

Relazione CareChain

Luca Rossini

Lorenzo Raia

`l.rossini18@campus.unimib.it`

`l.raia@campus.unimib.it`

September 6, 2026

Sommario

1	Glossario	6
2	Ideazione	8
2.1	Introduzione e visione del problema	8
2.2	Business case	8
2.3	Diagramma di Gantt	9
2.4	Attori del sistema	9
2.5	Definizione obiettivi utente	10
2.6	Analisi dei requisiti	10
2.6.1	Funzionalità	10
	Utente non registrato	10
	Volontario	11
	Organizzazione	12
	Operatore	13
2.6.2	Specifiche supplementari	14
	Usabilità	14
	Affidabilità	14
	Prestazioni	15
	Manutenibilità	15
	Sicurezza	15
	Portabilità	15
	Conformità normativa	16
2.7	Diagramma dei casi d'uso	17
2.8	Descrizione breve dei casi d'uso	17
2.9	Casi d'uso dettagliati	22
2.9.1	Crea richiesta	22
2.9.2	Crea offerta	25
2.9.3	Visualizza richiesta	27
2.9.4	Visualizza offerta	29
2.9.5	Invia messaggio	31
2.9.6	Login	33
2.10	Prototipazione dell'interfaccia	37
2.11	Pianificazione 1° Iterazione	38

3	1° iterazione	40
3.1	Diagramma di Gantt	40
3.2	Scelte architettureali	40
3.3	Modello di dominio	41
3.4	Diagrammi di sequenza di sistema (SSD)	42
3.4.1	Visualizza Richiesta	42
3.4.2	Visualizza Offerta	42
3.4.3	Login	43
3.5	Diagrammi di attività	44
3.5.1	Crea offerta	44
3.6	Contratti	45
3.6.1	Crea Richiesta	45
3.6.2	Crea Offerta	46
3.7	Modello di architettura logica	47
3.8	Diagramma delle classi di progetto	48
3.9	Diagrammi di sequenza (SD)	49
3.9.1	Crea richiesta	49
3.9.2	Crea offerta	49
3.10	Testing e analisi	49
3.11	Pianificazione 2° Iterazione	50
4	2° iterazione	50
4.1	Diagramma di Gantt	50
4.2	Casi d'uso dettagliati	51
4.2.1	Registra organizzazione	51
4.2.2	Registra volontario	53
4.2.3	Approva richiesta	56
4.2.4	Approva offerta	58
4.2.5	Segnala richiesta	60
4.2.6	Segnala offerta	61
4.3	Modello di dominio	64
4.4	Diagrammi di sequenza di sistema (SSD)	65
4.4.1	Registra volontario	65
4.4.2	Registra organizzazione	65
4.4.3	Segnala offerta	66
4.4.4	Segnala richiesta	66
4.4.5	Verifica segnalazione offerta	67
4.4.6	Verifica segnalazione richiesta	67

4.5	Diagrammi di attività	68
4.5.1	Verifica segnalazione richiesta	68
4.6	Contratti	69
4.6.1	RegistraVolontario	69
4.6.2	VerificaSegnalazioneRichiesta	69
4.7	Diagramma delle classi di progetto	71
4.8	Diagrammi di sequenza (SD)	72
4.8.1	Registra volontario	72
4.8.2	Verifica segnalazione richiesta	73
4.9	Pianificazione 3° Iterazione	73
5	3° iterazione	74
5.1	Diagramma di Gantt	74
5.2	Casi d'uso dettagliati	74
5.2.1	Ricerca richiesta	74
5.2.2	Visualizza richieste abbinate	76
5.3	Diagrammi di attività	78
5.3.1	Ricerca richiesta	78
5.4	Modello di architettura logica	79
5.5	Diagramma delle classi di progetto	80
5.6	Diagrammi di sequenza (SD)	81
5.6.1	Ricerca richiesta	81
5.6.2	Visualizza richieste abbinate	81
5.7	Testing e analisi	81
5.8	Pianificazione 4° Iterazione	82
6	4° iterazione	83
6.1	Diagramma di Gantt	83
6.2	Casi d'uso dettagliati	83
6.2.1	Modifica richiesta	83
6.2.2	Modifica dati profilo	86
6.3	Modello di dominio	88
6.4	Diagrammi di sequenza di sistema (SSD)	89
6.4.1	Invia messaggio	89
6.5	Diagrammi di attività	90
6.5.1	Modifica profilo volontario	90
6.6	Modello di architettura logica	91
6.7	Contratti	92

6.7.1	Modifica Richiesta	92
6.7.2	Crea chat	93
6.8	Diagramma delle classi di progetto	94
6.9	Diagrammi di sequenza (SD)	95
6.9.1	Crea chat	95
6.9.2	Modifica offerta	95
6.10	Diagrammi di macchine a stati	96
6.10.1	Richiesta	96
6.11	Testing e analisi	96
6.12	Pianificazione 4° Iterazione	97

1 Glossario

attività Si intende un'azione concreta svolta da un volontario per conto di un'organizzazione, in seguito all'accettazione di una richiesta di volontariato.. 9, 10, 12, 13, 18, 19

Better-Auth Libreria che aiuta a gestire l'autenticazione degli utenti in modo semplice e sicuro.. 40

express Libreria per Node.js che semplifica la creazione di server web e API.. 40

FURPS+ Modello usato per classificare e analizzare i requisiti software. È stato sviluppato da Hewlett-Packard (HP) e aiuta a valutare la qualità di un sistema in base a diversi aspetti. 10

MySQL Sistema per gestire database, usato per salvare e organizzare grandi quantità di dati.. 40

Next.js Next.js è un framework basato su React che semplifica la creazione di siti web e applicazioni. Offre strumenti utili per gestire le pagine, i dati e la navigazione.. 40

node.js Ambiente che permette di eseguire codice JavaScript fuori dal browser, ad esempio su un server.. 40

notifica Si intende un avviso inviato dal sistema a un utente per informarlo su eventi rilevanti, come nuove richieste, offerte, messaggi o aggiornamenti relativi al proprio profilo.. 14

offerta Si intende un annuncio pubblicato da un volontario per segnalare la propria disponibilità a svolgere attività di volontariato.. 10–14, 18–22, 25–27, 29–31, 58, 59, 61–63

operatore Si intende un utente con permessi speciali all'interno del sistema, incaricato della verifica e moderazione delle offerte e richieste, della gestione degli utenti e del monitoraggio delle attività della piattaforma.. 10–14, 18, 19, 21–23, 25, 33, 34, 51, 56–63

- organizzazione** Si intende qualsiasi tipo di associazione, ente o organizzazione registrata nel sistema, che richiede l'intervento di attività di volontariato, ossia gli utenti che necessitano di supporto o servizio da parte dei volontari.. 10–13, 17–19, 21–25, 27, 29–31, 33, 51, 52, 56–58, 61–63, 83–86
- Playwright** Framework open-source sviluppato da Microsoft per l'automazione di test end-to-end su applicazioni web. Supporta diversi browser (Chromium, Firefox, WebKit) e consente di simulare interazioni utente per verificare il corretto funzionamento dell'interfaccia e dei flussi applicativi. 49, 81, 96
- richiesta** Si intende un annuncio pubblicato da un'organizzazione per cercare volontari disponibili a offrire il proprio supporto in base a specifiche esigenze.. 10–14, 17–24, 27–29, 56–58, 60, 61, 74–77, 83–85
- sistema** Si intende l'insieme di componenti software e hardware che lavorano insieme per fornire le funzionalità richieste. 10–36, 51–63, 75–77, 84–87
- SonarQube** Strumento che analizza il codice sorgente per trovare errori, problemi di stile e vulnerabilità.. 49, 81, 96
- sviluppatore** Si intende la persona responsabile della progettazione, sviluppo e manutenzione dell'applicazione, garantendone il corretto funzionamento e l'evoluzione nel tempo. 24, 27, 28, 30, 33, 57, 59, 61, 63, 77, 85, 87
- utente** Si intende qualsiasi tipo persona registrata all'interno del sistema, quindi organizzazioni e volontari.. 9, 10, 14, 15, 19–21, 27–36, 51–55, 75–77, 86, 87
- utente non registrato** Si intende una persona che visita la piattaforma per la prima volta senza aver completato il processo di registrazione.. 10, 11, 19–21, 27, 29, 51–56, 74, 75
- volontario** Si intende una persona che sceglie liberamente di offrire il proprio tempo e impegno per attività di volontariato, senza compenso economico, per supportare associazioni, enti o organizzazioni.. 10–13, 18–22, 25–27, 29–31, 33, 34, 54–56, 58–61, 74, 76, 77, 83, 86

2 Ideazione

2.1 Introduzione e visione del problema

Nel mondo attuale il volontariato rappresenta un elemento fondamentale per il benessere della società, garantendo un supporto essenziale a tutte le persone o organizzazioni che si trovano in situazione di bisogno grazie al contributo di volontari disposti a dedicare il proprio tempo e competenze. Tuttavia ancora oggi molte opportunità di volontariato rimangono poco accessibili a chi vorrebbe partecipare, data la mancanza di visibilità o di piattaforme e strumenti che permettono l'incontro tra domanda e offerta. Questa difficoltà è particolarmente evidente in Lombardia, una regione che offre un'enorme varietà di opportunità per aiutare il prossimo, grazie alla presenza di numerose associazioni e iniziative benefiche. Tuttavia, nonostante questa ricchezza di possibilità, manca un sistema efficace e centralizzato che consenta di mettere in contatto in modo semplice e immediato chi desidera fare volontariato con le realtà che ne hanno bisogno. La frammentazione delle informazioni e l'assenza di strumenti adeguati rendono spesso complesso sia per i volontari che per le associazioni trovare l'incontro giusto.

Per rendere il volontariato più accessibile è fondamentale disporre di una piattaforma intuitiva, semplice ed efficace che faciliti l'incontro tra chi offre e chi cerca opportunità di volontariato, e che fornisca in particolare un supporto completo per entrambi i lati durante tutta l'esperienza di volontariato.

2.2 Business case

Il progetto *CareChain* nasce con l'obiettivo di facilitare l'incontro tra chi offre il proprio tempo e chi ha bisogno di supporto, grazie ad una piattaforma completa e funzionale che garantisce la gestione totale, dalla creazione di richieste all'avvio di attività, fornendo un supporto costante agli utenti. Questa piattaforma è dedicata al territorio lombardo ed è accessibile da qualsiasi dispositivo disponga di una connessione web, tramite browser oppure applicazione dedicata.

Con *CareChain* gli utenti possono pubblicare richieste e offerte, comunicare con altri utenti e avviare attività, il tutto all'interno della piattaforma, senza la necessità di strumenti esterni o intermediari. Oltre alle funzionalità principali, il servizio integra funzionalità tecniche innovative come una chat integrata, con lo scopo di migliorare l'esperienza d'uso per l'utente.

2.3 Diagramma di Gantt



Figure 1: Diagramma di Gantt per la fase Ideazione, 1° parte

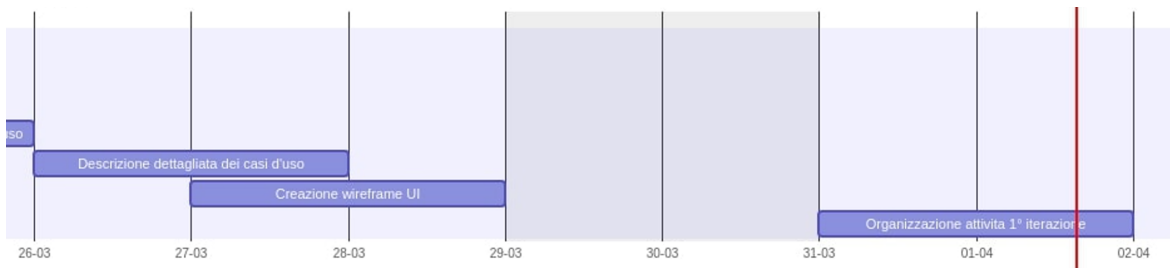


Figure 2: Diagramma di Gantt per la fase Ideazione, 2° parte

2.4 Attori del sistema

CareChain deve offrire le proprie funzionalità a diversi tipi di utenti:

- **Utente non registrato:** colui che utilizza il servizio per la prima volta e non ha ancora creato un account.
- **Volontario:** utente che offre le proprie competenze e tempo libero per attività di volontariato.
- **Organizzazione:** associazione, ente o organizzazione che richiede aiuto da parte di volontari per l'esecuzione di attività di volontariato.
- **Operatore:** utente facente parte del team di controllo a cui non sono destinate le funzionalità principali del sistema, ma effettua unicamente operazioni di controllo e verifica per garantirne la sicurezza ed efficienza.

2.5 Definizione obiettivi utente

Durante una prima analisi sono stati identificati i principali obiettivi che ogni tipologia di utente vuole raggiungere tramite l'utilizzo del sistema:

Lista Attore - Obiettivo	
Attore	Obiettivi
Utente non registrato	Registrarsi al sistema Visualizzare le richieste pubblicate Condividere le richieste sui social
Volontario	Pubblicare, modificare e cancellare offerte Trovare richieste adatte Comunicare con le organizzazioni Eseguire delle attività Condividere richieste e offerte sui social
Organizzazione	Pubblicare, modificare e cancellare richieste Trovare volontari adatti Comunicare con i volontari Avviare delle attività Condividere le richieste sui social
Operatore	Gestione e controllo di offerte, richieste e utenti Controllo del sistema

2.6 Analisi dei requisiti

Analizzando in maniera più precisa gli obiettivi individuati precedentemente sono stati identificati i requisiti che CareChain deve soddisfare, suddivisi secondo il modello FURPS+.

2.6.1 Funzionalità

CareChain deve offrire le seguenti funzionalità dedicate ad ogni tipologia di utente:

Utente non registrato

1. **Registrazione:** il sistema deve permettere agli utenti non registrati la possibilità di registrarsi al servizio come volontario o organizzazione fornendo tutti i dati

necessari per la creazione dell'account, quali email e password, e i dati necessari per la creazione del profilo con le sue caratteristiche.

2. **Ricerca avanzata di richieste:** il sistema deve permettere agli utenti non registrati la possibilità di effettuare ricerche mirate tramite parole chiave, dando anche la possibilità di utilizzare filtri avanzati quali periodo di tempo, posizione geografica e categorie.
3. **Visualizzazione dettagliata:** il sistema deve permettere agli utenti non registrati la visualizzazione in maniera dettagliata delle informazioni di una richiesta oppure della pagina profilo di una organizzazione.
4. **Integrazione con social media:** il sistema deve permettere agli utenti non registrati la condivisione di richieste sui principali social media (Facebook, Whatsapp, Instagram) con la generazione automatica di messaggi personalizzati, per aumentarne la visibilità tramite la condivisione.

Volontario

1. **Autenticazione:** il sistema deve permettere ai volontari la possibilità di autenticarsi al servizio con le credenziali inserite durante la fase di registrazione, e deve fornire anche la possibilità di reimpostare la password in caso di necessità tramite una verifica via e-mail.
2. **Pubblicazione di offerte:** il sistema deve permettere ai volontari la creazione di offerte di volontariato fornendo tutti i dati necessari, quali titolo, descrizione, categoria, posizione geografica, raggio di disponibilità e orari di disponibilità per ogni giorno della settimana. Queste offerte dovranno essere verificate dagli operatori prima della loro pubblicazione.
3. **Modifica e cancellazione di offerte:** il sistema deve permettere ai volontari la modifica o cancellazione di offerte create precedentemente.
4. **Ricerca avanzata di richieste:** il sistema deve permettere ai volontari la possibilità di effettuare ricerche mirate tramite parole chiave, dando anche la possibilità di utilizzare filtri avanzati quali periodo di tempo, posizione geografica e categorie.
5. **Sistema di abbinamento personalizzato:** il sistema, tramite l'utilizzo di algoritmi di matching, deve permettere ai volontari di visualizzare le richieste più adatte ad essi in base alle loro caratteristiche.

6. **Sistema di messaggistica integrato:** il sistema deve permettere ai volontari la comunicazione con le organizzazioni tramite un servizio di messaggistica integrato, che oltre ai classici messaggi deve offrire ulteriori funzionalità avanzate come l'invio di richieste di avvio attività.
7. **Sistema di segnalazione:** il sistema deve permettere ai volontari la segnalazione di richieste che verranno successivamente analizzate dagli operatori.
8. **Visualizzazione dettagliata:** il sistema deve permettere ai volontari la visualizzazione in maniera dettagliata delle informazioni di una richiesta oppure della pagina profilo di una organizzazione.
9. **Gestione dei preferiti:** il sistema deve permettere ai volontari l'aggiunta nei preferiti di richieste, in modo da potervi accedere in maniera più veloce e eventualmente ricevere notifiche in caso di modifiche.
10. **Sistema di feedback:** il sistema deve permettere ai volontari di inserire dei feedback, composti da punteggio da 1 a 5 stelle e da un commento opzionale, verso un'organizzazione dopo il completamento di un'attività, visualizzabili all'interno della pagina profilo della stessa.
11. **Integrazione con social media:** il sistema deve permettere ai volontari la condivisione di offerte e richieste sui principali social media (Facebook, Whatsapp, Instagram) con la generazione automatica di messaggi personalizzati, per aumentarne la visibilità tramite la condivisione.
12. **Gestione profilo:** il sistema deve permettere ai volontari la possibilità di gestire il proprio profilo, permettendo la visualizzazione e modifica dei propri dati e impostazioni.

Organizzazione

1. **Autenticazione:** il sistema deve permettere alle organizzazioni la possibilità di autenticarsi al servizio con le credenziali inserite durante la fase di registrazione, e deve fornire anche la possibilità di reimpostare la password in caso di necessità tramite una verifica via e-mail.
2. **Pubblicazione di richieste:** il sistema deve permettere alle organizzazioni la creazione di richieste di volontariato fornendo tutti i dati necessari, quali titolo, descrizione, categoria, competenze richieste, posizione geografica, orari di necessità per ogni giorno della settimana e foto. Queste richieste dovranno essere verificate dagli operatori prima della loro pubblicazione.

3. **Modifica e cancellazione di richieste:** il sistema deve permettere alle organizzazioni la modifica o cancellazione di richieste create precedentemente.
4. **Ricerca avanzata di offerte:** il sistema deve permettere alle organizzazioni la possibilità di effettuare ricerche mirate tramite parole chiave, dando anche la possibilità di utilizzare filtri avanzati quali periodo di tempo, posizione geografica e categorie.
5. **Sistema di abbinamento personalizzato:** il sistema, tramite l'utilizzo di algoritmi di matching, deve permettere alle organizzazioni di visualizzare le offerte più adatte ad essi in base alle loro caratteristiche.
6. **Sistema di messaggistica integrato:** il sistema deve permettere alle organizzazioni la comunicazione con i volontari tramite un servizio di messaggistica integrato, che oltre ai classici messaggi deve offrire ulteriori funzionalità avanzate come la gestione dell'avvio e terminazione delle attività correnti.
7. **Sistema di segnalazione:** il sistema deve permettere alle organizzazioni la segnalazione di offerte che verranno successivamente analizzate dagli operatori.
8. **Visualizzazione dettagliata:** il sistema deve permettere alle organizzazioni la visualizzazione in maniera dettagliata delle informazioni di un'offerta oppure della pagina profilo di un volontario.
9. **Sistema di feedback:** il sistema deve permettere alle organizzazioni di inserire dei feedback, composti da punteggio da 1 a 5 stelle e da un commento opzionale, verso un volontario dopo il completamento di un'attività, visualizzabili all'interno della pagina profilo dello stesso.
10. **Integrazione con social media:** il sistema deve permettere alle organizzazioni la condivisione di offerte e richieste sui principali social media (Facebook, Whatsapp, Instagram) con la generazione automatica di messaggi personalizzati, per aumentarne la visibilità tramite la condivisione.
11. **Gestione profilo:** il sistema deve permettere alle organizzazioni la possibilità di gestire il proprio profilo, permettendo la visualizzazione e modifica dei propri dati e impostazioni.

Operatore

1. **Autenticazione:** il sistema deve permettere agli operatori la possibilità di autenticarsi al servizio con le credenziali fornitesì.

2. **Gestione delle offerte e delle richieste:** il sistema deve permettere agli operatori la possibilità di controllare e verificare le offerte e le richieste degli utenti, e nel caso di rifiuto di una il sistema deve permettere all'operatore di fornire una motivazione. Il sistema dovrà successivamente inviare una notifica all'utente interessato contenente la motivazione del rifiuto.

2.6.2 Specifiche supplementari

Usabilità

1. **Interfaccia accessibile:** il sistema deve avere un'interfaccia intuitiva e essere accessibile da qualsiasi dispositivo.
2. **Usabilità migliorata:** il sistema deve avere design chiaro e una navigazione semplificata.
3. **Tempi di risposta rapidi:** il sistema deve avere un'interfaccia reattiva con tempi di caricamento minimi.
4. **Feedback immediato:** il sistema deve inviare notifiche e messaggi chiari dopo azioni dell'utente.

Affidabilità

1. **Backup regolari dei dati:** il sistema deve eseguire automaticamente e periodicamente dei backup per garantire la sicurezza delle informazioni.
2. **Monitoraggio continuo:** il sistema deve avere un monitoraggio attivo per rilevare e risolvere eventuali problemi in tempo reale.
3. **Tolleranza ai guasti:** il sistema deve continuare a funzionare anche in caso di malfunzionamento parziale.
4. **Sicurezza dei dati:** il sistema deve proteggere i dati tramite cifratura e protocolli di sicurezza robusti.
5. **Gestione delle versioni del sistema:** il sistema deve dare la possibilità di tornare a versioni precedenti in caso di aggiornamenti problematici.
6. **Logging e auditing:** il sistema deve registrare e archiviare i log per monitorare e analizzare le performance e gli errori.

Prestazioni

1. **Tempi di risposta rapidi:** il sistema deve rispondere alle richieste dell'utente entro 2 secondi nel 95% dei casi.
2. **Tempo di ripristino:** in caso di crash, il sistema deve riavviarsi e ripristinare l'operatività entro 30 secondi.

Manutenibilità

1. **Modularità del codice:** il sistema deve essere sviluppato con un'architettura modulare per facilitare l'aggiunta o la modifica di funzionalità senza impattare il resto del codice.
2. **Documentazione tecnica completa:** il sistema deve fornire una documentazione dettagliata dell'architettura, delle API e delle configurazioni dello stesso.
3. **Commenti e leggibilità del codice:** il sistema deve avere il codice sorgente che sia scritto seguendo standard di leggibilità e deve contenere commenti significativi per facilitarne la comprensione.
4. **Gestione delle versioni:** il sistema deve supportare il versionamento del codice e del database per permettere rollback in caso di errori.

Sicurezza

1. **Autenticazione a più fattori:** il sistema deve supportare l'autenticazione a due fattori (2FA) per tutti gli utenti.
2. **Controllo degli accessi:** il sistema deve avere dei permessi specifici per l'utente in base al ruolo assegnato.

Portabilità

1. **Compatibilità multi-piattaforma:** il sistema deve funzionare su Windows, macOS e Linux senza necessità di modifiche.
2. **Adattabilità ai browser:** il sistema deve essere compatibile con Chrome, Firefox, Edge, Safari e derivati.

Conformità normativa

1. **Conformità al GDPR:** il sistema deve garantire il trattamento dei dati personali in conformità con il Regolamento Generale sulla Protezione dei Dati (UE).
2. **Archiviazione secondo standard ISO:** il sistema deve memorizzare i dati secondo lo standard ISO 27001 per la sicurezza delle informazioni.

2.7 Diagramma dei casi d'uso

Sulla base dei requisiti precedentemente identificati è stata realizzata una prima versione del diagramma dei casi d'uso:

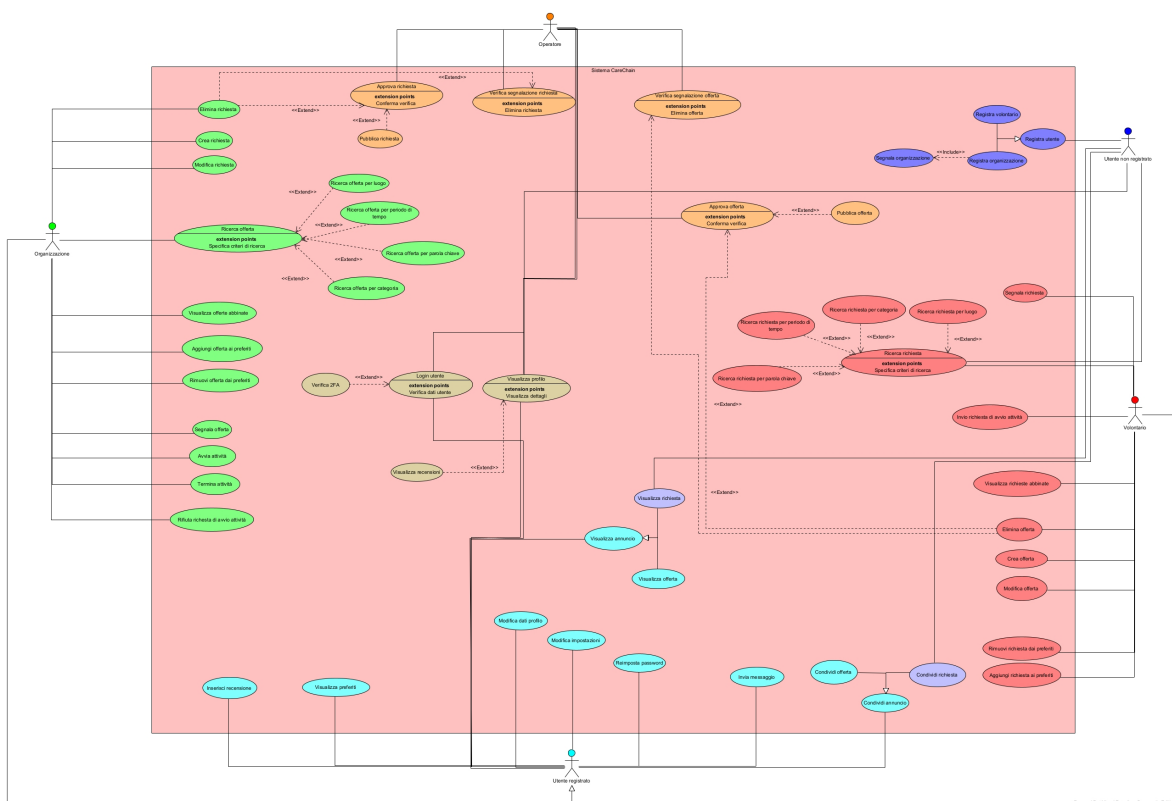


Figure 3: Diagramma dei casi d'uso

2.8 Descrizione breve dei casi d'uso

- **Elimina richiesta:** Un'organizzazione vuole eliminare una sua richiesta. Apre la richiesta e la elimina. Il sistema cancella la richiesta.
- **Crea richiesta:** Un'organizzazione vuole creare una richiesta. Dopo che l'organizzazione ha inserito tutti i dati il sistema li verifica. Se sono validi e il titolo non esiste tra le richieste già esistenti dell'organizzazione il sistema crea la richiesta, altrimenti notifica l'organizzazione che i dati non sono validi e propone di re-inserirli.
- **Modifica richiesta:** Un'organizzazione vuole modificare una richiesta. Inserisce

i dati da modificare, il sistema li controlla e, se validi, modifica la richiesta, altrimenti notifica l'organizzazione che i dati non sono validi, e propone di re-inserirli.

- **Ricerca offerta:** Un'organizzazione vuole ricercare un'offerta. La ricerca eventualmente filtrando per parola chiave, periodo di tempo, categoria o luogo, e il sistema mostra i risultati all'organizzazione.
- **Visualizza offerte abbinate:** Un'organizzazione vuole consultare le offerte più pertinenti alle proprie competenze e preferenze. Il sistema analizza le offerte disponibili, confrontandole con il profilo dell'organizzazione, e mostra quelle maggiormente compatibili.
- **Aggiungi offerta ai preferiti:** Un'organizzazione vuole aggiungere ai preferiti un'offerta. Apre l'offerta e la salva nei preferiti.
- **Rimuovi offerta ai preferiti:** Un'organizzazione vuole rimuovere dai preferiti un'offerta. Apre l'offerta e la toglie dai preferiti.
- **Segnala offerta:** Un'organizzazione vuole segnalare un'offerta in particolare. Dopo aver aperto l'offerta l'organizzazione la segnala e il sistema la manda in approvazione agli operatori.
- **Avvia attività:** Un'organizzazione vuole avviare un'attività. Apre la chat con il volontario con cui vuole iniziare l'attività, decide di avviarla, il sistema chiede la conferma e l'organizzazione accetta. Il sistema notifica il volontario l'inizio dell'attività.
- **Termina attività:** Un'organizzazione vuole terminare un'attività. Apre la chat con il volontario con cui vuole terminare l'attività, decide di terminarla, il sistema chiede la conferma e l'organizzazione accetta. Il sistema notifica il volontario la fine dell'attività.
- **Rifiuta richiesta di avvio attività:** Un'organizzazione vuole rifiutare una richiesta di avvio attività da parte di un volontario. Apre la chat con il volontario che ha inviato la richiesta e la rifiuta. Il sistema chiede la conferma e l'organizzazione l'accetta. Il sistema notifica il volontario il rifiuto della richiesta di inizio attività.
- **Approva richiesta:** Un operatore vuole verificare una richiesta. Apre la richiesta da verificare, controlla che tutti i dati siano validi e permette la pubblicazione. Se non sono validi, non permette la pubblicazione e la richiesta viene eliminata.

- **Verifica segnalazione richiesta:** Un operatore vuole verificare una segnalazione relativa ad una richiesta. Apre la segnalazione da verificare, analizza la motivazione e la richiesta segnalata. Se la ritiene valida elimina la segnalazione, altrimenti elimina la richiesta. In qualsiasi caso, il sistema notifica l'organizzazione l'esito dell'azione.
- **Approva offerta:** Un operatore vuole verificare un'offerta. Apre l'offerta da verificare, controlla che tutti i dati siano validi e permette la pubblicazione. Se non sono validi, non permette la pubblicazione e l'offerta viene eliminata.
- **Verifica segnalazione offerta:** Un operatore vuole verificare una segnalazione relativa ad un'offerta. Apre la segnalazione da verificare, analizza la motivazione e l'offerta segnalata. Se la ritiene valida elimina la segnalazione, altrimenti elimina l'offerta. In qualsiasi caso, il sistema notifica il volontario l'esito dell'azione.
- **Visualizza profilo:** Un utente o operatore vuole visualizzare il profilo di un altro utente. Il sistema recupera le informazioni e le mostra all'utente. L'utente o l'operatore può anche visualizzare le recensioni.
- **Invia messaggio:** Un volontario o un'organizzazione desidera inviare un messaggio a un altro utente. Se la conversazione non esiste, il sistema crea una nuova chat; altrimenti, il messaggio viene aggiunto alla chat esistente. L'utente scrive il testo, il sistema ne verifica la conformità ai requisiti e, se valido, lo invia e notifica il destinatario.
- **Segnala richiesta:** Un volontario vuole segnalare una richiesta in particolare. Dopo aver aperto la richiesta l'utente la segnala e il sistema la manda in approvazione agli operatori.
- **Ricerca richiesta:** Un utente non registrato o un volontario desiderano ricercare una richiesta. L'utente ricerca la richiesta eventualmente filtrando per parola chiave, periodo di tempo, categoria o luogo, e il sistema mostra i risultati all'utente.
- **Invio richiesta di avvio attività:** Un volontario vuole richiedere l'avvio di un'attività relativa ad una richiesta. Se la conversazione non esiste, il sistema crea una nuova chat; altrimenti, la richiesta di avvio attività viene aggiunta alla chat esistente. L'utente poi conferma l'invio della richiesta di avvio attività e il sistema notifica l'organizzazione della richiesta.
- **Visualizza richieste abbinate:** Un volontario desidera consultare le richieste più pertinenti alle proprie competenze e preferenze. Il sistema analizza le richieste

disponibili, confrontandole con il profilo del volontario, e mostra quelle maggiormente compatibili.

- **Elimina offerta:** Un volontario vuole eliminare una sua offerta. Apre l'offerta e la elimina. Il sistema cancella l'offerta.
- **Crea offerta:** Un volontario vuole creare un'offerta. Inserisce tutti i dati, il sistema li controlla e, se validi, crea l'offerta, altrimenti notifica il volontario che i dati non sono validi, e propone di re-inserirli.
- **Modifica offerta:** Un volontario vuole modificare un'offerta. Inserisce i dati da modificare, il sistema li controlla e, se validi, modifica l'offerta, altrimenti notifica il volontario che i dati non sono validi, e propone di re-inserirli.
- **Aggiungi richiesta ai preferiti:** Un volontario vuole aggiungere ai preferiti una richiesta. Apre la richiesta e la mette nei preferiti.
- **Rimuovi richiesta ai preferiti:** Un volontario vuole rimuovere dai preferiti una richiesta. Apre la richiesta e la toglie dai preferiti.
- **Inserisci recensione:** Un utente vuole fare una recensione ad un altro utente. Al termine di un attività l'utente crea la recensione. Il sistema la salva e la rende visualizzabile nella pagina profilo.
- **Visualizza preferiti:** Un utente vuole visualizzare i preferiti. L'utente apre la pagina dei preferiti e il sistema la mostra.
- **Modifica dati profilo:** Un utente vuole modificare i propri dati. L'utente entra nel proprio profilo e modifica i dati desiderati. Il sistema verifica la validità dei dati e li salva.
- **Modifica impostazioni:** Un utente vuole modificare le impostazioni. L'utente entra nel proprio profilo e aggiorna le impostazioni desiderate. Il sistema le salva.
- **Visualizza offerta:** Un utente vuole visualizzare nel dettaglio un'offerta. Il sistema recupera le informazioni relative all'offerta e le presenta all'utente.
- **Visualizza richiesta:** Un utente o un utente non registrato vuole visualizzare nel dettaglio una richiesta. Il sistema recupera le informazioni relative alla richiesta e le mostra all'utente.

- **Reimposta password:** Un utente vuole reimpostare la propria password. Dopo che l'utente seleziona di voler reimpostare la password il sistema invia una mail all'utente dove viene fornito l'accesso alla modifica. L'utente inserisce una nuova password e il sistema dopo averne verificato la validità la salva.
- **Condividi offerta:** Un utente vuole condividere una offerta sui social. L'utente visualizza l'offerta e la condivide, scegliendo il social desiderato. Il sistema genera automaticamente un messaggio che verrà condiviso sul social.
- **Condividi richiesta:** Un utente o utente non registrato vuole condividere una richiesta sui social. L'utente visualizza la richiesta e la condivide, scegliendo il social desiderato. Il sistema genera automaticamente un messaggio che verrà condiviso sul social.
- **Registra volontario:** Un utente non registrato vuole registrarsi nel sistema come volontario. L'utente non registrato accede alla pagina di registrazione, compila i campi richiesti e conferma i dati. Il sistema verifica la correttezza delle informazioni, controlla che l'email non sia già registrata e salva l'account. Infine, invia un'email di conferma all'utente, che deve cliccare sul link ricevuto per attivare il proprio profilo da volontario.
- **Registra organizzazione:** Un utente non registrato vuole registrarsi nel sistema come organizzazione. L'utente non registrato accede alla pagina di registrazione e inserisce i dati richiesti. Il sistema verifica la correttezza delle informazioni e l'assenza di duplicati. Dopo il salvataggio, la richiesta viene inviata agli operatori per l'approvazione. Se approvata, l'organizzazione riceve un'email con un link di conferma per attivare l'account.
- **Login utente:** Un utente o operatore vuole autenticarsi nella piattaforma. Dopo che l'utente ha inserito le proprie credenziali il sistema le verifica. Se l'autenticazione ha successo l'utente viene portato alla pagina principale, in caso negativo viene avvisato che le credenziali non sono valide. Se la verifica 2FA è attiva il sistema invia una email contenente un codice che l'utente dovrà inserire per potersi autenticare.

2.9 Casi d'uso dettagliati

Dato che il sistema si basa principalmente sulla gestione di richieste e offerte di volontariato, è stato ritenuto fondamentale descrivere fin da subito nel dettaglio i casi d'uso che regolano queste funzionalità, in modo da iniziare già ad avere una chiara visione di come organizzazioni e volontari interagiranno con la piattaforma:

2.9.1 Crea richiesta

Nome: Crea richiesta

Attore primario: Organizzazione

Attori e interessi:

- *Organizzazione*: vuole pubblicare una nuova richiesta; vuole un'interfaccia chiara ed un inserimento veloce e intuitivo.
- *Volontario*: vuole avere accesso a richieste valide.

Precondizioni: L'organizzazione è autenticata nel sistema.

Postcondizioni: La richiesta è stata creata e inviata in approvazione agli operatori.

Scenario principale:

1. L'organizzazione inizia la creazione della richiesta.
2. Il sistema mostra la pagina di creazione in cui bisogna inserire tutti i dati obbligatori per la creazione della richiesta (titolo, descrizione, categoria, competenze richieste, periodo di tempo, orari di disponibilità, posizione geografica e foto di copertina).
3. L'organizzazione inserisce i dati richiesti.
4. Il sistema verifica che i campi siano compilati correttamente.
5. L'organizzazione conferma la creazione della richiesta.
6. Il sistema controlla che non esistano altre richieste dell'organizzazione con lo stesso titolo.

7. Il sistema salva i dati della richiesta.
8. Il sistema invia agli operatori la richiesta per l'approvazione.
9. Il sistema notifica l'organizzazione di attendere per email e/o notifica l'esito dell'operazione.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica l'organizzazione che ha inserito dei dati non validi.
 2. Il sistema propone all'organizzazione di modificare i dati inseriti.
 3. Si torna al punto 3.
- 6a. Il sistema rileva che è presente un'altra richiesta dell'organizzazione con lo stesso titolo.
 1. Il sistema notifica l'organizzazione che il titolo inserito è già stato utilizzato.
 2. Il sistema propone all'organizzazione di inserire un nuovo titolo.
 3. Si torna al punto 3.
- 6b. Il sistema non riesce a verificare se è presente un'altra richiesta con lo stesso titolo.
 1. Il sistema notifica l'organizzazione che non riesce a verificare i dati.
 2. Il sistema invita l'organizzazione a riprovare più tardi.
 3. Si torna al punto 5.
7. Il sistema non riesce a salvare i dati della richiesta:
 1. Il sistema notifica l'organizzazione che ha riscontrato un errore nel salvataggio dei dati.
 2. Il sistema invita l'organizzazione a riprovare più tardi.
 3. Si torna al punto 5.
- *a. L'organizzazione vuole annullare l'operazione di creazione:

1. L'organizzazione annulla la creazione della richiesta.
2. Il sistema annulla l'operazione di creazione e riporta l'organizzazione alla pagina principale.

*b. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'organizzazione che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'organizzazione a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il sistema deve garantire un tempo massimo di 24-48 ore per la verifica della richiesta.
- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose creazioni di richieste durante la giornata.

2.9.2 Crea offerta

Nome: Crea offerta

Attore primario: Volontario

Attori e interessi:

- *Volontario*: vuole pubblicare una nuova offerta; vuole un'interfaccia chiara ed un inserimento veloce e intuitivo.
- *Organizzazione*: vuole avere accesso a offerte valide.

Precondizioni: Il volontario è autenticato nel sistema.

Postcondizioni: L'offerta è stata creata ed inviata in approvazione agli operatori.

Scenario principale:

1. Il volontario inizia la creazione dell'offerta.
2. Il sistema mostra la pagina di creazione in cui bisogna inserire tutti i dati obbligatori per la creazione dell'offerta (titolo, descrizione, categoria, posizione geografica, raggio di disponibilità e orari di disponibilità per ogni giorno della settimana).
3. Il volontario inserisce i dati richiesti.
4. Il sistema verifica che i campi siano compilati correttamente.
5. Il volontario conferma la creazione dell'offerta.
6. Il sistema controlla che non esistano altre offerte del volontario con lo stesso titolo.
7. Il sistema salva i dati dell'offerta.
8. Il sistema invia agli operatori l'offerta per l'approvazione.
9. Il sistema notifica il volontario di attendere per email e/o notifica l'esito dell'operazione.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica il volontario che ha inserito dei dati non validi.
 2. Il sistema propone al volontario di modificare i dati inseriti.
 3. Si torna al punto 3.
- 6a. Il sistema rileva che è presente un'altra offerta del volontario con lo stesso titolo.
 1. Il sistema notifica il volontario che il titolo inserito è già stato utilizzato.
 2. Il sistema propone al volontario di inserire un nuovo titolo.
 3. Si torna al punto 3.
- 6b. Il sistema non riesce a verificare se è presente un'altra offerta con lo stesso titolo.
 1. Il sistema notifica il volontario che non riesce a verificare i dati.
 2. Il sistema invita il volontario a riprovare più tardi.
 3. Si torna al punto 5.
7. Il sistema non riesce a salvare i dati dell'offerta:
 1. Il sistema notifica il volontario che ha riscontrato un errore nel salvataggio dei dati.
 2. Il sistema invita il volontario a riprovare più tardi.
 3. Si torna al punto 5.
- *a. Il volontario vuole annullare l'operazione di creazione:
 1. Il volontario annulla la creazione dell'offerta.
 2. Il sistema annulla l'operazione di creazione e riporta il volontario alla pagina principale.
- *b. Errore di sistema o fallimento imprevisto:
 1. Il sistema notifica il volontario che l'operazione non è andata a buon fine.

2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita il volontario a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il sistema deve garantire un tempo massimo di 24-48 ore per la verifica dell'offerta.
- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose creazioni di offerte durante la giornata.

2.9.3 Visualizza richiesta

Nome: Visualizza richiesta

Attore primario: Volontario, organizzazione, utente non registrato

Attori e interessi:

- *Organizzazione*: vuole visualizzare le proprie richieste; vuole un'interfaccia intuitiva.
- *Volontario*: vuole scoprire richieste per potersi candidare; vuole un'interfaccia intuitiva e che mostri tutti i dettagli.
- *Utente non registrato*: vuole scoprire delle richieste ancora prima di registrarsi; vuole un'interfaccia intuitiva e che mostri tutti i dettagli.

Precondizioni:

- La richiesta deve esistere ed essere verificata.

Postcondizioni: Il sistema mostra la richiesta all'utente mostrando le opzioni disponibili in base al tipo di utente che la sta visualizzando.

Scenario principale:

1. L'utente apre una qualsiasi pagina in cui sono visualizzate delle richieste.
2. L'utente seleziona una richiesta in particolare.
3. Il sistema mostra la richiesta nel dettaglio con tutte le sue informazioni.

Scenari alternativi:

3a. Il sistema non riesce a trovare la richiesta:

1. Il sistema notifica l'utente che la richiesta non è stata trovata.
2. Il sistema propone all'utente di tornare indietro oppure se la richiesta è propria di mandare una email agli sviluppatori.
3. Si torna al punto 2.

3b. Il sistema non riesce a mostrare la richiesta per errore interno.

1. Il sistema notifica l'utente che c'è stato un problema interno e che non riesce a mostrare la richiesta in quel momento.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.
4. Si torna al punto 2.

*. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il sistema deve garantire che ogni utente possa visualizzare solo le richieste a cui ha diritto:

- L'organizzazione può visualizzare solo le proprie richieste verificate.
- Il volontario e l'utente non registrato possono visualizzare solo le richieste verificate.
- Il tempo di risposta del sistema per mostrare la richiesta deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose visualizzazioni di richieste durante la giornata.

2.9.4 Visualizza offerta

Nome: Visualizza offerta

Attore primario: Volontario, organizzazione

Attori e interessi:

- *Organizzazione:* vuole scoprire le offerte per cercare un possibile volontario; vuole un'interfaccia intuitiva e che mostri tutti i dettagli.
- *Volontario:* vuole visualizzare le proprie offerte; vuole un'interfaccia intuitiva.

Precondizioni: L'offerta deve esistere ed essere verificata.

Postcondizioni: Il sistema mostra l'offerta all'utente mostrando le opzioni disponibili in base al tipo di utente che la sta visualizzando.

Scenario principale:

1. L'utente apre una qualsiasi pagina in cui sono visualizzate delle offerte.
2. L'utente seleziona un'offerta in particolare.
3. Il sistema mostra l'offerta nel dettaglio con tutte le sue informazioni.

Scenari alternativi:

3a. Il sistema non riesce a trovare l'offerta:

1. Il sistema notifica l'utente che l'offerta non è stata trovata.
2. Il sistema propone all'utente di tornare indietro oppure se l'offerta è propria di mandare una email agli sviluppatori.
3. Si torna al punto 2.

3b. Il sistema non riesce a mostrare l'offerta per errore interno.

1. Il sistema notifica l'utente che c'è stato un problema interno e che non riesce a mostrare l'offerta in quel momento.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.
4. Si torna al punto 2.

*. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il sistema deve garantire che ogni utente possa visualizzare solo le offerte a cui ha diritto:
 - L'organizzazione può visualizzare solo le offerte verificate.
 - Il volontario può visualizzare solo la proprie offerte verificate.
- Il tempo di risposta del sistema per mostrare l'offerta deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose visualizzazioni di offerte durante la giornata.

2.9.5 Invia messaggio

Nome: Invia messaggio

Attore primario: Volontario, organizzazione

Attori e interessi:

- *Volontario*: vuole inviare un messaggio ad un'organizzazione; vuole un'interfaccia chiara ed intuitiva.
- *Organizzazione*: vuole inviare un messaggio ad un volontario; vuole un'interfaccia chiara ed intuitiva.

Precondizioni: L'utente è autenticato nel sistema. L'utente ha aperto la pagina profilo di un altro utente.

Postcondizioni: Il messaggio è stato salvato nel database e ricevuto dal destinatario per la visualizzazione.

Scenario principale:

- Scenario principale 1: Invio di un messaggio in una nuova chat
 1. L'utente apre la pagina profilo di un altro utente.
 2. Il sistema mostra la pagina richiesta.
 3. L'utente avvia una nuova conversazione.
 4. Il sistema crea una nuova chat vuota e la mostra all'utente.
 5. L'utente compone il messaggio da inviare.
 6. Il sistema verifica che il messaggio inserito rispetti tutti i requisiti.
 7. L'utente invia il messaggio.
 8. Il sistema salva il messaggio e invia una notifica al destinatario.

- Scenario principale 2: Invio di un messaggio in una chat già avviata
 1. L'utente apre la sezione chat.
 2. Il sistema mostra tutte le chat già avviate dall'utente.
 3. L'utente seleziona una chat tra quelle proposte.
 4. Il sistema mostra all'utente la chat selezionata.
 5. L'utente compone il messaggio da inviare.
 6. Il sistema verifica che il messaggio inserito rispetti tutti i requisiti.
 7. L'utente invia il messaggio.
 8. Il sistema salva il messaggio e invia una notifica al destinatario.

Scenari alternativi:

- 4a. Il sistema non riesce a creare una nuova chat:
 1. Il sistema notifica l'utente che non riesce a creare una nuova chat.
 2. Il sistema invita l'utente a riprovare più tardi.
 3. Si torna al punto 3.
- 4b. Il sistema non riesce a ricavare le informazioni della chat:
 1. Il sistema notifica l'utente che non riesce a ricavare le informazioni della chat.
 2. Il sistema invita l'utente a riprovare più tardi.
 3. Si torna al punto 3.
6. Il sistema rileva nel messaggio dei requisiti non rispettati:
 1. Il sistema notifica l'utente che ha inserito un messaggio non valido.
 2. Il sistema propone all'utente di modificare il messaggio.
 3. Si torna al punto 5.
8. Il sistema non riesce a salvare il messaggio:
 1. Il sistema notifica l'utente che non riesce a salvare ed inviare il messaggio.

2. Il sistema invita l'utente a riprovare più tardi.

3. Si torna al punto 7.

*a. L'utente vuole annullare l'operazione:

1. L'utente decide di non inviare più il messaggio.

2. Il sistema riporta l'utente alla schermata precedente oppure alla pagina principale.

*b. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.

2. Se possibile, il sistema registra l'errore nei log per la diagnosi.

3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il messaggio deve avere una lunghezza massima di 300 caratteri.
- Le notifiche devono essere inviate in tempo reale

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire l'invio di numerosi messaggi da parte degli utenti durante la giornata.

2.9.6 Login

Nome: Login utente

Attore primario: Volontario, organizzazione e operatore

Attori e interessi:

- *Organizzazione:* vuole autenticarsi velocemente; vuole un'interfaccia intuitiva.

- *Volontario*: vuole autenticarsi velocemente; vuole un'interfaccia intuitiva.
- *Operatore*: vuole autenticarsi velocemente; vuole un'interfaccia intuitiva.
- *Sistema*: vuole autenticare gli utenti.

Precondizioni: L'utente deve essere già registrato.

Postcondizioni: L'utente ha eseguito correttamente il login. Il sistema ha autenticato l'utente.

Scenario principale:

1. L'utente inizia il login.
2. Il sistema mostra la pagina in cui bisogna inserire i dati necessari per la autenticazione (email e password).
3. L'utente inserisce i suoi dati.
4. Il sistema verifica che i campi siano compilati correttamente.
5. L'utente conferma il login.
6. Il sistema controlla che l'email sia già registrata e che la password sia quella collegata alla email.
7. Il sistema autentica l'utente.
8. Il sistema notifica l'utente di aver completato con successo l'autenticazione e lo porta alla home.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica l'utente che ha inserito dei dati non validi.
 2. Il sistema propone all'utente di inserire nuovi dati.
 3. Si torna al punto 3.
- 6a. Il sistema rileva che la password non coincide con l'email.

1. Il sistema notifica l'utente che la password è sbagliata.
 2. Il sistema propone all'utente di inserire una nuova password.
 3. Si torna al punto 3.
- 6b. Il sistema rileva che la email non è collegata a nessun account esistente.
1. Il sistema notifica l'utente che non esiste nessun account che abbia quella email.
 2. Il sistema propone all'utente di inserire una nuova email.
 3. Si torna al punto 3.
- 6c. Il sistema non riesce a verificare se la email e la password siano corretti.
1. Il sistema notifica l'utente che non riesce a verificare la validità dei dati.
 2. Il sistema invita l'utente a riprovare più tardi.
 3. Si torna al punto 3.
- 7a. Il sistema non riesce ad autenticare l'utente:
1. Il sistema notifica l'utente che ha riscontrato un errore nel processo di autenticazione dell'utente.
 2. Il sistema invita l'utente a riprovare più tardi.
 3. Si torna al punto 5.
- 7b. L'utente ha attivato la funzione 2FA:
1. Il sistema invia all'utente un email contenente il codice di verifica per l'autenticazione.
 2. Il sistema chiede all'utente di inserire il codice inviato.
 3. L'utente inserisce il codice di verifica e conferma l'inserimento.
 4. Il sistema verifica che il codice inserito sia valido.
 5. Si passa al punto 8.
- 7b1. L'utente non ha ricevuto il codice di verifica:

1. L'utente chiede al sistema di inviare nuovamente il codice di verifica.
2. Il sistema invia all'utente una nuova email contenente il codice di verifica per l'autenticazione.
3. Si torna al punto 7b2.

7b4a. L'utente ha inserito un codice di verifica non valido:

1. Il sistema notifica l'utente che il codice inserito non è valido.
2. Il sistema propone all'utente di inserire nuovamente il codice di verifica ricevuto.
3. Si torna al punto 7b2.

7b4b. Il sistema non riesce a verificare la validità del codice inserito:

1. Il sistema notifica l'utente che non riesce a verificare la validità del codice inserito.
2. Il sistema invita l'utente a riprovare più tardi.
3. Si torna al punto 7b3.

*. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email al team di controllo.

Requisiti speciali:

- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Potenzialmente infinita.

2.10 Prototipazione dell'interfaccia

Per avere un'idea iniziale della UI sono stati realizzati dei wireframe relativi alle funzionalità più critiche del sistema, analizzate precedentemente nel dettaglio. Questi wireframe forniscono una rappresentazione preliminare delle schermate principali, in modo da valutarne l'usabilità.



Figure 4: Schede riassuntive di richieste e offerte



Figure 5: Visualizzazione dettagliata di richieste e offerte

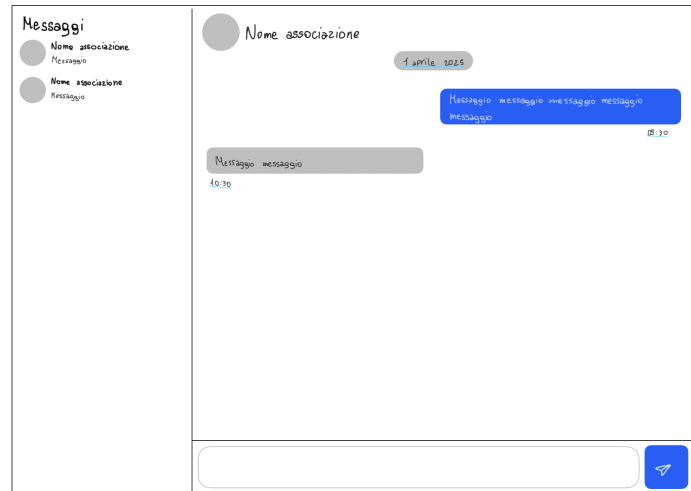


Figure 6: Visualizzazione lista e dettaglio chat

Crea nuova richiesta di volontariato

[Vai alla anteprima](#)

TITOLO *

Esempio

DESCRIZIONE *

Descrizione dettagliata

o/non unitari

Categoria *

Seleziona una categoria

Posizione geografica *

Milano

Periodo *

Gennaio - Luglio 2025

Competenze richieste *

Esempio - Esempio

Orari di disponibilità

Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato	Domenica
10:00 - 12:00	...	...	...	...	...	...

Foto

Inserisci foto

[Pubblica](#)

Crea nuova offerta di volontariato

[Vai alla anteprima](#)

TITOLO *

Esempio

DESCRIZIONE *

Descrizione dettagliata

o/non unitari

Categoria *

Seleziona una categoria

Posizione e disponibilità

Milano

Raggio di disponibilità

0 km - 10 km

Orari di disponibilità

Lunedì	Martedì	Mercoledì	Giovedì	Venerdì	Sabato	Domenica
10:00 - 12:00	...	...	...	...	...	...

[Pubblica](#)

Figure 7: Pagine per la creazione di richieste e offerte

2.11 Pianificazione 1° Iterazione

Dopo aver analizzato e dato un grado di importanza a tutte le funzionalità che devono essere realizzate sono stati identificati i casi d'uso che verranno sviluppati durante la prima iterazione. In particolare durante questa iterazione verranno stabilite le fon-

damenta del progetto, e sviluppate le funzionalità relative ai seguenti casi d'uso, data la loro centralità nel sistema:

- Crea offerta
- Crea richiesta
- Visualizza offerta
- Visualizza richiesta

Oltre a queste è stato scelto di sviluppare la funzionalità di login che, nonostante non rappresenti una funzione chiave per l'ambito del sistema, è necessaria per il test e il corretto sviluppo di tutte le altre funzionalità. In particolare il caso d'uso relativo ad essa è:

- Login

3 1° iterazione

In questa prima iterazione, come analizzato nella fase precedente di ideazione, sono state stabilite le fondamenta del progetto, e sviluppati i primi casi d'uso ritenuti più centrali.

3.1 Diagramma di Gantt

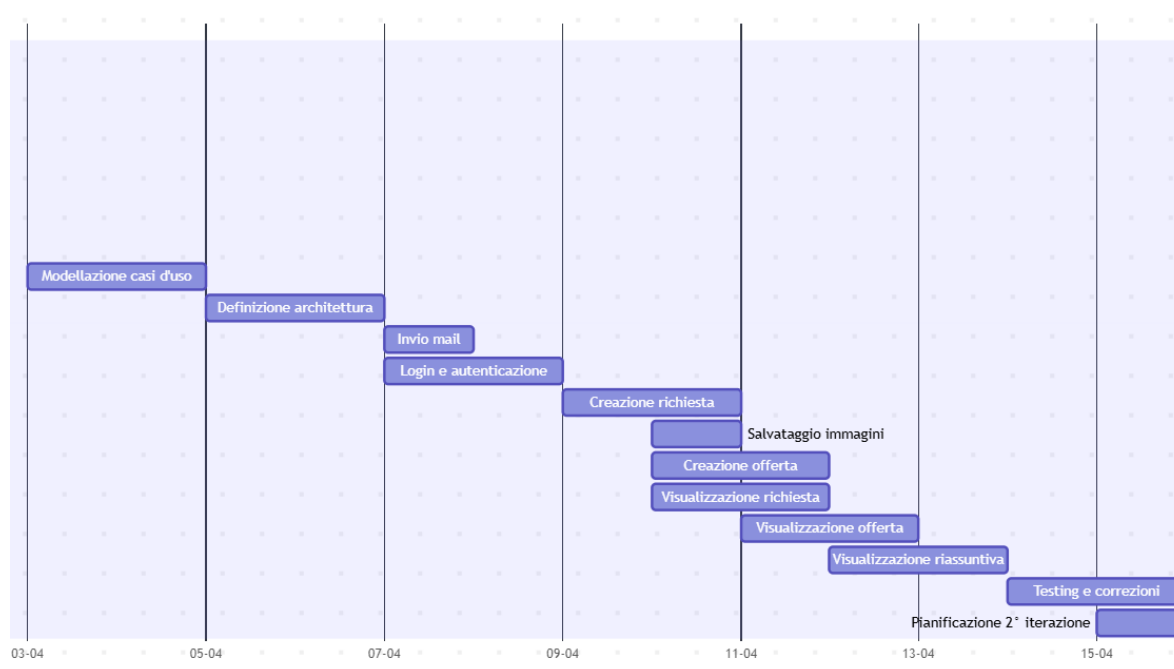


Figure 8: Diagramma di Gantt per la 1° iterazione

3.2 Scelte architetturali

Per la piattaforma si è optati per una architettura client-server realizzata in Javascript:

- **Frontend** sviluppato con il framework *Next.js*
- **Backend** organizzato in servizi separati e sviluppato in *node.js* con il framework *express*. Per l'autenticazione si è optato per l'utilizzo della libreria *Better-Auth*, che permette una gestione funzionale e ottimizzata delle sessioni.
- **Database** relazionale basato su *MySQL*

3.3 Modello di dominio

Per lo sviluppo delle funzionalità di **login**, **creazione** e **visualizzazione** di richieste e offerte, sono state identificate le seguenti classi concettuali, che hanno permesso la realizzazione di un primo prototipo di modello di dominio:

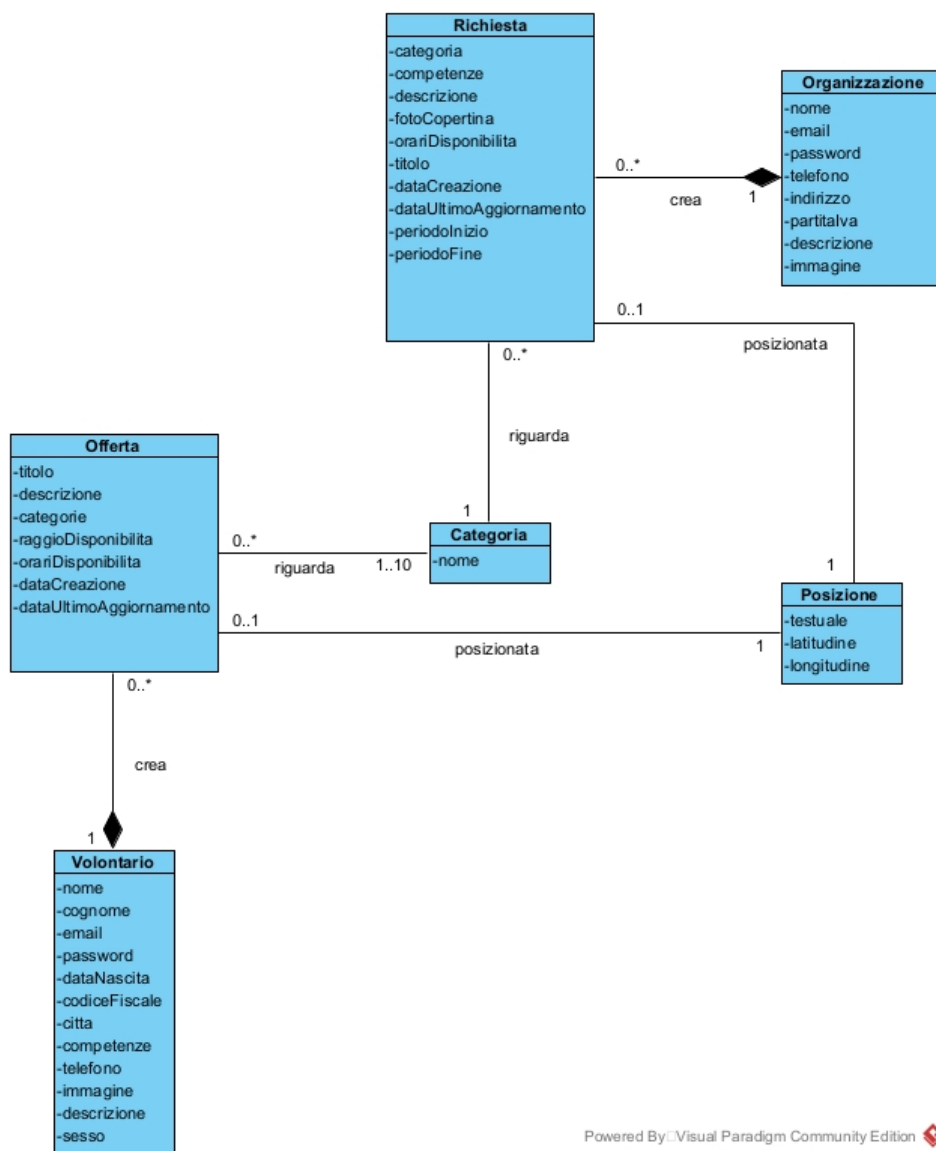


Figure 9: Iterazione 1 - Modello di dominio

3.4 Diagrammi di sequenza di sistema (SSD)

3.4.1 Visualizza Richiesta

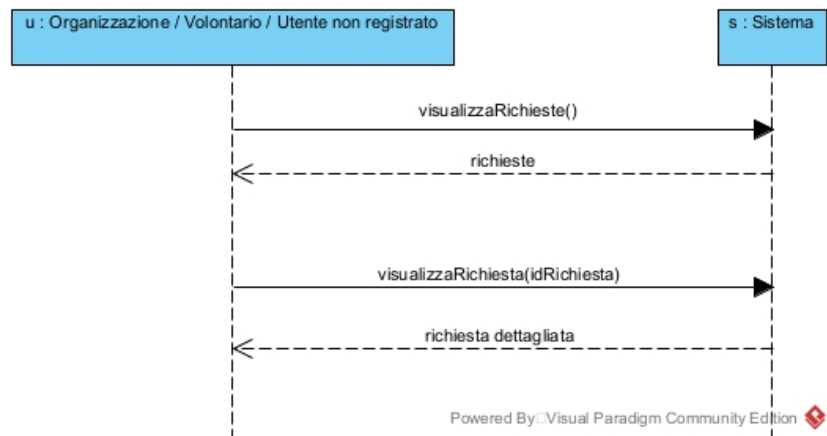


Figure 10: SSD Visualizza Richiesta

3.4.2 Visualizza Offerta

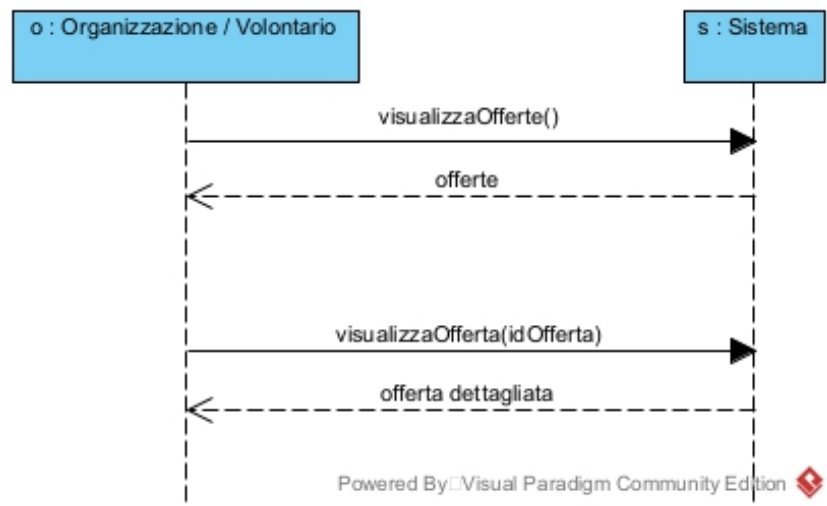


Figure 11: SSD Visualizza Offerta

3.4.3 Login

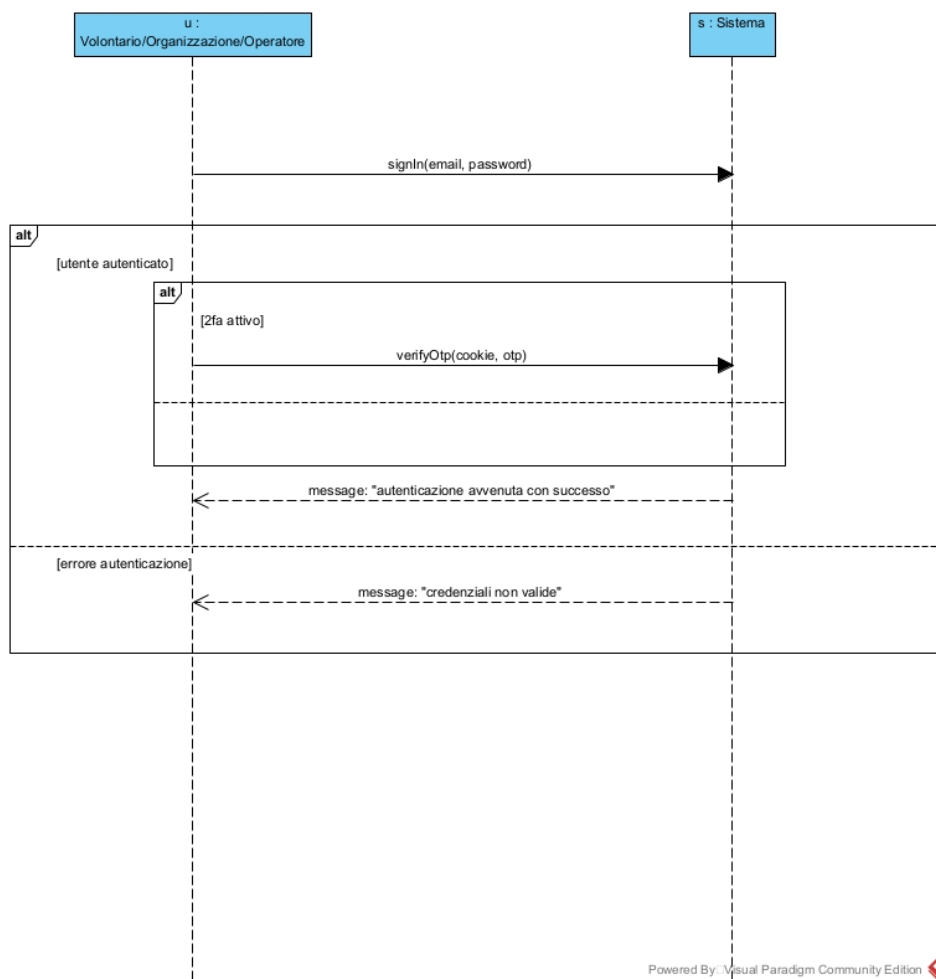


Figure 12: SSD Login

3.5 Diagrammi di attività

3.5.1 Crea offerta

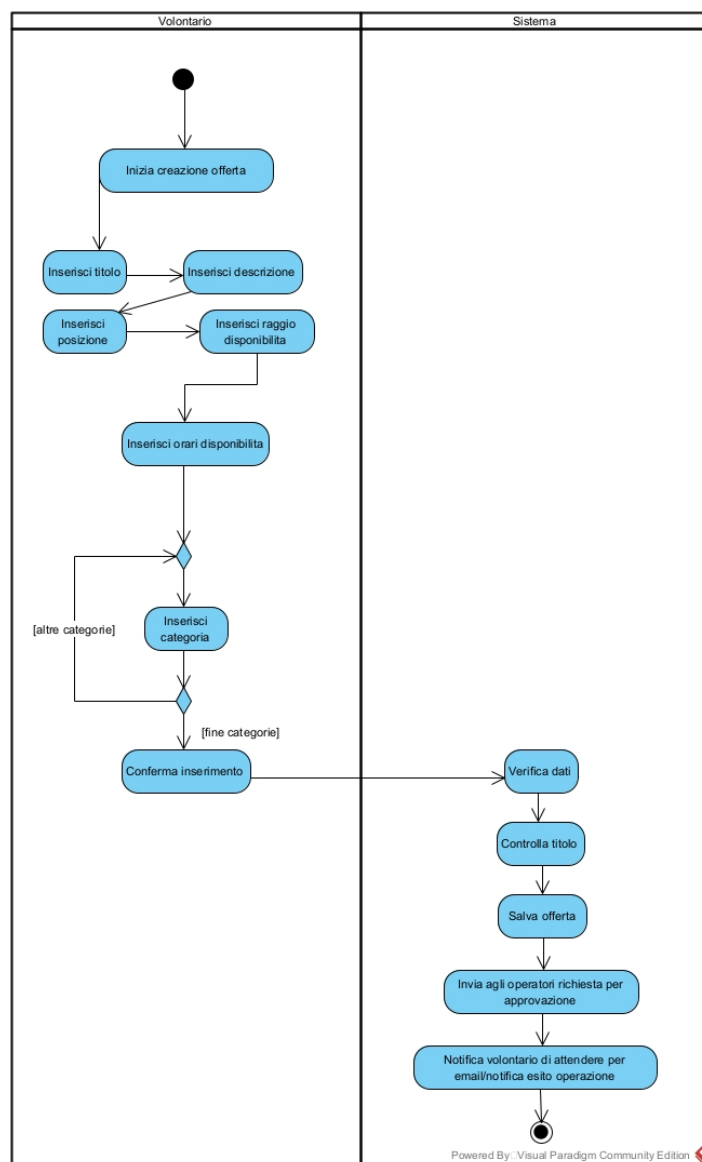


Figure 13: AD Crea offerta

3.6 Contratti

3.6.1 Crea Richiesta

- **Operazione:** `creaRichiesta`(*titolo*: String, *descrizione*: String, *categoria*: int, *periodoInizio*: DateTime, *periodoFine*: DateTime, *posizione*: String, *fotoCopertina*: String, *orariDisponibilita*: OrariDisponibilita, *competenze*: String[])
- **Riferimenti:** casi d'uso: Crea richiesta
- **Pre-condizioni:** È in corso la creazione di una Richiesta *r*.
- **Post-condizioni:**
 - è stata creata un'istanza *r* di Richiesta (**creazione di oggetto**)
 - *r.titolo* è diventato *titolo* (**modifica di attributo**)
 - *r.descrizione* è diventato *descrizione* (**modifica di attributo**)
 - *r.categoria* è diventato *categoria* (**modifica di attributo**)
 - *r.periodoInizio* è diventato *periodoInizio* (**modifica di attributo**)
 - *r.periodoFine* è diventato *periodoFine* (**modifica di attributo**)
 - *r.posizione* è diventato *posizione* (**modifica di attributo**)
 - *r.fotoCopertina* è diventato *fotoCopertina* (**modifica di attributo**)
 - *r.orariDisponibilita* è diventato *orariDisponibilita* (**modifica di attributo**)
 - *r.competenze* è diventato *competenze* (**modifica di attributo**)
 - *r.dataCreazione* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - *r.dataUltimoAggiornamento* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - è stata creata un'istanza *p* di Posizione (**creazione di oggetto**)
 - *p.testuale* è diventato *posizione* (**modifica di attributo**)
 - *p.latitudine* e *p.longitudine* sono stati impostati con la conversione dell'indirizzo testuale (**modifica di attributo**)
 - la Posizione *p* è stata associata con la Richiesta *r* (**formazione di collegamento**)
 - la Richiesta *r* è stata associata con l'Organizzazione (**formazione di collegamento**)

3.6.2 Crea Offerta

- **Operazione:** `creaOfferta(titolo: String, descrizione: String, posizione: String, raggioDisponibilita: int, orariDisponibilita: OrariDisponibilita, categorie: int[])`
- **Riferimenti:** casi d'uso: Crea offerta
- **Pre-condizioni:** È in corso la creazione di un'Offerta *o*.
- **Post-condizioni:**
 - è stata creata un'istanza *o* di Offerta (**Creazione di oggetto**)
 - *o.titolo* è diventato *titolo* (**modifica di attributo**)
 - *o.descrizione* è diventato *descrizione* (**modifica di attributo**)
 - *o.raggioDisponibilita* è diventato *raggioDisponibilita* (**modifica di attributo**)
 - *o.orariDisponibilita* è diventato *orariDisponibilita* (**modifica di attributo**)
 - *o.categorie* è diventato *categorie* (**modifica di attributo**)
 - *o.dataCreazione* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - *o.dataUltimoAggiornamento* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - è stata creata un'istanza *p* di Posizione (**creazione di oggetto**)
 - *p.testuale* è diventato *posizione* (**modifica di attributo**)
 - *p.latitudine* e *p.longitudine* sono stati impostati con la conversione dell'indirizzo testuale (**modifica di attributo**)
 - la Posizione *p* è stata associata con l'Offerta *o* (**formazione di collegamento**)
 - l'Offerta *o* è stata associata con il Volontario (**formazione di collegamento**)

3.7 Modello di architettura logica

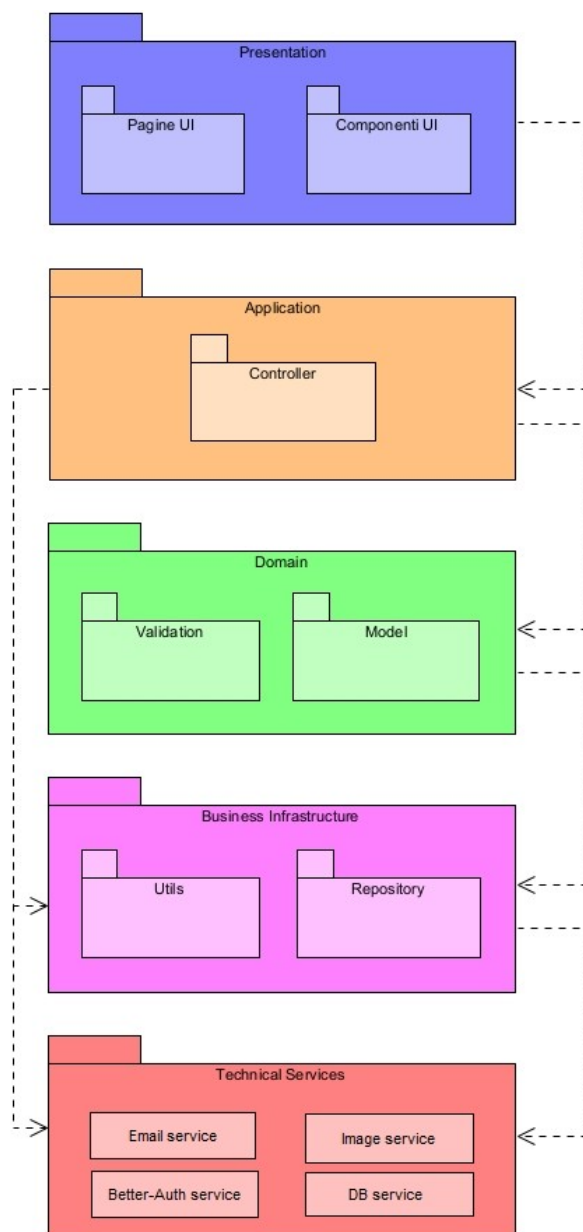


Figure 14: Modello di architettura logica 1° iterazione

3.8 Diagramma delle classi di progetto

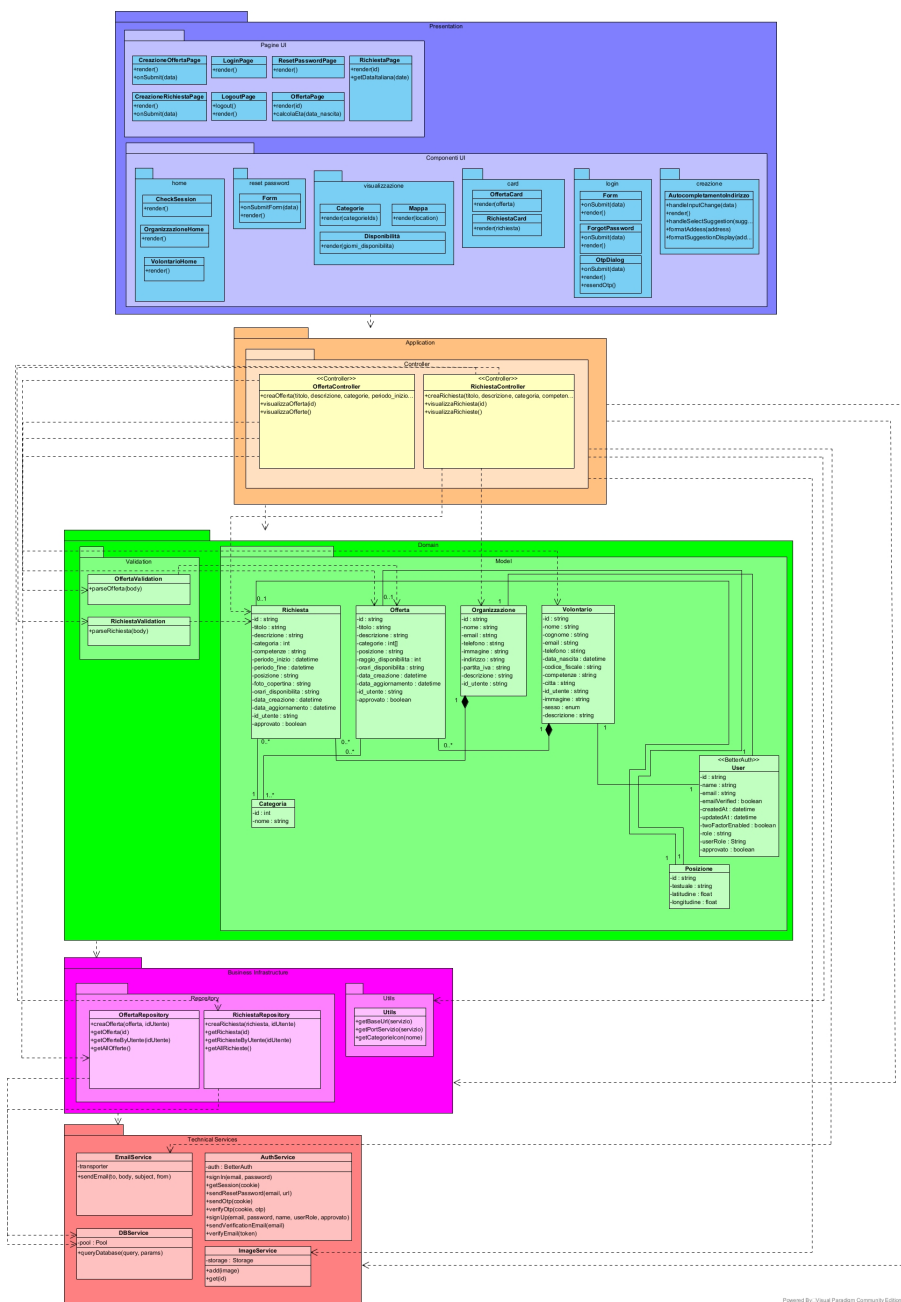
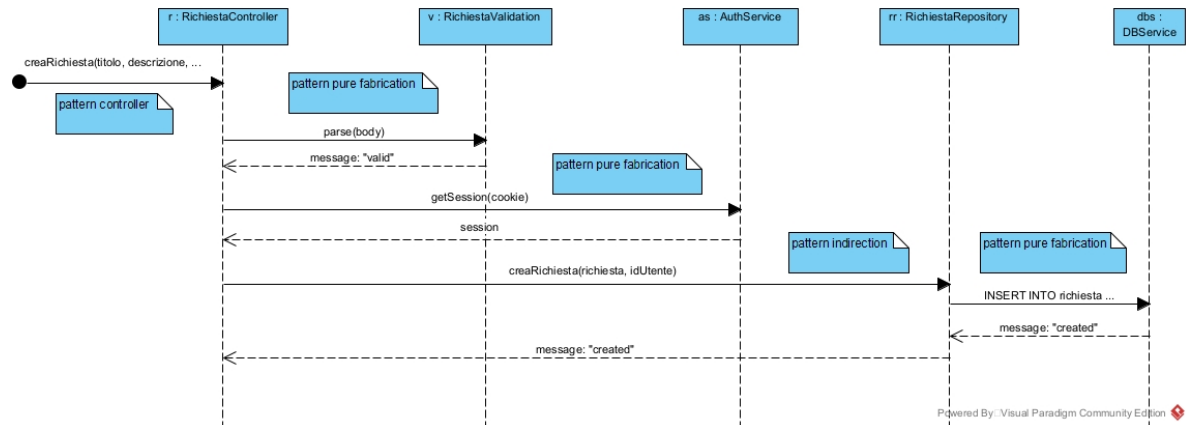


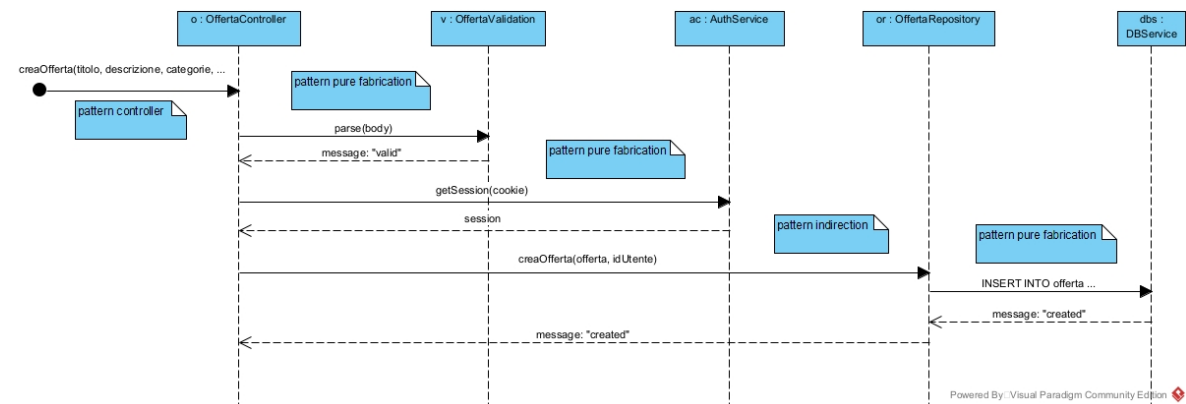
Figure 15: Diagramma delle classi di progetto 1° iterazione

3.9 Diagrammi di sequenza (SD)

3.9.1 Crea richiesta



3.9.2 Crea offerta



3.10 Testing e analisi

Per verificare il corretto funzionamento delle funzionalità implementate sono stati realizzati dei test end-to-end utilizzando Playwright, ed è stata anche condotta un'analisi del codice mediante SonarQube grazie al quale sono stati identificati e corretti alcuni code smell.

3.11 Pianificazione 2° Iterazione

Per la seconda iterazione è stato deciso di sviluppare le funzionalità relative ai seguenti casi d'uso:

- Registra volontario
- Registra organizzazione
- Verifica richiesta
- Verifica offerta
- Segnala richiesta
- Segnala offerta

4 2° iterazione

4.1 Diagramma di Gantt

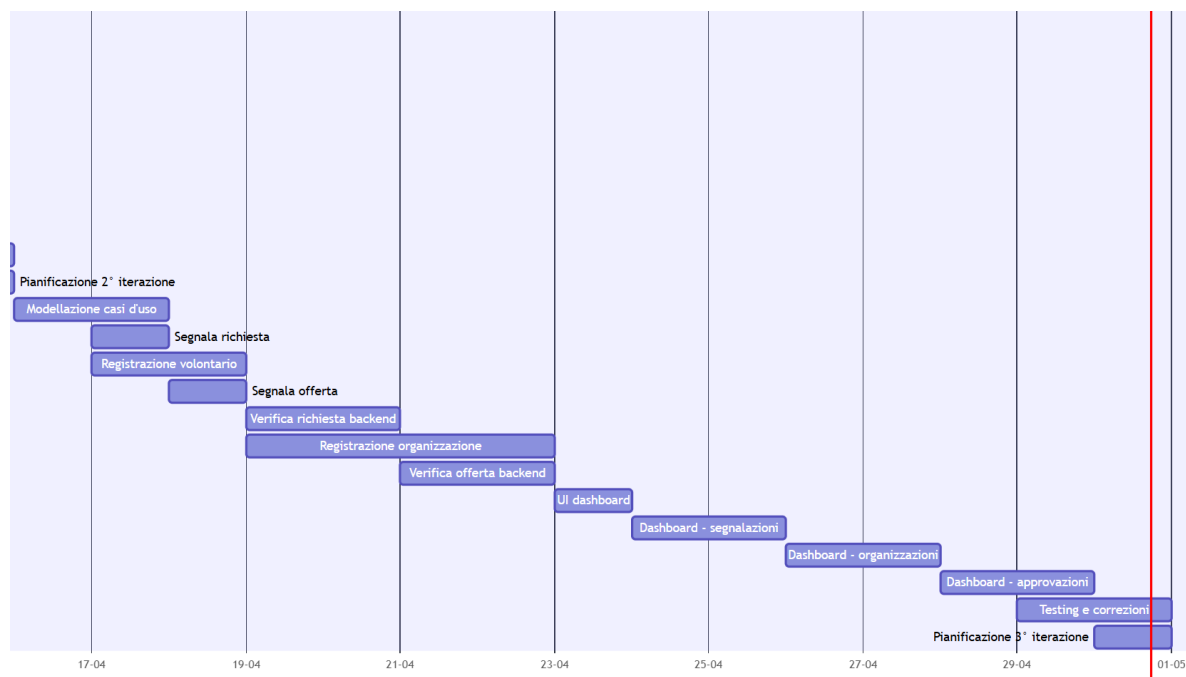


Figure 18: Diagramma di Gantt per la 2° iterazione

4.2 Casi d'uso dettagliati

4.2.1 Registra organizzazione

Nome: Registra organizzazione

Attore primario: Utente non registrato

Attori e interessi:

- *Utente non registrato:* vuole registrarsi velocemente e senza errori; vuole un'interfaccia chiara.

Precondizioni: L'utente non deve essere già registrato. L'utente deve avere un'email valida.

Postcondizioni: L'organizzazione è stata registrata. I suoi dati sono salvati nel database.

Scenario principale:

1. L'utente non registrato inizia la registrazione.
2. Il sistema chiede all'utente non registrato di inserire i dati necessari per la registrazione (nome dell'organizzazione, email, password, telefono, indirizzo sede legale e PIVA).
3. L'utente non registrato inserisce i suoi dati.
4. Il sistema verifica che i campi siano compilati correttamente.
5. L'utente non registrato conferma la registrazione.
6. Il sistema controlla che la email, telefono o PIVA non siano già registrati.
7. Il sistema salva i dati dell'organizzazione.
8. Il sistema invia una richiesta di approvazione al team di controllo.
9. Il sistema notifica l'organizzazione di attendere per email l'esito dell'operazione.
10. (Dopo che un operatore ha approvato la richiesta) Il sistema invia un'email con l'esito dell'operazione all'utente.

11. L'utente attiva l'account.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:

1. Il sistema notifica l'utente non registrato che ha inserito dei dati non validi.
2. Il sistema propone all'utente non registrato di inserire nuovi dati oppure modificare i presenti.
3. Si torna al punto 3.

6a. Il sistema rileva che la email e/o telefono e/o PIVA sono già collegati ad un altro account esistente.

1. Il sistema notifica l'utente non registrato che il dato inserito è già stato utilizzato.
2. Il sistema propone all'utente non registrato di inserire un nuovo dato.
3. Si torna al punto 3.

6b. Il sistema non riesce a verificare se la email e/o telefono e/o PIVA sono già collegati ad un altro account.

1. Il sistema notifica l'utente non registrato che non riesce a verificare la validità dei dati.
2. Il sistema invita l'utente non registrato a riprovare più tardi.
3. Si torna al punto 3.

7. Il sistema non riesce a salvare i dati dell'organizzazione:

1. Il sistema notifica l'utente non registrato che ha riscontrato un errore nel salvataggio dei dati.
2. Il sistema invita l'utente non registrato a riprovare più tardi.
3. Si torna al punto 5.

10. La richiesta di registrazione non è stata approvata:

1. Il sistema invia un'email all'utente con i dettagli della non approvazione e invita l'utente a riprovare con dati diversi o ad inviare un'email al team di controllo.
- *. L'utente non registrato vuole annullare l'operazione di registrazione:
1. L'utente non registrato decide di annullare la registrazione.
 2. Il sistema annulla l'operazione di registrazione.
- *. Errore di sistema o fallimento imprevisto:
1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'utente non registrato a riprovare più tardi e ad inviare una segnalazione per email al team di controllo.

Requisiti speciali:

- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna

Frequenza di ripetizione: Un utente può registrarsi una sola volta, ma il sistema deve essere in grado di gestire molte registrazioni in una giornata.

4.2.2 Registra volontario

Nome: Registra volontario

Attore primario: Utente non registrato

Attori e interessi:

- *Utente non registrato:* vuole registrarsi velocemente e senza errori; vuole un'interfaccia chiara.

Precondizioni: L'utente non deve essere già registrato. L'utente deve avere un'email valida.

Postcondizioni: Il volontario è stato registrato. I suoi dati sono salvati nel database.

Scenario principale:

1. L'utente non registrato inizia la registrazione.
2. Il sistema chiede all'utente non registrato di inserire i dati per la registrazione (nome, cognome, email, password, data di nascita, codice fiscale e città).
3. L'utente non registrato inserisce i suoi dati.
4. Il sistema verifica che i campi siano compilati correttamente.
5. Il sistema controlla che la email non sia già registrata.
6. L'utente non registrato conferma la registrazione.
7. Il sistema salva i dati del volontario.
8. Il sistema invia un'email di conferma al volontario.
9. L'utente attiva l'account.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica l'utente non registrato che ha inserito dei dati non validi.
 2. Il sistema propone all'utente non registrato di inserire nuovi dati oppure modificare i presenti.
 3. Si torna al punto 3.
- 6a. Il sistema rileva che la email è già collegata ad un altro account esistente.
 1. Il sistema notifica l'utente non registrato che la email inserita è già stata utilizzata.

2. Il sistema propone all'utente non registrato di inserire una nuova email oppure provare a fare il login.
 3. Si torna al punto 3.
- 6b. Il sistema non riesce a verificare se l'email è già collegata ad un altro account.
1. Il sistema notifica l'utente non registrato che non riesce a verificare la validità della email.
 2. Il sistema invita l'utente non registrato a riprovare più tardi.
 3. Si torna al punto 3.
9. Il sistema rileva degli errori nella compilazione dei campi nel caso siano compilati:
1. Il sistema notifica l'utente non registrato che ha inserito dei dati non validi.
 2. Il sistema propone all'utente non registrato di inserire nuovi dati.
 3. Si torna al punto 8.
11. Il sistema non riesce a salvare i dati del volontario:
1. Il sistema notifica l'utente non registrato che ha riscontrato un errore nel salvataggio dei dati.
 2. Il sistema invita l'utente non registrato a riprovare più tardi.
 3. Si torna al punto 10.
- *. L'utente non registrato vuole annullare l'operazione di registrazione:
1. L'utente non registrato decide di annullare la registrazione.
 2. Il sistema annulla l'operazione di registrazione.
- *. Errore di sistema o fallimento imprevisto:
1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'utente non registrato a riprovare più tardi e ad inviare una segnalazione per email al team di controllo.

Requisiti speciali:

- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati:

- L'utente non registrato può inserire i dati usando una tastiera.

Frequenza di ripetizione: Un utente non registrato può registrarsi una sola volta, ma il sistema deve essere in grado di gestire molte registrazioni in una giornata.

4.2.3 Approva richiesta

Nome: Approva richiesta

Attore primario: Operatore

Attori e interessi:

- *Operatore*: vuole assicurarsi che le richieste rispettino le linee guida e siano appropriate prima della pubblicazione.
- *Organizzazione*: desidera che la propria richiesta venga approvata in tempi rapidi.
- *Volontario*: vuole visualizzare solo richieste affidabili e verificate.

Precondizioni: È presente almeno una richiesta in attesa di approvazione.

Postcondizioni: La richiesta è stata approvata oppure rifiutata. L'organizzazione riceve una notifica con l'esito della verifica.

Scenario principale:

1. Il sistema mostra le richieste in attesa di approvazione.
2. L'operatore seleziona una richiesta da verificare.
3. Il sistema mostra tutti i dettagli della richiesta.

4. L'operatore esamina le informazioni e decide se approvare o rifiutare la richiesta.
5. L'operatore fornisce una motivazione in caso di rifiuto.
6. Il sistema salva l'esito della verifica.
7. Il sistema notifica l'organizzazione con l'esito.

Scenari alternativi:

6. L'operatore non fornisce una motivazione nel caso di rifiuto:
 1. Il sistema richiede all'operatore di inserire una motivazione obbligatoria.
 2. Si torna al punto 6.
7. Il sistema non riesce a salvare l'esito della verifica:
 1. Il sistema notifica l'operatore di un errore nel salvataggio.
 2. Il sistema invita l'operatore a riprovare più tardi.
 3. Si torna al punto 6.
- *a. L'operatore decide di annullare la verifica:
 1. L'operatore abbandona la verifica della richiesta.
 2. Il sistema non applica modifiche allo stato della richiesta.
 3. Il sistema annulla l'operazione di verifica.
- *b. Errore di sistema o fallimento imprevisto:
 1. Il sistema notifica l'operatore che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'operatore a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- La motivazione del rifiuto deve essere obbligatoria.

- Il sistema deve garantire che nessuna richiesta venga pubblicata senza essere prima verificata.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose approvazioni di richieste durante la giornata.

4.2.4 Approva offerta

Nome: Approva offerta

Attore primario: Operatore

Attori e interessi:

- *Operatore:* vuole assicurarsi che le offerte rispettino le linee guida e siano appropriate prima della pubblicazione.
- *Volontario:* desidera che la propria offerta venga approvata in tempi brevi.
- *Organizzazione:* vuole visualizzare solo offerte affidabili e verificate.

Precondizioni: È presente almeno una offerta in attesa di approvazione.

Postcondizioni: La offerta è stata approvata oppure rifiutata. Il volontario riceve una notifica con l'esito della verifica.

Scenario principale:

1. Il sistema mostra le offerte in attesa di approvazione.
2. L'operatore seleziona un'offerta da verificare.
3. Il sistema mostra tutti i dettagli della offerta.
4. L'operatore esamina le informazioni e decide se approvare o rifiutare la offerta.
5. L'operatore fornisce una motivazione in caso di rifiuto.

6. Il sistema salva l'esito della verifica.
7. Il sistema notifica il volontario con l'esito.

Scenari alternativi:

6. L'operatore non fornisce una motivazione nel caso di rifiuto:
 1. Il sistema richiede all'operatore di inserire una motivazione obbligatoria.
 2. Si torna al punto 6.
7. Il sistema non riesce a salvare l'esito della verifica:
 1. Il sistema notifica l'operatore di un errore nel salvataggio.
 2. Il sistema invita l'operatore a riprovare più tardi.
 3. Si torna al punto 6.
- *a. L'operatore decide di annullare la verifica:
 1. L'operatore abbandona la verifica della offerta.
 2. Il sistema non applica modifiche allo stato della offerta.
 3. Il sistema annulla l'operazione di verifica.
- *b. Errore di sistema o fallimento imprevisto:
 1. Il sistema notifica l'operatore che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'operatore a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- La motivazione del rifiuto deve essere obbligatoria.
- Il sistema deve garantire che nessuna offerta venga pubblicata senza essere prima verificata.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose verifiche di offerte durante la giornata.

4.2.5 Segnala richiesta

Nome: Segnala richiesta

Attore primario: Volontario

Attori e interessi:

- *Volontario:* Vuole segnalare una richiesta. Vuole un'interfaccia intuitiva.
- *Operatore:* Vuole ricevere segnalazioni valide.

Precondizioni: La richiesta deve esistere ed essere verificata. Il volontario sta visualizzando la richiesta nel dettaglio.

Postcondizioni: Il sistema ha inviato agli operatori la segnalazione che dovrà poi essere verificata.

Scenario principale:

1. Il volontario decide di segnalare la richiesta.
2. Il sistema chiede al volontario di inserire il motivo della segnalazione.
3. Il volontario inserisce il motivo della segnalazione.
4. Il volontario conferma l'invio della segnalazione.
5. Il sistema verifica la validità della motivazione inserita.
6. Il sistema salva la segnalazione e la invia agli operatori.

Scenari alternativi:

5. Il sistema rileva che la motivazione inserita non è valida:
 1. Il sistema notifica il volontario che la motivazione inserita non è valida oppure mancante.
 2. Il sistema invita il volontario a inserire una nuova motivazione.
 3. Si torna al punto 3.

6. Il sistema non riesce a verificare la validità della motivazione inserita:
 1. Il sistema notifica il volontario che non riesce a verificare la validità della motivazione inserita.
 2. Il sistema invita il volontario a riprovare più tardi.
 3. Si torna al punto 4.
- *. Errore di sistema o fallimento imprevisto:
 1. Il sistema notifica il volontario che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita il volontario a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il tempo di risposta del sistema durante l'invio della segnalazione non deve superare i 2 secondi.
- La segnalazione deve essere immediatamente resa disponibile per la revisione da parte degli operatori.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose segnalazioni di richieste durante la giornata.

4.2.6 Segnala offerta

Nome: Segnala offerta

Attore primario: Organizzazione

Attori e interessi:

- *Organizzazione:* Vuole segnalare un'offerta. Vuole un'interfaccia intuitiva.

- *Operatore*: Vuole ricevere segnalazioni valide.

Precondizioni: L'offerta deve esistere ed essere verificata. L'organizzazione sta visualizzando l'offerta nel dettaglio.

Postcondizioni: Il sistema ha inviato agli operatori la segnalazione che dovrà poi essere verificata.

Scenario principale:

1. L'organizzazione decide di segnalare l'offerta.
2. Il sistema chiede all'organizzazione di inserire il motivo della segnalazione.
3. L'organizzazione inserisce il motivo della segnalazione.
4. L'organizzazione conferma l'invio della segnalazione.
5. Il sistema verifica la validità della motivazione inserita.
6. Il sistema salva la segnalazione e la invia agli operatori.

Scenari alternativi:

5. Il sistema rileva che la motivazione inserita non è valida:
 1. Il sistema notifica l'organizzazione che la motivazione inserita non è valida oppure mancante.
 2. Il sistema invita l'organizzazione a inserire una nuova motivazione.
 3. Si torna al punto 3.
 6. Il sistema non riesce a verificare la validità della motivazione inserita:
 1. Il sistema notifica l'organizzazione che non riesce a verificare la validità della motivazione inserita.
 2. Il sistema invita l'organizzazione a riprovare più tardi.
 3. Si torna al punto 4.
- *. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'organizzazione che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'organizzazione a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il tempo di risposta del sistema durante l'invio della segnalazione non deve superare i 2 secondi.
- La segnalazione deve essere immediatamente resa disponibile per la revisione da parte degli operatori.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose segnalazioni di offerte durante la giornata.

4.3 Modello di dominio

Per lo sviluppo delle funzionalità di registrazione, segnalazione e approvazione il modello di dominio è stato aggiornato con l'aggiunta di due nuove classi concettuali: **Segnalazione** e **Operatore**

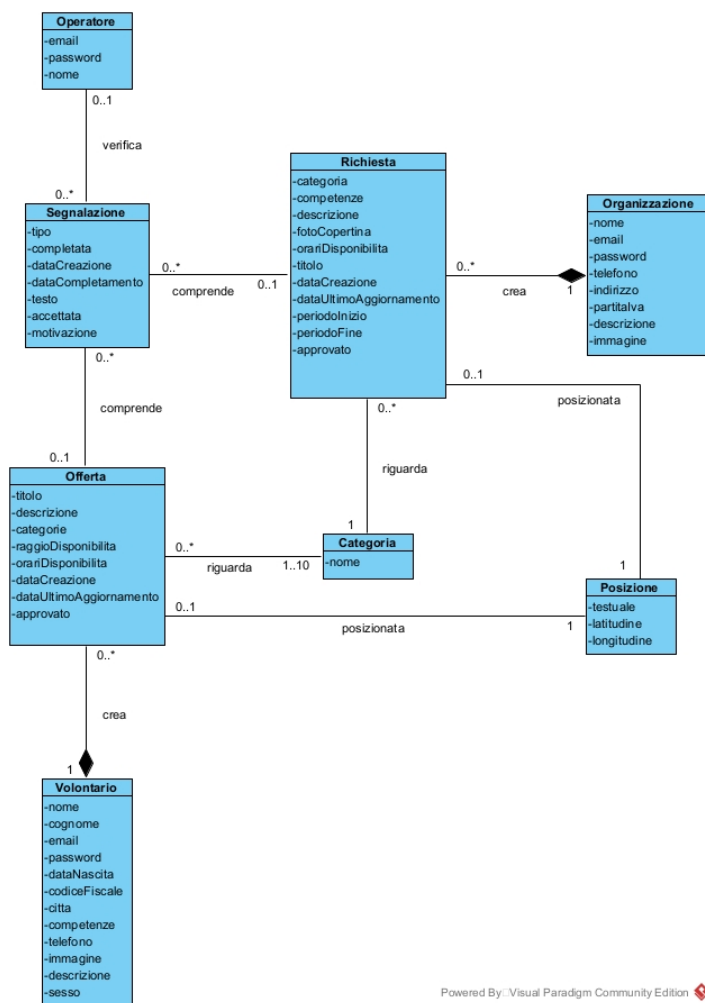


Figure 19: Modello di dominio 2° Iterazione

4.4 Diagrammi di sequenza di sistema (SSD)

4.4.1 Registra volontario

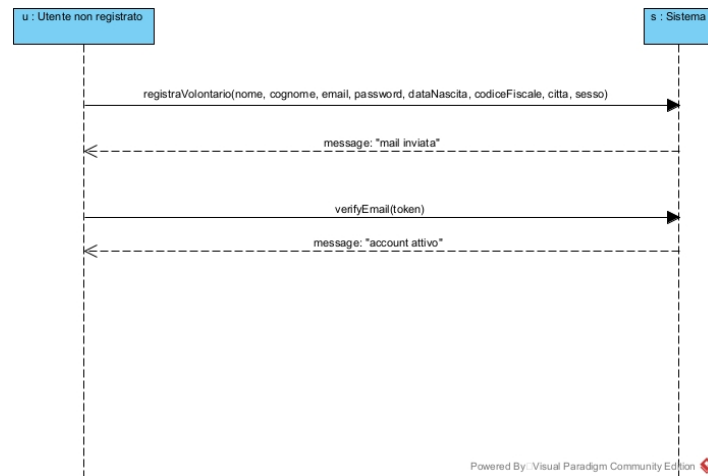


Figure 20: SSD Registra volontario

4.4.2 Registra organizzazione

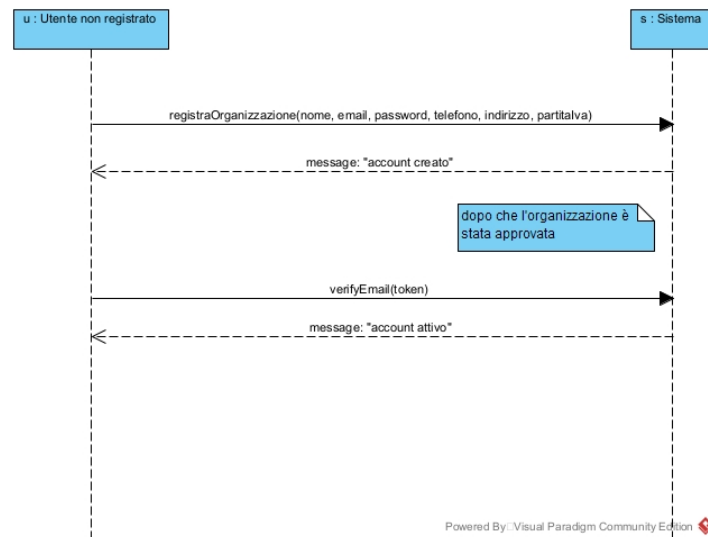


Figure 21: SSD Registra organizzazione

4.4.3 Segnala offerta

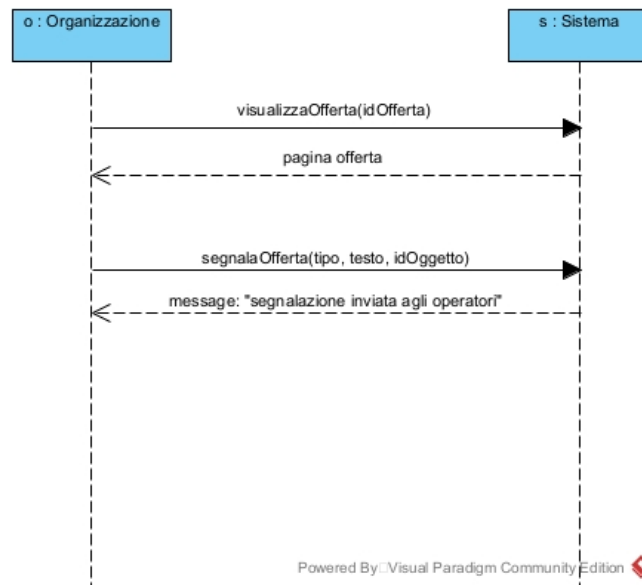


Figure 22: SSD Segnala offerta

4.4.4 Segnala richiesta

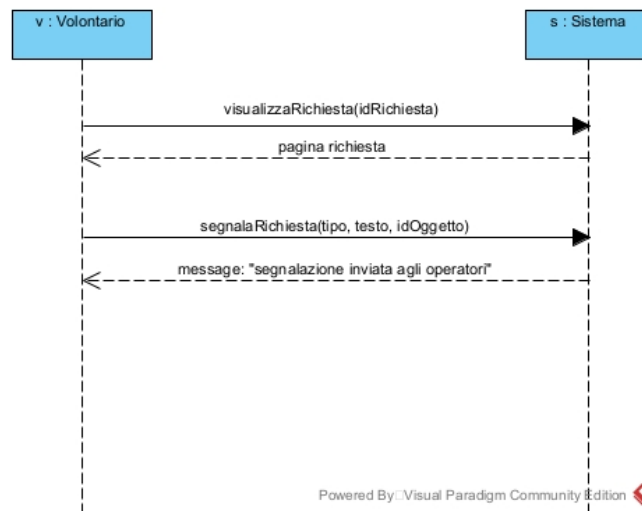


Figure 23: SSD Segnala richiesta

4.4.5 Verifica segnalazione offerta

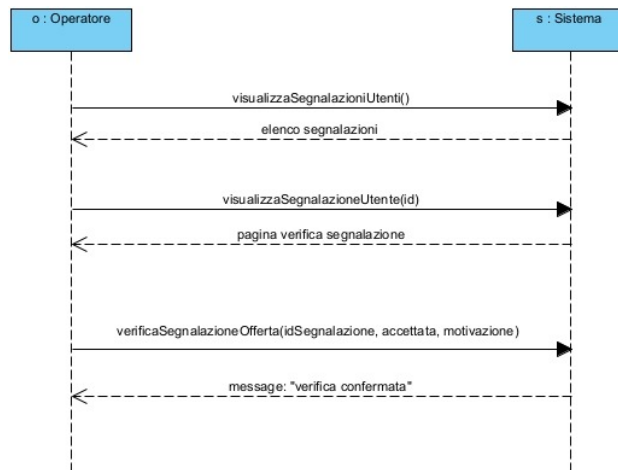


Figure 24: SSD Segnala richiesta

4.4.6 Verifica segnalazione richiesta

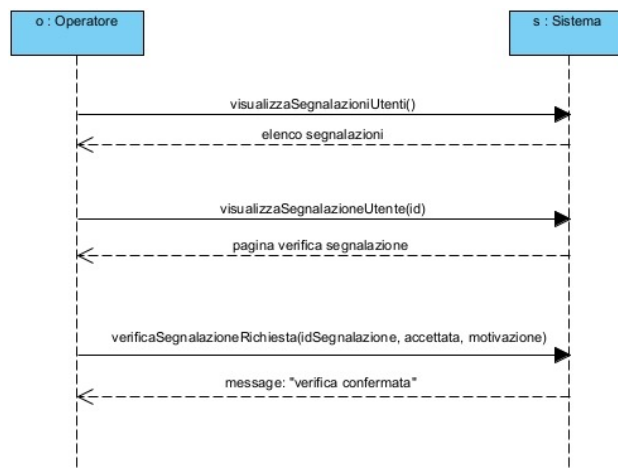


Figure 25: SSD Segnala richiesta

4.5 Diagrammi di attività

4.5.1 Verifica segnalazione richiesta

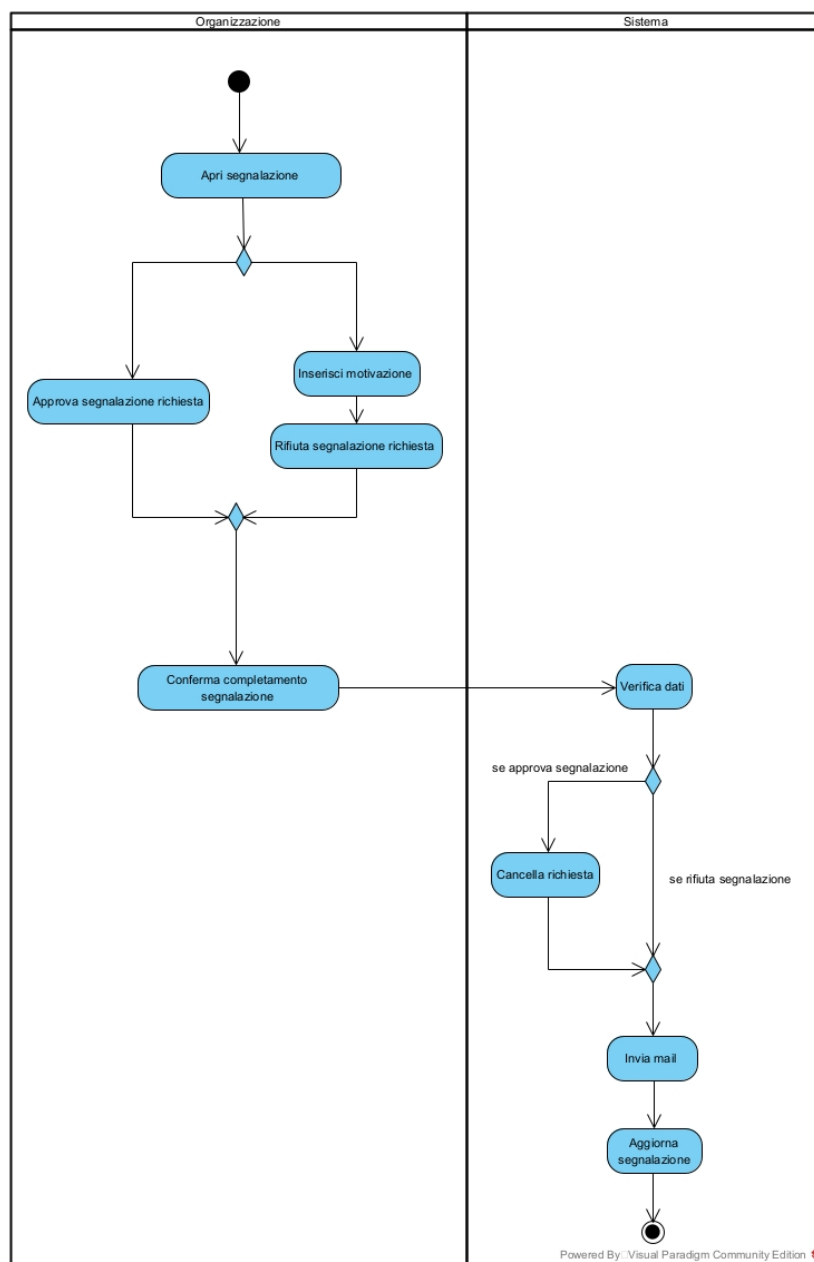


Figure 26: AD Verifica segnalazione richiesta

4.6 Contratti

4.6.1 RegistraVolontario

- **Operazione:** `registraVolontario(nome: String, cognome: String, email: String, password: String, dataNascita: String, codiceFiscale: String, citta: String, sesso: String, telefono: String)`
- **Riferimenti:** casi d'uso: Registra volontario
- **Pre-condizioni:** È in corso la creazione di un Volontario *v*.
- **Post-condizioni:**
 - è stata creata un'istanza *v* di Volontario (**creazione di oggetto**)
 - *v.nome* è diventato *nome* (**modifica di attributo**)
 - *v.cognome* è diventato *cognome* (**modifica di attributo**)
 - *v.email* è diventato *email* (**modifica di attributo**)
 - *v.password* è diventato *password* (**modifica di attributo**)
 - *v.dataNascita* è diventato *dataNascita* (**modifica di attributo**)
 - *v.codiceFiscale* è diventato *codiceFiscale* (**modifica di attributo**)
 - *v.citta* è diventato *citta* (**modifica di attributo**)
 - *v.sesso* è diventato *sesso* (**modifica di attributo**)
 - *v.telefono* è diventato *telefono* (**modifica di attributo**)

4.6.2 VerificaSegnalazioneRichiesta

- **Operazione:** `verificaSegnalazioneRichiesta(idSegnalazione: String, accettata: Boolean, motivazione: String)`
- **Riferimenti:** casi d'uso: Verifica segnalazione richiesta
- **Pre-condizioni:** È in corso la cancellazione di una Richiesta *r*. È in corso la modifica di una Segnalazione *s*.
- **Post-condizioni:**
 - è stata eliminata un'istanza *r* di Richiesta (**eliminazione di oggetto**)
 - *s.completata* è diventato `true` (**modifica di attributo**)

- *s.accettata* è diventato **true** (**modifica di attributo**)
- *s.dataCompletamento* è stato impostato con la data e ora correnti (**modifica di attributo**)
- la Segnalazione *s* è stata associata con l'Operatore (**formazione di collegamento**)

4.7 Diagramma delle classi di progetto

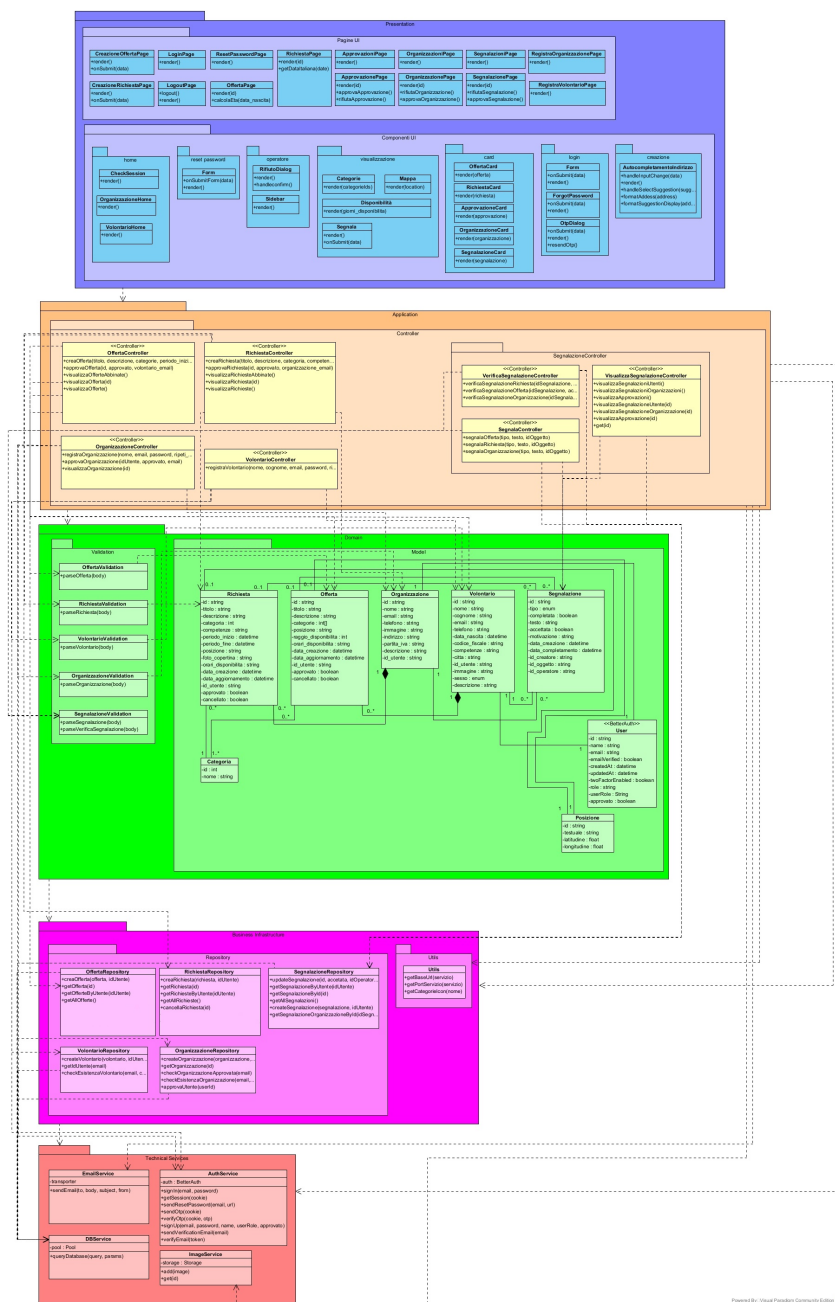


Figure 27: Diagramma delle classi di progetto 2° iterazione

4.8 Diagrammi di sequenza (SD)

4.8.1 Registra volontario

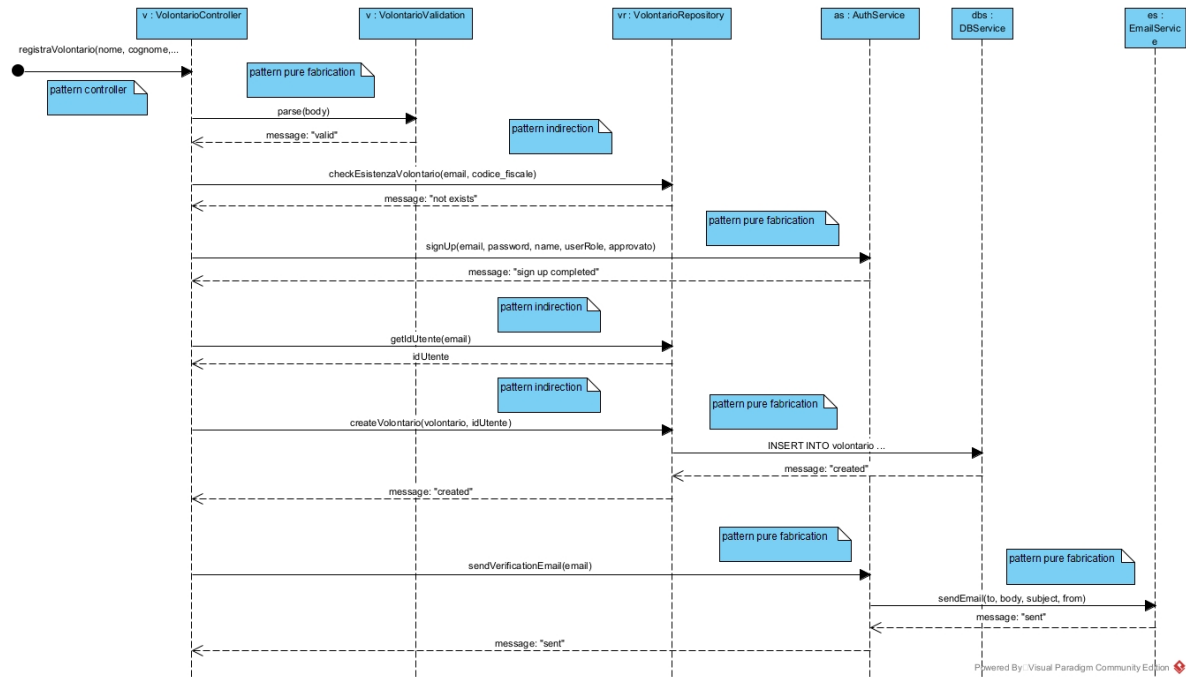
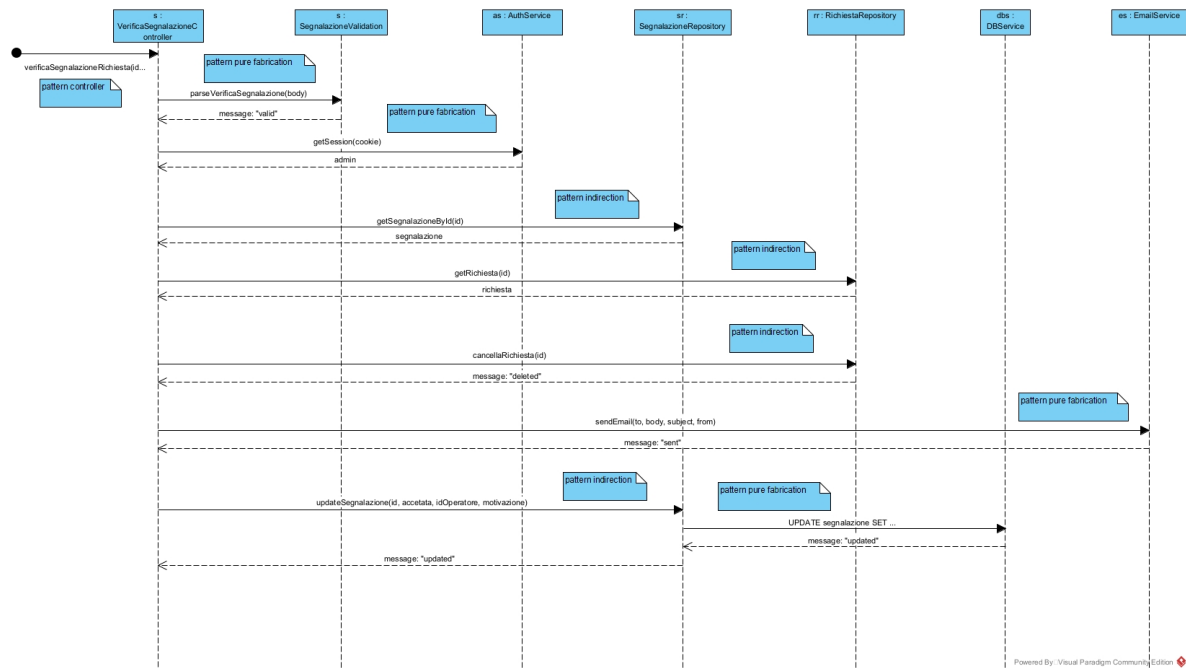


Figure 28: SD Registra volontario

4.8.2 Verifica segnalazione richiesta



4.9 Pianificazione 3° Iterazione

Per la terza iterazione è stato deciso di sviluppare le funzionalità relative ai seguenti casi d'uso:

- Ricerca offerta
- Ricerca richiesta
- Visualizza offerte abbinate
- Visualizza richieste abbinate

5 3° iterazione

5.1 Diagramma di Gantt

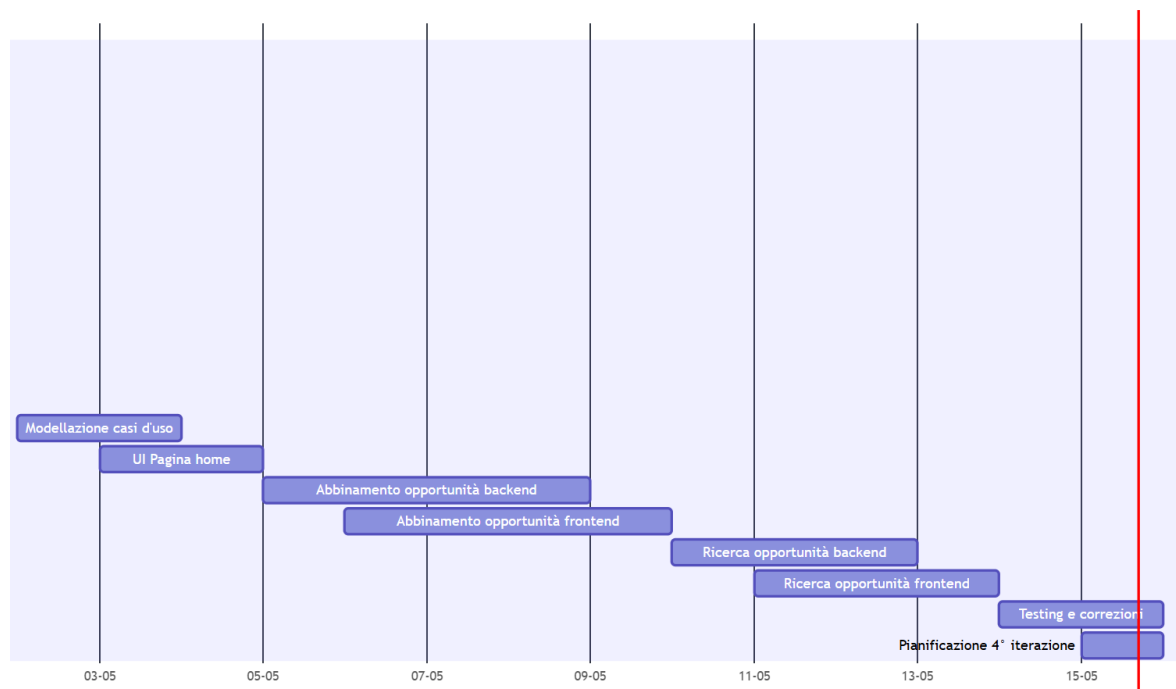


Figure 30: Diagramma di Gantt per la 3° iterazione

5.2 Casi d'uso dettagliati

5.2.1 Ricerca richiesta

Nome: Ricerca richiesta

Attore primario: Utente non registrato, Volontario

Attori e interessi:

- *Utente non registrato, Volontario:* vuole cercare una richiesta in maniera semplice, eventualmente filtrando e ordinando i risultati.

Precondizioni: Il volontario deve essere autenticato nel sistema.

Postcondizioni: La richiesta è stata trovata, è possibile visualizzarla.

Scenario principale:

1. L'utente inizia la ricerca.
2. L'utente inserisce il testo di ricerca.
3. Il sistema mostra i primi risultati suggeriti.
4. L'utente conferma la ricerca.
5. Il sistema mostra tutti i risultati della ricerca.

Scenari alternativi:

2a. L'utente vuole filtrare i risultati per categoria:

1. L'utente seleziona le categorie che vuole visualizzare.
2. L'utente conferma l'applicazione dei filtri selezionati.
3. Si passa al punto 3.

2b. L'utente vuole filtrare i risultati per posizione:

1. L'utente inserisce la posizione da cui vuole partire.
2. L'utente inserisce il raggio di validità.
3. L'utente conferma l'applicazione dei filtri selezionati.
4. Si passa al punto 3.

2c. L'utente vuole filtrare i risultati per periodo:

1. L'utente inserisce la data di inizio e di fine del periodo di tempo.
2. L'utente conferma l'applicazione dei filtri selezionati.
3. Si passa al punto 3.

5. Il sistema non riesce a ricercare le richieste:

1. Il sistema notifica l'utente non registrato che ha riscontrato un errore nella ricerca delle richieste.

2. Il sistema invita l'utente a riprovare più tardi.
 3. Si torna al punto 4.
- *. Errore di sistema o fallimento imprevisto:
1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email al team di controllo.

Requisiti speciali:

- Il tempo di esecuzione del sistema per ogni ricerca effettuata deve essere minore di 5 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna

Frequenza di ripetizione: Il sistema deve essere in grado di gestire un numero molto elevato di ricerche in una giornata.

5.2.2 Visualizza richieste abbinate

Nome: Visualizza richieste abbinate

Attore primario: Volontario

Attori e interessi:

- *Volontario*: vuole scoprire richieste abbinate per potersi candidare; vuole un'interfaccia intuitiva e che mostri tutti i dettagli.

Precondizioni:

- La richiesta deve esistere ed essere verificata.

Postcondizioni: Le richieste abbinate al volontario sono visualizzabili.

Scenario principale:

1. L'utente apre la pagina principale.
2. Il sistema mostra le richieste abbinate in forma abbreviata.

Scenari alternativi:

2a. Il sistema non riesce a trovare nessuna richiesta abbinata:

1. Il sistema notifica il volontario che non ci sono richieste abbinate.
2. Si torna al punto 1.

2b. Il sistema non riesce a mostrare le richieste abbinate per errore interno.

1. Il sistema notifica il volontario che c'è stato un problema interno e che non riesce a mostrare le richieste abbinate in quel momento.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.
4. Si torna al punto 1.

*. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica il volontario che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita il volontario a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il tempo di esecuzione del sistema per visualizzare le richieste abbinate deve essere minore di 5 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose visualizzazioni di richieste abbinate durante la giornata.

5.3 Diagrammi di attività

5.3.1 Ricerca richiesta

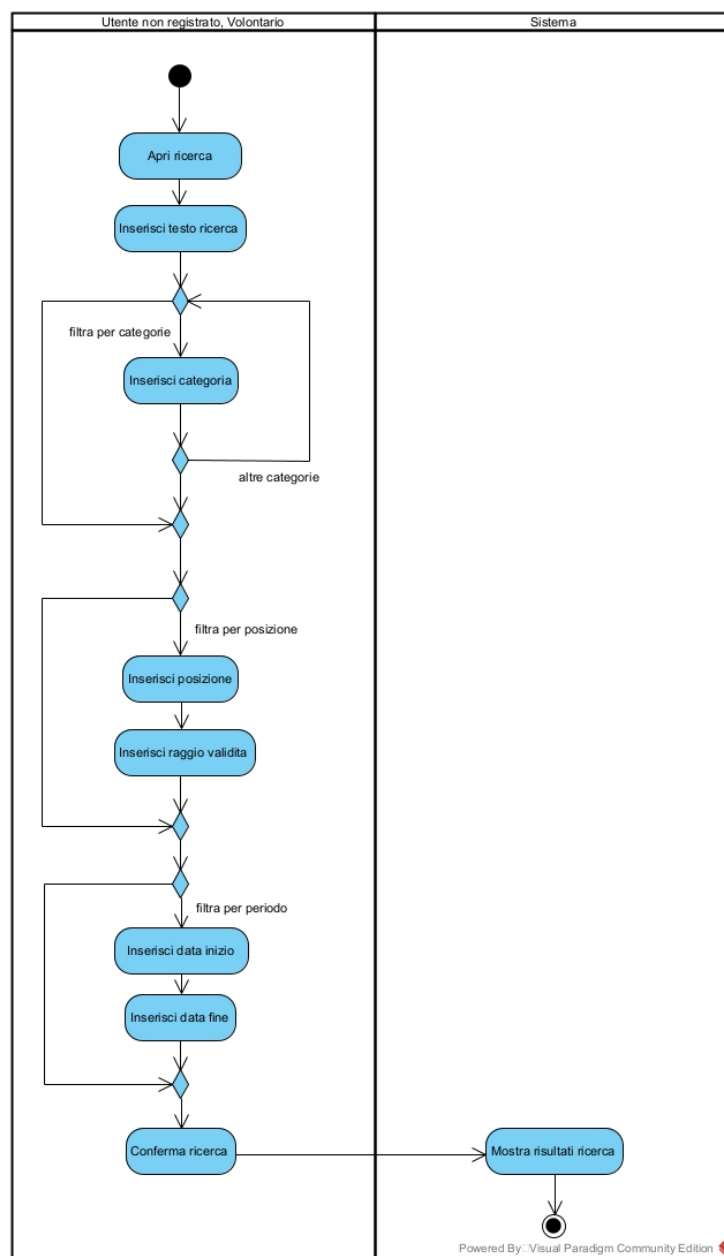


Figure 31: AD Ricerca richiesta

5.4 Modello di architettura logica

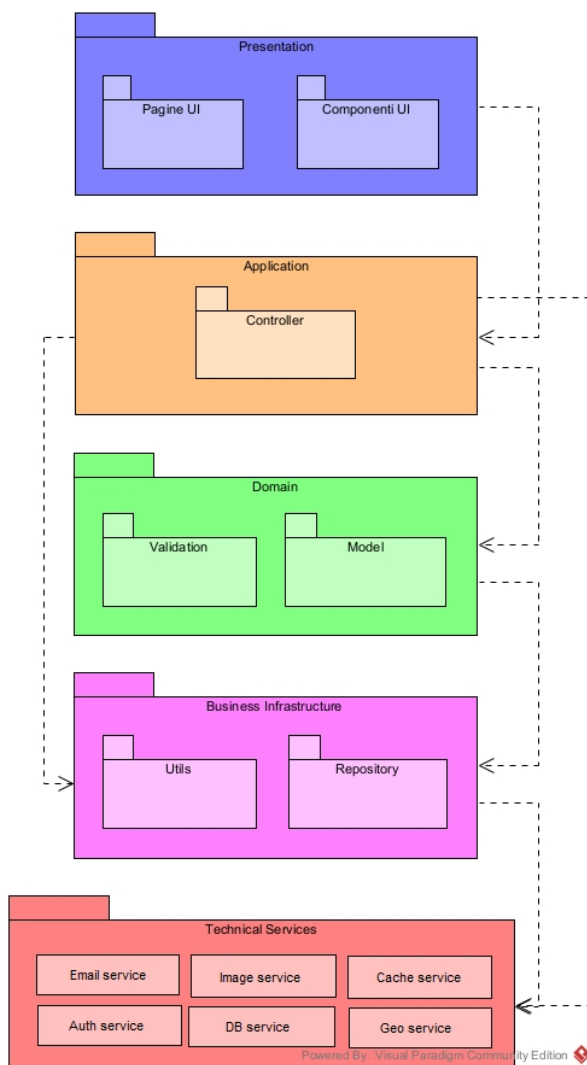


Figure 32: Modello di architettura logica 3° iterazione

5.5 Diagramma delle classi di progetto

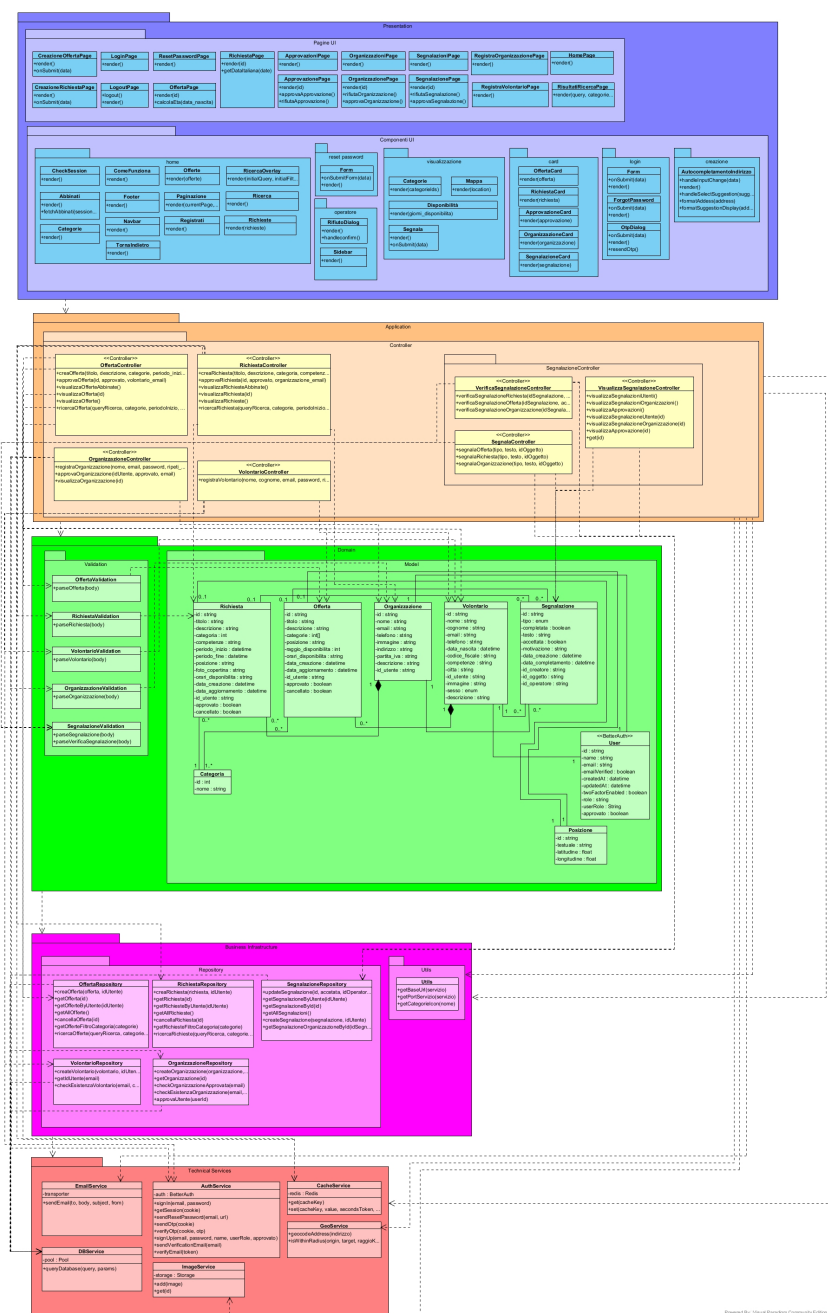
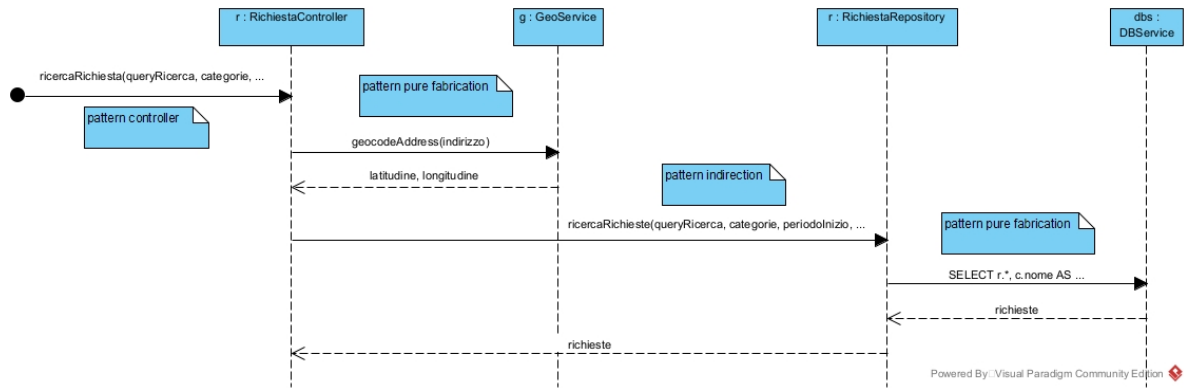


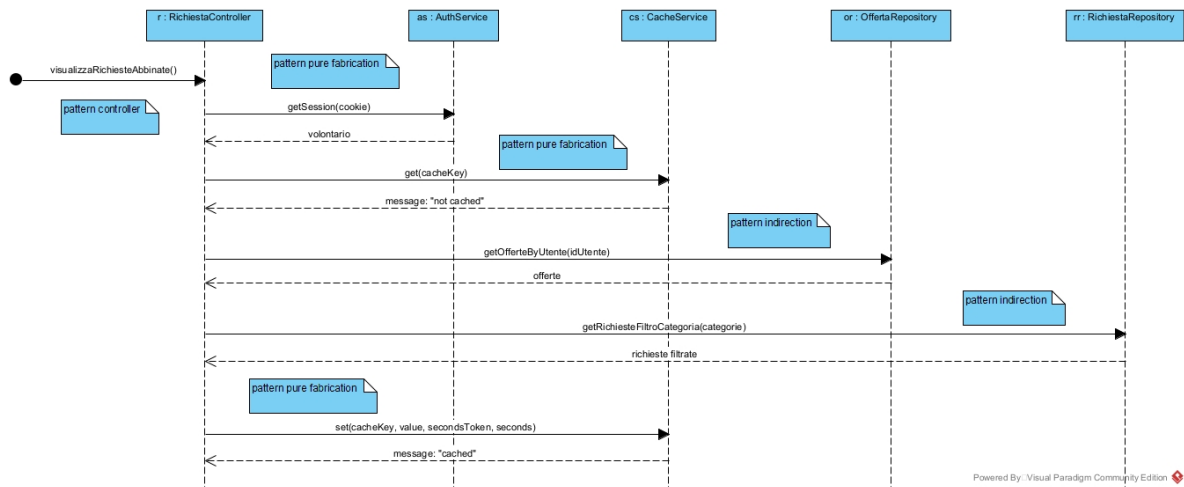
Figure 33: Diagramma delle classi di progetto 3° iterazione

5.6 Diagrammi di sequenza (SD)

5.6.1 Ricerca richiesta



5.6.2 Visualizza richieste abbinate



5.7 Testing e analisi

Oltre al testing delle funzionalità eseguito con Playwright sono state anche corrette delle reliability issues analizzate tramite SonarQube. Queste riguardavano principal-

mente l'uso non corretto di link, l'utilizzo di componenti non completamente accessibili tramite tastiera e l'assegnazione di parametri inutilizzati.

5.8 Pianificazione 4° Iterazione

Per la quarta iterazione è stato deciso di sviluppare le funzionalità relative alla chat integrata e tutte quelle eseguibili dalla pagina profilo, più altre funzionalità di minore importanza. Esse sono rappresentate dai seguenti casi d'uso:

- Invia messaggio
- Modifica offerta
- Modifica richiesta
- Cancella offerta
- Cancella richiesta
- Visualizza profilo
- Modifica dati profilo
- Modifica impostazioni
- Condividi annuncio
- Aggiungi offerta ai preferiti
- Rimuovi offerta dai preferiti
- Aggiungi richiesta ai preferiti
- Rimuovi richiesta dai preferiti

6 4° iterazione

6.1 Diagramma di Gantt

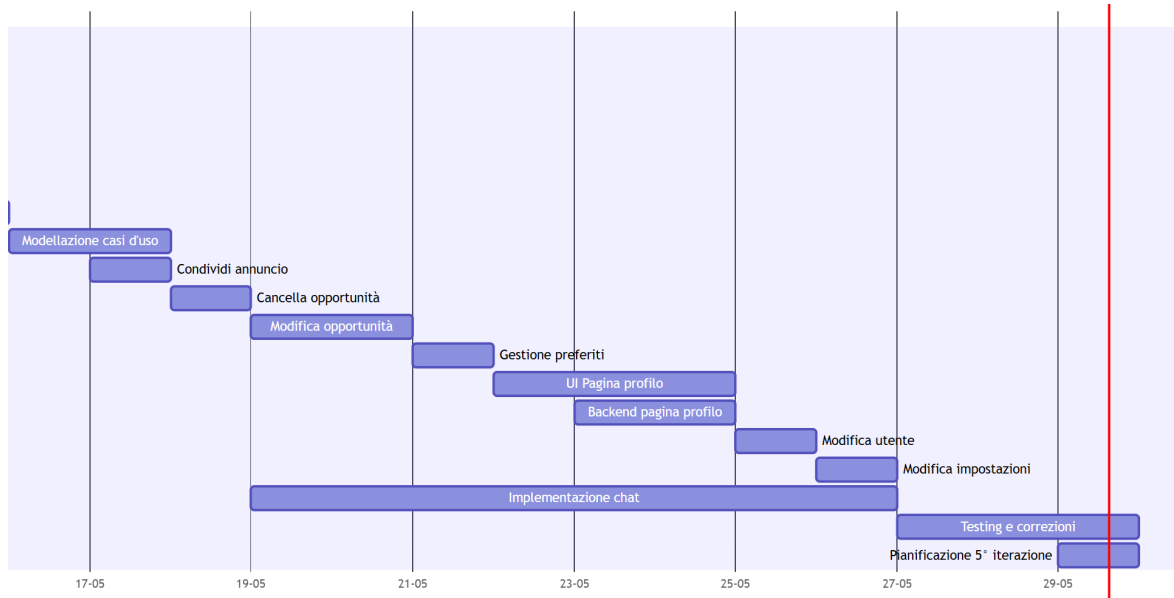


Figure 36: Diagramma di Gantt per la 4° iterazione

6.2 Casi d'uso dettagliati

6.2.1 Modifica richiesta

Nome: Modifica richiesta

Attore primario: Organizzazione

Attori e interessi:

- *Organizzazione*: vuole modificare una richiesta precedentemente pubblicata; vuole un'interfaccia chiara ed un inserimento veloce e intuitivo.
- *Volontario*: vuole avere accesso a richieste valide e aggiornate.

Precondizioni: L'organizzazione è autenticata nel sistema.

Postcondizioni: La richiesta è stata aggiornata con i nuovi dati inseriti.

Scenario principale:

1. L'organizzazione inizia la modifica della richiesta.
2. Il sistema mostra la pagina di modifica in cui vengono mostrati tutti i dati precedentemente inseriti della richiesta (titolo, descrizione, categoria, competenze richieste, periodo di tempo, orari di disponibilità, posizione geografica e foto di copertina).
3. L'organizzazione aggiorna i dati voluti.
4. Il sistema verifica che i campi siano compilati correttamente.
5. L'organizzazione conferma l'aggiornamento della richiesta.
6. Il sistema controlla che non esistano altre richieste dell'organizzazione con lo stesso titolo.
7. Il sistema aggiorna i dati della richiesta.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica l'organizzazione che ha inserito dei dati non validi.
 2. Il sistema propone all'organizzazione di modificare i dati inseriti.
 3. Si torna al punto 3.
- 6a. Il sistema rileva che è presente un'altra richiesta dell'organizzazione con lo stesso titolo.
 1. Il sistema notifica l'organizzazione che il titolo inserito è già stato utilizzato.
 2. Il sistema propone all'organizzazione di inserire un nuovo titolo.
 3. Si torna al punto 3.
- 6b. Il sistema non riesce a verificare se è presente un'altra richiesta con lo stesso titolo.

1. Il sistema notifica l'organizzazione che non riesce a verificare i dati.
 2. Il sistema invita l'organizzazione a riprovare più tardi.
 3. Si torna al punto 5.
7. Il sistema non riesce a aggiornare i dati della richiesta:
1. Il sistema notifica l'organizzazione che ha riscontrato un errore nell'aggiornamento dei dati.
 2. Il sistema invita l'organizzazione a riprovare più tardi.
 3. Si torna al punto 5.
- *a. L'organizzazione vuole annullare l'operazione di modifica:
1. L'organizzazione annulla la modifica della richiesta.
 2. Il sistema annulla l'operazione di modifica e riporta l'organizzazione alla pagina della richiesta.
- *b. Errore di sistema o fallimento imprevisto:
1. Il sistema notifica l'organizzazione che l'operazione non è andata a buon fine.
 2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
 3. Il sistema invita l'organizzazione a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose modifiche di richieste durante la giornata.

6.2.2 Modifica dati profilo

Nome: Modifica dati profilo

Attore primario: Volontario, organizzazione

Attori e interessi:

- *Organizzazione:* vuole modificare i dati del proprio profilo; vuole avere accesso a profili con informazioni valide e aggiornate
- *Volontario:* vuole modificare i dati del proprio profilo; vuole avere accesso a profili con informazioni valide e aggiornate

Precondizioni: L'organizzazione / volontario è autenticata nel sistema.

Postcondizioni: Il profilo è stato aggiornato con i nuovi dati inseriti.

Scenario principale:

1. L'utente inizia la modifica del proprio profilo.
2. Il sistema mostra la pagina di modifica in cui vengono mostrati tutti i dati precedentemente inseriti del proprio profilo.
3. L'utente aggiorna i dati voluti.
4. Il sistema verifica che i campi siano compilati correttamente.
5. L'utente conferma l'aggiornamento del profilo.
6. Il sistema aggiorna i dati del profilo.

Scenari alternativi:

4. Il sistema rileva degli errori nella compilazione dei campi:
 1. Il sistema notifica l'utente che ha inserito dei dati non validi.
 2. Il sistema propone all'utente di modificare i dati inseriti.
 3. Si torna al punto 3.

6. Il sistema non riesce a aggiornare i dati del profilo:

1. Il sistema notifica l'utente che ha riscontrato un errore nell'aggiornamento dei dati.
2. Il sistema invita l'utente a riprovare più tardi.
3. Si torna al punto 5.

*a. L'utente vuole annullare l'operazione di modifica:

1. L'utente annulla la modifica del profilo.
2. Il sistema annulla l'operazione di modifica e riporta l'utente alla pagina profilo.

*b. Errore di sistema o fallimento imprevisto:

1. Il sistema notifica l'utente che l'operazione non è andata a buon fine.
2. Se possibile, il sistema registra l'errore nei log per la diagnosi.
3. Il sistema invita l'utente a riprovare più tardi e ad inviare una segnalazione per email agli sviluppatori.

Requisiti speciali:

- Il tempo di risposta del sistema per ogni azione effettuata deve essere minore di 2 secondi.

Elenco delle varianti tecnologiche e dei dati: Nessuna.

Frequenza di ripetizione: Il sistema deve essere in grado di gestire numerose modifiche di profili durante la giornata.

6.3 Modello di dominio

Per lo sviluppo delle funzionalità relative alla chat, alla gestione dei preferiti e delle impostazioni profilo, il modello di dominio è stato aggiornato con l'aggiunta di tre nuove classi concettuali: **Chat**, **Messaggio** e **Impostazioni**

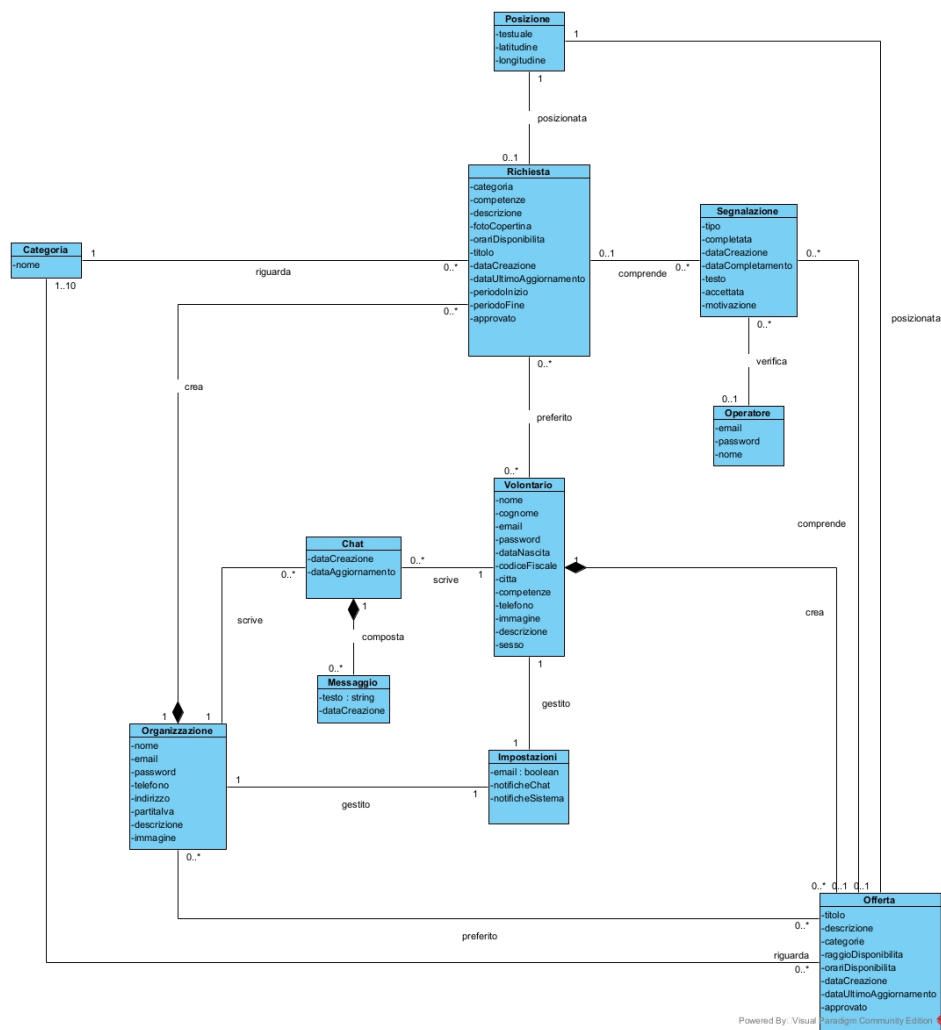


Figure 37: Modello di dominio 4° Iterazione

6.4 Diagrammi di sequenza di sistema (SSD)

6.4.1 Invia messaggio

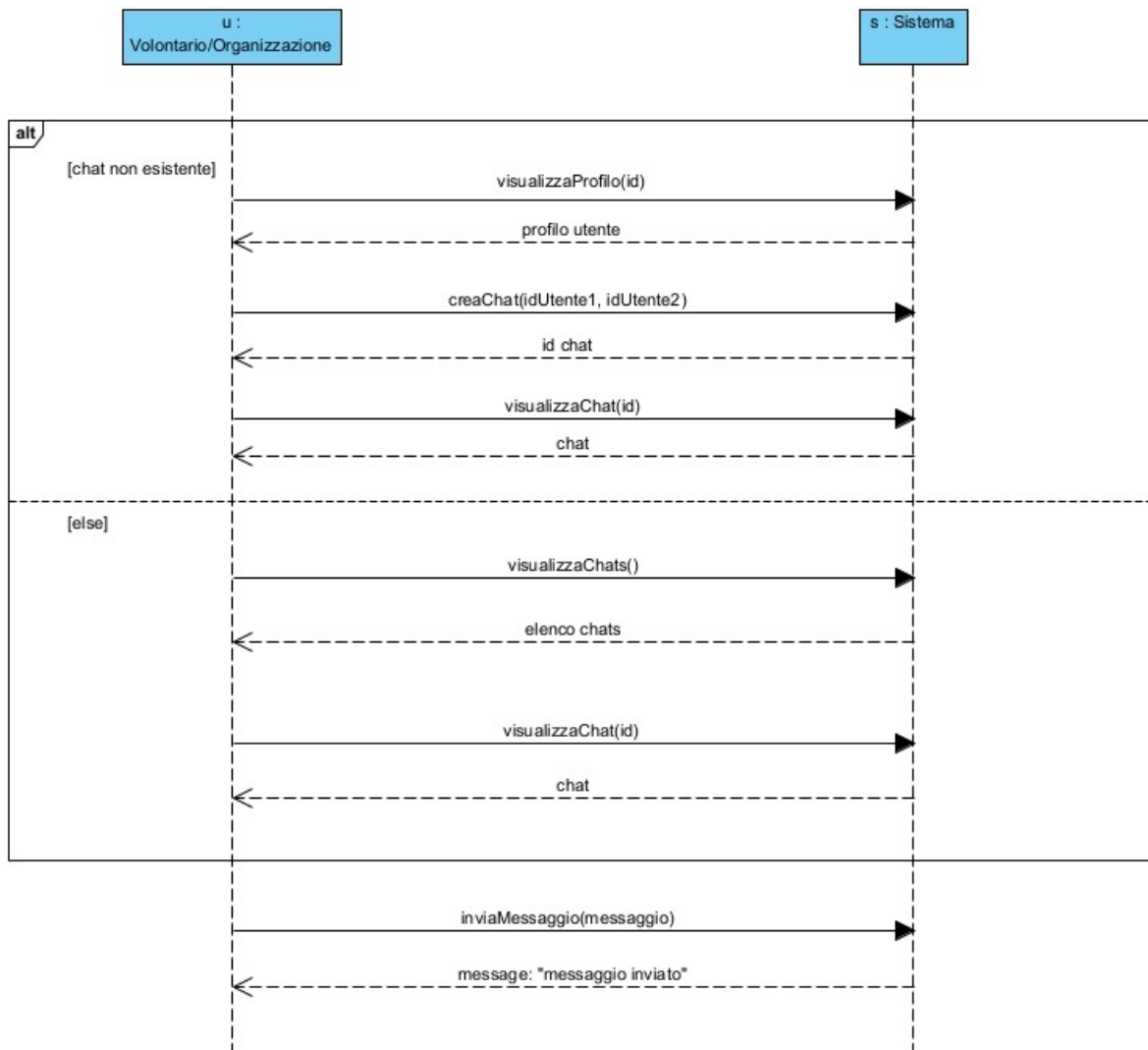


Figure 38: SSD Invia messaggio

6.5 Diagrammi di attività

6.5.1 Modifica profilo volontario

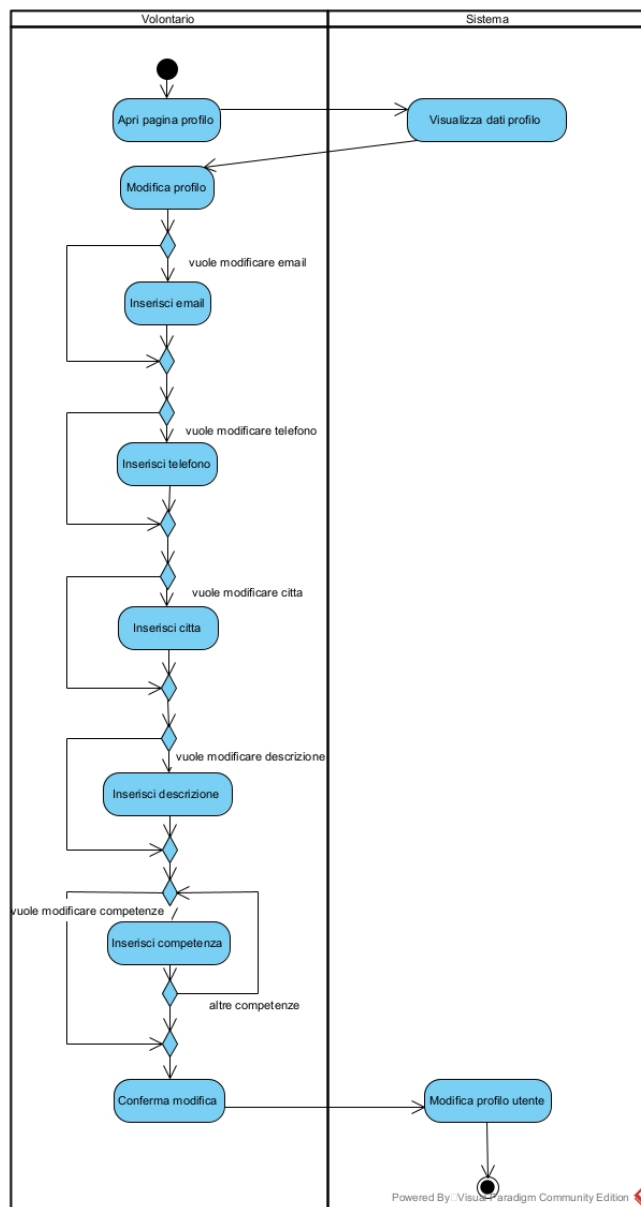


Figure 39: AD Modifica profilo volontario

6.6 Modello di architettura logica

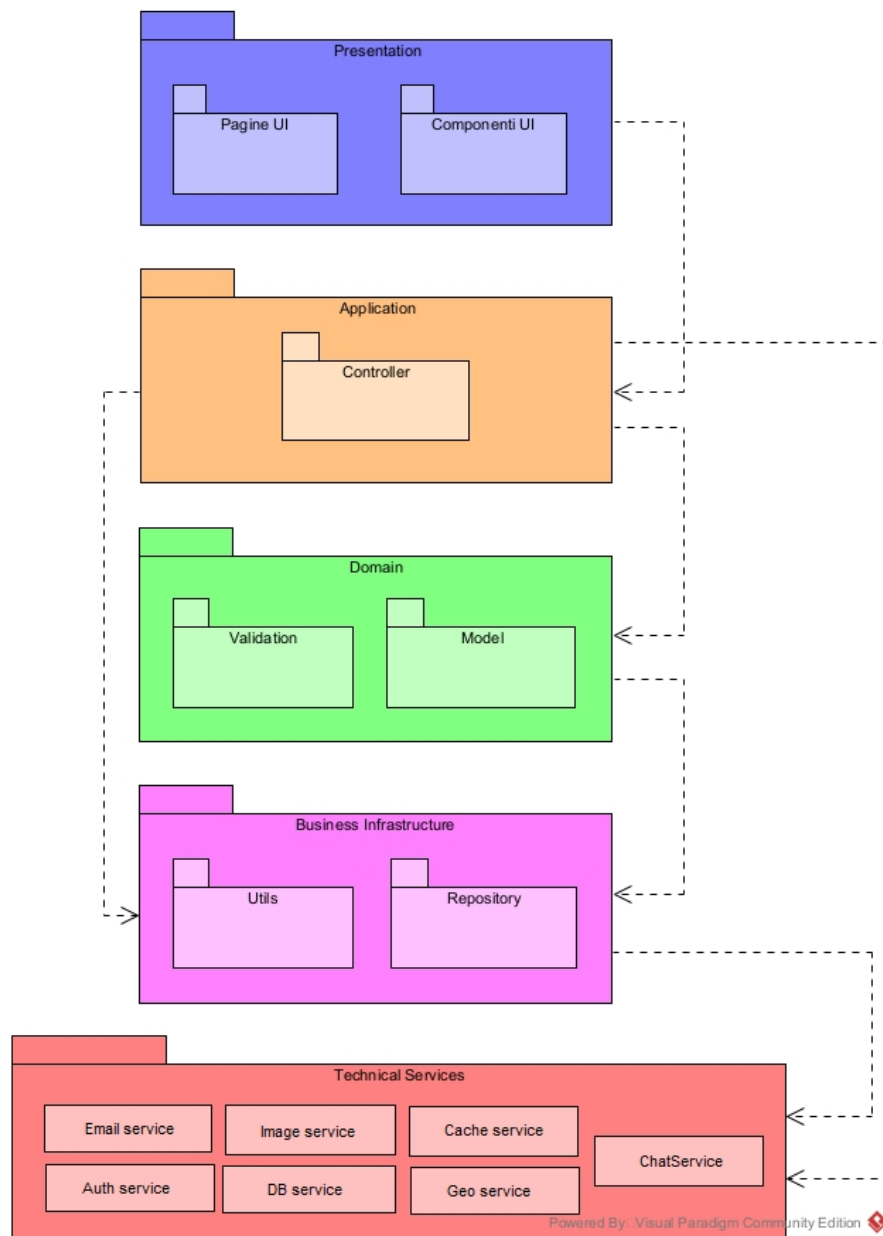


Figure 40: Modello di architettura logica 4° iterazione

6.7 Contratti

6.7.1 Modifica Richiesta

- **Operazione:** `modificaRichiesta(id: String, titolo: String, descrizione: String, categoria: int, periodoInizio: DateTime, periodoFine: DateTime, posizione: String, fotoCopertina: String, orariDisponibilita: OrariDisponibilita, competenze: String[])`
- **Riferimenti:** casi d'uso: Modifica richiesta
- **Pre-condizioni:** È in corso la modifica di una Richiesta *r*.
- **Post-condizioni:**
 - *r.titolo* è diventato *titolo* (**modifica di attributo**)
 - *r.descrizione* è diventato *descrizione* (**modifica di attributo**)
 - *r.categoria* è diventato *categoria* (**modifica di attributo**)
 - *r.periodoInizio* è diventato *periodoInizio* (**modifica di attributo**)
 - *r.periodoFine* è diventato *periodoFine* (**modifica di attributo**)
 - *r.posizione* è diventato *posizione* (**modifica di attributo**)
 - *r.fotoCopertina* è diventato *fotoCopertina* (**modifica di attributo**)
 - *r.orariDisponibilita* è diventato *orariDisponibilita* (**modifica di attributo**)
 - *r.competenze* è diventato *competenze* (**modifica di attributo**)
 - *r.dataUltimoAggiornamento* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - è stata creata un'istanza *p* di Posizione (**creazione di oggetto**)
 - *p.testuale* è diventato *posizione* (**modifica di attributo**)
 - *p.latitudine* e *p.longitudine* sono stati impostati con la conversione dell'indirizzo testuale (**modifica di attributo**)
 - la Posizione *p* è stata associata con la Richiesta *r* (**formazione di collegamento**)

6.7.2 Crea chat

- **Operazione:** `creaChat(idUtente1: String, idUtente2: String)`
- **Riferimenti:** casi d'uso: Invia messaggio
- **Pre-condizioni:** È in corso l'invio di un messaggio.
- **Post-condizioni:**
 - è stata creata un'istanza *c* di Chat (**Creazione di oggetto**)
 - la chat *c* è stata associata con l'utente di id *idUtente1* (**formazione di collegamento**)
 - la chat *c* è stata associata con l'utente di id *idUtente2* (**formazione di collegamento**)
 - *o.dataCreazione* è stato impostato con la data e ora correnti (**modifica di attributo**)
 - *o.dataAggiornamento* è stato impostato con la data e ora correnti (**modifica di attributo**)

6.8 Diagramma delle classi di progetto

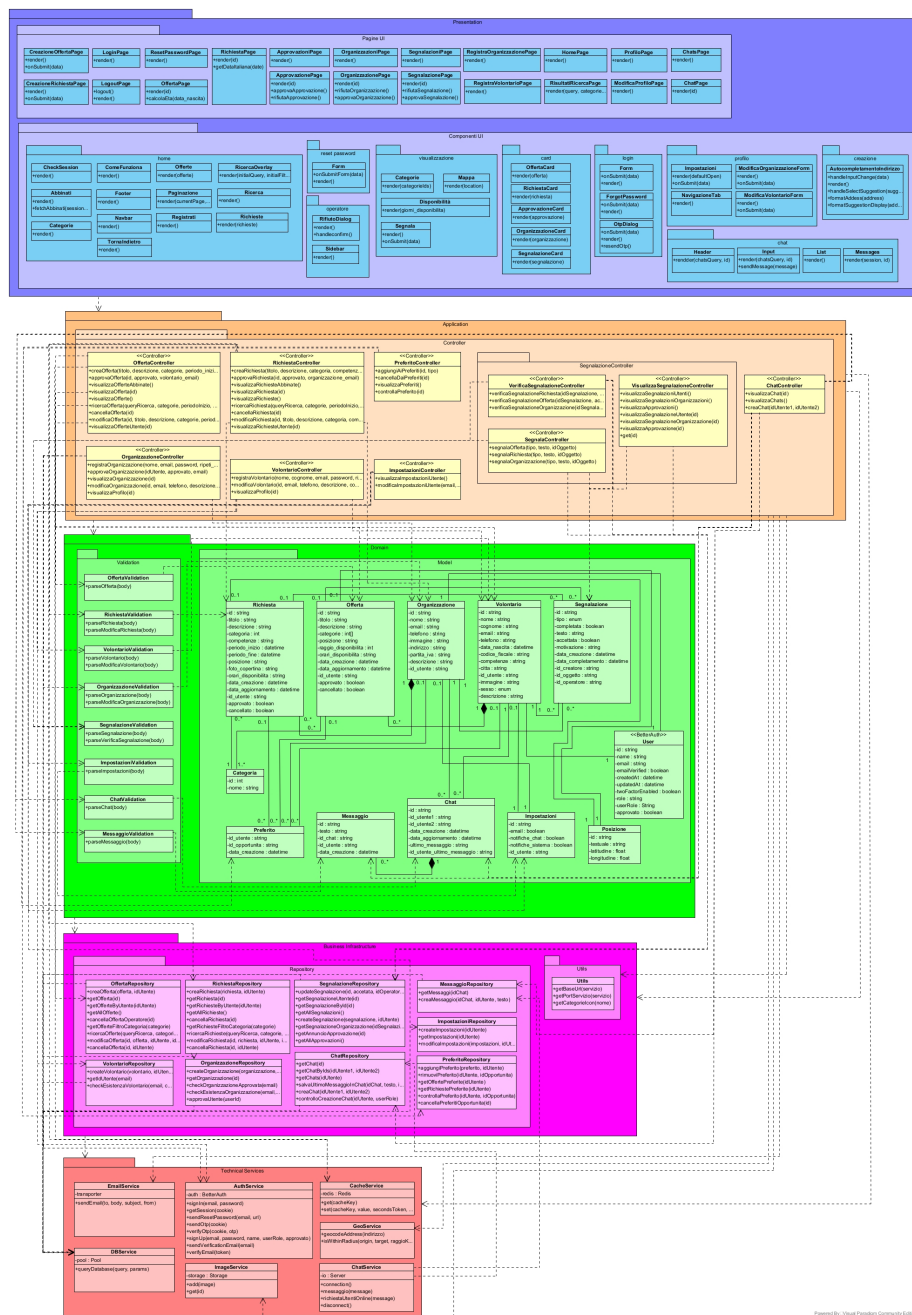


Figure 41: Diagramma delle classi di progetto 4° iterazione

6.9 Diagrammi di sequenza (SD)

6.9.1 Crea chat

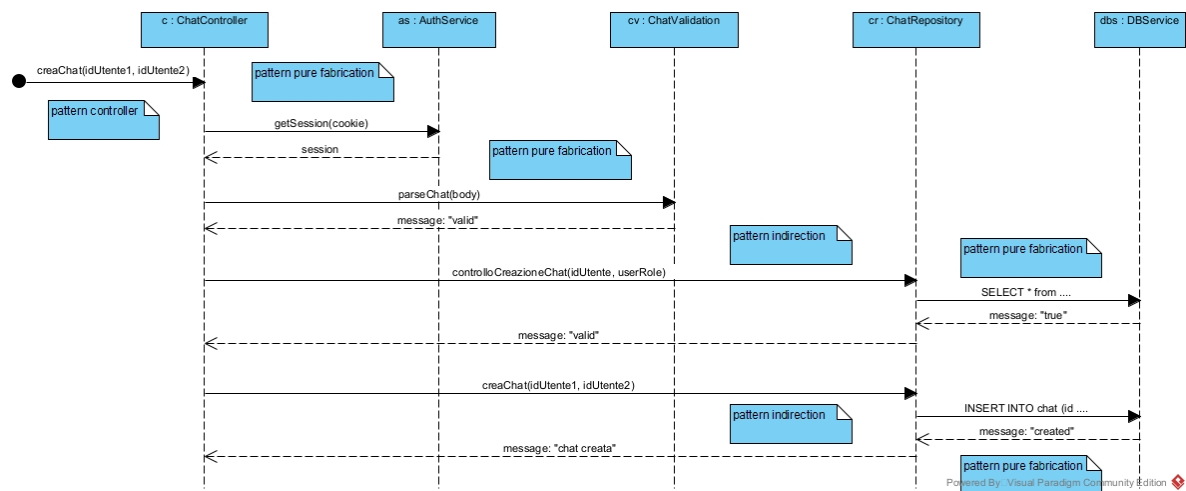


Figure 42: SD Crea chat

6.9.2 Modifica offerta

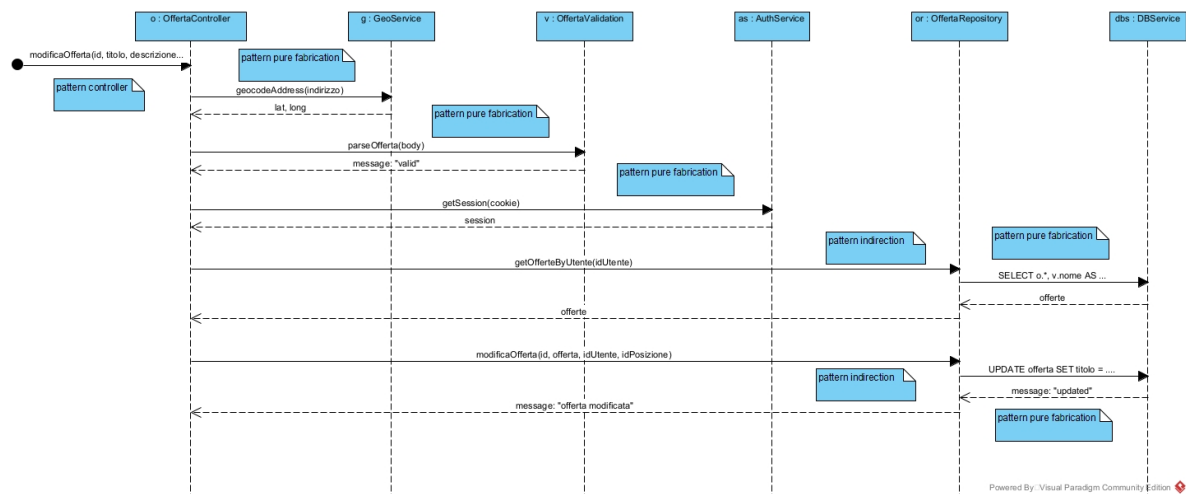


Figure 43: SD Modifica offerta

6.10 Diagrammi di macchine a stati

6.10.1 Richiesta

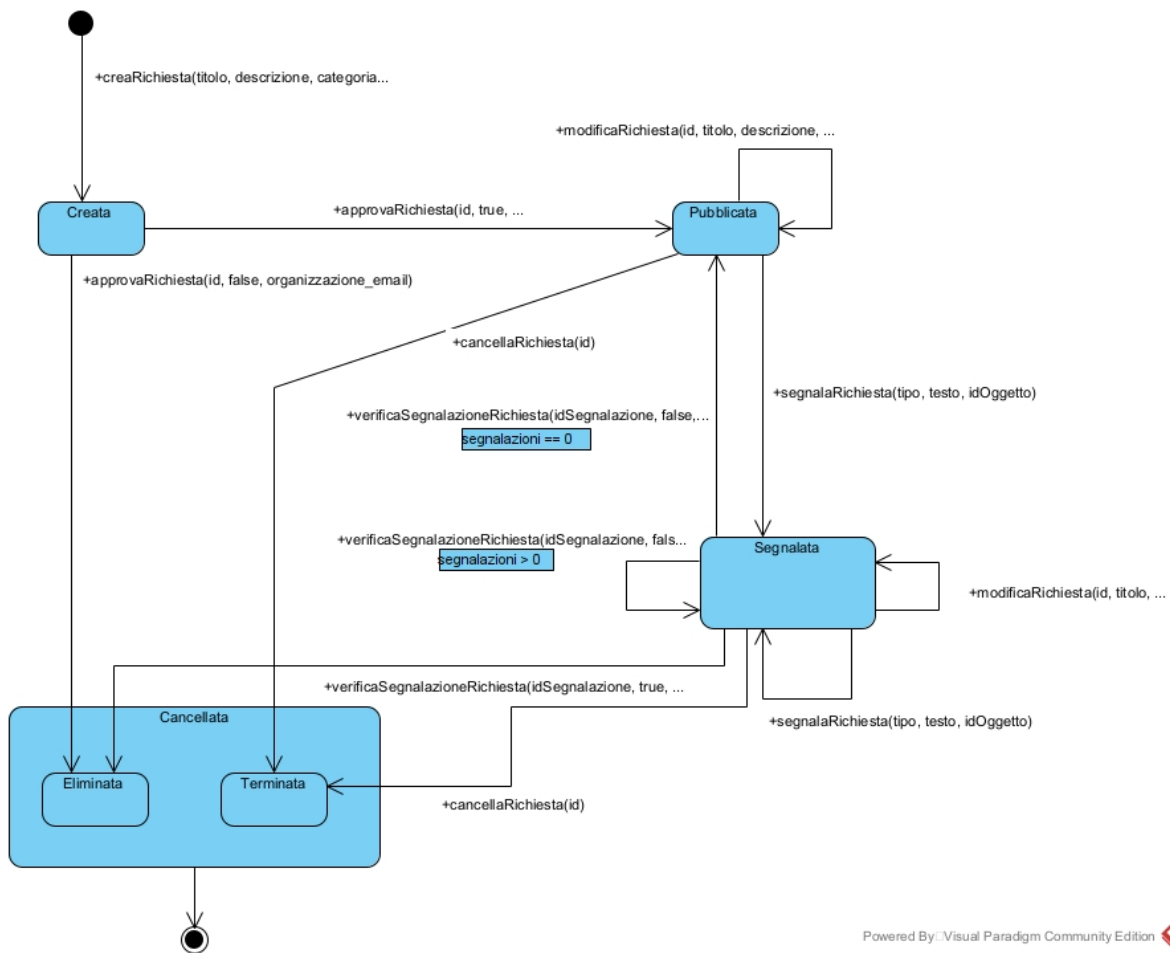


Figure 44: SM Richiesta

6.11 Testing e analisi

Oltre al testing delle funzionalità eseguito con Playwright sono state anche corrette delle reliability issues analizzate tramite SonarQube.

Sono stati risolte 20 issues raggruppate in queste categorie:

- 12 di *Maintainability*: tra questi ci sono problemi di variabili locali dichiarate e

non utilizzate, problemi di elementi cliccabili non accessibili a persone con disabilità.

- 8 di *Reliability*: tra questi ci sono problemi di espressioni booleane sempre vere, elementi interattivi che devono avere ruoli per essere selezionati.

È stato corretto il code smell *Duplicated code* estraendo il codice duplicato e creato un metodo apposito.

6.12 Pianificazione 4° Iterazione

Per la quarta iterazione è stato deciso di sviluppare le funzionalità relative alla gestione delle attività di volontariato, delle recensioni, delle notifiche e al supporto app con funzionalità offline.