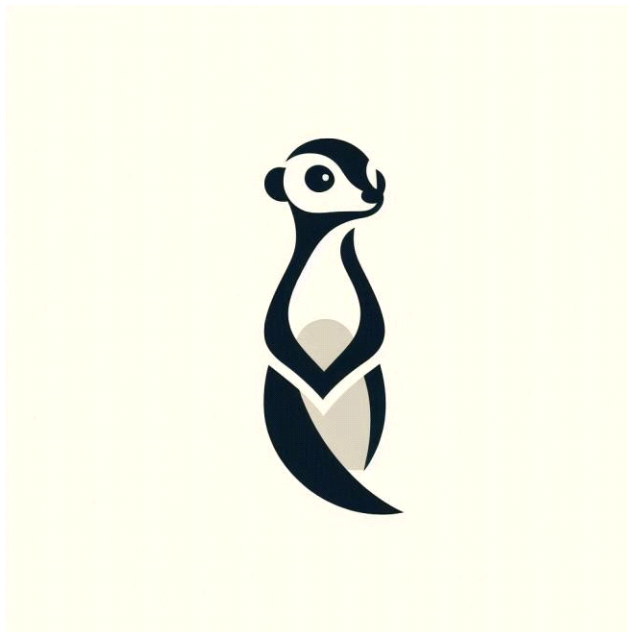


**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**MeerKat**  
**Requirements Analysis Document**  
**Versione 0.1.7**



Data: 10/11/2024



Coordinatore del progetto:

Nome	Matricola

Partecipanti:

Nome	Matricola
Stefano Nicolò Zito	0512117457
Gabriel Tabasco	0512116905
Giuseppe Ballacchino	0512116596
Francesco Giuseppe Trotta	0512116788

Scritto da:	Stefano Nicolò Zito
-------------	---------------------

Revision History

Data	Version	Descrizione	Autore
19/10/2024	0.0	Scrittura indice.	Stefano Nicolò Zito
22/10/2024	0.1	Rifacimento indice, aggiunta requisiti e scenari, modifiche scenari, scrittura casi d'uso 5,6,9	Stefano Nicolò Zito
23/10/2024	0.1.1	Aggiunti UC7 ed UC8 e sistemazione elenchi	Gabriel Tabasco
25/10/2024	0.1.2	Aggiunti UC1 ed UC4 con relativi diagrammi	Giuseppe Ballachino
25/10/2024	0.1.3	Part 1 Final Revision	Stefano Nicolò Zito, Giuseppe Ballachino, Gabriel Tabasco, Francesco Giuseppe Trotta
28/10/2024	0.1.4	Review	Ballacchino Giuseppe
08/11/2024	0.1.5	Indice aggiornato + sequence diagram 5,6,9	Stefano Nicolò Zito
09/11/2024	0.1.6	Object Model + Sequence Diagrams	Stefano Nicolò Zito, Gabriel Tabasco
10/11/2024	0.1.7	Mock Up + Navigation Path	Trotta Francesco Giuseppe



## Indice

1. Introduzione .....	8
1. Purpose of the System .....	8
2. References .....	8
2. Current System .....	8
3. Sistema Proposto .....	9
3. Overview .....	9
4. Functional Requirements .....	9
3. Non Functional Requirements .....	11
3.1. Usability .....	11
3.2. Reliability .....	11
3.3. Performance .....	11
3.4. Packaging .....	11
3.5. Legal .....	11
4. System Models .....	12
4.1. Scenari .....	12
4.1.1. Scenario registrazione utente .....	12
4.1.2. Scenario Creazione di un team .....	12
4.1.3. Scenario Creazione di Task .....	13
4.1.4. Scenario: Svolgimento di task .....	13
4.1.5. Scenario: messaggio di gruppo al team .....	14
4.1.6. Scenario: Messaggio individuale tra utenti .....	14

4.2. Use Case Model .....	15
4.2.1. Use Case Diagram .....	15
4.2.2. UC1: Registrazione .....	16
4.2.3. UC2: Create Team .....	16
4.2.4. UC3: Modifica info di un team .....	18
4.2.5. UC4: AddTask .....	18
4.2.6. UC5: CreateTaskList .....	20
4.2.7. UC6: MooveTaskInList .....	21
4.2.8. UC7: TaskDelivery .....	22
4.2.9. UC8: StartTimer .....	23
4.2.10. UC9: SendMessage .....	24
4.3. Object Model .....	25
4.3.1. Class diagram .....	25
4.4. Dynamic Model .....	26
4.4.1. SD1: Registrazione .....	26
4.4.2. SD2: Create Team .....	26
4.4.3. SD3: Modifica info di un team .....	26
4.4.4. SD4: AddTask .....	26
4.4.5. SD5: CreateTaskList .....	27
4.4.6. SD6: MooveTaskInList .....	27
4.4.7. SD7: TaskDelivery .....	27
4.4.8. SD8: StartTimer .....	28
4.4.9. SD9: SendMessage .....	28

4.4.1.SCD1: Task.....

28

4.4.2.SCD2 Timer.....

29

4.5. User Interface .....

30

## **1. Introduzione**

### ***1. Purpose of the System***

Lo scopo del sistema MeerKat è creare un unico luogo in cui le aziende possano organizzare progetti e gruppi di lavoro, consentendo di gestire e creare task, comunicare con gli altri membri del progetto in modo organizzato e ricevere dati in tempo reale sul progresso del team e sull'andamento del progetto.

### ***2. References***

- MeerKat Problem Statement

## **2. Current System**

Alla situazione corrente non esiste un sistema che permetta di eseguire funzioni di gestione del team, gestione dei task e comunicazione testuale tutto insieme, questo richiede agli utenti di svolgere ognuna di queste attività separatamente, richiedendo quindi agli utenti di registrarsi su molti sistemi software sia desktop che web based, risultato in un maggior uso di tempo dall'utente, soprattutto se una singola attività svolta va riportata su sistemi diversi non integrati fra di loro.



### 3. Sistema Proposto

#### 3. *Overview*

Questa piattaforma integra tutte le funzionalità necessarie per la gestione dei progetti e del lavoro di gruppo in un'unica applicazione, ottimizzata per semplificare il lavoro quotidiano. Con MeerKat, i manager possono facilmente creare gruppi di lavoro, assegnare membri a specifici team promuovendo così un ambiente di lavoro ben strutturato.

#### 4. *Functional Requirements*

Gestione dei progetti

- **FR1** (Priorità Alta) Un manager può creare un nuovo progetto specificando nome, descrizione ed eventuale data di scadenza.
- **FR2** (Priorità Alta) Il manager può modificare i dettagli del progetto in qualsiasi momento.
- **FR3** (Priorità Alta) Il manager può visualizzare un elenco di tutti i progetti attivi.

Gestione del team:

- **FR3** (Priorità Alta) Il manager può assegnare e rimuovere dipendenti a un progetto.
- **FR4** (Priorità Alta) Il sistema mostra un elenco aggiornato dei membri del team per ogni progetto.

Gestione dei task:

- **FR5** (Priorità Alta) Il manager può creare task all'interno di un progetto, specificando titolo, descrizione e scadenza.
- **FR6** (Priorità Alta) I task sono raccolti in liste.
- **FR7** (Priorità Alta) I membri del team possono aggiornare lo stato dei task, notificandone il completamento.
- **FR8** (Priorità Media) Il sistema fornisce una vista d'insieme dei task per progetto, mostrando il loro stato di avanzamento.

Monitoraggio dell'avanzamento:

- **FR9** (Priorità Alta) Il sistema genera report sull'avanzamento del progetto, mostrando la percentuale di completamento dei task.
- **FR10** (Priorità Alta) Il manager può visualizzare grafici e statistiche sull'andamento del progetto nel tempo.
- **FR11** (Priorità Alta) I membri del team possono visualizzare grafici e statistiche sull'andamento del progetto e le proprie prestazioni.

Time tracking:

- **FR12 (Priorità Media)** Il sistema è in grado di tracciare il tempo speso dai dipendenti a lavorare, in generale e su specifiche task.
- **FR14 (Priorità Alta)** I dati di time tracking sono aggregati e visualizzabili dal manager per ogni dipendente e per ogni progetto.

### **3. *Non Functional Requirements***

#### **3.1. Usability**

- **NFR1** (Priorità Alta) Il sistema implementa un robusto sistema di autenticazione e autorizzazione.
- **NFR2** (Priorità Alta) I dati dei progetti sono protetti e accessibili solo agli utenti autorizzati.
- **NFR3** (Priorità Alta) Il sistema ha un uptime del 99.9% durante le ore lavorative.
- **NFR4** (Priorità Alta) In caso di crash, il sistema può essere ripristinato entro 15 minuti senza perdita di dati.

#### **3.2. Reliability**

- **NFR5** (Priorità Alta) Il sistema risponde alle richieste degli utenti entro 2 secondi in condizioni normali di carico.
- **NFR6** (Priorità Media) Il sistema può gestire almeno 100 utenti concorrenti senza degradazione delle prestazioni.

#### **3.3. Performance**

- **NFR7** (Priorità Alta) Il sistema risponde alle richieste degli utenti entro 2 secondi

#### **3.4. Packaging**

- **NFR8** (Priorità media) L'applicazione viene distribuita con un installer .net standard e/o una versione portable

#### **3.5. Legal**

- **NFR9** (Priorità Alta) Il sistema deve rispettare le normative base sulla privacy
- **NFR10** (Priorità Media) Tutte le librerie utilizzate devono avere licenze compatibili con uso open source

## **4. *System Models***

### **4.1. Scenari**

#### **4.1.1.Scenario registrazione utente**

A Mario Rossi è stato comunicato dall'azienda per cui lavora che dovrà usare Meerkat per organizzare i suoi progetti. Dopo aver installato l'applicazione, Mario si trova nella schermata di creazione dell'account, dove gli viene chiesto di inserire il proprio nome e cognome, l'indirizzo e-mail, la data di nascita e una password, che deve essere ripetuta due volte per conferma. Dopo aver inserito i dati, l'operazione si completa e il sistema invia a Mario un'e-mail di conferma.

#### **4.1.2.Scenario Creazione di un team**

Mario Rossi lavora per un'azienda di produzione di sistemi software e deve creare il team per partire con un nuovo progetto di lavoro da remoto, per prima cosa esegue l'accesso all'applicazione Meerkat inserendo nella pagina d'accesso la sua mail aziendale e la sua password, dopo aver fatto ciò si ritrova nella schermata di scelta del team, dove può visualizzare da un elenco organizzato in una griglia dei team di cui fa parte, ognuno con un titolo e un'immagine, qui sceglie l'opzione per creare un nuovo team e inserisce al interno del form che si è aperto davanti a lui: un nome, una breve descrizione, l'immagine del team e la attuale data di consegna del progetto, inseriti questi dati gli si apre un nuovo form da cui ha la possibilità di invitare i suoi collaboratori nel team, attraverso una funzione di ricerca tramite e-mail o un link di invito associato al progetto, Mario sceglie la prima opzione, e digita le email dei suoi 4 collaboratori nella barra di ricerca, a questo punto Mario viene portato alla homepage del suo nuovo team, dove attraverso un menu al lato della pagina, potrà selezionare se accedere alla schermata delle statistiche di lavoro, quella per scrivere un messaggio di gruppo ai suoi collaboratori o a quella per l'elenco e l'organizzazione delle task.

Filippo Bianchi, è uno dei colleghi di Mario che sono stati scelti per collaborare con Francesco sul progetto software, riceve da Francesco un messaggio che gli chiede di confermare il suo ingresso al team Meerkat, anche Filippo apre l'applicazione di Meerkat e accede con e-mail aziendale e password, come Mario si ritrova nella schermata con l'elenco dei team di cui fa parte, da qui accede alla sezione per gli inviti a un nuovo team, vede la notifica del team creato da Mario e accetta l'invito, venendo anche lui portato nella schermata principale del team.

#### 4.1.3. Scenario Creazione di Task

Mario Rossi deve creare i primi task per il suo nuovo progetto. Esegue i passi iniziali già descritti nel punto 3.4.1.2; nella schermata dell'elenco dei progetti sceglie il progetto a cui desidera accedere, ritrovandosi così nella homepage. Da qui, tramite il menu, accede alla pagina di organizzazione dei task. In questa schermata può vedere una serie di blocchi che rappresentano le liste di task: ognuno di questi ha un titolo in cima (il nome della lista) e una serie di blocchetti più piccoli impilati uno sotto l'altro, che sono i task, ciascuno rappresentato da un nome e una data di scadenza. Al momento sono presenti quattro liste, ognuna delle quali è una to-do list per uno dei collaboratori di Mario.

Mario desidera creare un nuovo task per Filippo Bianchi, quindi seleziona l'opzione per aggiungere un task alla to-do list di Filippo. Si apre un form in cui può inserire il nome del task, una descrizione dettagliata e la data di scadenza. Dopo aver compilato i campi necessari, completa l'operazione e viene reindirizzato alla pagina di organizzazione dei task, dove può vedere che il nuovo task è stato aggiunto alla lista di Filippo.

Dopo aver creato il task in questo modo, Mario ci ripensa e decide che, essendo un lavoro a lungo termine, il task dovrebbe trovarsi in una lista differente. Seleziona quindi l'opzione per creare una nuova lista di task e si apre un form in cui può inserire il titolo della lista e selezionare un task preesistente da usare come primo task della nuova lista. Mario assegna un nome alla lista e sceglie il task appena creato; quindi salva la scelta e torna alla pagina di gestione dei task, dove può vedere che il task è stato spostato dalla to-do list di Filippo alla nuova lista.

#### 4.1.4. Scenario: Svolgimento di task

Filippo Bianchi, grafico del team di Mario Rossi, apre l'app di MeerKat seguendo i passi descritti nel punto 3.4.1.3 fino ad arrivare alla schermata di gestione dei task. Qui controlla la sua to-do list e nota l'aggiunta di un nuovo task. Clicca sul blocco corrispondente, aprendo una pagina di informazioni sul task, dove oltre al nome e alla data di scadenza, può leggere la descrizione scritta dal suo project manager. Inoltre, la pagina contiene un campo per caricare un file — il prodotto da consegnare al team — e uno per segnalare il completamento del task.

Dopo aver esaminato il compito assegnatogli, che consiste nel creare un'icona per l'applicazione in fase di sviluppo, Filippo esce dalla pagina di dettaglio del task e apre invece la sezione dedicata alle prestazioni. Qui può vedere in alto una barra che rappresenta il progresso del progetto rispetto alla scadenza; al momento, la barra non è completamente riempita, poiché ci sono task ancora non completati. Oltre alla barra di avanzamento, Filippo trova anche dei grafici che mostrano le sue performance rispetto ai task assegnati e il tempo lavorativo speso durante la settimana, insieme a un timer che indica il tempo lavorato oggi, attualmente impostato a 0 minuti.

Filippo avvia il timer cliccandoci sopra; appare un piccolo widget mobile sullo schermo che mostra il progresso del tempo e un pulsante di pausa. Riduce quindi MeerKat a icona e inizia a lavorare, tenendo d'occhio il timer.

Dopo qualche ora, Filippo ha completato il task. Ferma il timer, riapre MeerKat, accede alla pagina del task, carica il file dell'immagine creata e notifica il sistema del completamento dell'incarico.

#### **4.1.5.Scenario: messaggio di gruppo al team**

Mario Rossi deve scrivere un messaggio al suo team per informare i membri di un meeting programmato per la prossima settimana. Seguendo i passi descritti nel punto 3.4.1.2, accede a MeerKat e si sposta sulla schermata home del team interessato, quindi accede alla sezione dedicata alla chat. Qui, un menu laterale gli permette di scegliere se inviare un messaggio di gruppo o uno individuale agli altri membri. Mario seleziona l'opzione di gruppo, e nella pagina compare un elenco di topic di conversazione, simile a quello di un forum.

Mario sceglie di creare un nuovo topic e gli viene richiesto di inserire un nome. Lo intitola "Meeting 20-10-2024", e il topic viene immediatamente creato e aperto, portandolo in una schermata simile a quella di molte chat testuali. Qui, una barra in fondo alla pagina consente di scrivere il messaggio, mentre uno spazio sopra mostra i messaggi inviati in precedenza, attualmente vuoto.

Mario scrive il messaggio di avviso per il meeting e lo invia, vedendolo apparire nella sezione dei messaggi inviati, vicino al bordo destro della pagina.

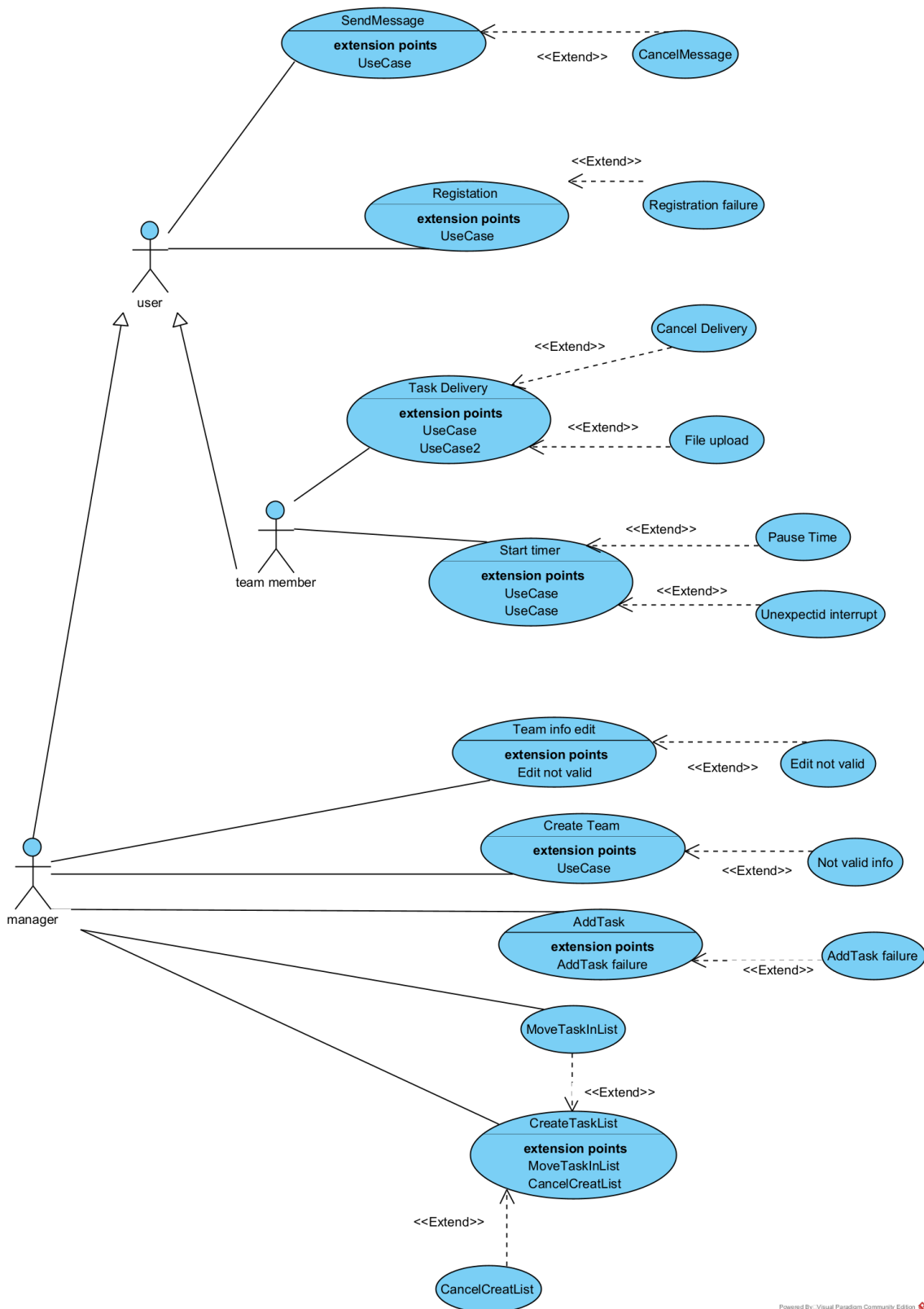
#### **4.1.6.Scenario: Messaggio individuale tra utenti**

Filippo Bianchi ha bisogno di chiarimenti da Mario Rossi su uno degli argomenti discussi nell'ultimo meeting. Segue quindi i passi descritti in 3.4.1.2 per accedere a MeerKat e quelli indicati in 3.4.1.5 per arrivare al menu delle chat. Qui seleziona l'opzione per inviare un messaggio individuale a Mario; sul lato destro dello schermo appare una chat simile a quella descritta in 3.4.1.5.

Filippo scrive il suo messaggio e preme invio. Per qualche istante osserva il messaggio inviato e nota che lo stato indicato sotto di esso passa da "inviato" a "visualizzato". Poco dopo, nello spazio dei messaggi compare la risposta di Mario, distinguibile per il colore e per la posizione a sinistra, in contrasto con il messaggio di Filippo posizionato vicino al bordo destro della schermata.

## 4.2. Use Case Model

### 4.2.1. Use Case Diagram



### 4.2.2.UC1: Registrazione

#### Attori coinvolti

- utente

#### Entry condition

L'utente si trova sulla home page con la schermata di registrazione

#### Flusso di eventi:

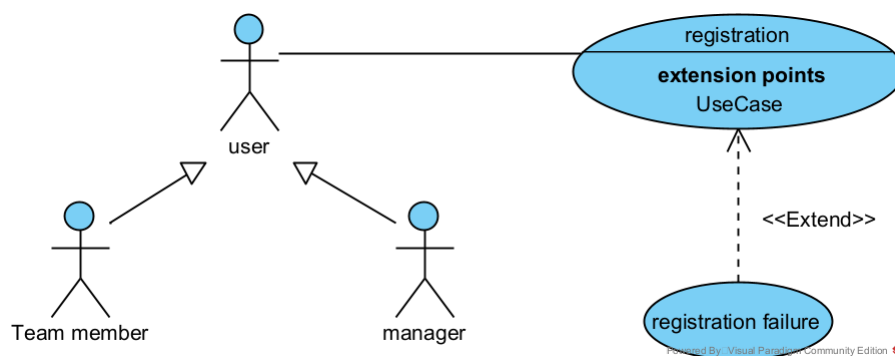
1. L'utente inserisce il proprio nome, cognome, la propria e-mail, data di nascita e una password, che va ripetuta due volte
2. L'utente invia i dati nel sistema
3. Il sistema controlla i dati, il controllo ha esito positivo
4. Il sistema invia una mail di conferma
5. L'utente accetta la conferma tramite il link inviato nella mail

#### Exit condition

L'utente è registrato e si trova nella home page

#### Flussi di eventi/ Eccezioni:

Se al punto 3 il sistema rileva credenziali non corrette, il sistema mostrerà un messaggio di errore "nome, cognome... non corretto" e ripresenterà la schermata di registrazione (UC1.1 Registrazione fallita).



### 4.2.3.UC2: Create Team

#### Attori coinvolti

- Manager



### Entry Condition

Il Manager è nella schermata di visualizzazione dei gruppi.

### Flusso di eventi:

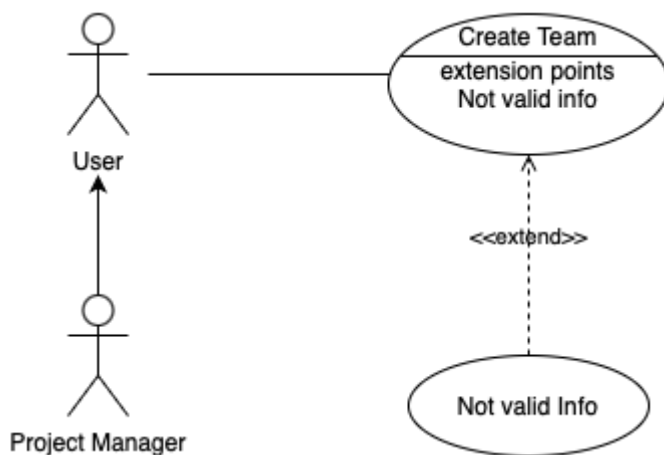
1. Il Manager seleziona l'opzione di creazione di un nuovo gruppo.
2. Il Manager inserisce i dettagli del gruppo, il nome, una descrizione, la data di consegna del progetto e un'immagine
  - 2.1. Il Manager conferma la creazione del gruppo.
  - 2.2. Il Manager salva e crea il nuovo gruppo.
  - 2.3. Il sistema conferma che il gruppo è stato creato con successo.

### Exit Condition:

Un nuovo gruppo è stato creato e registrato nel sistema.

### Flussi alternativi / Eccezioni:

**UC2.1: Not valid info**, Se al punto 2.1, il Manager inserisce informazioni già esistenti o non completa i campi obbligatori, il sistema mostrerà un messaggio di errore e chiederà di inserire delle informazioni valide.



#### 4.2.4.UC3: Modifica info di un team

**Attore:**

- Manager

**Entry Condition:** Il Manager è nella schermata di visualizzazione dei gruppi.

**Flusso di eventi:**

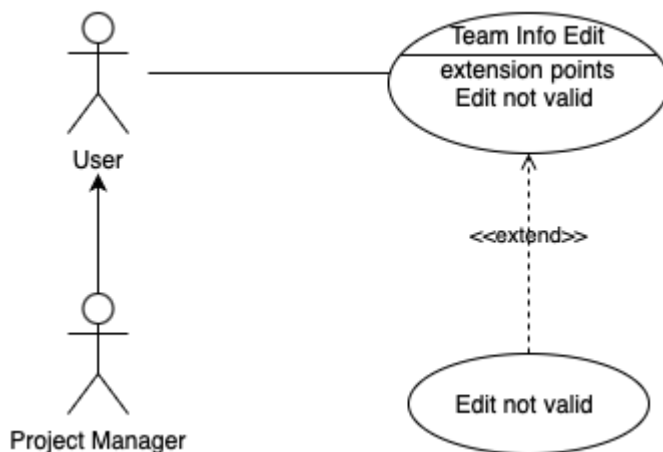
1. Il Manager seleziona il gruppo da modificare.
2. Il sistema visualizza le attuali informazioni del gruppo, nome, descrizione, data di consegna del progetto e l'immagine.
1. Il Manager modifica le informazioni del gruppo desiderate.
2. Il Manager conferma le modifiche.
3. Il sistema salva le modifiche e aggiorna le informazioni del gruppo.
4. Il sistema conferma che le informazioni sono state aggiornate con successo.

**Exit Condition:**

Le informazioni del gruppo sono state aggiornate nel sistema.

**Flussi alternativi / Eccezioni:**

**UC3.1: Edit not valid**, Se al punto 2.1 le modifiche non sono valide, il sistema notificherà l'errore al Manager.



#### 4.2.5.UC4: AddTask

### Attori coinvolti

- Manager.

### Entry condition

Il manager accede alla pagina di organizzazione dei task

### Flusso di eventi:

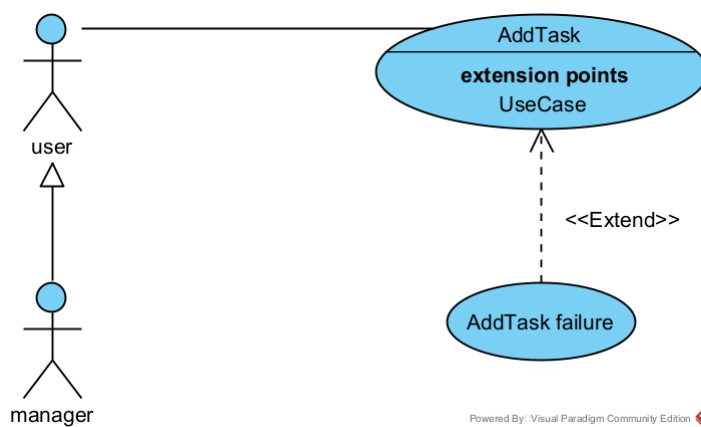
1. sceglie l'opzione di aggiungere un task alla lista
2. Il sistema
3. Il manager inserisce nome, descrizione e data di scadenza del task.
4. Il manager conferma l'inserimento
5. Il sistema controlla il nome e la data di scadenza, il controllo ha esito positivo
6. Il sistema salva i dati

### Exit condition

Il task è salvato e il manager viene riportato alla pagina di organizzazione dove può vedere il nuovo task aggiunto

### Flussi Alternativi / Eccezioni

Se al punto 3 il nome è già stato inserito o la data di scadenza è incoerente con le precedenti manda un messaggio di errore (UC4.1 AddTask fallita)



#### 4.2.6.UC5: CreateTaskList

##### Attori coinvolti

- Manager

##### entry condition

questo use case inizia quando il Manager sta guardando la schermata di organizzazione delle Task.

##### Flusso di eventi

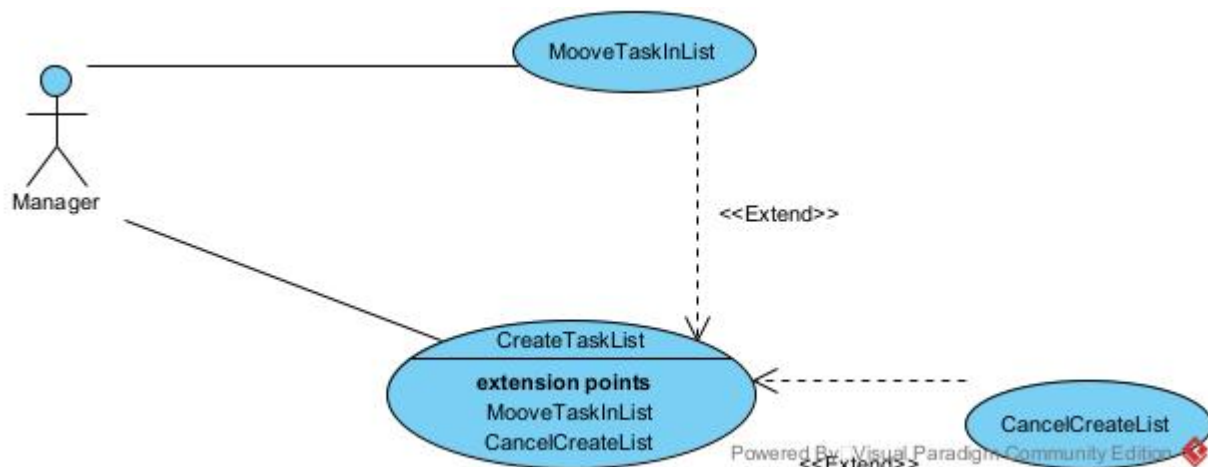
- 1) IL Manager selezione la funzionalità di creazione di una nuova lista
- 2) il sistema porta il Manager in una pagina dove può inserire i dettagli della lista
- 3) il Manager inserisce il nome della lista di Task
- 4) il Manager sceglie una delle Task già esistenti da spostare nella lista (Opzionale)
- 5) Il Manager salva i dati inseriti
- 6) Il Sistema crea la nuova Lista di Task con i dati assegnatogli e la rende visibile nella pagina gestione Task
- 7) Il Manager esce dalla pagina di creazione Lista Task

##### Exit condition

questo caso d'uso termina quando la lista e creata con i dati desiderati e il Manager torno alla schermata di gestione delle Task.

##### Flussi Alternativi/ Eccezioni

- Durante il punto 5) il Manager può decidere di uscire, senza salvare i dati, in questo caso il punto 6) viene saltato e si va direttamente al 7) (UC5.1 CancelCreateList)
- Se il punto 4 è avvenuto, durante il punto 6) viene eseguito UC6.



#### 4.2.7.UC6: MooveTaskInList

##### Attori Coinvolti

- Manager

##### entry condition

questo use case inizia quando il Manager sta guardando la schermata di organizzazione delle Task.

##### Flusso di eventi

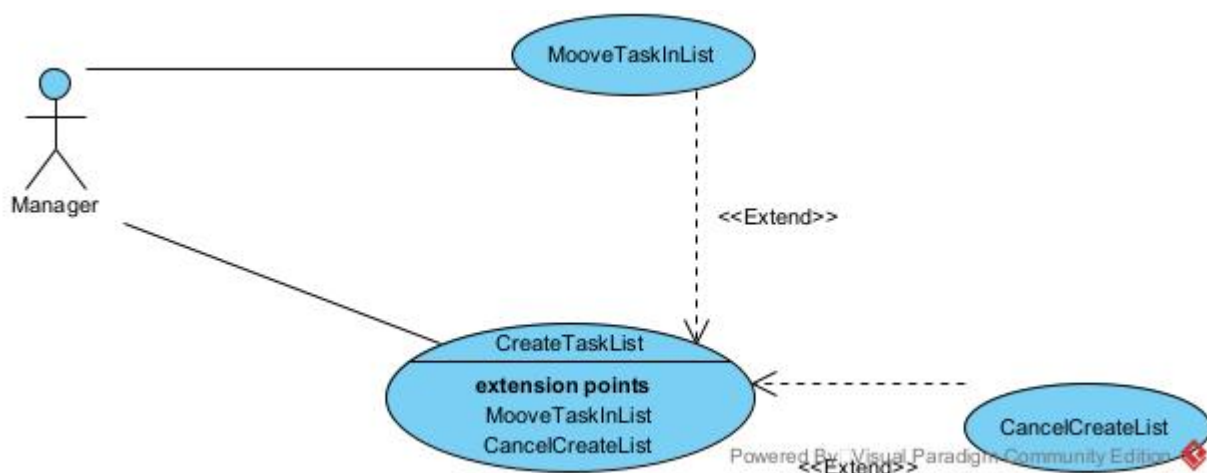
- 1) Il Manager seleziona la Task che desidera inserire in una lista e la posiziona nella lista adeguata dall'interfaccia grafica
- 2) il sistema inserisce effettivamente il task nella Lista desiderata e la elimina dalla lista che la conteneva in precedenza

##### Exit condition

questo caso d'uso termina quando la Task viene inserita nella Lista desiderata dal Manager.

##### Flussi Alternativi/ Eccezioni

- Quando questo Use Case e seguito come parte di UC 5, il punto 4) di UC 5 prende il posto del punto 1) di questo Use Case.



### 4.2.8.UC7: TaskDelivery

#### Attori Coinvolti

- Team Member

#### Entry condition

Il team member è autenticato e si trova nella pagina di dettaglio del task

#### Flusso di eventi

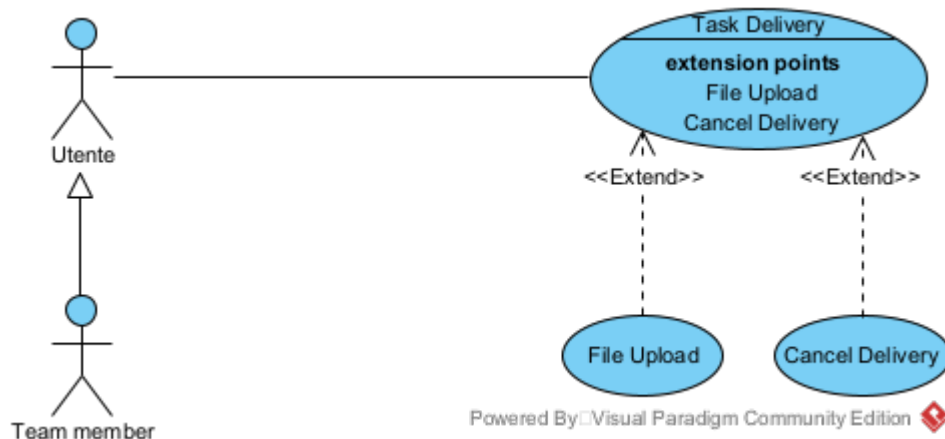
- 1) L'utente visualizza i dettagli del task
- 2) Caricamento eventuale di file allegati (opzionale)
- 3) L'utente seleziona l'opzione per segnalare il completamento del task
- 4) Il sistema chiede conferma del completamento
- 5) L'utente conferma il completamento
- 6) Il sistema aggiorna lo stato del task e le statistiche di progetto

#### Exit condition

Il task è stato marcato come completato.

#### Flussi Alternativi/ Eccezioni

- Se al punto 2) si sceglie di caricare dei file allegati viene eseguito UC 7.1 (UC 7.1 File Upload)
- Se al punto 5) l'utente annulla la conferma, il sistema mantiene lo stato corrente del task (UC 7.2 CancelDelivery)



#### 4.2.9.UC8: StartTimer

##### Attori Coinvolti

- Team Member

##### Entry condition

Il team member è autenticato e si trova nella pagina delle prestazioni

##### Flusso di eventi

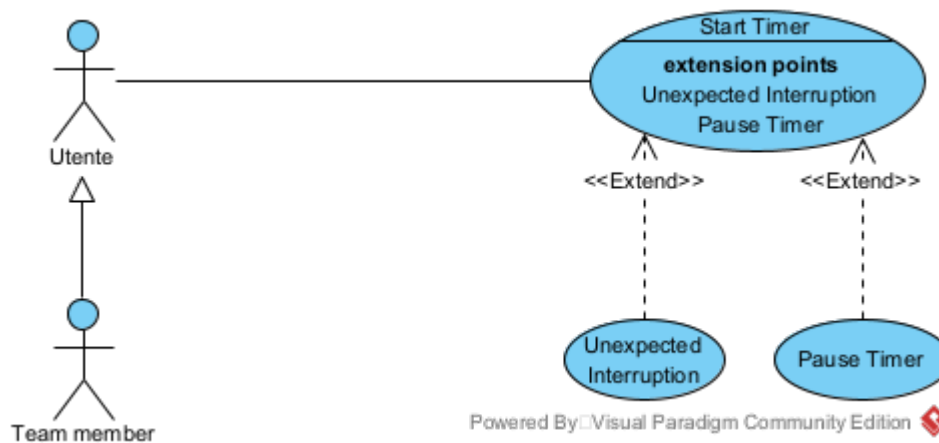
- 1) L'utente seleziona il pulsante di avvio del timer nella pagina delle prestazioni
- 2) Il sistema avvia il conteggio del tempo
- 3) Il sistema mostra il tempo trascorso aggiornandolo in tempo reale
- 4) L'utente può selezionare il pulsante di pausa quando necessario
- 5) Il sistema interrompe il conteggio del tempo mantenendo il totale

##### Exit condition

Il timer è attivo e sta tracciando il tempo di lavoro.

##### Flussi Alternativi/ Eccezioni

- Se al punto 4 l'utente mette in pausa, il sistema sospende il conteggio fino alla ripresa (UC 8.1 PauseTimer)
- Se durante il tracking si verifica una disconnessione, il sistema salva il tempo registrato fino a quel momento (UC 8.2 UnexpectedInterruption)



#### 4.2.10.UC9: SendMessage

##### Attori Coinvolti

- Utente (Manager o Team member)

entry condition

questo use case inizia quando l'Utente sta scorrendo l'elenco di chat room del progetto.

##### Flusso di eventi

- 1) L'utente sceglie la chat room, di gruppo o individuale, in cui deve inviare un messaggio
- 2) il sistema apre la schermata relativa alla chat e rende visibile i messaggi inviati in precedenza dagli Utenti
- 3) L'utente inserisce il corpo del suo messaggio e preme il pulsante di invio
- 4) il sistema invia il messaggio agli altri membri della chat e li notifica del suo arrivo

##### Exit condition

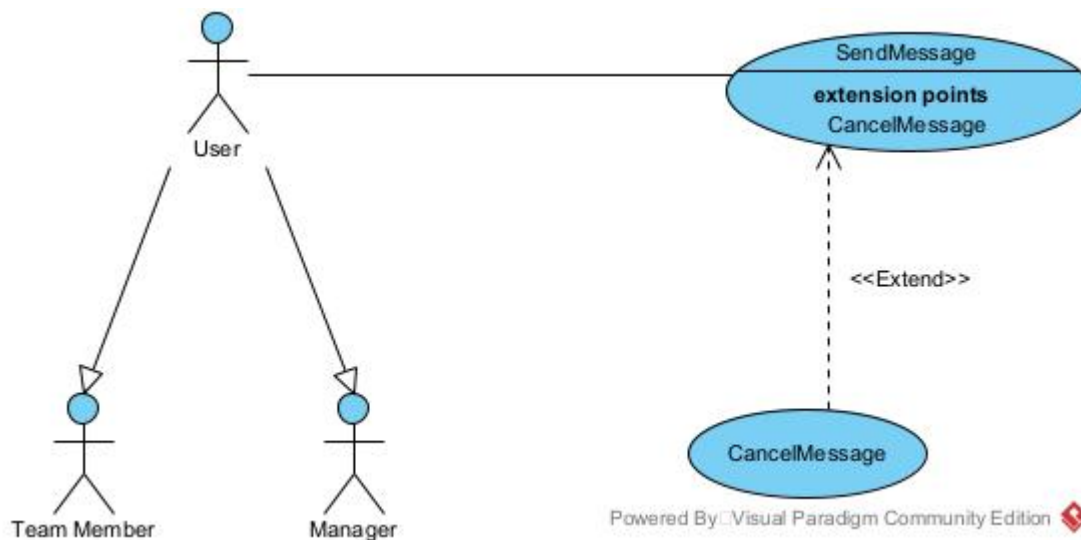
questo caso d'uso termina quando il messaggio è stato inviato con successo dal sistema.

##### Flussi Alternativi/ Eccezioni

- Al punto 3) l'utente può smettere di scrivere e cambiare pagina senza premere il pulsante di invio, in quel caso il testo del messaggio viene eliminato e il caso d'uso termina. (UC9.1 CancelMessage)

##### Requisiti Speciali

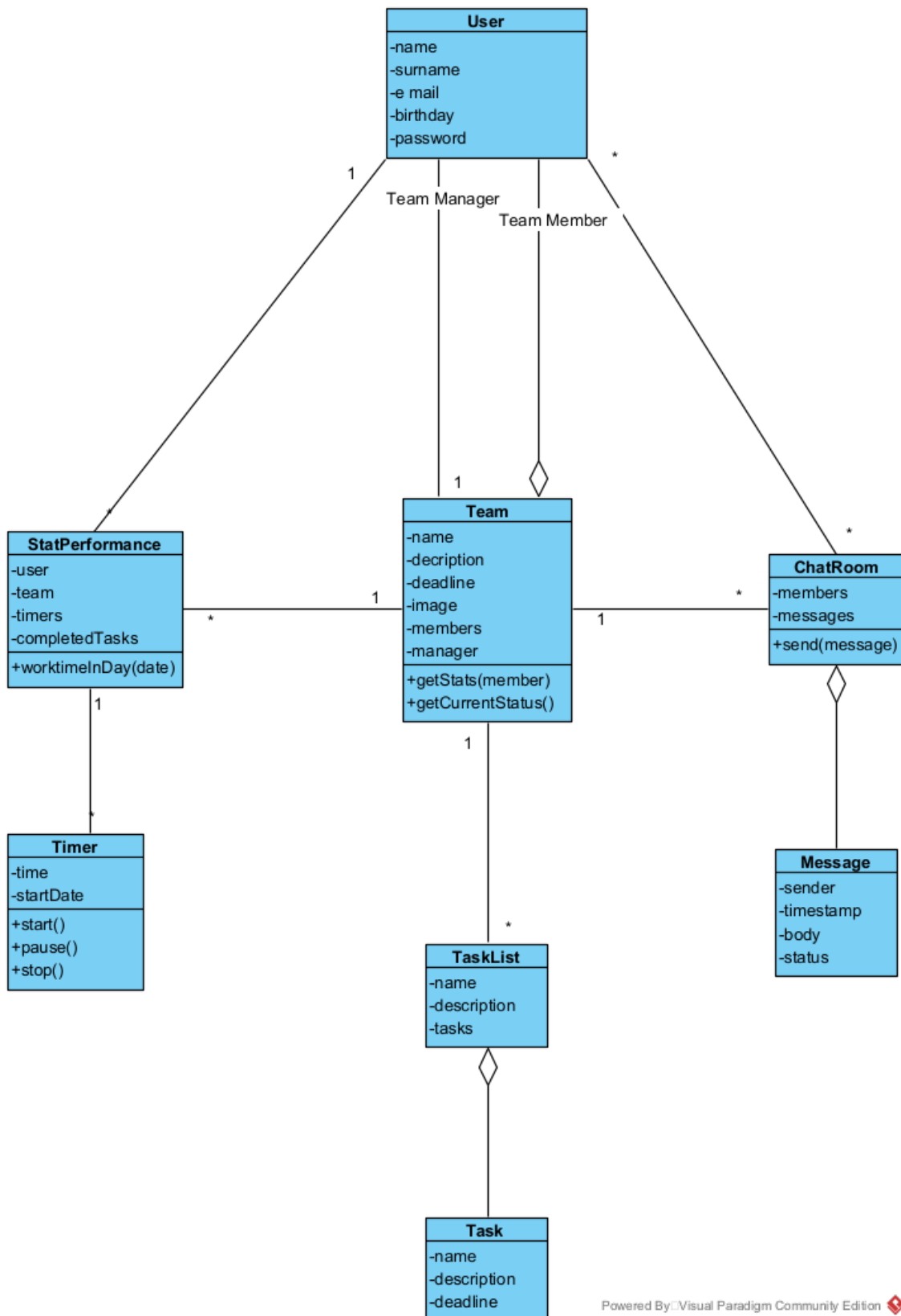
- Il sistema deve inviare il Messaggio con successo in un tempo di 10 secondi dal comando dell'utente





### 4.3. Object Model

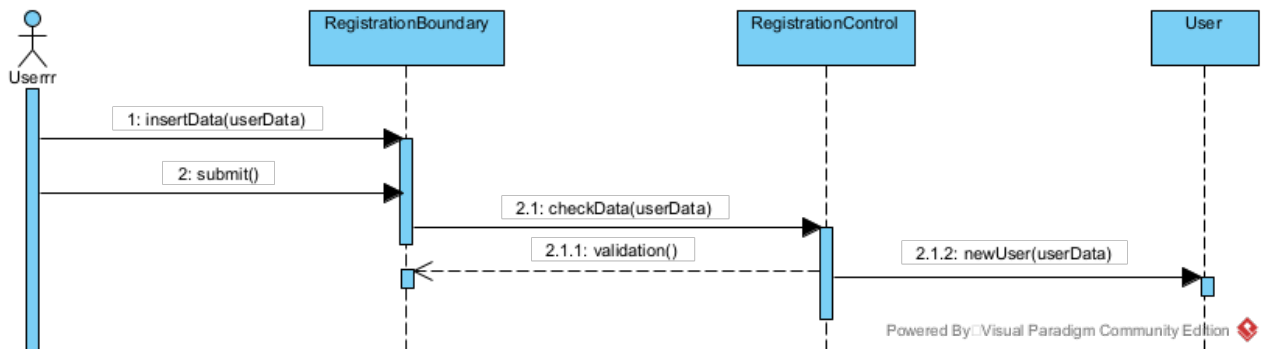
#### 4.3.1. Class diagram



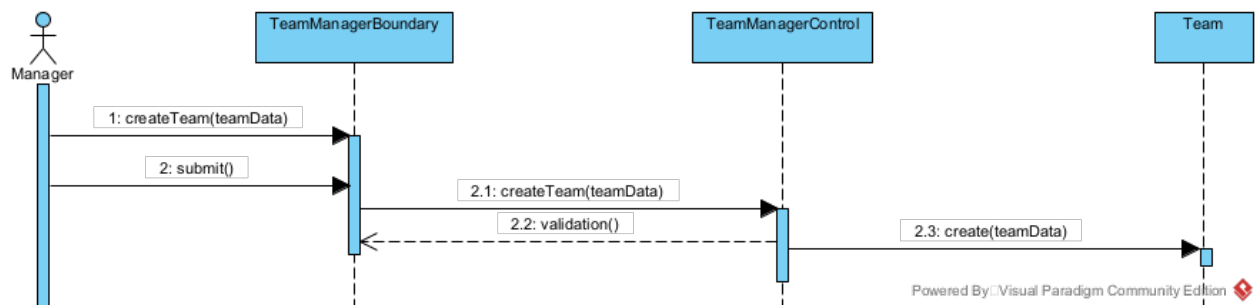
Powered By: Visual Paradigm Community Edition

## 4.4. Dynamic Model

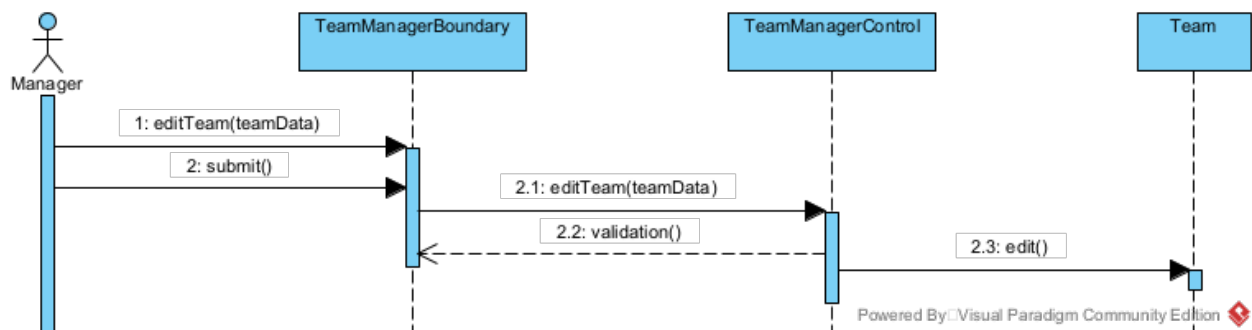
### 4.4.1.SD1: Registrazione



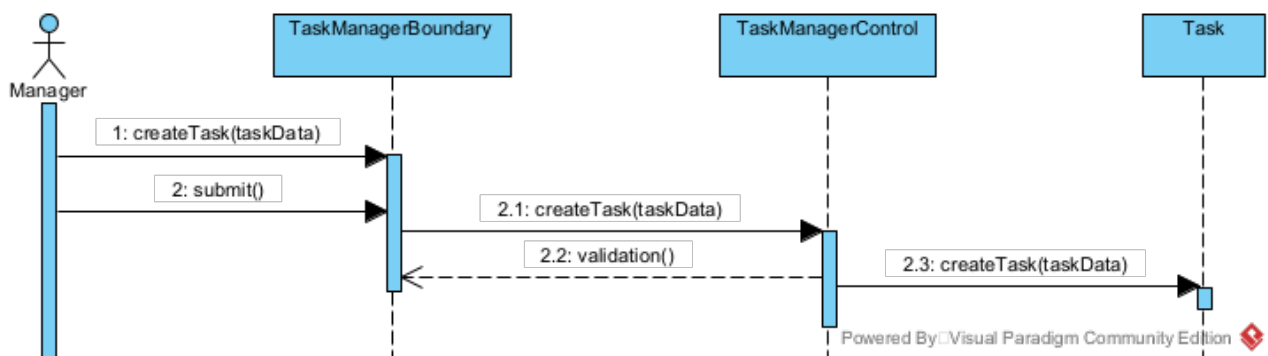
### 4.4.2.SD2: Create Team

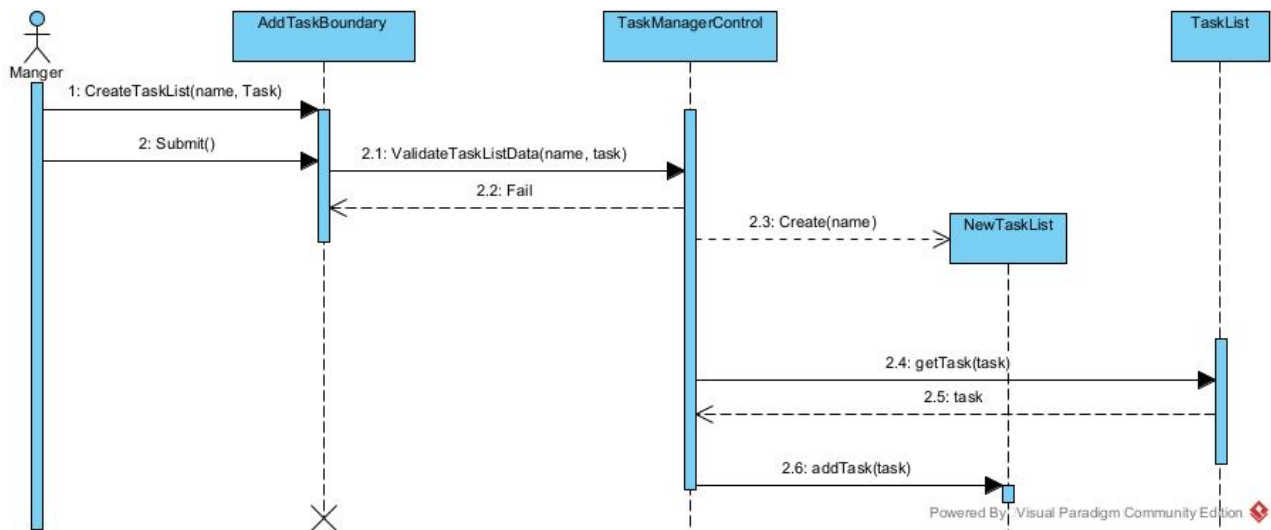
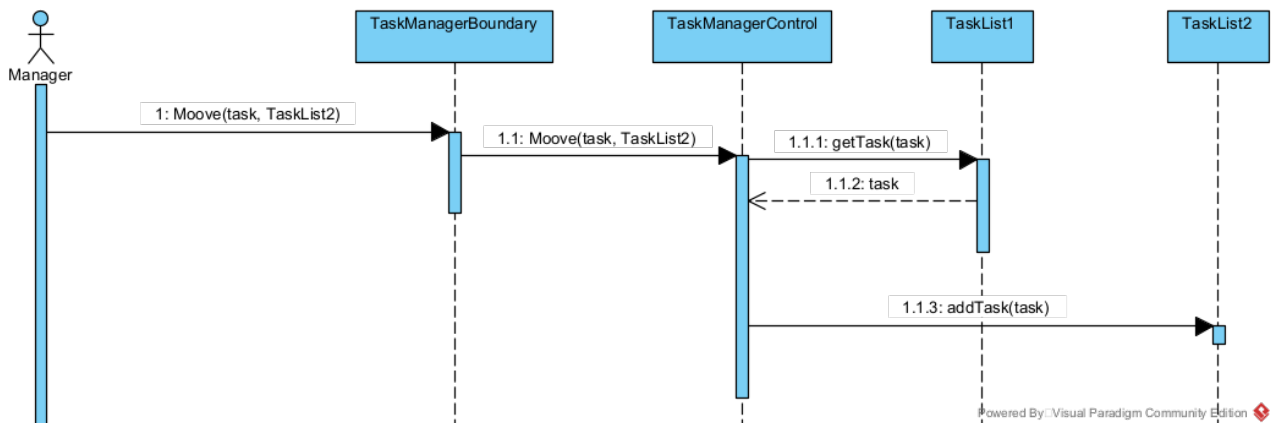
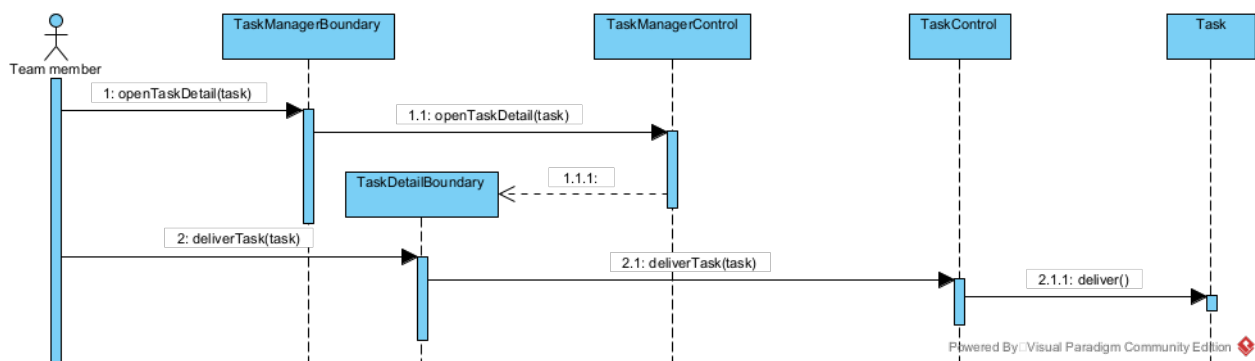


### 4.4.3.SD3: Modifica info di un team

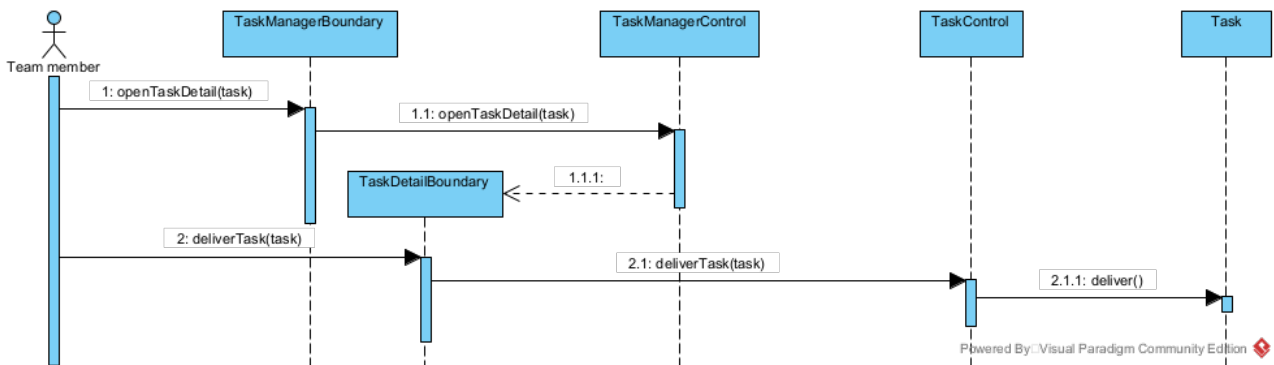


### 4.4.4.SD4: AddTask

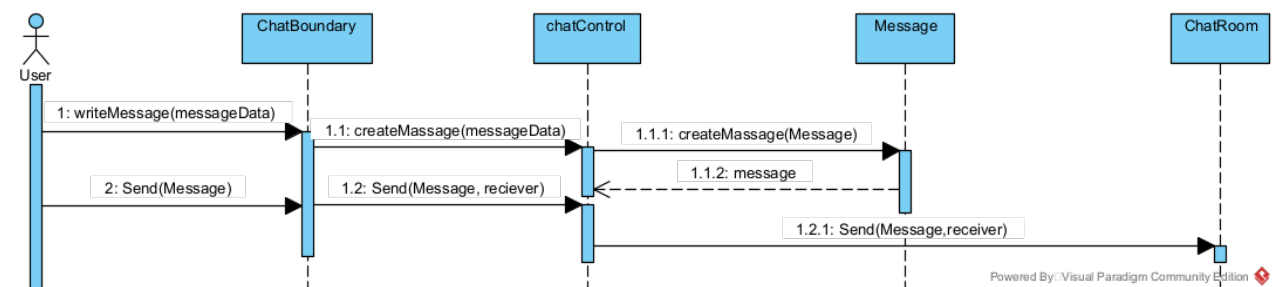


**4.4.5.SD5: CreateTaskList****4.4.6.SD6: MooveTaskInList****4.4.7.SD7: TaskDelivery**

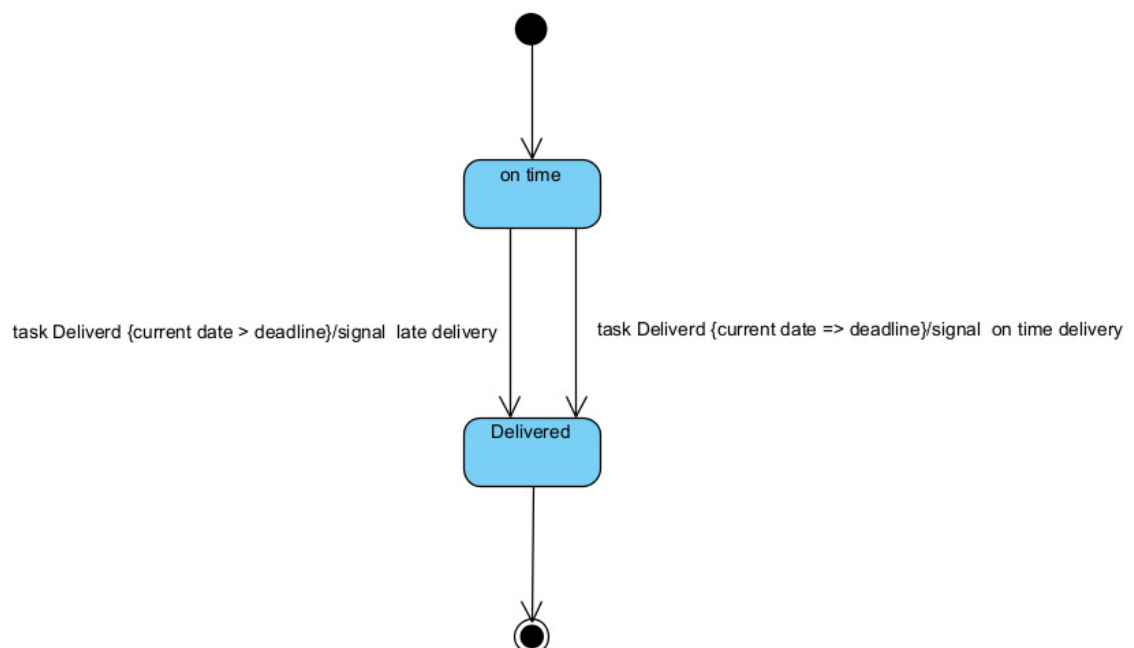
#### 4.4.8.SD8: StartTimer



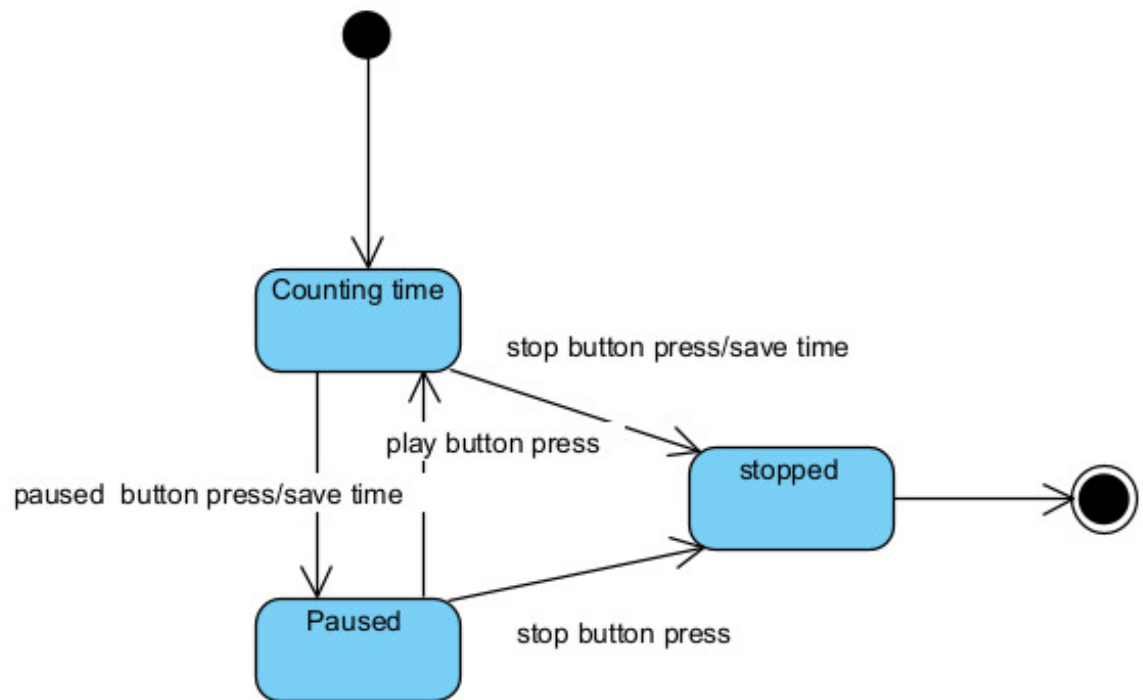
#### 4.4.9.SD9: SendMessage



#### 4.4.1.SCD1: Task

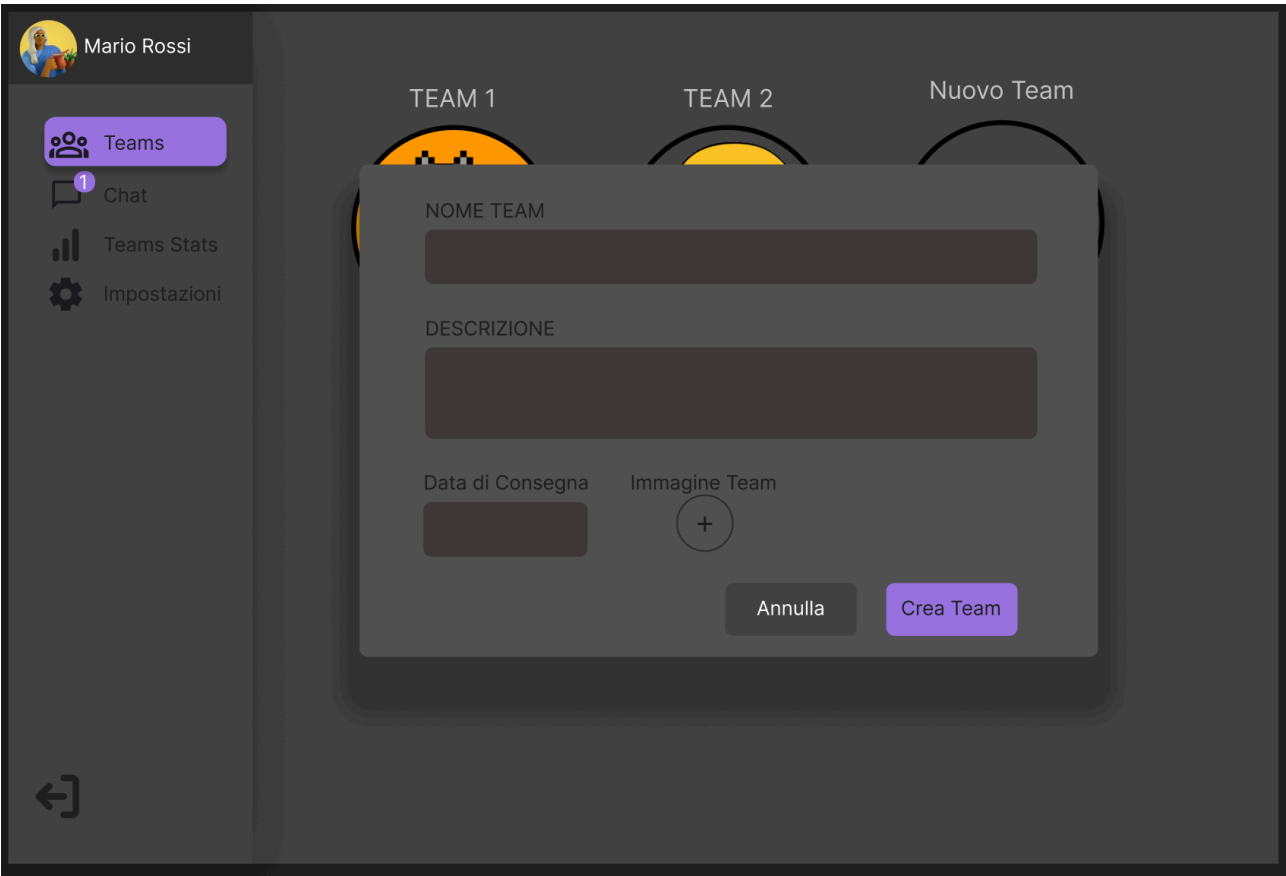


#### 4.4.2.SCD2 Timer

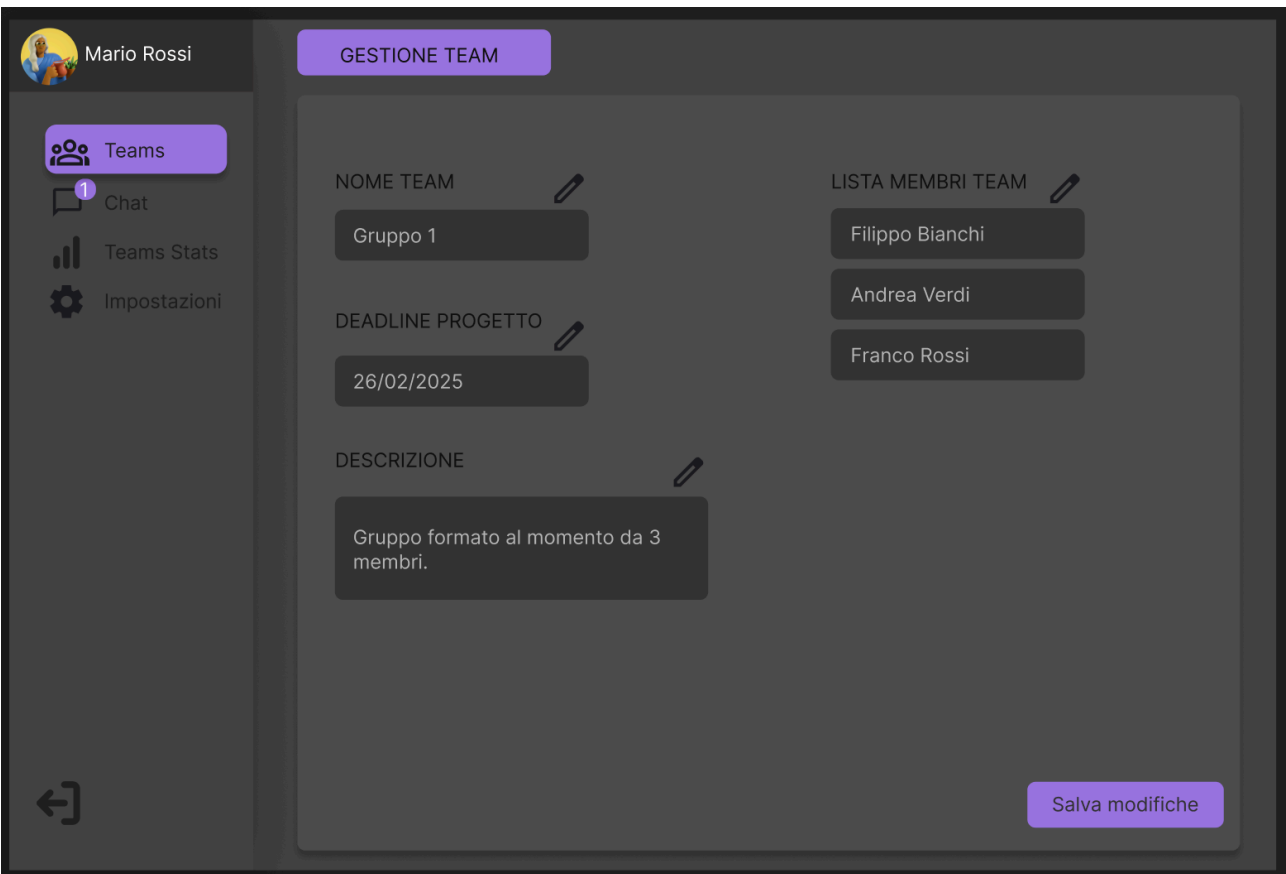


4.5. User Interface

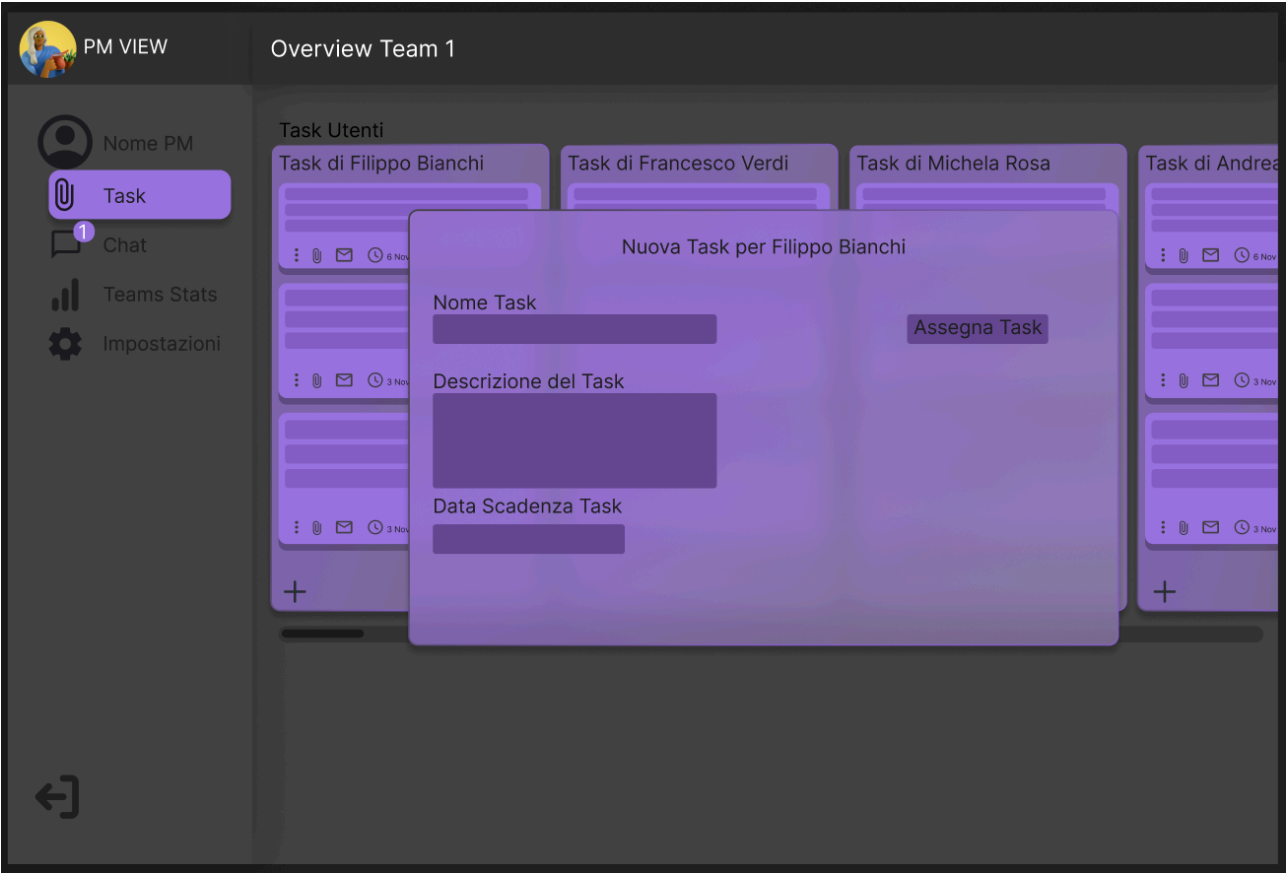
Creazione di un Team



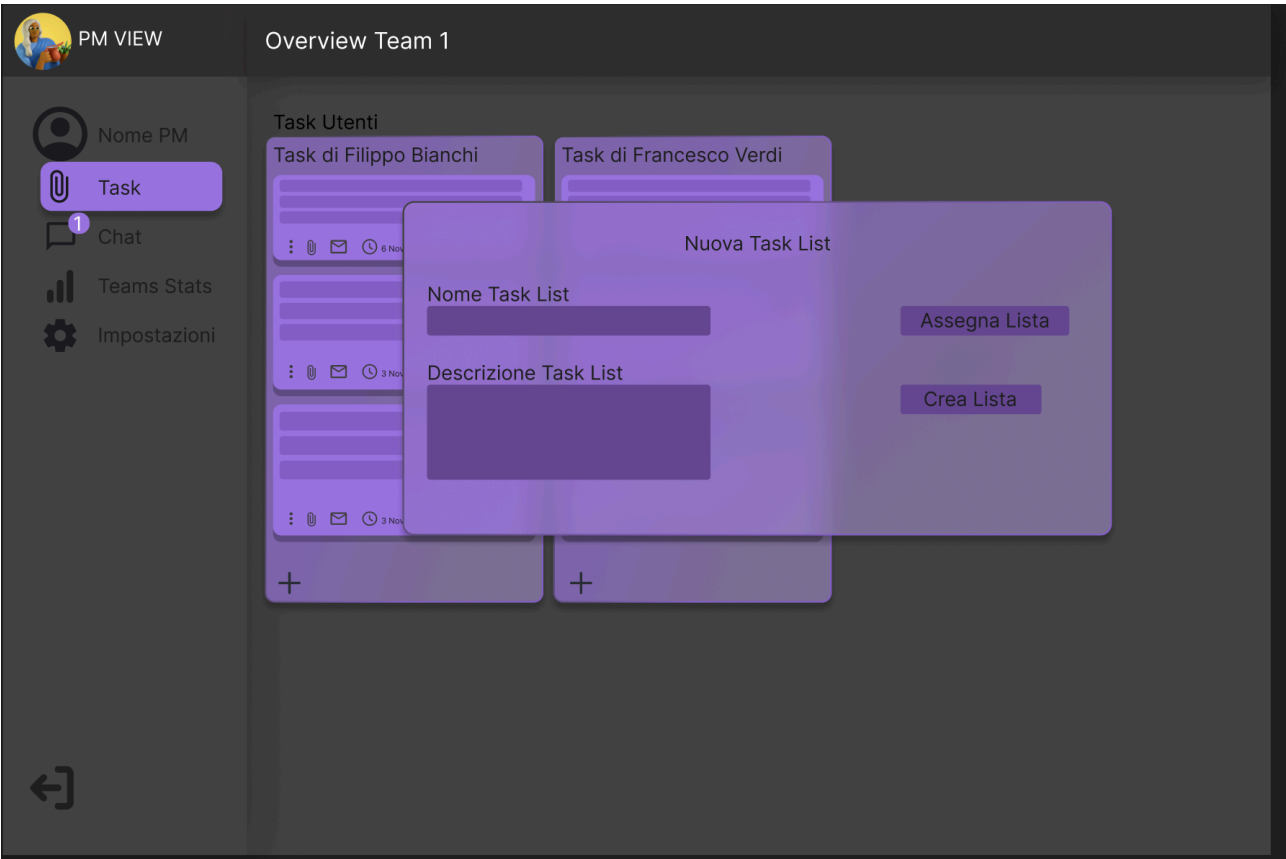
Modifica Info Team



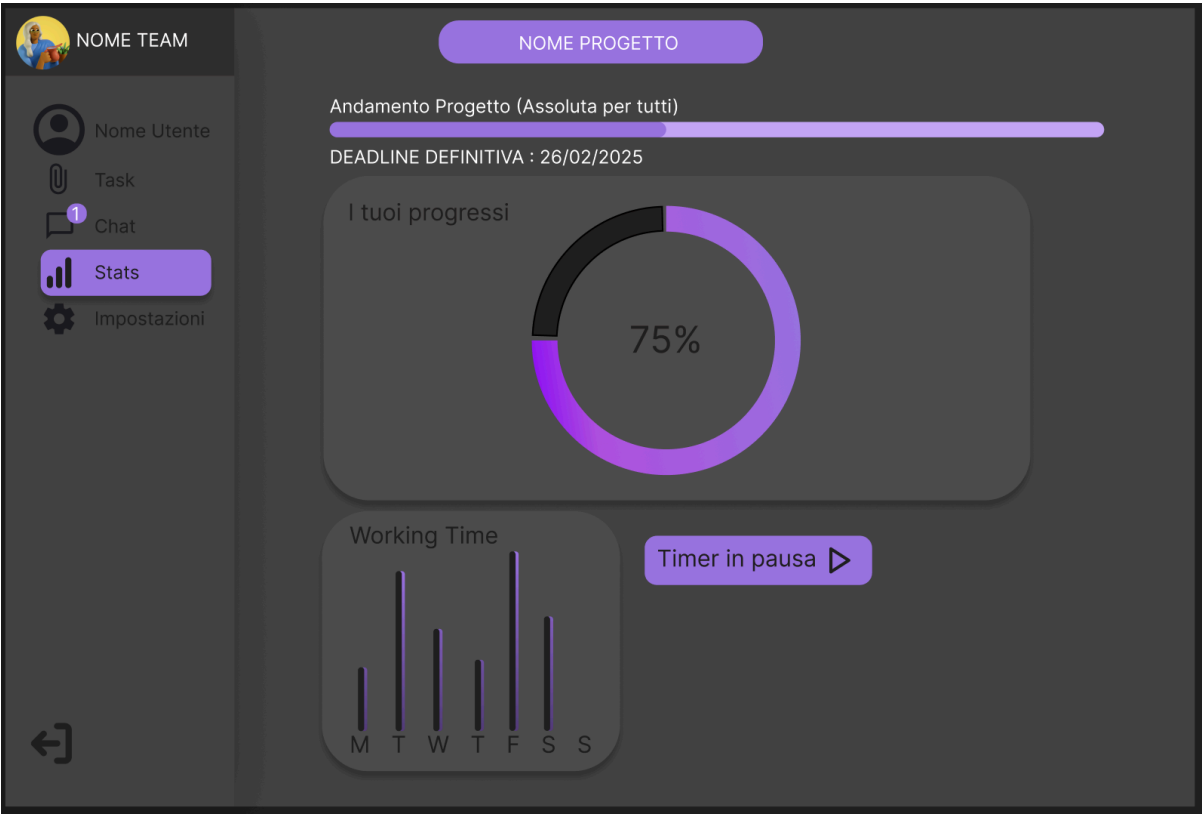
Add Task



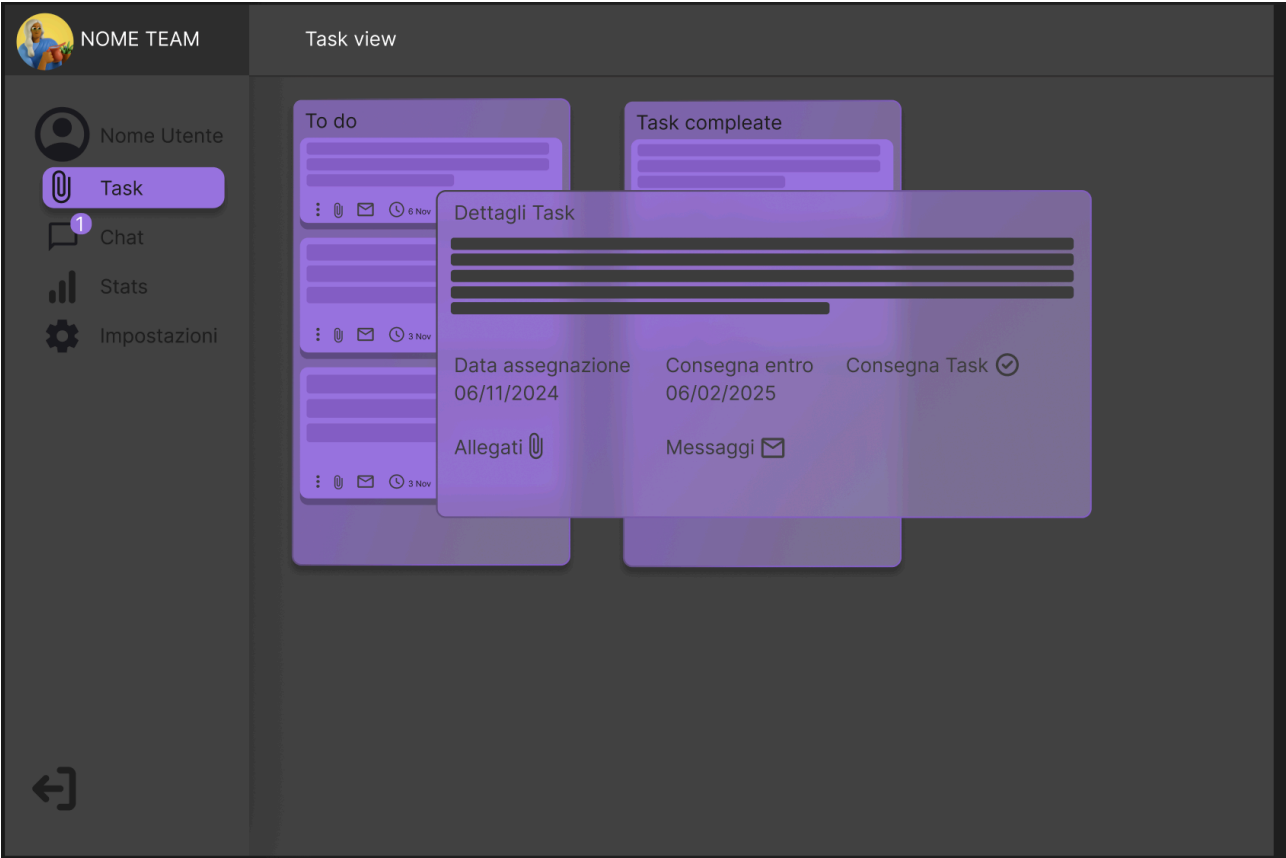
Creazione di una Lista di Task



Utilizzo Timer

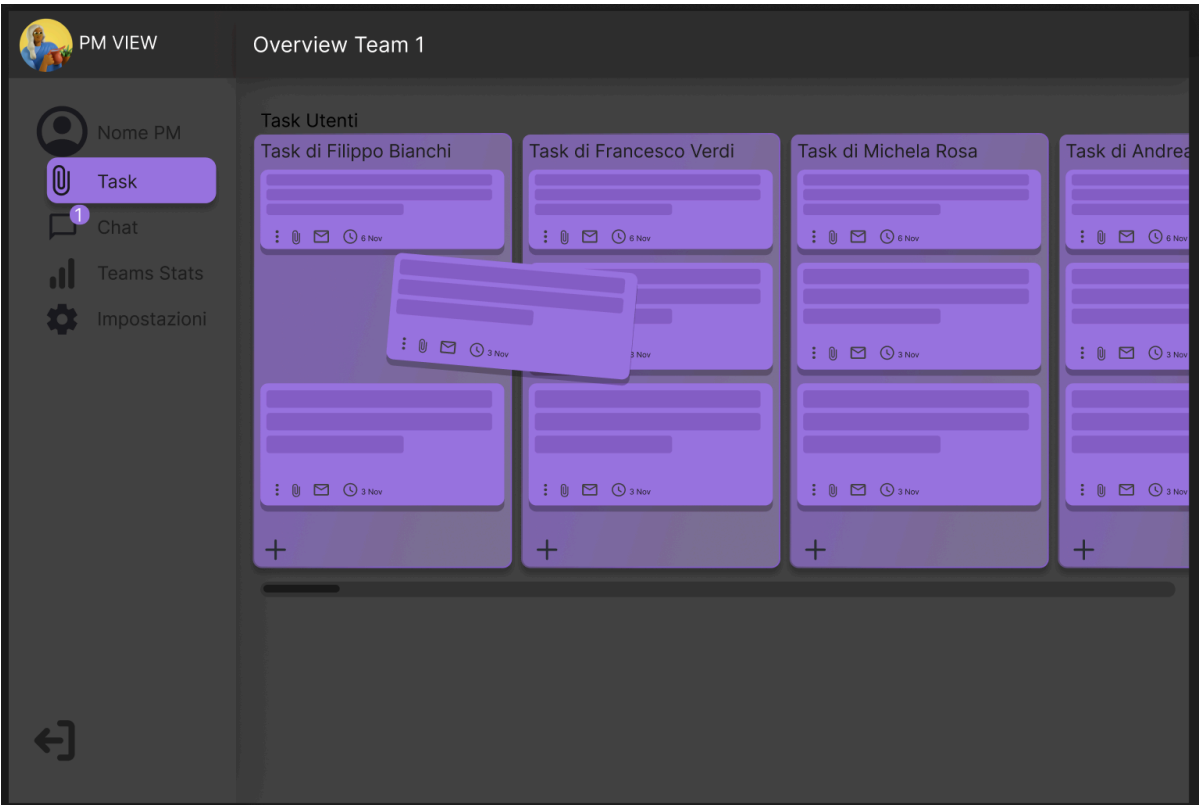


Task Delivery

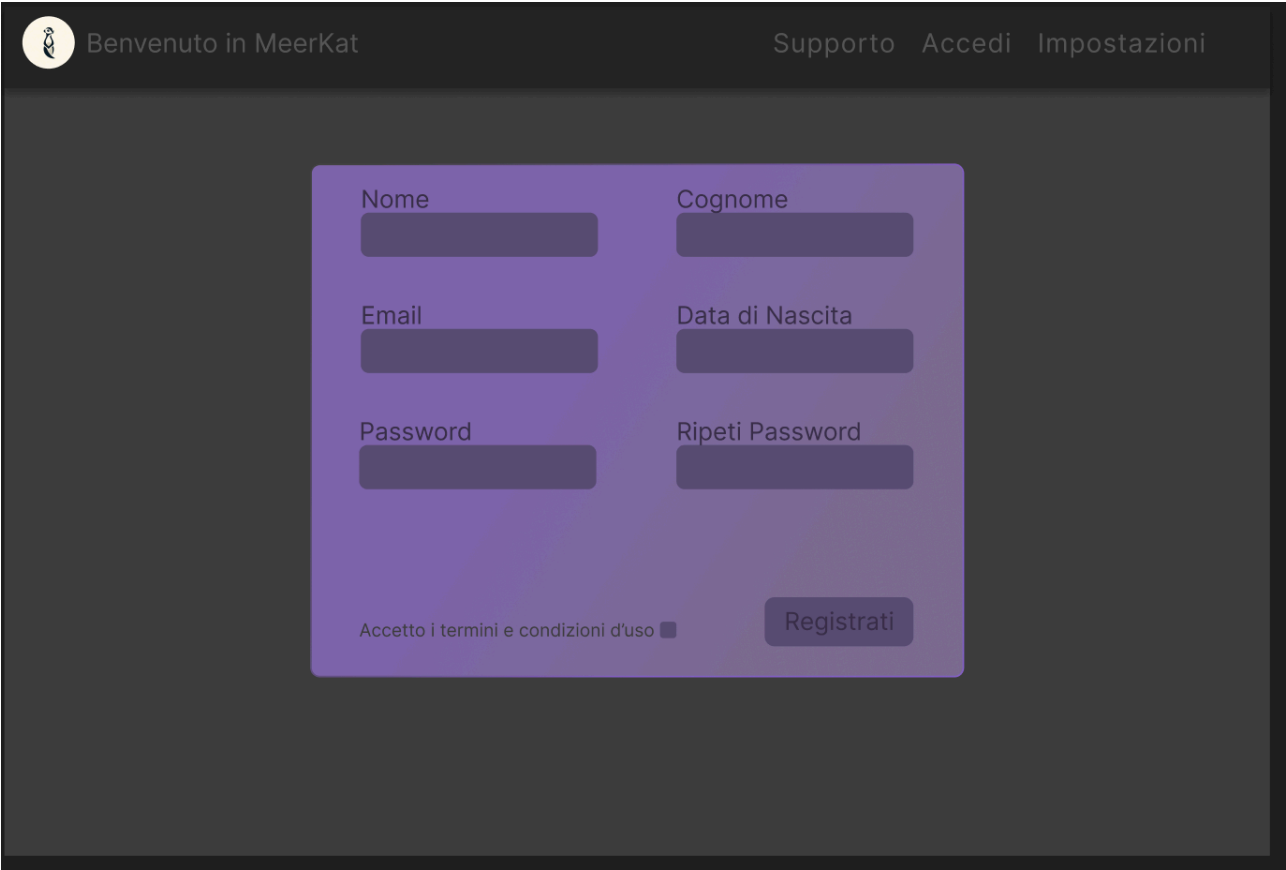




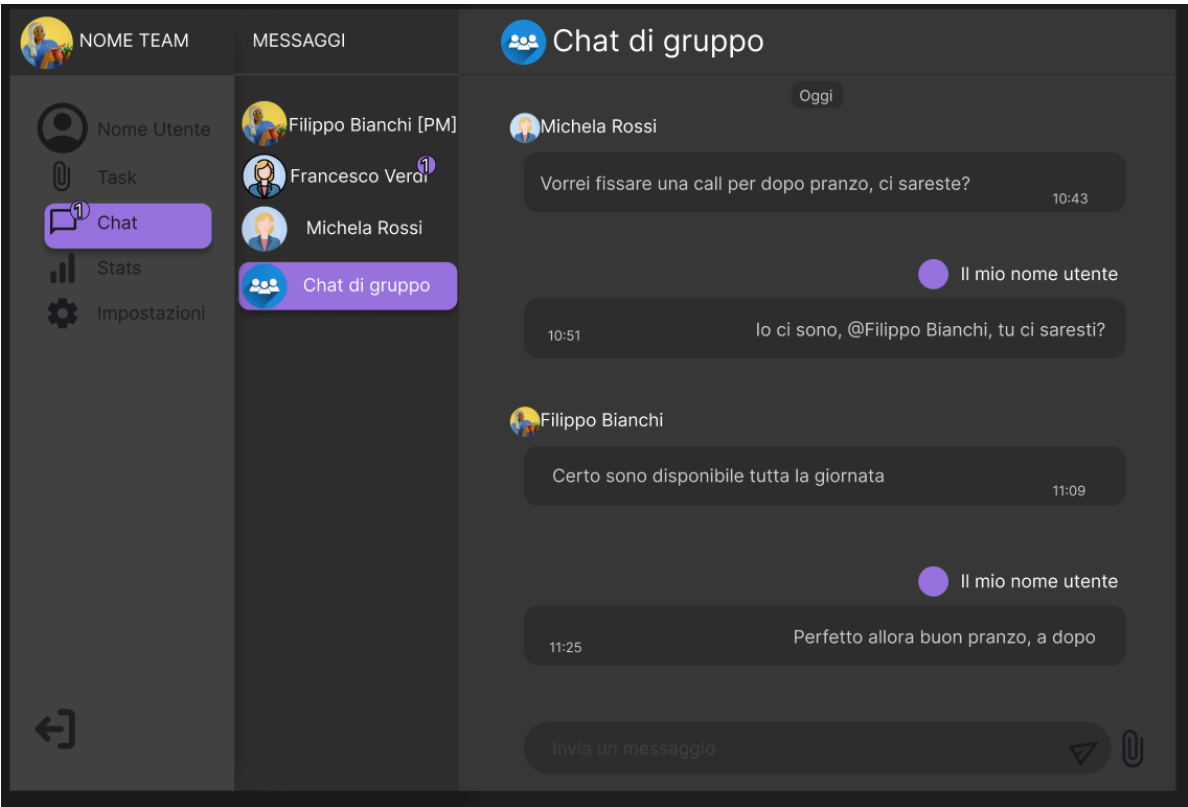
Spostamento di un Task



Registrazione Utente



Invio messaggio



Navigation Path

