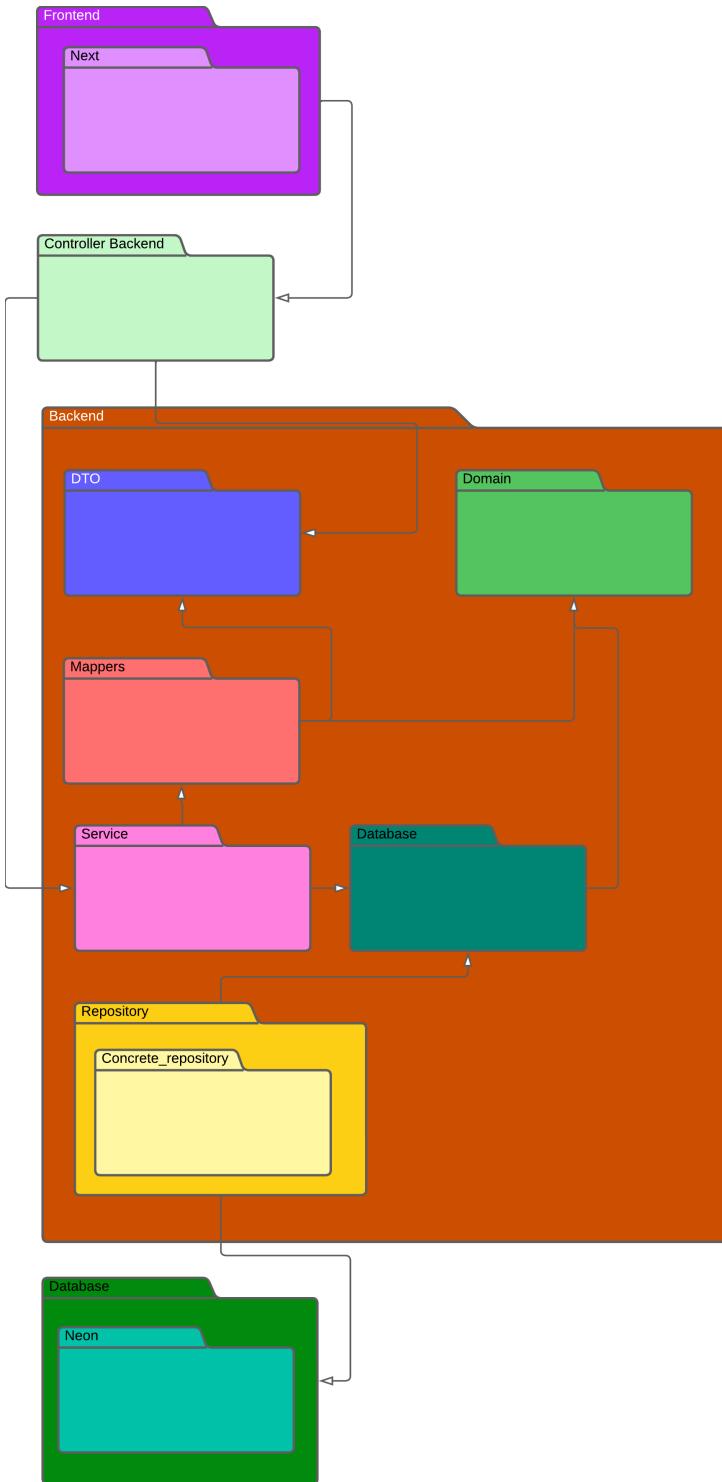


Figure 55: Diagramma dell'Architettura Logica  
104

### 3 Pattern

Pattern Matteo Cervini Daniele Buser Andrea Cozzi Gabriele Groppo Scadenza: 24 Gennaio 2025

- **Assigned** Matteo Cervini Daniele Buser Andrea Cozzi Gabriele Groppo
- **Blocked by** Requisiti
- **Due** 18 Gennaio, 2025 1:00 AM (GMT+1) → 24 Gennaio, 2025 11:00 PM (GMT+1)
- **Priority** Alta
- **Projects** DoIT-2

#### 3.1 Design Patterns

Frontend:

- **Hydration** (Passare dinamicamente i dati dal server al client per renderizzare la pagine e i componenti)
- **React Context** (Stato globale e gestione del contesto)
- **Dynamic Import**

Backend:

- **Singleton**
- **Template** (authRedirectService)
- **DTO**
- **Strategy** (Availability)
- **Mapper**

### 3.2 Pattern Architetturali

Frontend:

- File-Based Routing
- Atomic Design Pattern
- Advanced Folder Structure

Backend:

- Repository
- Service
- Dependency Injection
- Inversion of Control (Programming Principle)

### 3.3 Buone Pratiche

Frontend:

- Skeleton Loading
- Cache e Pre-Fetching
- React Hooks
- Environment Variables
- Styling (Tailwind CSS)
- API Fetching Client Side

Backend:

- Repository
- Service
- DTO
- Dependency Injection
- Inversion of Control (Programming Principle)

### 3.4 Design Principles

Frontend e Backend:

- REP - The Release Reuse Equivalency Principle
- CCP - The Common Closure Principle

# Documenti dei Requisiti nella loro versione finale

## 1 Specifica Supplementare

Specifiche Supplementari: Matteo Cervini, Andrea Cozzi, Gabriele Groppo, Daniele Buser  
Scadenza: 11 Gennaio, 2025

### 1.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-2

### 1.2 Requisiti Funzionali

#### Registrazione e Gestione degli Utenti

1. Gli utenti devono poter creare uno o entrambi gli account scegliendo tra i seguenti profili:
  - Volontario singolo
  - Associazione o organizzazione di volontariato o beneficiario singolo
2. Devono essere richieste informazioni obbligatorie per la registrazione, quali:
  - Nome e cognome (o ragione sociale per aziende/enti)
  - Email
  - Password sicura con requisiti minimi di complessità
3. Ogni utente deve avere la possibilità di completare un profilo personalizzato, comprensivo di:
  - Disponibilità orarie
  - Competenze specifiche
  - Località

## **Gestione delle Richieste di Volontariato**

1. I beneficiari devono poter creare richieste di supporto specificando:
  - Categoria di aiuto
  - Descrizione dettagliata della necessità
  - Data e orario preferiti
  - Localizzazione geografica tramite mappa interattiva
2. Le richieste devono poter essere modificate o cancellate fino a quando non vengono accettate da un volontario.

## **Gestione delle Offerte di Volontariato**

1. I volontari devono poter indicare la propria disponibilità attraverso:
  - Calendario settimanale con fasce orarie selezionabili
  - Tipologia di attività offerte
  - Zone geografiche preferite
2. Devono poter modificare o aggiornare le proprie disponibilità in qualsiasi momento.

## **Sistema di Abbinamento (Matching)**

1. La piattaforma deve includere un algoritmo di matching che:
  - Confronti le richieste dei beneficiari con le disponibilità dei volontari
  - Utilizzi parametri quali categoria, localizzazione e orario per proporre abbinamenti
2. Gli utenti devono ricevere notifiche per ogni proposta di abbinamento rilevante.

## **Sistema di Messaggistica**

1. Deve essere integrato un sistema di chat sicuro per consentire la comunicazione tra volontari e beneficiari.
2. Le funzionalità principali della chat includono:
  - Invio di messaggi testuali
  - Notifiche in tempo reale
  - Cronologia dei messaggi

## **Sistema di Valutazione e Feedback**

1. Dopo il completamento di un'attività, entrambe le parti devono poter lasciare una valutazione.
2. Le recensioni devono includere:
  - Punteggio da 1 a 5 stelle
  - Commento opzionale
3. I punteggi devono essere visibili sui profili pubblici degli utenti.

## **Comitato di Controllo e Validazione**

1. La piattaforma deve consentire a un team dedicato di:
  - Verificare le richieste e le offerte inserite dagli utenti
  - Segnalare e rimuovere contenuti inappropriati
2. Deve essere previsto un sistema per ricevere e gestire segnalazioni dagli utenti.

## **Integrazione con Social Media**

1. Gli utenti devono poter condividere richieste e offerte di volontariato sui principali social network.

### **1.3 Requisiti Non Funzionali**

#### **Prestazioni**

1. Il sistema deve garantire tempi di risposta inferiori a 2 secondi per il caricamento delle principali pagine e operazioni.
2. L'algoritmo di matching deve essere in grado di elaborare richieste e offerte in meno di 5 secondi.
3. Aggiornamento in tempo reale delle notifiche di chat

#### **Scalabilità**

1. Deve essere possibile aumentare la capacità del sistema con l'aumento del numero di utenti.

## **Sicurezza**

1. I dati degli utenti devono essere protetti attraverso crittografia sia a riposo che in transito.

## **Compatibilità**

1. L'app deve essere compatibile con:
  - iOS
  - Android
2. Se previsto un portale web, deve essere ottimizzato per i principali browser.

## **Usabilità**

1. L'interfaccia deve essere intuitiva e accessibile
2. Deve essere garantita un'esperienza coerente su tutti i dispositivi
3. Il processo di offerta deve essere completabile in pochi passaggi intuitivi

## **Manutenibilità**

1. Il codice deve essere scritto seguendo standard di programmazione modulari
2. Deve essere garantito il supporto per aggiornamenti regolari della piattaforma

## **Disponibilità e Affidabilità**

1. La piattaforma deve essere operativa almeno il 99,5% del tempo su base mensile.

## **Privacy**

1. I dati personali del volontario devono essere condivisi con l'organizzazione solo dopo l'accettazione dell'offerta
2. Le conversazioni nella chat devono essere protette e accessibili solo alle parti coinvolte

## 1.4 Requisiti Hardware

**Server e Servizi di Hosting** Il progetto DoIT utilizzerà una soluzione di hosting cloud per garantire scalabilità e affidabilità:

- **Cloud Provider:** Render è usato per hostare il server
- **Database:**
  - Neon Database hosta il database remoto
  - PostgreSQL per dati strutturati

**Esigenze di Storage** Per la gestione dei dati, il sistema utilizzerà:

- PostgreSQL per dati strutturati
- Neon per messaggistica
- Neon per file utente

## 1.5 Requisiti Software

### Tecnologie Preferite per lo Sviluppo

- **Frontend:**
  - HTML, CSS e TailwindCSS
  - TypeScript e Next.js
- **Backend:**
  - Java con Spring Boot
  - Spring Security
  - Spring Data JPA
  - OpenAPI
- **Database:**
  - PostgreSQL
  - Flyway

- **Tools/Utilities:**
  - Git
  - Maven
  - npm
  - SonarQube
  - Understand
  - Mermaid Live e Lucidchart

## Integrazioni con Altre Piattaforme o API Esterne

- Geolocalizzazione (Google Maps o Mapbox)
- Social Media (Facebook, Twitter, LinkedIn)

### 1.6 Specifiche di Interfaccia

#### Requisiti per il Design dell'Interfaccia Utente

##### 1. Design Responsivo:

- Interfaccia completamente responsive
- Utilizzo di TailwindCSS

##### 2. Estetica Pulita e Semplice:

- Design minimale
- Colori chiari e contrastanti
- Interfaccia coerente

##### 3. Navigazione Intuitiva:

- Barra di navigazione chiara
- Funzionalità di ricerca avanzata
- Sistema di notifiche visibile

##### 4. Iconografia Chiara:

- Icônes compréhensibles

- Stile uniforme

##### 5. Accessibilità:

- Conformità WCAG 2.1
- Supporto screen reader
- Font leggibili

#### Linee Guida per l'Esperienza Utente (UX)

1. Semplicità e Chiarezza
2. Flusso Intuitivo
3. Feedback Visivo e Interattivo
4. Personalizzazione
5. Chiarezza nelle Interazioni
6. Performance e Velocità

#### 1.7 Vincoli di Progetto

##### Budget disponibile per sviluppo e manutenzione

- Progetto universitario senza budget specifico
- Sviluppo da parte di team accademico

##### Tempi di rilascio per MVP e versioni successive

- **MVP:** Completamento entro 1 mese
- **Versioni successive:** Cicli mensili di aggiornamento

## **1.8 Gestione del Rischio**

### **Rischi Aggiuntivi**

- 1. Mancanza di partecipazione da parte degli utenti**
- 2. Problemi di sicurezza e privacy dei dati**
- 3. Difficoltà tecniche nell'integrazione**
- 4. Scalabilità limitata nell'MVP**
- 5. Ritardi nello sviluppo**

### **Strategie per Mitigare i Rischi**

- 1. Strategia per l'integrazione tecnica:**
  - Seguire guide ufficiali
  - Creare prototipi
- 2. Strategia per la scalabilità:**
  - Architettura modulare
  - Load testing
- 3. Strategia per i ritardi:**
  - Pianificazione agile
  - Valutazione risorse aggiuntive

## **2 Glossario**

Specifico Supplementare Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele Buser Scadenza: 11 Gennaio, 2025

## 2.1 Nota Introduttiva

- **Disciplina:** Requisiti
- **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
- **Priority:** Medio
- **Projects:** DoIT-2

## 2.2 Glossario Tecnico

- **Autenticazione**

Processo di verifica dell'identità dell'utente effettuato tramite email e password. È un passaggio obbligatorio durante la registrazione che garantisce la sicurezza e l'unicità dell'account.

- **Crittografia**

Tecnica di sicurezza utilizzata per proteggere i dati sensibili degli utenti nel sistema. È un requisito non funzionale che garantisce la protezione delle informazioni personali memorizzate.

- **Dashboard Personale**

Interfaccia utente principale mostrata dopo il login, contenente:

- Opportunità di volontariato consigliate
- Riepilogo del profilo utente
- Altre funzionalità personalizzate

- **Form di Registrazione**

Modulo di registrazione che raccoglie informazioni specifiche del volontario. Include:

- Campi obbligatori da compilare
- Validazione in tempo reale

- **Login Automatico**

Processo che avviene dopo la registrazione completata con successo, permettendo l'accesso immediato alla dashboard personale senza necessità di inserire nuovamente le credenziali.

- **Opportunità**

Attività di volontariato disponibili sulla piattaforma che vengono:

- Consigliate ai volontari sulla dashboard
- Filtrate in base al profilo dell'utente e area geografica
- Presentate dopo la registrazione
- Pubblicate solo da Organizzazioni verificate
- Non sovrapponibili temporalmente per lo stesso volontario

- **Organizzazione**

Tipologia di account alternativa a quella del volontario, con funzionalità e processo di registrazione specifici.

- **Profilo**

Insieme delle informazioni che caratterizzano il volontario, composto da:

- Informazioni aggiuntive fornite durante la registrazione
- Stato di completezza e verifica

- **Registrazione**

Processo di creazione di un nuovo account volontario che include:

- Verifica dei prerequisiti (account Google)
- Raccolta dati tramite form
- Validazione in tempo reale
- Tempo massimo di completamento: 5 minuti

- **Storage Sicuro**

Sistema di memorizzazione dei dati personali che implementa:

- Crittografia delle informazioni sensibili

- Protezione dei dati personali
- Conformità con i requisiti di sicurezza

- **Validazione Real-time**

Sistema di controllo immediato dei dati inseriti che:

- Verifica la correttezza delle informazioni
- Mostra messaggi di errore specifici
- Impedisce la sottomissione di dati non validi

- **Volontario**

Utente registrato nella piattaforma che:

- Ha completato il processo di registrazione
- Può accedere alle opportunità di volontariato
- Può offrire disponibilità per attività non sovrapposte temporalmente
- Deve specificare competenze e disponibilità temporali

- **Area Geografica**

Zona territoriale in cui il volontario è disponibile ad operare, specificata durante l'offerta di disponibilità per un'attività di volontariato.

- **Chat Integrata**

Sistema di messaggistica interno alla piattaforma che permette la comunicazione diretta tra Volontari e Organizzazioni dopo l'offerta di disponibilità.

- **Competenze**

Capacità e abilità specifiche richieste dalle Organizzazioni e selezionabili dai Volontari durante l'offerta di disponibilità per un'attività.

- **Mappa Interattiva**

Visualizzazione geografica delle richieste di volontariato disponibili che permette ai volontari di:

- Visualizzare la distribuzione delle opportunità sul territorio
- Interagire direttamente con le richieste

- Filtrare le attività per zona
  - **Organizzazione Verificata**

Status speciale assegnato alle organizzazioni che hanno completato un processo di verifica nel sistema, requisito necessario per pubblicare richieste di volontariato.
  - **Controllo Duplicato Account**

Sistema che verifica l'esistenza di un account già registrato associato alla stessa organizzazione, impedendo la duplicazione degli account e fornendo all'utente le alternative per risolvere eventuali conflitti.
  - **Errore di Registrazione**

Messaggio che il sistema mostra quando il processo di registrazione non può essere completato, generalmente dovuto a dati mancanti, errati o duplicati. L'errore fornisce solitamente dettagli specifici per correggere il problema.
  - **Conferma di Registrazione**

La fase finale del processo di registrazione in cui l'organizzazione approva i dati inseriti e conferma la creazione del proprio account, permettendo al sistema di procedere con il login automatico e il reindirizzamento alla dashboard.
  - **Piattaforma**

Il sistema complessivo che include sia le funzionalità per i volontari che quelle per le organizzazioni, come la gestione dei profili, l'accesso alle opportunità di volontariato e le interazioni tra le diverse entità.
- ### 3 Visione
- Specifica Supplementare Matteo Cervini Andrea Cozzi Gabriele Groppo Daniele  
Buser Scadenza: 11 Gennaio, 2025
- #### 3.1 Nota Introduttiva
- **Disciplina:** Requisiti
  - **Due:** 11 Gennaio, 2025 1:00 AM (GMT+1) → 11 Gennaio, 2025 7:00 PM (GMT+1)
  - **Priority:** Medio
  - **Projects:** DoIT-2

**Scopo del Documento** Il presente documento descrive la visione per il progetto **DoIT**, una piattaforma innovativa dedicata alla gestione delle richieste e delle offerte di volontariato. L’obiettivo principale è facilitare la connessione tra chi offre il proprio tempo e chi necessita di supporto, con un focus iniziale sulla città di Milano e, successivamente, sull’intera Lombardia. Il progetto è ispirato all’app “Attivati”, già operativa nel Trentino, ma mira a proporre un’evoluzione adattata al contesto lombardo, introducendo nuove funzionalità tecniche avanzate e rispondendo alle specifiche necessità locali.

## Descrizione del Problema

- **Contesto attuale:** La gestione del volontariato soffre spesso di una mancata connessione tra chi offre aiuto e chi lo richiede. Le opportunità di volontariato non sempre vengono adeguatamente pubblicizzate o rese accessibili, creando un disallineamento tra domanda e offerta.
- **Necessità principali:** Vi è il bisogno di una piattaforma centralizzata e user-friendly che consenta di gestire in modo efficace le richieste e le offerte di volontariato.
- **Benefici attesi:** Un aumento della partecipazione al volontariato, una migliore distribuzione delle risorse e una maggiore trasparenza e sicurezza per tutti i partecipanti.

## Stakeholder

- **Cittadini:** Persone disposte a offrire il proprio tempo per il volontariato.
- **Associazioni di volontariato:** Organizzazioni che coordinano e gestiscono attività di volontariato.
- **Aziende ed enti:** Realtà interessate a promuovere il volontariato aziendale.
- **Beneficiari:** Persone o gruppi che necessitano di supporto.
- **Comitato di controllo:** Team responsabile di verificare la veridicità delle informazioni e garantire la sicurezza della piattaforma.

## **Obiettivi del Sistema**

1. Creare un ambiente digitale sicuro e intuitivo per la gestione del volontariato.
2. Facilitare il matching tra richieste e offerte di volontariato.
3. Garantire la sicurezza e la qualità delle interazioni tra gli utenti.
4. Promuovere il volontariato come pratica sociale attraverso l'uso di tecnologie avanzate.

## **Funzionalità Principali**

### **1. Registrazione degli utenti:**

- Creazione di profili per volontari, associazioni, aziende ed enti.
- Inserimento di informazioni su disponibilità, competenze e preferenze.

### **2. Gestione delle richieste di volontariato:**

- Inserimento di richieste specifiche (es. supporto per anziani, studenti, ecc.).
- Dettaglio di durata, orari e posizione geografica.
- Descrizione dettagliata delle necessità.

### **3. Gestione delle offerte di volontariato:**

- Inserimento di disponibilità di tempo e competenze.
- Localizzazione della disponibilità e delle aree di intervento.

### **4. Sistema di abbinamento e ricerca:**

- Algoritmi di matching basati su categorie, orari e posizione geografica.
- Filtri per ricerca avanzata.

### **5. Comitato di controllo e validazione:**

- Verifica delle richieste e delle offerte.
- Segnalazione di contenuti inappropriati.

### **6. Sistema di messaggistica integrato:**

- Chat per comunicazioni dirette tra volontari e richiedenti.
- Notifiche in tempo reale.

#### **7. Sistema di valutazione e feedback:**

- Recensioni degli utenti per garantire qualità e affidabilità.

#### **8. Integrazione con social media:**

- Condivisione delle attività e delle disponibilità sui social network.

### **Ambito del Sistema**

- **Incluso:**

- Funzionalità principali elencate sopra.
- Focus iniziale su Milano con espansione successiva alla Lombardia.

- **Escluso:**

- Gestione di richieste fuori dal territorio lombardo (nella fase iniziale).
- Supporto per lingue diverse dall’italiano.

### **Vincoli**

- **Tecnologici:** L’app deve essere compatibile con i principali sistemi operativi (iOS e Android).
- **Economici:** Rispetto del budget assegnato per lo sviluppo e la manutenzione ossia 0.
- **Temporali:** Lancio della prima versione entro 1 mese dall’inizio dello sviluppo.

### **Rischi e Assunzioni**

- **Rischi:**

- Mancata partecipazione degli utenti.
- Problemi tecnici durante lo sviluppo.
- Difficoltà nel garantire la sicurezza delle interazioni.

- **Assunzioni:**

- Gli utenti disporranno di smartphone per accedere all'app.
- Esiste una domanda significativa di volontariato nella regione.

**Conclusione** Il progetto **DoIT** si propone come uno strumento essenziale per favorire il volontariato nella regione Lombardia, con particolare attenzione alla città di Milano. Grazie all'integrazione di tecnologie avanzate e a un approccio centrato sugli utenti, l'app rappresenta un passo avanti verso una comunità più solidale e connessa.

## Understand

### 1 Analisi Generale

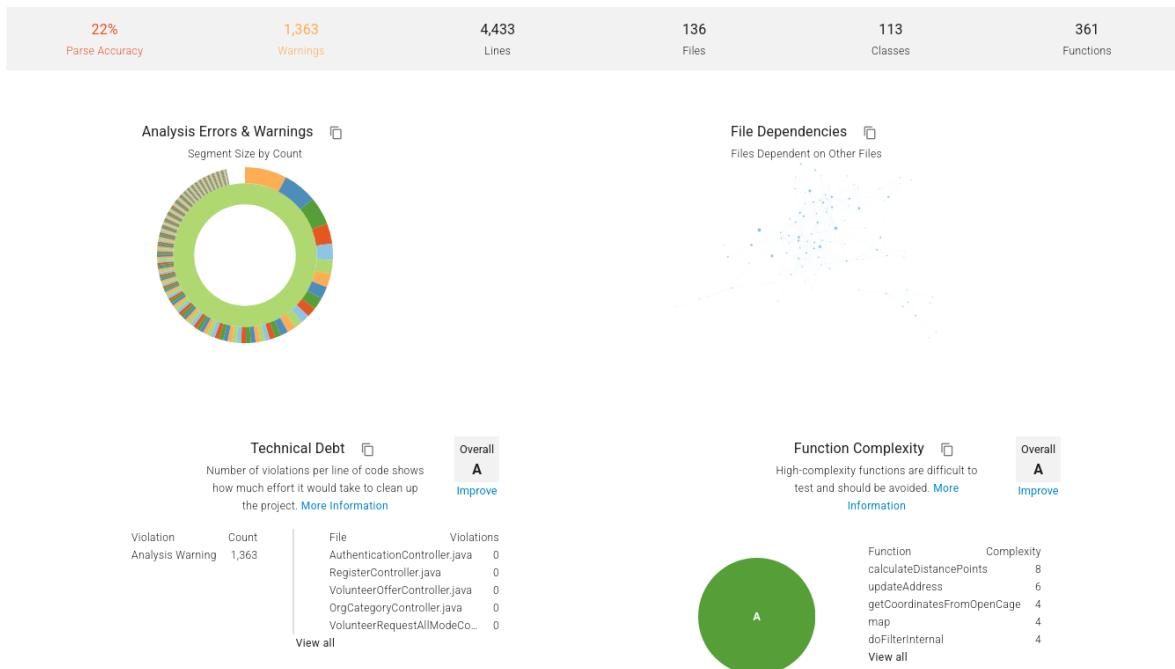


Figure 56: Immagine dell'Analisi Generale

### 2 Metrics Treemaps

Treemap che mostra le metriche **Countline: 90** e **CountLineCode: 75** individuate da Understand

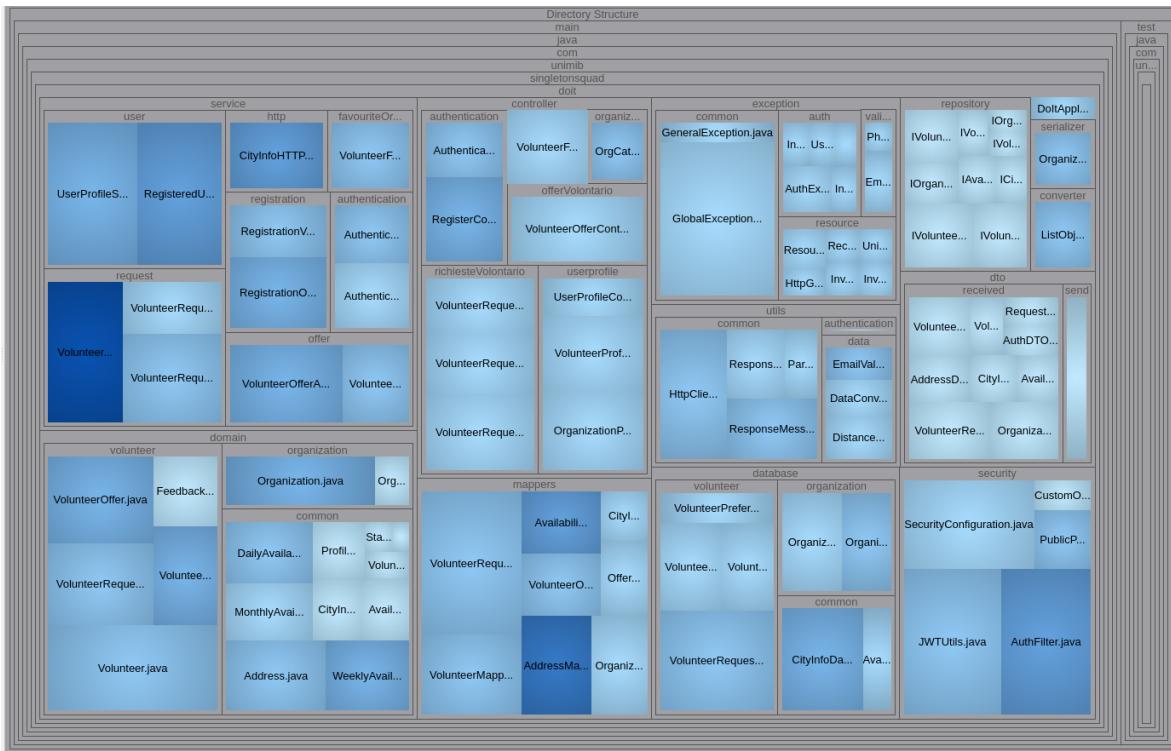


Figure 57: Immagine dell’Analisi sulle Linee di Codice

### 3 Metriche Generali

Le metriche individuate da Understand sul package **doit** sono le seguenti:

- **Blank Lines:** 759
- **Classes:** 113
- **Code Lines:** 3529
- **CodeCheck Violation Density by Code Lines:** 0.00
- **CodeCheck Violation Density by Lines:** 0.00
- **CodeCheck Violation Types:** 0
- **CodeCheck Violation Types:** 0
- **Comment Lines:** 160
- **Comments to Code Ratio:** 0.04
- **Declarative Statements:** 1717
- **Executable Statements:** 879
- **Executable Statements:** 107

Questa analisi ci comunica che abbiamo un assenza di violazioni, densità: 0.00. Ciò significa che il codice è ben strutturato e segue standard definiti, 113 classi, 1717 dichiarazioni, e 879 istruzioni eseguibili indicano un codice con una complessità moderata. Il rapporto commenti/codice di 0.04 è leggermente basso ma comunque accettabile. Il codice è ben strutturato e non presenta violazioni di standard, il che è fondamentale per la manutenibilità e la qualità del software.

### 4 Metriche del file: **VolunteerRequestMatchingService**

Analisi del file: **VolunteerRequestMatchingService** in quanto è il file con l'indicatore di colore più alto.

- **Blank Lines:** 759
- **Classes:** 113
- **Code Lines:** 3529
- **CodeCheck Violation Density by Code Lines:** 0.00
- **CodeCheck Violation Density by Lines:** 0.00
- **CodeCheck Violation Types:** 0
- **Comment Lines:** 160
- **Comments to Code Ratio:** 0.04
- **Declarative Statements:** 1717
- **Executable Statements:** 879
- **Executable Statements:** 107

Questa analisi in un file con un alto indicatore di colore ci comunica che abbiamo un assenza di violazioni, densità: 0.00. È infatti ben strutturato e privo di violazioni delle regole di codifica. Ha una complessità logica moderata, con un numero significativo di dichiarazioni e istruzioni eseguibili. Concludendo abbiamo una buona qualità codice.

## 5 Conclusione

L'analisi ci ha portato a concludere i seguenti Anti-Patterns:

- **Carenza di Documentazione:** Comments to Code Ratio: 0.04
- **God Object:** Numero elevato di Classi: 113