



Corso di Laurea Magistrale in Ingegneria Informatica

Progetto per il corso di Gestione dei Sistemi ICT

Work Permissions

Docente:

Chiar.ma Prof.ssa Bonfanti Silvia

Componenti del gruppo:

Filippo Bordogna

Andrea Cassina

Anno Accademico 2023-2024

Elenco delle figure

1.1	Architettura Odoo	3
2.1	Diagramma MoSCoW must have	5
2.2	Diagramma MoSCoW should have	6
2.3	Diagramma MoSCoW could have	6
2.4	Diagramma MoSCoW won't have	6
4.1	Diagramma delle classi	15
5.1	Diagramma delle attività - Login	17
5.2	Diagramma delle attività - Admin	18
5.3	Diagramma delle attività - User	19
5.4	Diagramma delle attività - Sistema	20
6.1	Login	21
6.2	Visualizzazione richieste - User	22
6.3	Visualizzazione richieste - Admin	22
6.4	Creazione/modifica richiesta - User	23
6.5	Creazione/modifica richiesta - Admin	23
6.6	Ricerca richiesta	24
6.7	Raggruppamento richieste	24
6.8	Visualizzazione tipi richieste	25
6.9	Creazione/modifica tipi richieste	25

Indice

Elenco delle figure	I
Indice	1
1 Introduzione	3
2 Analisi dei requisiti	5
3 Use Cases	7
4 Diagramma delle classi	15
5 Diagramma delle attività	17
5.0.1 Login	17
5.0.2 Attività utente	18
6 Conclusioni	21
6.0.1 Note operative	21
6.1 Stato corrente	21
6.1.1 Login	21
6.1.2 Visualizzazioni Richieste	22
6.1.3 Creazione/Modifica Richiesta	23
6.1.4 Ricerca Richiesta	24
6.1.5 Raggruppamento Richieste	24
6.1.6 Visualizzazione tipi richieste	25
6.1.7 Creazione/Modifica tipi richieste	25
6.2 Sviluppi futuri	25
Bibliografia	27

1 | Introduzione

Il progetto che abbiamo sviluppato consiste in un modulo Odoo che consente la gestione dei permessi lavorativi.

Il lavoro si è svolto in due fasi:

- Comprensione dell'architettura e delle tecnologie utilizzate all'interno del framework Odoo
- Progettazione e implementazione del modulo `WorkPermissions`

L'ERP si basa su un'architettura multilivello formata da

- Presentation tier: interfaccia utente
- Logic tier: elaborazione dati e trasferimento delle informazioni tra Presentation e Data tier
- Data tier: le informazioni sono salvate su **un** Database e File System

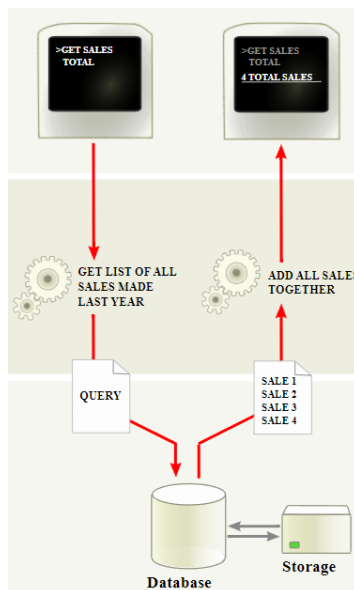


Figura 1.1: Architettura Odoo

Un modulo è un insieme di funzioni e dati che mirano a raggiungere un obiettivo o a risolvere un problema. Il modulo sviluppato come progetto contiene:

- Business Objects: classi python che rappresentano oggetti che tramite l'ORM vengono trasformati in tabelle nel Database
- Object Views: file XML o HTML che definiscono l'interfaccia utente
- Data Files: file XML o CSV che dichiarano i modelli di
 - View e Report
 - Dati di configurazione (parametri e regole d'accesso)

2 | Analisi dei requisiti

Per analizzare i requisiti abbiamo utilizzato il modello MoSCoW che classifica ... in:

- Must have: requisito indispensabile per il raggiungimento dell'obiettivo
- Should have: aspetto altamente prioritario, che vorremmo essere incluso nel lavoro svolto
- Could have: requisito auspicabile ma non indispensabile (da sviluppare se rimangono tempo e risorse)
- Won't have: ciò che per il momento può essere accantonato ed eventualmente ripreso in futuro (non rientra nelle priorità del progetto)



Must have
Accesso all'applicazione distinguendo tra i ruoli di 'User' e 'Admin'
Visualizzazione  le richieste di permesso lavorativo inserite
 Inserimento una nuova richiesta di permesso lavorativo
Eliminazione di una richiesta
Accettazione / Rifiuto di una richiesta
Inserimento di un nuovo tipo di richieste
Visualizzazione dei tipi di richieste
Eliminazione di un tipo di richiesta

Figura 2.1: Diagramma MoSCoW must have

Should have
Modifica di una richiesta
Ricerca / Filtraggio / Raggruppamento delle richieste
Modifica di un tipo di richiesta

Figura 2.2: Diagramma MoSCoW should have

Could have
Segnalazione di visualizzato su una richiesta
Scadenza delle richieste (sistema)

Figura 2.3: Diagramma MoSCoW could have

Won't have
Invio di una mail per informare l'user che la sua richiesta è stata accettata / rifiutata
Visualizzazione del calendario con richieste di permesso accettate
Visualizzazione del totale ore di permessi per ogni utente (admin)

Figura 2.4: Diagramma MoSCoW won't have

3 | Use Cases

Di seguito sono specificati i casi d'uso implementati.

Quando si cita l'Utente si intende l'entità loggata, qualsiasi sia il suo ruolo, mentre laddove l'operazione è performata da uno specifico ruolo questo verrà indicato come 'User' o 'Admin'.

Nome	UC1 - Autenticazione
Descrizione	L'utente esegue l'accesso al sistema
Attori	Utente
Precondizioni	L'utente è già in possesso di un account
Flusso eventi	<ol style="list-style-type: none">1. L'utente accede alla schermata di login2. L'utente esegue il login utilizzando la sua email e la sua password.
Postcondizioni	L'utente è loggato come 'Admin' o come 'User'.
Eccezioni	L'utente non possiede un account

Tabella 3.1: UC1 - Autenticazione

Nome	UC2 - Visualizzazione richieste utente
Descrizione	L'utente visualizza le richieste di permessi/ferie in ordine di stato della richiesta
Attori	User, Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma
Flusso eventi	<ol style="list-style-type: none"> 1. Se l'utente è loggato come 'User' visualizza solamente le sue richieste. Se l'utente è loggato come 'Admin' visualizza tutte le richieste
Postcondizioni	
Eccezioni	Non sono presenti richieste

Tabella 3.2: UC2 - Visualizzazione richieste

Nome	UC3 - Inserimento di una nuova richiesta
Descrizione	L'utente crea e inserisce nel sistema una nuova richiesta di permesso/ferie
Attori	User, Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma
Flusso eventi	<ol style="list-style-type: none"> 1. L'utente seleziona il tipo di richiesta da inserire 2. L'utente inserisce la data di inizio della richiesta 3. L'utente seleziona la durata della richiesta in ore o giorni
Postcondizioni	Il sistema salva la richiesta con stato 'New'
Eccezioni	Inserimento di una durata negativa

Tabella 3.3: UC3 - Inserimento nuova richiesta

Nome	UC4 - Modifica di una richiesta
Descrizione	L'utente modifica una richiesta
Attori	User, Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma La richiesta è nello stato 'New' o 'Pending'
Flusso eventi	<ol style="list-style-type: none"> 1. Se l'utente è loggato come 'User' può selezionare solamente una delle sue richieste. Se l'utente è loggato come 'Admin' può selezionare una delle richieste 2. L'utente modifica la richiesta
Postcondizioni	Il sistema salva la richiesta
Eccezioni	Inserimento di una durata negativa

Tabella 3.4: UC4 - Modifica richiesta

Nome	UC5 - Eliminazione di una richiesta
Descrizione	L'utente cancella una richiesta
Attori	User, Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma
Flusso eventi	<ol style="list-style-type: none"> 1. Se l'utente è loggato come 'User' può selezionare solamente una delle sue richieste. Se l'utente è loggato come 'Admin' può selezionare una delle richieste 2. L'utente elimina la richiesta
Postcondizioni	
Eccezioni	

Tabella 3.5: UC5 - Cancellazione richiesta

Nome	UC6 - Ricerca di una richiesta
Descrizione	L'utente ricerca una richiesta
Attori	User, Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma
Flusso eventi	<p>1. Se l'utente è loggato come 'User' specifica il valore di uno tra i seguenti campi: data e ora di inizio, data e ora di fine, tipo della richiesta, titolo della richiesta.</p> <p>Se l'utente è loggato come 'Admin' specifica il valore di uno tra i seguenti campi: data e ora di inizio, data e ora di fine, tipo della richiesta, titolo della richiesta, richiedente, data di creazione</p>
Postcondizioni	
Eccezioni	

Tabella 3.6: UC6 - Ricerca

Nome	UC7 - Raggruppamento richieste
Descrizione	L'Admin raggruppa le richieste
Attori	Admin
Precondizioni	L'Utente ha eseguito l'accesso alla piattaforma come 'Admin'
Flusso eventi	L'amministratore seleziona uno dei filtri predefiniti che raggruppano le richieste per utente, per tipo o per entrambi
Postcondizioni	
Eccezioni	

Tabella 3.7: UC7 - Raggruppamento

Nome	UC8 - Modifica stato richiesta
Descrizione	L'amministratore modifica lo stato della richiesta
Attori	Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma come 'Admin' La richiesta è nello stato 'New' o 'Pending'
Flusso eventi	<ol style="list-style-type: none"> 1. L'admin seleziona la richiesta 2. L'admin seleziona lo stato tra 'Accepted', 'Refused' o 'Pending' <p>Oppure</p> <ol style="list-style-type: none"> 1. L'admin seleziona lo stato tra 'Accepted', 'Refused' o 'Pending' direttamente dalla vista iniziale
Postcondizioni	
Eccezioni	

Tabella 3.8: UC8 - Richiesta visualizzata

Nome	UC9 - Scadenza richiesta
Descrizione	Il sistema imposta la richiesta come 'Refused' dopo 3 giorni
Attori	Sistema
Precondizioni	
Flusso eventi	<ol style="list-style-type: none"> 1. Il sistema verifica se la richiesta è scaduta ovvero se sono passati 3 giorni dalla creazione e questa non è stata né accettata né rifiutata 2. Il sistema imposta la richiesta come 'Refused'
Postcondizioni	
Eccezioni	

Tabella 3.9: UC9 - Scadenza richiesta

Nome	UC10 - Visualizzazione tipi di richieste
Descrizione	L'amministratore visualizza i tipi di richieste selezionabili
Attori	Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma come 'Admin'
Flusso eventi	1. L'amministratore visualizza le tipologie di richieste
Postcondizioni	
Eccezioni	

Tabella 3.10: UC10 - Visualizzazione tipologie richieste

Nome	UC11 - Inserimento di un nuovo tipo di richiesta
Descrizione	L'amministratore crea e inserisce nel sistema un nuovo tipo di richiesta
Attori	Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma come 'Admin'
Flusso eventi	1. L'amministratore inserisce il nome del nuovo tipo di richiesta 2. L'amministratore salva il nuovo tipo di richiesta
Postcondizioni	
Eccezioni	Inserimento di un tipo già esistente

Tabella 3.11: UC11 - Inserimento nuovo tipo di richiesta

Nome	UC12 - Modifica di un tipo di richiesta
Descrizione	L'amministratore modifica un tipo richiesta
Attori	Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma come 'Admin'
Flusso eventi	<ol style="list-style-type: none"> 1. L'amministratore seleziona uno dei tipi di richiesta. 2. L'amministratore modifica il tipo di richiesta
Postcondizioni	Il sistema salva la richiesta
Eccezioni	

Tabella 3.12: UC12 - Modifica tipo di richiesta

Nome	UC13 - Eliminazione di un tipo di richiesta
Descrizione	L'amministratore cancella un tipo di richiesta
Attori	Admin
Precondizioni	L'utente ha eseguito l'accesso alla piattaforma come 'Admin'
Flusso eventi	<ol style="list-style-type: none"> 1. L'utente seleziona un tipo di richiesta. 2. L'utente elimina il tipo di richiesta
Postcondizioni	
Eccezioni	

Tabella 3.13: UC13 - Cancellazione tipo di richiesta

4 | Diagramma delle classi

Di seguito è mostrato il diagramma UML delle classi:

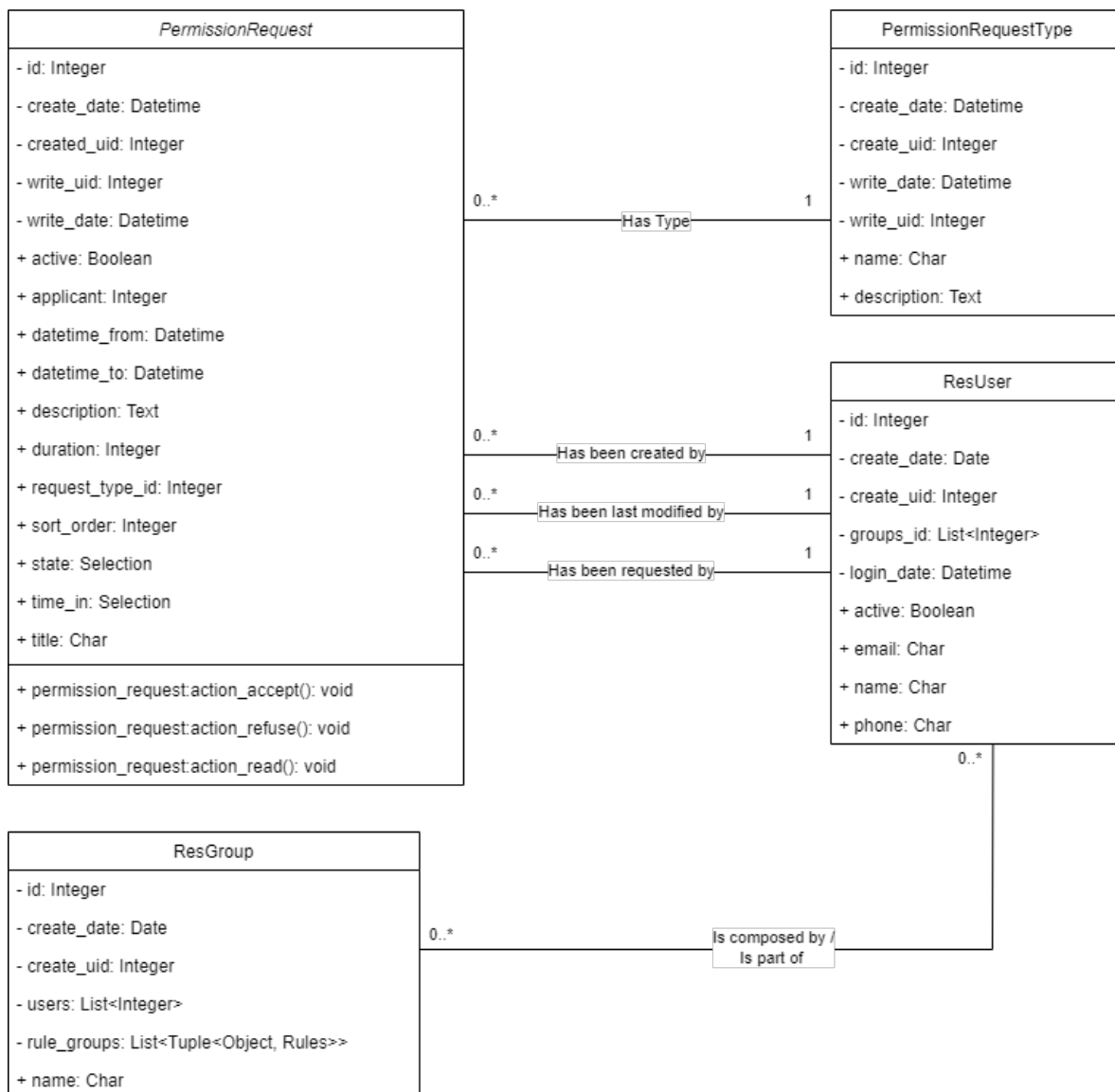


Figura 4.1: Diagramma delle classi

Una best practice che è stata seguita è rendere privati tutti quei campi che non possono/devono essere modificati direttamente dall'utente.

La classe `PermissionRequest` rappresenta la richiesta di permesso lavorativo.

Il campo `applicant` specifica chi è il richiedente, che in caso di inserimento da parte dell'admin, potrebbe non corrispondere con il campo `create_uid` che indica l'utente che ha creato la richiesta. Entrambi i campi sopracitati sono chiavi esterne che si riferiscono al campo `id` della classe `ResUser`

Il campo `datetime_to` non è modificabile dall'utente ma viene calcolato in base ai dati inseriti nei campi `datetime_from` e `duration` che specificano rispettivamente data ed ora di inizio e durata del permesso.

Il campo `time_in` discrimina tra permessi giornalieri e orari, è di tipo Selection poiché può assumere solo determinati valori (nel nostro caso 'day' o 'hour').

Le funzioni `permission_request_action_accept`, `permission_request_action_refuse` e `permission_request_action_read` sono richiamate dalla vista nel momento in cui l'admin modifica lo stato della richiesta.

La classe `PermissionRequestType` rappresenta i possibili tipi di richiesta (ad esempio 'non pagate', 'pagate', 'malattia',...).

La classe `ResUser` rappresenta gli utenti del sistema. È una classe di default di Odoo della quale abbiamo specificato solo i campi più significativi.

In particolare il campo `groups_id` contiene una lista di id ognuno dei quali si riferisce al campo `id` della classe `ResGroup`.

La classe `ResGroups` rappresenta i gruppi del sistema. Anche questa è una classe di default di Odoo di cui abbiamo specificato solo i campi più significativi.

Il campo `rule_group` specifica le regole d'accesso associate al gruppo nella forma (Oggetto/Classe, permessi di lettura, scrittura, creazione ed eliminazione).

5 | Diagramma delle attività

5.0.1. Login

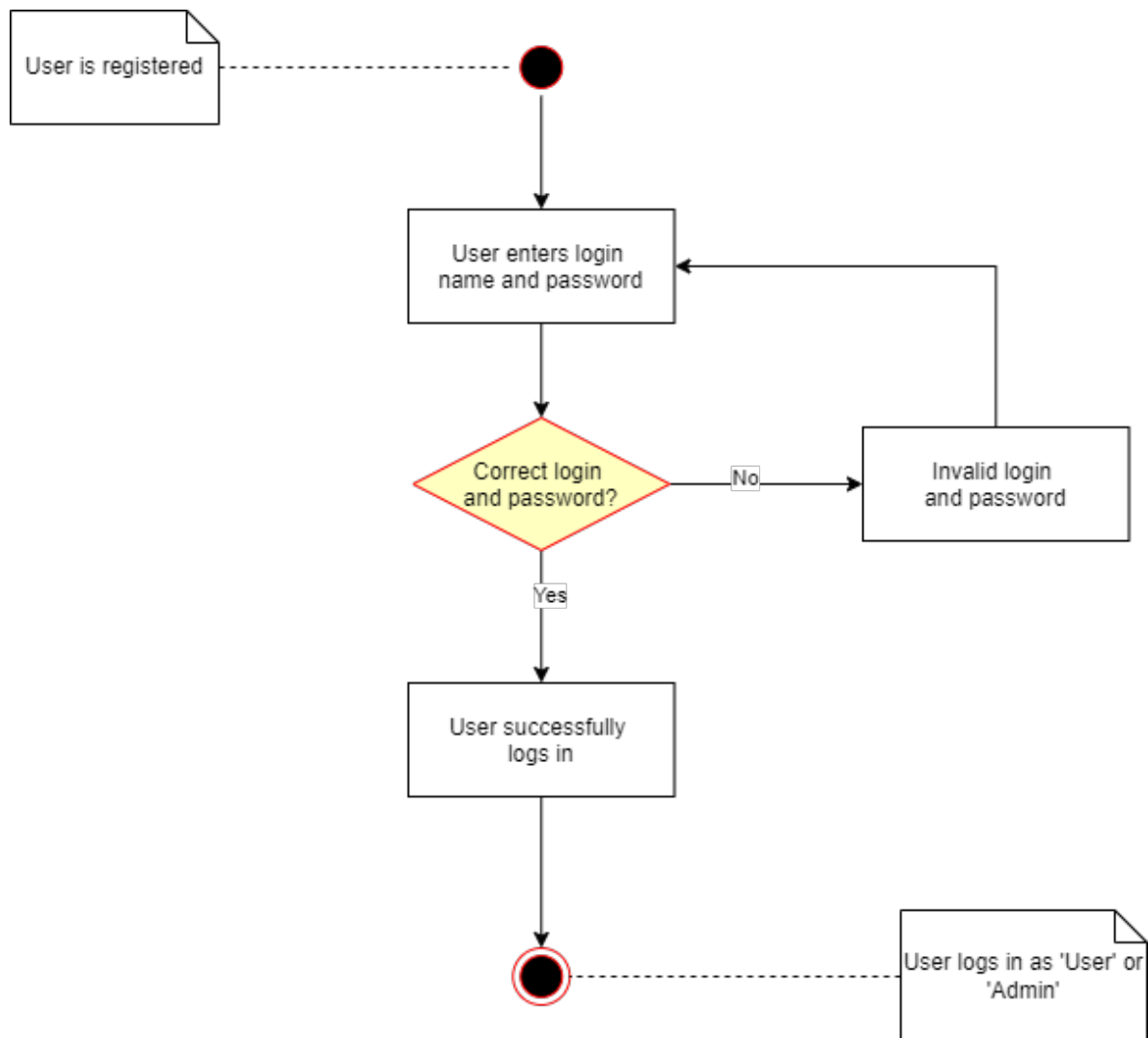


Figura 5.1: Diagramma delle attività - Login

5.0.2. Attività utente

Admin

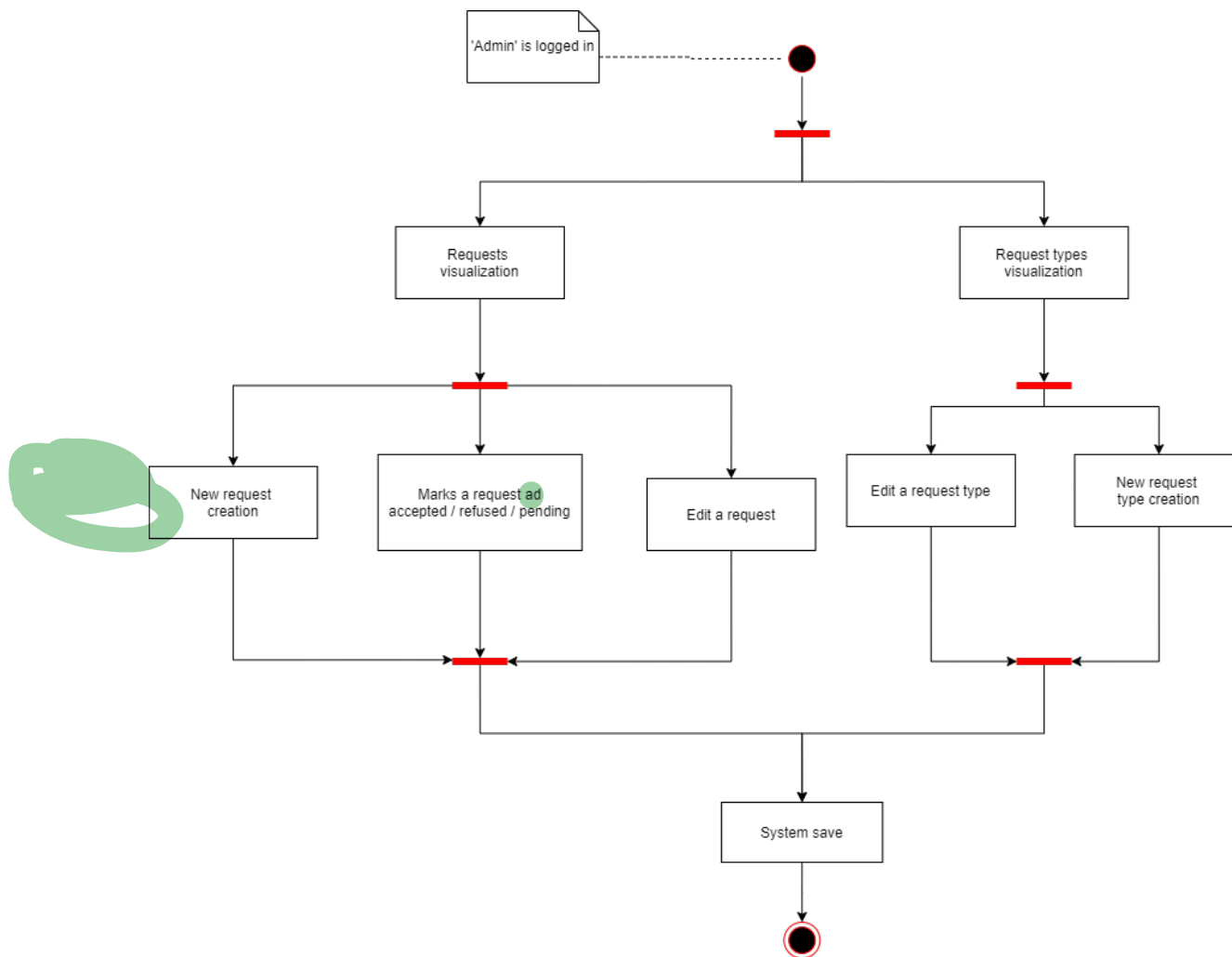


Figura 5.2: Diagramma delle attività - Admin

User

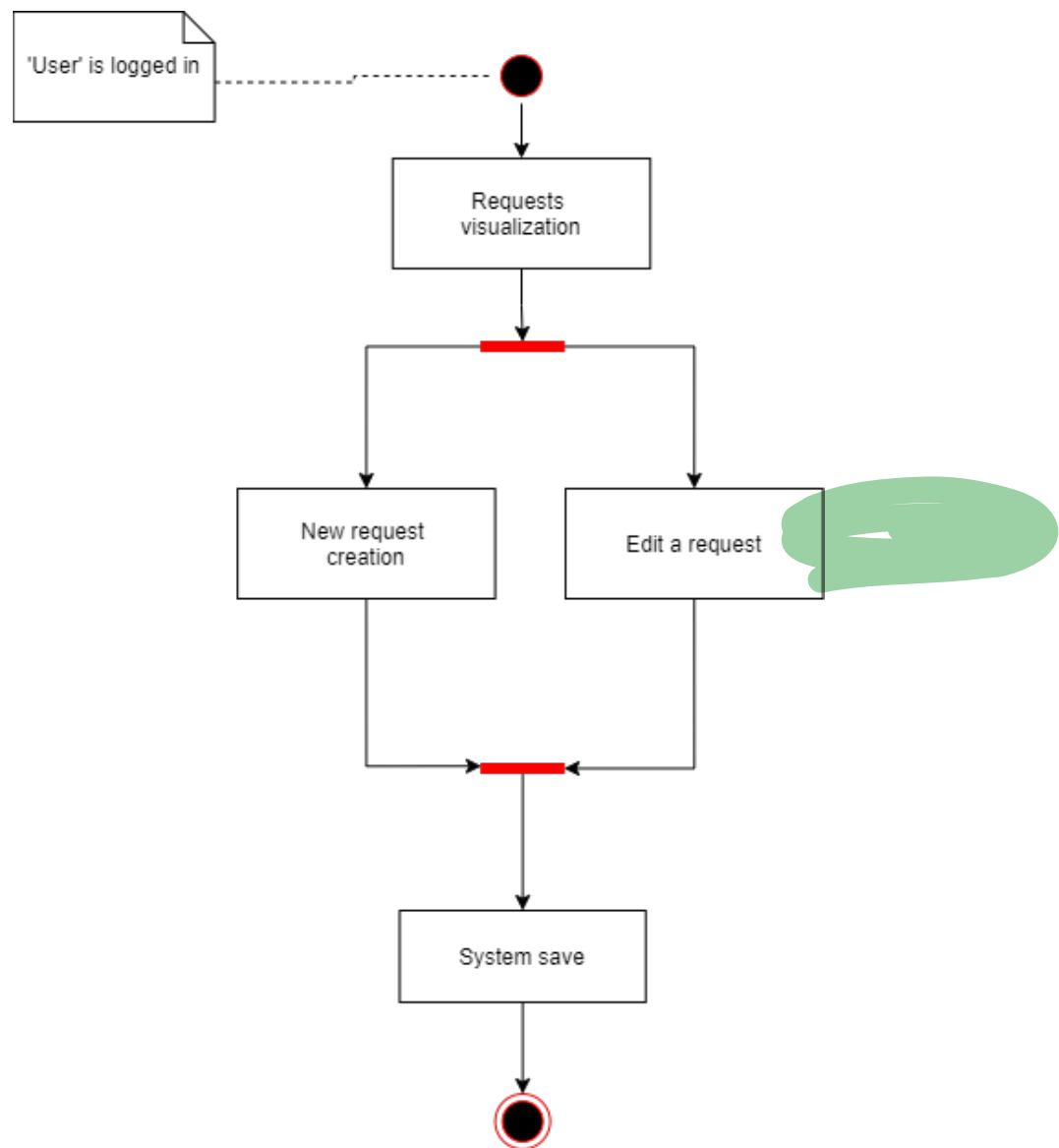


Figura 5.3: Diagramma delle attività - User

Sistema

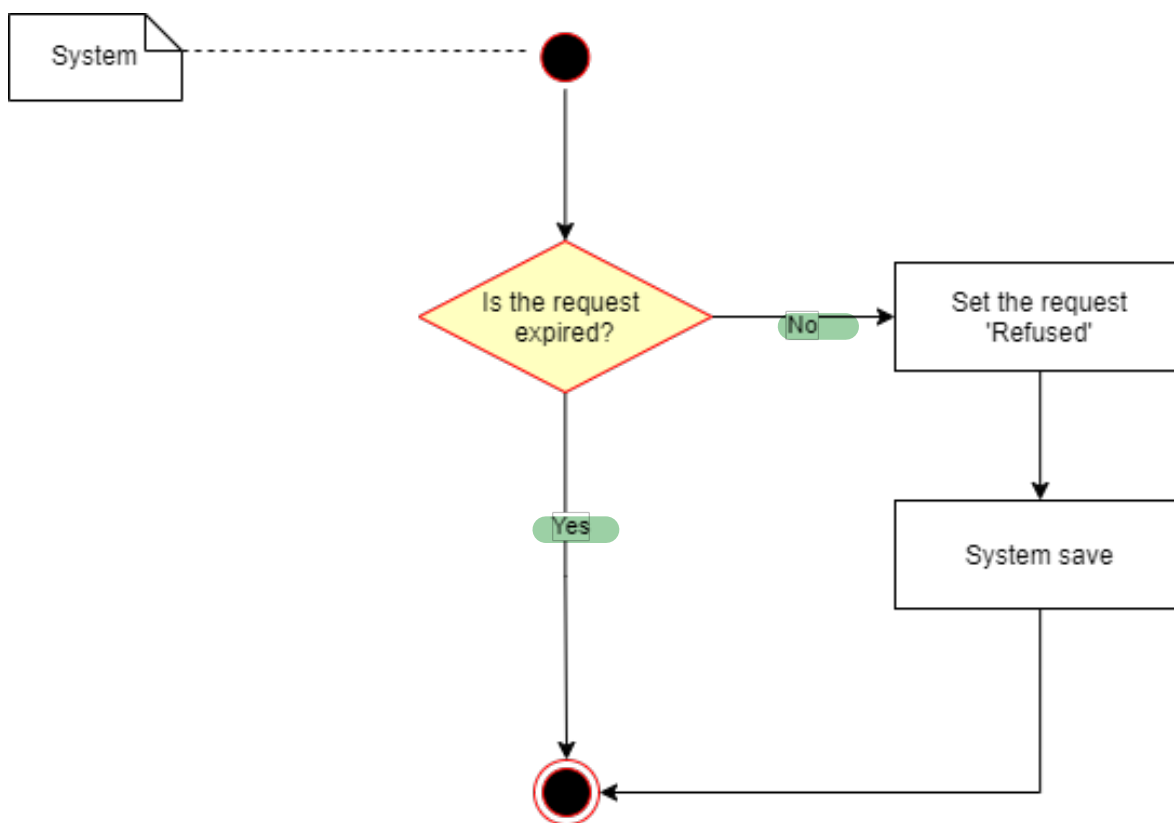


Figura 5.4: Diagramma delle attività - Sistema

6 | Conclusioni

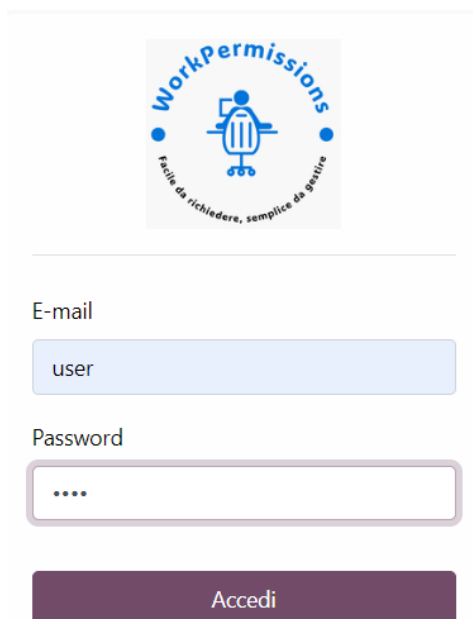
6.0.1. Note operative

Il progetto è sviluppato con Odoo 17 ed **eseguito** su server in locale situato all'interno del Windows Subsystem for Linux (WSL).

6.1. Stato corrente

Attualmente il sistema aiuta a gestire in modo efficiente, efficace ed intuitivo le richieste di permessi lavorativi. Di seguito sono mostrati degli screenshot che mostrano le varie viste raggiungibili tramite interfaccia utente.

6.1.1. Login



WorkPermissions

facile da richiedere, semplice da gestire

E-mail

user

Password

....

Accedi

Figura 6.1: Login

6.1.2. Visualizzazioni Richieste

User

Permission Requests			
<div> <div>Nuovo</div> <div>Permission Request</div> <div> <input type="text" value="Ricerca..."/> </div> <div>1-3 / 3</div> </div>			
<input type="checkbox"/> Title	Applicant	Type	Permission State
<input type="checkbox"/> [accepted] Andrea Cassina (Worker): Sick (14/07/2024 00:49:57)	Andrea Cassina (Worker)	Sick	Accepted
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Sick (14/07/2024 02:43:23)	Andrea Cassina (Worker)	Sick	Refused
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Paid (14/07/2024 02:48:04)	Andrea Cassina (Worker)	Paid	Refused

Figura 6.2: Visualizzazione richieste - User

Admin

Permission Requests			
<div> <div>Nuovo</div> <div>Permission Request</div> <div> <input type="text" value="Ricerca..."/> </div> <div>1-8 / 8</div> </div>			
<input type="checkbox"/> Title	Applicant	Type	Permission State
<input type="checkbox"/> [new] Bordogna Filippo (Admin): Paid (17/07/2024 18:03:11)	Bordogna Filippo (Admin)	Paid	New
<input type="checkbox"/> [pending] Bordogna Filippo (Admin): Unpaid (17/07/2024 18:03:39)	Bordogna Filippo (Admin)	Unpaid	Pending
<input type="checkbox"/> [accepted] Andrea Cassina (Worker): Sick (14/07/2024 00:49:57)	Andrea Cassina (Worker)	Sick	Accepted
<input type="checkbox"/> [accepted] Bordogna Filippo (Admin): Unpaid (14/07/2024 02:57:08)	Bordogna Filippo (Admin)	Unpaid	Accepted
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Sick (14/07/2024 02:43:23)	Andrea Cassina (Worker)	Sick	Refused
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Paid (14/07/2024 02:48:04)	Andrea Cassina (Worker)	Paid	Refused
<input type="checkbox"/> [refused] Bordogna Filippo (Admin): Unpaid (14/07/2024 00:44:05)	Bordogna Filippo (Admin)	Unpaid	Refused
<input type="checkbox"/> [refused] Bordogna Filippo (Admin): Paid (14/07/2024 02:26:17)	Bordogna Filippo (Admin)	Paid	Refused

Figura 6.3: Visualizzazione richieste - Admin

6.1.3. Creazione/Modifica Richiesta

User

Permission Requests

Nuovo Permission Request permission.request,99

1 / 4

Pending

[pending] Andrea Cassina (Worker): Unpaid (17/07/2024 18:03:39)

Description	From	22/07/2024 00:00:00	
Type	Unpaid	Duration	1
Time in	Day	To	22/07/2024 23:59:00

Figura 6.4: Creazione/modifica richiesta - User

Admin

Permission Requests Settings

Nuovo Permission Request permission.request,98

1 / 8

Read Accept Refuse

New

[new] Bordogna Filippo (Admin): Paid (17/07/2024 18:03:11)

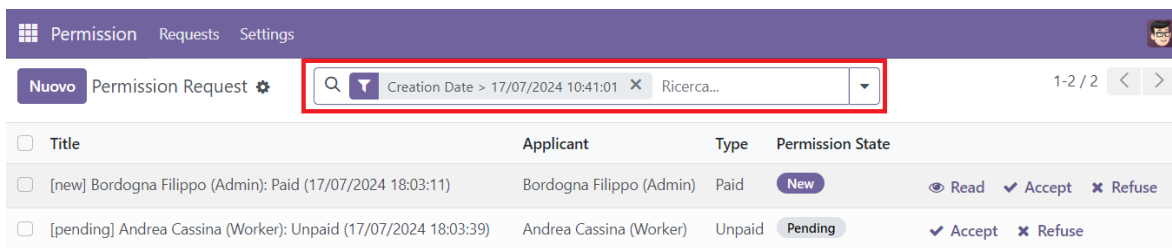
Description	From	17/07/2024 10:00:00	
Type	Paid	Duration	4
Time in	Hour	To	17/07/2024 14:00:00

Admin Fields

Applicant	Bordogna Filippo (Admin)
Active	<input checked="" type="checkbox"/>
Permission State ?	New
Sort Order	0
Creation Date	17/07/2024 18:03:11

Figura 6.5: Creazione/modifica richiesta - Admin

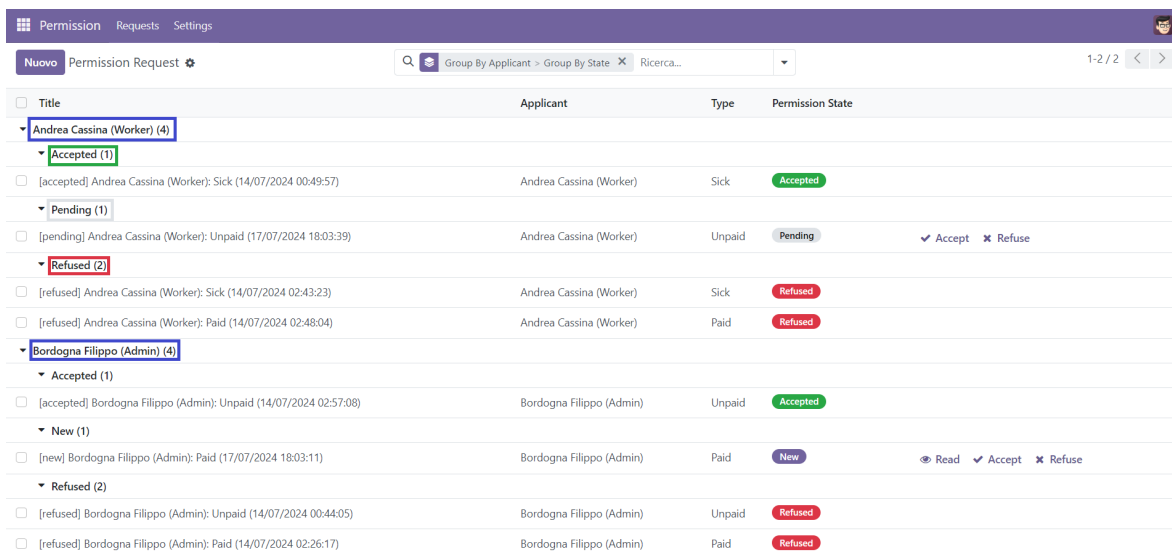
6.1.4. Ricerca Richiesta



<div> <div>Permission Requests Settings</div> <div> <div>Nuovo</div> <div>Permission Request</div> <div> <div> <div>Q</div> <div>Creation Date > 17/07/2024 10:41:01</div> <div>Ricerca...</div> </div> </div> <div>1-2 / 2</div> </div> </div>				
<input type="checkbox"/> Title	Applicant	Type	Permission State	
<input type="checkbox"/> [new] Bordogna Filippo (Admin): Paid (17/07/2024 18:03:11)	Bordogna Filippo (Admin)	Paid	New	Read Accept Refuse
<input type="checkbox"/> [pending] Andrea Cassina (Worker): Unpaid (17/07/2024 18:03:39)	Andrea Cassina (Worker)	Unpaid	Pending	Accept Refuse

Figura 6.6: Ricerca richiesta

6.1.5. Raggruppamento Richieste



<div> <div>Permission Requests Settings</div> <div> <div>Nuovo</div> <div>Permission Request</div> <div> <div> <div>Q</div> <div>Group By Applicant > Group By State</div> <div>Ricerca...</div> </div> </div> <div>1-2 / 2</div> </div> </div>				
<input type="checkbox"/> Title	Applicant	Type	Permission State	
<div>Andrea Cassina (Worker) (4)</div> <div>Accepted (1)</div>				
<input type="checkbox"/> [accepted] Andrea Cassina (Worker): Sick (14/07/2024 00:49:57)	Andrea Cassina (Worker)	Sick	Accepted	
<div>Pending (1)</div> <div>Refused (2)</div>				
<input type="checkbox"/> [pending] Andrea Cassina (Worker): Unpaid (17/07/2024 18:03:39)	Andrea Cassina (Worker)	Unpaid	Pending	Accept Refuse
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Sick (14/07/2024 02:43:23)	Andrea Cassina (Worker)	Sick	Refused	
<input type="checkbox"/> [refused] Andrea Cassina (Worker): Paid (14/07/2024 02:48:04)	Andrea Cassina (Worker)	Paid	Refused	
<div>Bordogna Filippo (Admin) (4)</div> <div>Accepted (1)</div> <div>New (1)</div> <div>Refused (2)</div>				
<input type="checkbox"/> [accepted] Bordogna Filippo (Admin): Unpaid (14/07/2024 02:57:08)	Bordogna Filippo (Admin)	Unpaid	Accepted	
<input type="checkbox"/> [new] Bordogna Filippo (Admin): Paid (17/07/2024 18:03:11)	Bordogna Filippo (Admin)	Paid	New	Read Accept Refuse
<input type="checkbox"/> [refused] Bordogna Filippo (Admin): Unpaid (14/07/2024 00:44:05)	Bordogna Filippo (Admin)	Unpaid	Refused	
<input type="checkbox"/> [refused] Bordogna Filippo (Admin): Paid (14/07/2024 02:26:17)	Bordogna Filippo (Admin)	Paid	Refused	

Figura 6.7: Raggruppamento richieste

6.1.6. Visualizzazione tipi richieste

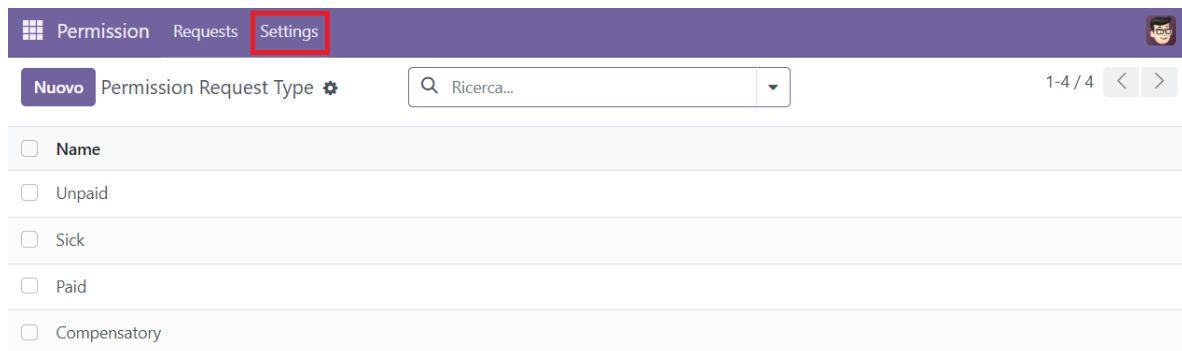


Figura 6.8: Visualizzazione tipi richieste

6.1.7. Creazione/Modifica tipi richieste

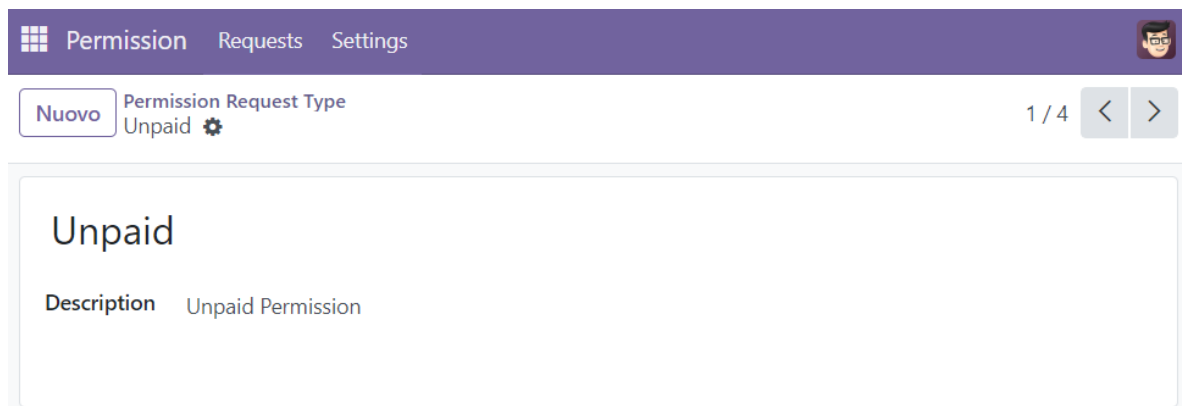


Figura 6.9: Creazione/modifica tipi richieste

6.2. Sviluppi futuri

Come specificato nell'analisi dei requisiti, alcune funzionalità non sono state implementate.

Tra queste possiamo trovare l'invio di email successivamente all'accettazione/rifiuto della richiesta (tramite Mail Plugin).

Altre funzionalità che sarebbe interessante inserire sono quelle di visualizzazione del totale delle ore di permesso richieste/effettuate da ogni utente (nella view dell'admin) e del calendario contenente i giorni di festa/ferie/permesso (nella vista dell'user).

Bibliografia

- [1] Odoo. Odoo setup, 2024. URL https://www.odoo.com/documentation/17.0/developer/tutorials/setup_guide.html.
- [2] Odoo. Odoo 17 architecture, 2024. URL https://www.odoo.com/documentation/17.0/developer/tutorials/server_framework_101/01_architecture.html.
- [3] Odoo. Odoo tutorial, 2024. URL https://www.odoo.com/documentation/17.0/developer/tutorials/server_framework_101.html.

