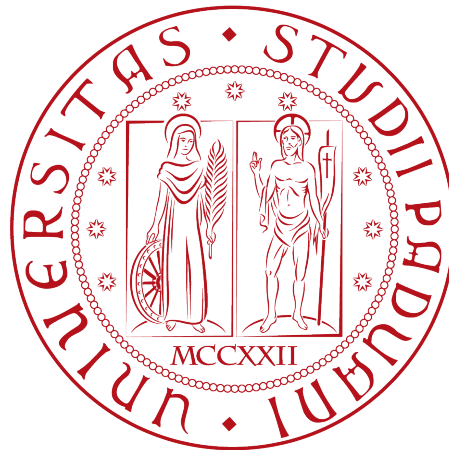


Università degli Studi di Padova

DIPARTIMENTO DI MATEMATICA "TULLIO LEVI-CIVITA"

CORSO DI LAUREA IN INFORMATICA



**"Analisi, Progettazione e Implementazione di
Nuove Funzionalità in un'applicazione Web
per la Schedulazione e Rendicontazione delle
Attività dei Team di Sviluppo Aziendale"**

Tesi di laurea

Relatore

Prof.ssa Ombretta Gaggi

Laureando

Jude Vensil Braceros 2011068

ANNO ACCADEMICO 2023-2024

Jude Vensil Bracerros: *"Analisi, Progettazione e Implementazione di Nuove Funzionalità in un'applicazione Web per la Schedulazione e Rendicontazione delle Attività dei Team di Sviluppo Aziendale"*, Tesi di laurea, © Febbraio 2024.

Sommario

Questa tesi documenta l'esperienza di stage di Jude Vensil Braceros presso l'azienda Wavelop, durante la quale sono state sviluppate nuove funzionalità per l'applicazione Timesheet, una webapp per la schedulazione e rendicontazione delle attività lavorative. Tra le nuove funzionalità sviluppate ci sono:

1. Creazione di gruppi di lavoro;
2. Visualizzazione di gruppi di lavoro;
3. Assegnazione e rimozione di un utente a un gruppo.

“That’s our duty, isn’t it? To leave our mark before we die?”

— Junji Ito, *Tomie*

Ringraziamenti

Per prima vorrei ringraziare la mia famiglia per avermi supportato durante tutti questi anni soprattutto per avermi aiutato a rialzarmi dai miei momenti più bui.

Desidero sinceramente ringraziare la Prof.ssa Ombretta Gaggi, relatrice della mia tesi, per l’inestimabile aiuto e sostegno fornito durante la stesura del mio lavoro.

Grazie a Nicola, Matteo e Franco per l’inestimabile supporto durante il mio stage. La vostra guida e collaborazione hanno contribuito in modo significativo alla mia crescita professionale e personale.

Infine ringrazio i miei amici con cui ho passato dei momenti di felicità che porterò con me per sempre.

Padova, Febbraio 2024

Un grazie sincero a tutti voi, Jude Vensil Braceros

Indice

1 Azienda	1
1.1 Servizi	1
1.2 Cultura aziendale	3
2 Progetto di Stage	4
2.1 Scopo dello Stage	8
2.2 Obiettivi	8
3 Tecnologie usate	9
3.1 Tecnologie per lo sviluppo	9
3.1.1 <i>MongoDB</i>	9
3.1.2 <i>Express.js</i>	9
3.1.3 <i>React</i>	10
3.1.4 <i>Node.js</i>	10
3.1.5 <i>Typescript</i>	10
3.2 Tecnologie di supporto e strumenti	11
3.2.1 <i>Docker</i>	11
3.2.2 <i>VSCode</i>	11
3.2.3 <i>MongoDB Compass</i>	11
4 Realizzazione del progetto	13
4.0.1 Analisi dei requisiti	13
4.1 Progettazione	18
4.1.1 Database	18
4.1.2 Struttura dei file	20
4.1.3 Tipi di utente	24
4.1.4 Pagina di modifica di un progetto	24
4.1.5 Report delle attività	24
4.2 Implementazione	25
4.2.1 Creazione di un gruppo	25
4.2.2 Gestione degli utenti all'interno di un gruppo	26
4.2.3 Assegnazione di un gruppo a un progetto	27
4.2.4 Report delle attività	28
4.2.5 Aggiunta di task	29
4.3 Verifica e Validazione	30
4.3.1 Frontend	30
4.3.2 Backend	30
4.3.3 Formatter	30

<i>INDICE</i>	vii
5 Conclusioni	31
5.1 Consuntivo delle attività	31
5.2 Raggiungimento degli obiettivi	32
5.3 Sviluppi futuri	32
5.4 Conoscenze acquisite	33
5.5 Valutazione personale	34
Glossario	35
Riferimenti Bibliografici	37

Elenco delle figure

1.1	Logo dell'azienda Wavelop Srls	1
1.2	Sprint cycle [1]	3
2.1	Login tramite SSO	4
2.2	Form inserimento attività	5
2.3	Form per la creazione di un nuovo spazio di lavoro	5
2.4	Form per invitare persone nel workspace	5
2.5	Form per modificare i dati di un cliente	6
2.6	Form per modificare i dati di un progetto	6
2.7	Grafico delle attività	7
3.1	Logo MongoDB	9
3.2	Logo ReactJS	10
3.3	Logo NodeJs	10
3.4	Logo TypeScript	11
3.5	Logo Docker	11
4.1	Use case gruppi	13
4.2	Use case 6 in dettaglio	14
4.3	Use case progetti	14
4.4	Documentazione interattiva delle API	22
4.5	Pagina utenti del workspace	25
4.6	Pagina gruppi del workspace	25
4.7	Form per la creazione di un gruppo	26
4.8	Form per l'aggiunta di un utente a un gruppo	26
4.9	Tabella degli utenti del gruppo	26
4.10	Select con utenti e gruppi	27
4.11	Tabella utenti e gruppi assegnati al progetto	27
4.12	Filtro utenti e gruppi nella pagina report	28
4.13	Form per l'aggiunta di task	29

Elenco delle tabelle

2.1	Prospetto degli obiettivi.	8
5.1	Tempi previsti ed effettivi dello svolgimento del progetto.	31
5.2	Stato di completamento degli obiettivi.	32

Elenco dei codici

4.1	codice per select all'interno della pagina gruppi	20
4.2	codice per lo stile della pagina dei gruppi	21
4.3	esempio di un test che controlla che l'operazione di POST vada a buon fine	30

Capitolo 1

Azienda



Figura 1.1: Logo dell'azienda Wavelop Srls

Wavelop Srls è un'azienda software fondata nel 2018 a Treviso, da tre *alumni* dell'università di Padova, che progetta e sviluppa soluzioni per business e startup di vario tipo.

1.1 Servizi

L'azienda offre diversi servizi ma il principale è quello di sviluppare soluzioni web tramite questi passaggi:

- Digital product design: Attraverso un approccio centrato sull'utente, ci si concentra sui bisogni dell'utente finale per garantire un'eccellente User Experience. Collaborando attivamente, vengono creati mock-up e prototipi per definire il prodotto desiderato, raccogliendo feedback e perfezionando il design in modo iterativo. L'obiettivo principale è creare soluzioni che soddisfino appieno le esigenze dell'utente, mettendolo costantemente al primo posto nel processo di sviluppo.
- Web development: che si divide in:
 - Frontend development: Utilizzando tecnologie JavaScript di avanguardia come React, Angular e Vue, la squadra si impegna nella creazione di soluzioni ad alte prestazioni. Attraverso queste piattaforme, trasformano i mock-up in esperienze tangibili, assicurando che l'utente goda della migliore User

Experience possibile. Grazie a questo approccio tecnologico, sono in grado di realizzare prodotti altamente performanti, offrendo agli utenti un'interazione fluida e coinvolgente con le soluzioni sviluppate.

- Backend development: La squadra si specializza nello sviluppo di sistemi Back-End, API e web app con rendering lato server, integrandoli senza problemi con una vasta gamma di database, sia SQL che NoSQL. Queste competenze vengono implementate attraverso l'uso di tecnologie consolidate come Node.js, Java o PHP.
- Aws development: La squadra adotta soluzioni Cloud di alta qualità, sostenute da robuste garanzie di sicurezza, seguendo l'approccio centrato sul cliente che caratterizza l'ecosistema Amazon.

Mobile Application Development La squadra sfrutta competenze nel mondo JavaScript per lo sviluppo di applicazioni mobile ibride, facendo uso di tecnologie come Ionic e React Native. Questa conoscenza avanzata consente di creare app fruibili su dispositivi Android e iOS. Grazie all'utilizzo di framework ibridi, il team ottimizza l'efficienza nello sviluppo, garantendo al contempo un'esperienza utente coerente e di qualità su entrambe le piattaforme.

Chat Bot Il team sfrutta i nuovi canali di comunicazione per potenziare vari processi di business, posizionando il cliente in modo competitivo nel mercato emergente. La filosofia del team è improntata all'innovazione, riconoscendo che essa gioca un ruolo chiave nel mantenere la competitività.

1.2 Cultura aziendale

Wavelop adotta un approccio di sviluppo flessibile seguendo la metodologia *scrum*_[G], che si basa su principi agili e che divide il lavoro in *sprint*_[G]. La struttura della metodologia Scrum consiste delle seguenti parti:

- Sprint Planning: riunione in cui si sceglie dal *product backlog*_[G] quali sono gli elementi più importanti che si metteranno nello *sprint backlog*_[G]
- Daily stand-up: riunione giornaliera dalla durata contenuta in cui ogni componente del team descrive cosa ha fatto il giorno precedente, cosa farà nel giorno corrente e se ha incontrato degli ostacoli durante il lavoro.
- Sprint review: Una revisione alla fine di ogni sprint in cui il team dimostra ciò che è stato realizzato e raccoglie feedback.
- Sprint retrospective: Una riflessione sullo sprint appena concluso, in cui il team valuta cosa è andato bene, cosa può essere migliorato e definisce azioni per le future iterazioni.

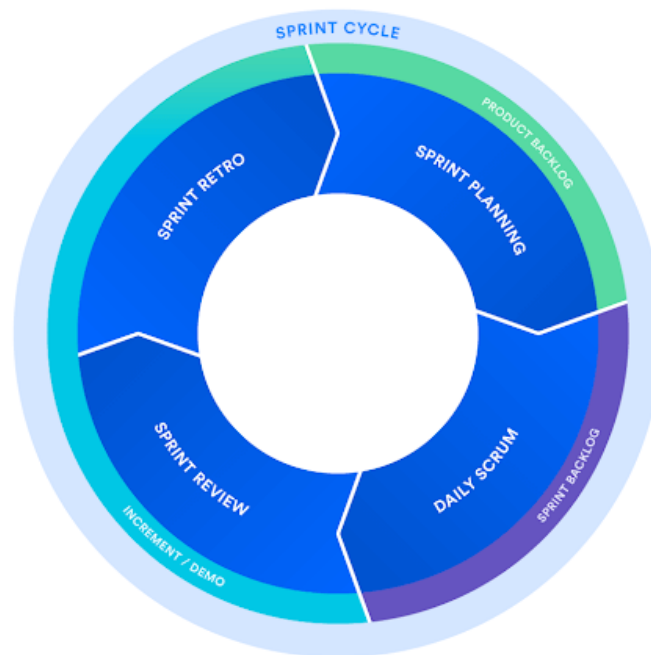


Figura 1.2: Sprint cycle [1]

Capitolo 2

Progetto di Stage

Il progetto di stage descritto in questa tesi consiste nello sviluppo di nuove funzionalità per la webapp *Timesheet* un prodotto interno dell'azienda per la gestione delle attività lavorative. Prima del mio stage l'applicazione permetteva di:

- Effettuare l'autenticazione tramite SSO_{IG} (Single sign on);

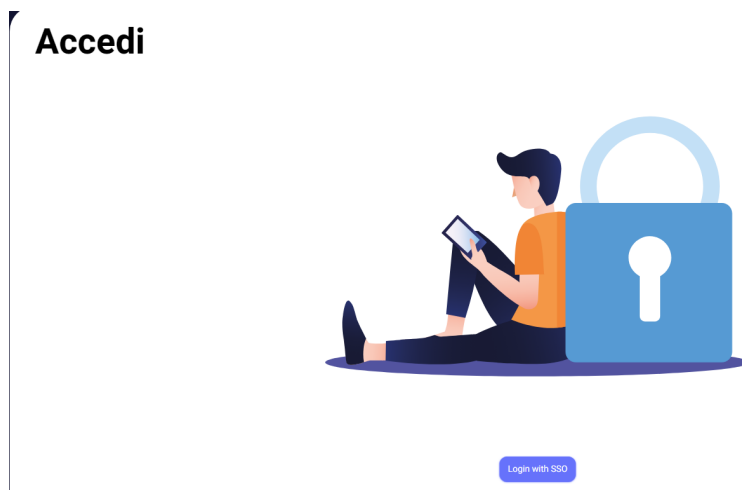


Figura 2.1: Login tramite SSO

- Rendicontare e pianificare le attività eseguite sui clienti e sui progetti;

Attività 03:08 Oggi, 2 febbraio

Data
02 febbraio

Ore
0 h 00 m

Progetto: Seleziona il progetto

Attività: Tipo di attività

Postazione: Postazione

Descrizione: Inserisci una descrizione

Link: Inserisci un link di riferimento

Pulisci Edit avanzato Salva e duplica Salva

Task 1 dicembre 30h

Tipo di attività	Progetto	Descrizione	Postazione	Ore	Azioni
Sviluppo	Porto	1 dic, 2023	Cliente	8h	[checkbox] [document] [trash]

Figura 2.2: Form inserimento attività

- Creare e gestire spazi di lavoro;

Nome
Nome dello spazio di lavoro

Descrizione
Inserisci una descrizione

Codice
Codice del workspace

Status
Attivo

Pulisci Indietro Salva

Figura 2.3: Form per la creazione di un nuovo spazio di lavoro

- Invitare tramite email persone allo spazio di lavoro;

Invito allo spazio di lavoro
Immettere l'email

Pulisci Invia

Figura 2.4: Form per invitare persone nel workspace

- Modificare i dati dei clienti e dei progetti;

Nome
interno

Descrizione
Inserisci una descrizione di riferimento

Status
Non attivo

Codice
1

Pulisci Indietro Salva

Figura 2.5: Form per modificare i dati di un cliente

Nome
Progetto 1

Guadagno
Guadagno previsto

Tempo
Tempo previsto

Descrizione
Inserisci una descrizione

Cliente
interno

Codice
i231

Status
Attivo

Pulisci Indietro Salva

Figura 2.6: Form per modificare i dati di un progetto

- Visualizzare un grafico delle attività e applicare dei filtri;
- Esportare il grafico come CSV o PDF.



Figura 2.7: Grafico delle attività

2.1 Scopo dello Stage

Le funzionalità da sviluppare sono quelle di poter creare gruppi di utenti, infatti l'azienda ha individuato come una necessità quella di poter gestire più efficientemente un gruppo di persone.

2.2 Obiettivi

Nel piano di lavoro l'azienda ha definito una serie di obiettivi e di aspettative da dover compiere entro la fine del percorso formativo. Essi sono stati suddivisi in tre categorie e vengono identificati con la seguente notazione:

- O - per i requisiti obbligatori, vincolanti in quanto obiettivo primario richiesto dall'azienda;
- D - per i requisiti desiderabili, non vincolanti o strettamente necessari, ma dal riconoscibile valore aggiunto;
- F - per i requisiti facoltativi, rappresentanti valore aggiunto non strettamente competitivo.

Nella Tabella 2.1 vengono riportati tutti gli obiettivi richiesti; l'indicazione del tipo di requisito sarà correlata da un indice numerico sequenziale che lo identifica.

Tabella 2.1: Prospetto degli obiettivi.

<i>Obiettivo</i>	<i>Definizione</i>
O01	Apprendimento delle tecnologie di sviluppo come React e Node.js e per il versionamento del progetto come git.
O02	Creazione di gruppi di utenti.
O03	Visualizzazione dei gruppi di utenti con relativi dati.
O04	Assegnazione e rimozione di un utente da un gruppo.
O05	Visualizzazione dei progetti per assegnazione diretta o per gruppo.
D01	Definizione di uno o più manager per gruppo.
D02	Visualizzazione report con capability da Manager.
F01	Implementare il multilingua.
F02	Visualizzare progetti abilitati.
F03	Effettuare il <i>refactoring</i> _[G] dei form.

Capitolo 3

Tecnologie usate

In questa sezione vengono definite quali sono state le tecnologie usate durante il progetto di stage.

3.1 Tecnologie per lo sviluppo

Per realizzare la webapp è stato utilizzato lo *stack tecnologico*_[G] MERN [2], uno stack JavaScript molto usato dall'azienda ed è un acronimo che sta per:

3.1.1 *MongoDB*



Figura 3.1: Logo MongoDB

MongoDB è un database management system(*DBMS*_[G]) *NoSQL*_[G] che memorizza i dati in un formato simile a *JSON*_[G] chiamato BSON (Binary JSON). Conosciuto soprattutto per la sua scalabilità, flessibilità e facilità d'uso. MongoDB è progettato per gestire grandi volumi di dati ed è particolarmente adatto per applicazioni che richiedono schemi flessibili, come quelli negli ambienti di sviluppo agili.

3.1.2 *Express.js*

Express.js, o semplicemente Express, è un framework minimale e flessibile per applicazioni web in Node.js che fornisce un insieme robusto di funzionalità per lo sviluppo di applicazioni web e mobile. È progettato per creare rapidamente applicazioni web e API in Node.js con codice minimale.

3.1.3 *React*



Figura 3.2: Logo ReactJS

React, anche conosciuto come React.js o ReactJS, è una libreria JavaScript open-source sviluppata da Meta. Utilizzata per la costruzione di interfacce utente (UI) per applicazioni a pagina singola, React è particolarmente popolare per la creazione di componenti riutilizzabili e la gestione efficiente dello stato dell'interfaccia. React utilizza una sintassi chiamata JSX che consente di scrivere la struttura dell'interfaccia utente utilizzando una sintassi simile all'XML o HTML all'interno del codice JavaScript.

3.1.4 *Node.js*



Figura 3.3: Logo NodeJs

Node.js è un ambiente di runtime open-source basato su JavaScript che consente l'esecuzione di codice JavaScript lato server, rendendo possibile la condivisione di codice e competenze tra il frontend e il backend. Node.js è ampiamente utilizzato per la creazione di applicazioni web, API, applicazioni di streaming in tempo reale, applicazioni IoT (Internet of Things), applicazioni desktop e altro ancora.

3.1.5 *Typescript*

TypeScript è un linguaggio di programmazione open source sviluppato da Microsoft. Essenzialmente, si tratta di un'estensione di JavaScript che aggiunge funzionalità come tipi statici, classi, interfacce e moduli. Questo linguaggio tipizzato consente ai programmatori di definire chiaramente la struttura dei dati e di fornire una documentazione più precisa, consentendo inoltre a TypeScript di rilevare potenziali errori nel codice durante la fase di sviluppo.

Un aspetto distintivo di TypeScript rispetto ad altri linguaggi è che è un superset di JavaScript. Ciò significa che qualsiasi codice JavaScript esistente è già compatibile



Figura 3.4: Logo TypeScript

con TypeScript, consentendo ai programmatori di integrare gradualmente le nuove funzionalità di TypeScript nei loro progetti senza dover riscrivere completamente il codice. Questa caratteristica favorisce l'adozione di TypeScript da parte di sviluppatori che sono già familiari con JavaScript e desiderano migliorare la manutenibilità e la sicurezza del proprio codice senza dover imparare un nuovo linguaggio da zero

3.2 Tecnologie di supporto e strumenti

3.2.1 *Docker*



Figura 3.5: Logo Docker

Docker è una piattaforma open source che permette di automatizzare il deployment (il rilascio) e la gestione di applicazioni all'interno di contenitori software. I contenitori sono ambienti leggeri e isolati che contengono tutto il necessario per eseguire un'applicazione, inclusi il codice, le dipendenze e le librerie, garantendo che l'applicazione funzioni in modo coerente su qualsiasi ambiente in cui venga eseguita. Utilizzato per avviare il database.

3.2.2 *VSCode*

VSCode è un popolare IDE sviluppato da Microsoft. Utilizzato per preferenza personale.

3.2.3 *MongoDB Compass*

MongoDB Compass è un'interfaccia grafica utente (GUI) sviluppata da MongoDB Inc. per l'interazione con i database MongoDB. Essa fornisce agli sviluppatori e agli amministratori un'esperienza visiva per esplorare e interagire con i dati all'interno di

un database MongoDB. Con MongoDB Compass, è possibile eseguire operazioni come la visualizzazione dei dati, la creazione di query, l'ottimizzazione delle prestazioni e la gestione degli indici, il tutto tramite un'interfaccia intuitiva e facile da usare.

Capitolo 4

Realizzazione del progetto

4.0.1 Analisi dei requisiti

Per la parte di analisi dei requisiti ho creato dei diagrammi *use case*_[G] come fase preliminare per identificare come sarebbero state sviluppate le nuove funzionalità.

Attori

- Amministratore: gli utenti amministratori hanno i permessi di accedere alle nuove funzionalità introdotte per la gestione dei gruppi.
- Utente autenticato: questo tipo di utente può essere assegnato a dei gruppi e poi visualizzare i progetti di appartenenza di tali gruppi.

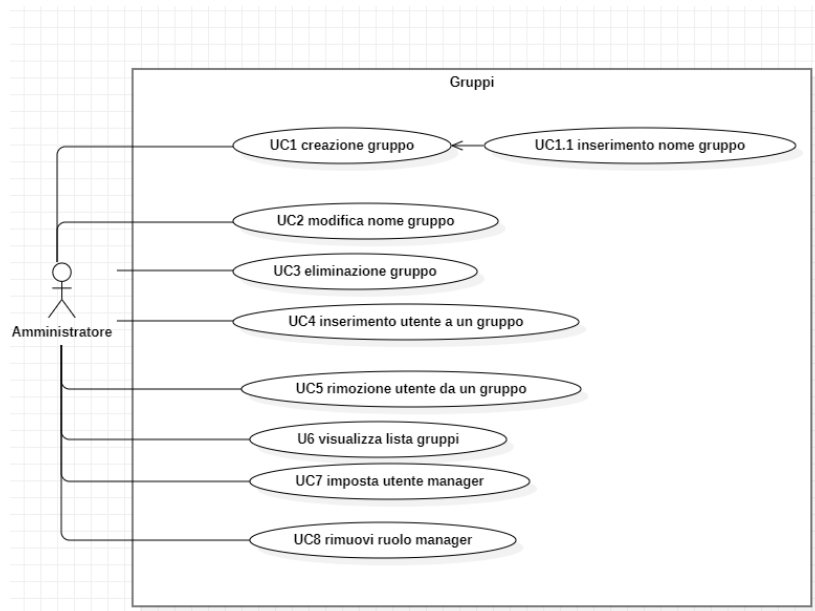


Figura 4.1: Use case gruppi

Nel diagramma UML 4.1 vengono descritti gli use case trovati nell'interazione degli amministratori con la pagina di gestione dei gruppi.

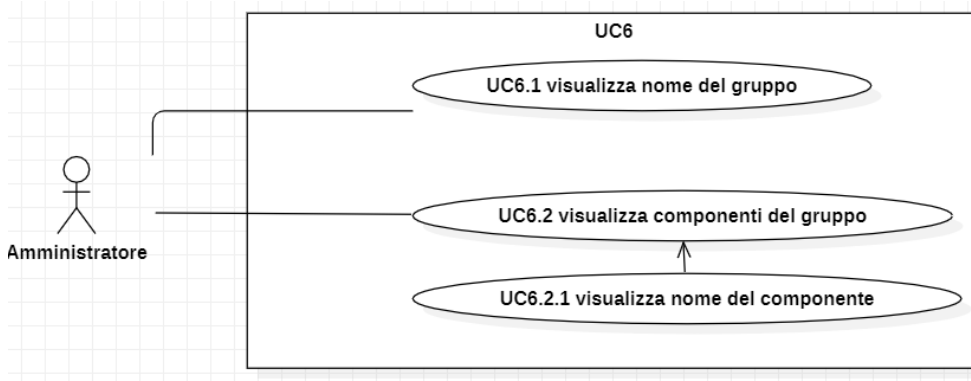


Figura 4.2: Use case 6 in dettaglio

Nel diagramma UML 4.2 viene descritto nello specifico lo use case 6 che definisce come dovrebbero essere visualizzati i singoli gruppi.

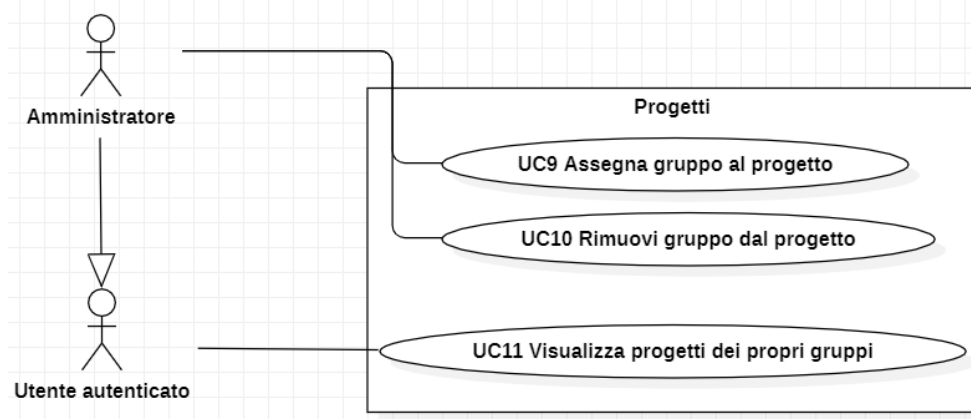


Figura 4.3: Use case progetti

Nel diagramma UML 4.3 vengono descritti gli use case riguardanti i progetti, e che riguardano nuove funzionalità introdotte dalla implementazione dei gruppi.

Dai diagrammi use case sono poi state definite delle *user stories*_[G], che sono orientate verso gli obiettivi degli utenti e sono più focalizzate sul "perché" e "per chi" di una funzionalità rispetto agli use case. Ogni user story è scritta in un linguaggio informale e segue questa specifica struttura:

- Titolo (peso)
- Descrizione del tipo: "Come [tipo di utente], l'utente deve poter [obiettivo] così che [ragione]"
- Lista di task per completare la user story.

Il peso, rappresentato da un numero della serie di Fibonacci, simboleggia quanto in termini di ore ci si aspetta che la user story possa impiegare. La user story con un peso di:

- 1: richiede poco tempo, è immediato
- 2: richiede almeno un'ora di lavoro, non immediato
- 3: richiede almeno mezza giornata di lavoro
- 5: richiede una giornata di lavoro intera
- 8: richiede più giorni di lavoro
- 13: richiede una settimana intera di lavoro

User stories

Creazione gruppo (8)

Come utente admin che si trova nella tab Gruppi, l'utente deve poter creare un gruppo di utenti, inserendo il nome del gruppo, questo consente di gestire più utenti contemporaneamente, semplificando l'assegnazione e la gestione delle attività per gruppi specifici di utenti nel sistema.

TASK:

- aggiungere bottone "aggiungi gruppo".
- aggiungere input per nome del gruppo.
- implementare backend per la creazione del gruppo.

Eliminazione gruppo (2)

Come utente admin che si trova nella tab Gruppi, l'utente deve poter eliminare un gruppo di utenti, cliccando il pulsante con l'icona di un cestino, questo consente di gestire i gruppi eliminando quelli obsoleti.

TASK:

- Aggiungere bottone elimina gruppo con icona
- Implementa backend per eliminazione del gruppo

Aggiunta utente a un gruppo (2)

Come amministratore nella tab "Gruppi", l'utente deve poter avere la possibilità di assegnare un utente a un gruppo di cui non fa parte, ciò consente di gestire in modo flessibile l'appartenenza degli utenti ai gruppi.

TASK:

- Aggiungere bottone aggiungi al gruppo
- Aggiungere select con membri del workspace

Rimozione utente da un gruppo(2)

Come utente amministratore nella tab "Gruppi", l'utente deve poter avere la possibilità di rimuovere un utente da un gruppo specifico. Questa funzionalità consente di gestire in modo flessibile l'appartenenza degli utenti ai gruppi, permettendo di rimuovere membri quando necessario, garantendo una gestione precisa e aggiornata delle risorse di gruppo.

TASK:

- Aggiungere bottone per eliminare dal gruppo sulla riga del componente
- Implementare logica di query nel backend

Modifica nome di un gruppo(2)

Come utente admin che si trova nella tab Gruppi l'utente deve poter modificare il nome di un gruppo, questa funzionalità consente di gestire in modo flessibile e aggiornare i nomi dei gruppi per riflettere eventuali cambiamenti organizzativi o di denominazione all'interno del sistema.

TASK:

- Aggiungere bottone modifica nome gruppo
- Implementare logica di query nel backend

Gestione manager gruppo(3)

Come utente admin che si trova nella tab Gruppi l'utente deve poter impostare un utente come un manager del gruppo o rimuovergli questa posizione, questa funzionalità consente di gestire l'autorità e le responsabilità all'interno del gruppo, garantendo un'organizzazione chiara e efficace delle dinamiche di gruppo.

TASK:

- Aggiungere switch per impostare se manager o meno
- Implementare logica di query nel backend

Modifica Assegnazione Gruppo Progetto(5)

Come utente amministratore nella sezione "Modifica Progetti", l'utente deve poter avere la possibilità sia di assegnare un gruppo specifico al progetto corrente, sia di rimuovere un gruppo già assegnato. Questa funzionalità consente di gestire in modo flessibile l'assegnazione di gruppi ai progetti, garantendo che solo i gruppi pertinenti siano associati ai progetti specifici, e permettendo di modificare queste assegnazioni secondo le necessità del progetto e delle risorse disponibili.

TASK:

- Verificare che si possa aggiungere i gruppi al select già presente per l'aggiunta di utenti:
 - in caso positivo aggiungere i nomi dei gruppi come opzioni per il select

- in caso negativo implementare select con opzioni i nome dei gruppi e bottone per confermare l'aggiunta
- Implementare nuova sezione “groups” nella tabella già presente o implementare nuova tabella completamente.

Tab Gruppi(8)

Come utente autenticato che si trova nella sezione “Team” l'utente deve poter vedere e cliccare sulla tab gruppi.

TASK:

- Implementare componente tab Gruppi
- Implementare componente tab Members che contiene quello che c'era nella pagina Team
- Implementare componente pagina che contenga le tab “Members” e “Gruppi”
- Creare collezione “Groups”

Visualizzazione lista gruppi(3)

Come utente autenticato che si trova nella tab gruppi, l'utente deve poter visualizzare la lista dei gruppi del corrente workspace e per ogni gruppo deve poter visualizzare il nome completo di ogni membro, in modo da poter facilmente identificare i gruppi a cui appartiene e identificare in modo chiaro e completo i suoi componenti.

TASK:

- Implementare tabella lista gruppi
- Aggiungere colonna “componenti” nella tabella gruppi
- Implementare chiamata al backend per richiedere i dati

Visualizzazione dei progetti(3)

Come utente autenticato che si trova nella sezione progetti e attività, l'utente deve poter visualizzare i progetti ai quali è assegnato tramite un gruppo, in modo che non debba essere assegnato direttamente a ogni progetto.

TASK:

- Implementare chiamata al backend per richiedere i dati
- Aggiornare i dati nella tabella projects con i progetti assegnati tramite gruppo

Visualizzazione attività membri dei gruppi di cui sono manager (8)

Come Manager di un gruppo nella sezione Reports, l'utente deve poter visualizzare le attività di tutti gli utenti all'interno dei propri gruppi, in modo da poter monitorare le prestazioni complessive del gruppo e apportare eventuali modifiche necessarie.

TASK:

- Aggiungere filtro user nella pagina reports se utente è manager di qualche gruppo del corrente workspace
- Aggiungere filtro team nella pagina reports se utente è manager di qualche gruppo del corrente workspace
- Implementare logica di query nel backend

4.1 Progettazione

4.1.1 Database

Il primo problema affrontato durante la progettazione è stato quello di decidere come implementare i gruppi a livello di database. Per l'implementazione dei gruppi a livello del database ho creato una nuova collezione chiamata `groups` e per rappresentare la relazione tra `users` e `groups` c'erano 2 strade:

1. Includere "groups" nella collezione `users`. In questa opzione, ogni documento nella collezione "users" conterrà un campo "gruppi" che è un array di ID di gruppi a cui appartiene l'utente.

```
{
  _id: ObjectId,
  //campi già esistenti
  groups: [groupId1, groupId2, ...]
}
```

Vantaggi:

- Semplicità: Un singolo campo nell'oggetto utente rende l'implementazione più semplice e può essere più veloce per le query relative agli utenti.
- Facilità di gestione: l'aggiornamento del gruppo di un utente coinvolge solo una modifica nel suo documento utente.

Svantaggi:

- Dimensioni dei documenti: se un utente è membro di molti gruppi, il campo "gruppi" all'interno del documento utente potrebbe diventare molto grande, portando a documenti di dimensioni notevoli. Ciò può rallentare le operazioni di lettura/scrittura dei documenti e richiedere più risorse di sistema.
- Complessità delle Query: le query che coinvolgono gli utenti e i gruppi potrebbero diventare complesse, specialmente quando si devono eseguire operazioni di ricerca, aggiornamento o eliminazione basate su gruppi specifici. Questo può richiedere query più intricate e quindi più lente.

2. Includere "users" nella collezione `groups`.

In questa opzione, ogni documento nella collezione "groups" conterrà un campo "users" che è un array di ID di utenti che fanno parte del gruppo.

```
{
  _id: ObjectId,
  groupName: String,
  users: [userId1, userId2, ...]
}
```

Vantaggi:

- Flessibilità nelle query: Le query che coinvolgono i gruppi e gli utenti possono essere più efficienti in quanto si basano su collezioni separate, specialmente quando si devono gestire grandi quantità di dati.

Svantaggi:

- Complessità nella Gestione delle Relazioni: Gestire le relazioni tra gli utenti e i gruppi richiede un po' più di complessità nella logica del tuo sistema. Devi garantire che le associazioni tra utenti e gruppi siano gestite in modo coerente e preciso.
- Possibili Incongruenze dei Dati: Se non gestite attentamente, potrebbero verificarsi inconsistenze nei dati, come utenti che sono nel campo "users" di un gruppo ma non hanno un riferimento corrispondente nel documento utente. Ciò richiede una gestione rigorosa dell'integrità referenziale.

Esempio di come è stato implementato Workspaces. Per l'implementazione di Workspaces è stato adottato il metodo di includere in "users" un array di oggetti con la seguente struttura:

```
{
  "_id": id univoco
  "id": id del workspace a cui ci sta riferendo
  "role": indica se l'utente è un amministratore o meno
}
```

Seguendo questo esempio possiamo implementare Groups in questo modo:

```
{
  "_id: id univoco"
  "id": id del gruppo a cui ci sta riferendo
  "role": indica se l'utente è un manager o meno
}
```

Alla fine discutendo con l'azienda siamo arrivati alla conclusione che la seconda opzione sarebbe stata la migliore per l'implementazione.

```
{
  "_id: id univoco"
  "name: nome del gruppo"
  "users: array di Id degli utenti del gruppo"
  "workspace:id del workspace a cui il gruppo appartiene"
  "createdAt:data e orario in cui il gruppo è stato creato"
  "updatedAt:data e orario in cui il gruppo è stato aggiornato l'ultima volta"
}
```

4.1.2 Struttura dei file

Client

Il progetto presenta questa struttura per la parte del client:

```
src
├── assets
├── components
├── i18n
├── pages
├── services
├── theme
├── utils
└── index.tsx
```

- assets: contiene tutti i file immagine in forma vettoriale e tutti i font utili per la webapp.
- components: cartella contenente i componenti che vengono riutilizzati più spesso.
- i18n: contiene i file utili per la internazionalizzazione del sito e per ora supporta l'italiano e l'inglese.
- pages: cartella principale, contenente i file che compongono ogni pagina, e per la quale esiste una cartella propria che è strutturata come segue:

```
Groups
├── Edit
├── Groups.config.tsx
├── Groups.style.tsx
└── Groups.tsx
```

Group.config.tsx è il codice responsabile per la configurazione dei componenti contenuti nella pagina ad esempio per i dati all'interno di una *Select* la funzione è:

```
1   export const select: any = (t: any, data: any, icons: any) =>
2   {
3   const result = [
4   {
5     name: "cell2",
6     tag: "row",
7     customRow: CustomRow,
8     fields: [
9     {
10      name: "user",
11      label: t("Wavelop.application.groupForm.userLabel"),
12      helperText: data.users.length
13      ? t("Wavelop.application.groupForm.userPlaceholder")
14      : t("Wavelop.application.groupForm.noUserPlaceholder"),
15      type: ["icon", icons.ClientIcon],
16      attributes: {
17        class: "large"
18      },
19      tag: Select,
```

```
19         options: data.users
20     }
21 ]
22 }
23
24 ];
25
26     return result;
27 };
28
```

Listing 4.1 codice per select all'interno della pagina gruppi

Group.style.tsx invece è il codice responsabile per utilizzare la libreria *styled-components*, per stilizzare i componenti React utilizzando dei tagged template literals. Consente di scrivere CSS nei file JavaScript definendo componenti stilizzati, che sono componenti React regolari con stili ad essi allegati.

```
1  import styled from "styled-components";
2  export const EditStyled = styled.div`
3      padding-top: ${({ theme }): string => theme.spacing.unit(3)
4  `;
5  export const LastRow = styled.div`
6      position: relative;
7      float: right;
8      margin-top: ${({ theme }): string => theme.spacing.unit(25)
9  `;
10 export const Buttons = styled.div`
11     text-align: right;
12     margin-top: ${({ theme }): string => theme.spacing.unit(0.5)
13 `;
14 export const UserSelect = styled.div`
15     margin-top: ${({ theme }): string => theme.spacing.unit(5.0)};
16 `;
17 export const GroupMembersTable = styled.div`
18     margin-top: ${({ theme }): string => theme.spacing.unit(5.0)};
19 `;
20
```

Listing 4.2 codice per lo stile della pagina dei gruppi

Groups.tsx. è il file principale della pagina dove risiede la logica e il componente React principale.

Server

Per la parte di server invece la struttura è la seguente:

```
server
├── test
├── documentation
├── src
│   ├── controllers
│   ├── middlewares
│   ├── models
│   ├── repository
│   ├── services
│   └── index.ts
```

- test: cartella contenente tutti i test scritti come descritto a 4.3.2.
- documentation: contiene un file json che descrive la struttura dell'API, includendo la sua versione, il titolo, la descrizione e i modelli di dati (schemas) utilizzati all'interno dell'API secondo le specifiche *OpenAPI*. L'OpenAPI Specification definisce un modo standardizzato per descrivere la struttura di un'API RESTful, inclusi endpoint, metodi HTTP supportati, parametri richiesti, formati di dati accettati e restituiti, autenticazione e autorizzazione, e così via. Questo schema fornisce una documentazione dettagliata e strutturata delle API, che può essere utilizzata per generare automaticamente una documentazione interattiva.

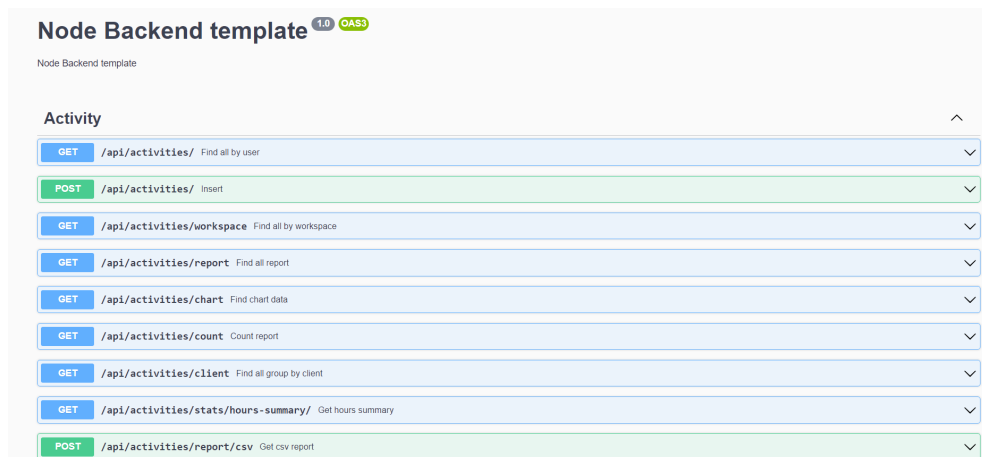


Figura 4.4: Documentazione interattiva delle API

- **controllers:** cartella contenente tutti i controller, i quali sono responsabili per la gestione delle richieste in arrivo, interpretare i parametri di richiesta, per poi richiamare i servizi che performano l'operazione richiesta e che inviano una risposta HTTP al client.
- **middlewares:** i middleware sono componenti software che si collocano tra la richiesta HTTP in arrivo e la risposta HTTP in uscita. Questi intermediari eseguono operazioni di elaborazione, manipolazione o filtraggio delle richieste e delle risposte. I middleware implementati nel backend sono per:
 - autenticazione;
 - autorizzazione;
 - gestire gli errori;
 - verificare che gli id passati siano validi;
 - verificare che il workspace passato esista.
- **models:** contiene le interfacce che definiscono come sono gli oggetti presenti nel database;
- **repository:** contiene i file repository, i quali forniscono uno strato di astrazione per l'accesso e la manipolazione dei dati. Essi racchiudono la logica per interagire con il livello di memorizzazione dei dati (ad esempio, database, API, sistemi di file) e forniscono un'interfaccia pulita per il resto dell'applicazione per eseguire operazioni CRUD (Create, Read, Update, Delete) sulle entità di dati. I repository aiutano a separare la logica di business dell'applicazione dai dettagli dell'implementazione della memorizzazione dei dati, facilitando il passaggio tra diverse tecnologie di memorizzazione o fonti di dati fittizie per scopi di test.
- **services:** I servizi racchiudono la logica di business dell'applicazione e eseguono operazioni specifiche richieste dai controller. Si occupano di attività come la validazione, la manipolazione e il recupero dei dati, oltre ad altre operazioni necessarie per soddisfare i requisiti dell'applicazione. I servizi favoriscono la riutilizzabilità e la manutenibilità centralizzando le funzionalità comuni e separandole dai controller.
- **index.ts:** è il file principale che fa partire il backend.

4.1.3 Tipi di utente

All'interno di ogni workspace nella webapp Timesheet ci sono diversi profili di utenti che possono interagirvi, ed ognuno di questo tipo di utenti ha delle esigenze diverse che verranno elencate di seguito.

- Utenti amministratori: questo tipo di utenti ha libero controllo su ogni funzione dell'applicazione, hanno all'interno della propria sezione report non solo le loro attività ma di tutti gli utenti registrati al corrente workspace e soprattutto per la parte implementata durante il mio progetto, solo gli utenti amministratori possono creare dei gruppi all'interno del workspace dove sono amministratori. Ho poi dato la possibilità agli amministratori di rendere un utente all'interno di un gruppo manager e anche di rendere un utente project manager.
- Utenti standard: questo tipo di utenti hanno solo la possibilità di immettere le proprie attività e quindi poi visualizzare i propri report.
- Project manager: utenti che sono project manager possono visualizzare le attività di tutti gli utenti partecipanti al progetto, sia che vi siano assegnati singolarmente o che un gruppo a cui appartengono sia assegnato ad esso.
- Group: come per i project manager questi utenti possono visualizzare le attività di tutti gli utenti appartenenti al gruppo di cui sono manager.

4.1.4 Pagina di modifica di un progetto

All'interno di questa pagina c'è voluto uno studio di quale fosse il modo migliore per implementare la possibilità di assegnare un gruppo al progetto. Come prima cosa ho pensato di aggiungere una nuova sezione che replicasse la sezione per l'assegnazione di un singolo utente, cioè avere una select è una tabella solo per i gruppi, ma questo avrebbe generato una pagina eccessivamente lunga in verticale. Quindi dopo averne discusso con il tutor, la soluzione migliore che abbiamo trovato era quella di implementare una select che includesse sia gli utenti, sia i gruppi.

4.1.5 Report delle attività

Con l'introduzione dei gruppi e le figure di group manager e project manager ho dovuto rivisitare la logica dietro alla visualizzazione del report delle attività. Un utente se è admin vede le attività di tutti. Se un utente è project manager deve vedere tutte le attività di tutti gli utenti sui progetti che gestisce. Se un utente è group manager, oltre che alle sue, deve vedere tutte le attività di tutte le persone sui progetti a cui sono associati i gruppi di cui è manager

4.2 Implementazione

Qui verrà descritto come è stato implementato e i vari workflow per le varie nuove funzionalità.

4.2.1 Creazione di un gruppo

Per la creazione del gruppo, invece di creare un nuovo link nella navbar, ho deciso insieme all'azienda che sarebbe stato più logico incorporarlo nella sezione Team, perchè si tratta sempre di gestire le persone all'interno del workspace. Ma avere tutto in una pagina risultava troppo ingombrante e quindi su consiglio del mio tutore aziendale ho utilizzato la sidebar per implementare un bottone che permette di passare dalla pagina dove vengono visualizzati singolarmente le persone del workspace alla pagina di visualizzazione dei gruppi.

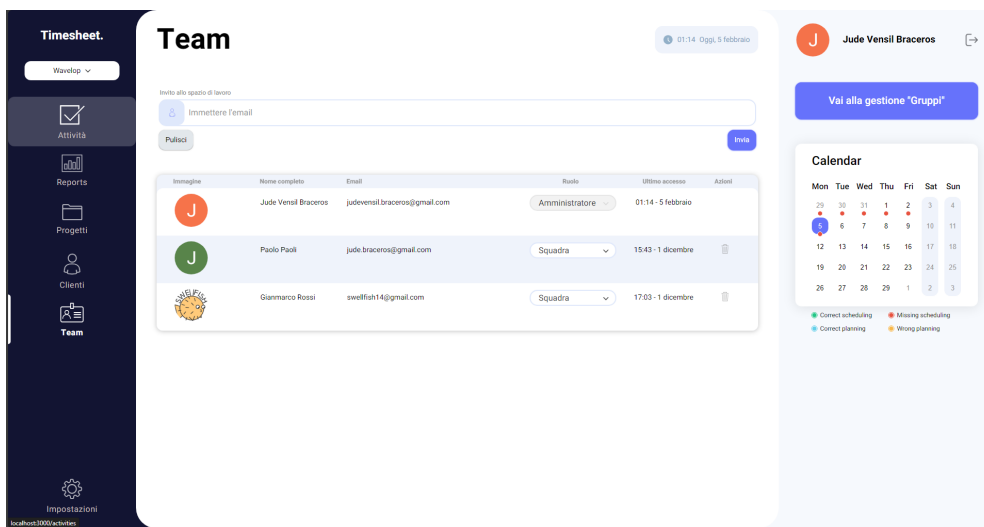


Figura 4.5: Pagina utenti del workspace

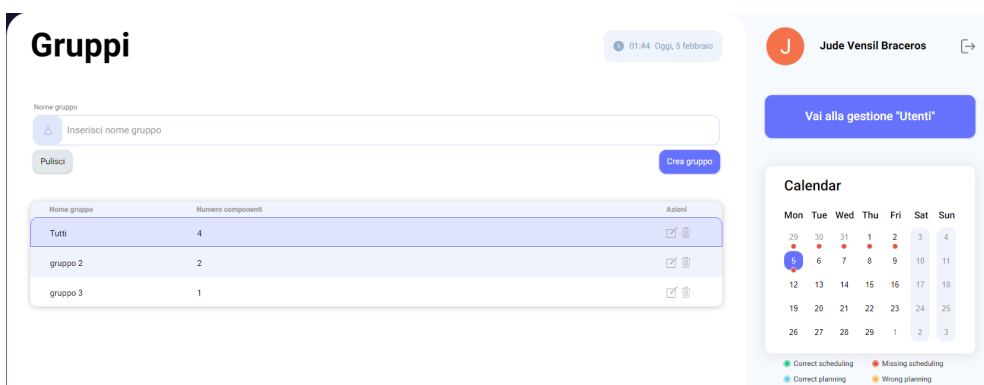
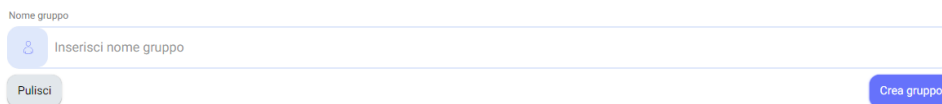


Figura 4.6: Pagina gruppi del workspace



Nome gruppo

Inserisci nome gruppo

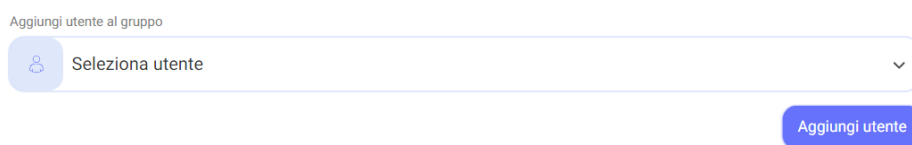
Pulisci

Crea gruppo

Figura 4.7: Form per la creazione di un gruppo

Per creare un gruppo bisogna immettere il nome che si vuole assegnare nell'input mostrato a 4.7 e poi premere crea gruppo. Una volta creato un gruppo si potrà modificare il nome del gruppo e aggiungere o rimuovere utenti al gruppo.

4.2.2 Gestione degli utenti all'interno di un gruppo



Aggiungi utente al gruppo

Seleziona utente

Aggiungi utente

Figura 4.8: Form per l'aggiunta di un utente a un gruppo




Nome completo	Email	Manager	Azioni
Jude Vensil Braceros	judevensil.braceros@gmail.com	<input type="checkbox"/>	
Paolo Paoli	jude.braceros@gmail.com	<input type="checkbox"/>	
Gianmarco Rossi	swellfish14@gmail.com	<input type="checkbox"/>	

Figura 4.9: Tabella degli utenti del gruppo

Per aggiungere un utente al gruppo bisogna selezionarlo dal select (4.8) e poi cliccare il tasto "Aggiungi utente". Una volta aggiunto l'utente verrà poi aggiunto alla tabella (4.9), dove verranno visualizzati:

- il nome completo;
- l'email;
- uno switch per impostare se l'utente sia un manager o meno;
- un pulsante per rimuovere l'utente.

4.2.3 Assegnazione di un gruppo a un progetto

Nella pagina di modifica di un progetto per implementare la possibilità di assegnarci un gruppo ho implementato un select come descritto a 4.1.4.

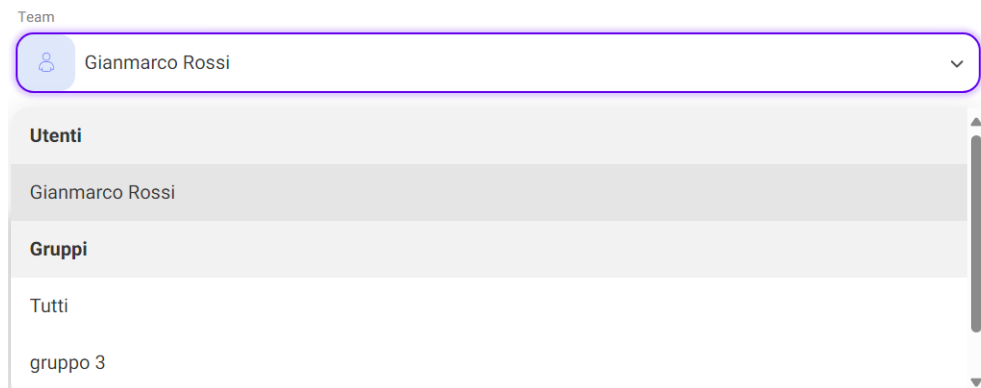


Figura 4.10: Select con utenti e gruppi




Nome completo	Email	Project manager	Azioni
Jude Vensil Braceros	judevensil.braceros@gmail.com	<input type="checkbox"/>	
Paolo Paoli	jude.braceros@gmail.com	<input checked="" type="checkbox"/>	
gruppo 2 Gruppo	Vedi gruppo		

Figura 4.11: Tabella utenti e gruppi assegnati al progetto

Per differenziare tra utenti e gruppi nella tabella(4.11) mi è stato consigliato di utilizzare un *chip*, un componente della libreria MUI [3].

4.2.4 Report delle attività

Oltre a implementare le query come descritte a 4.1.5 ho dovuto includere nel filtro utenti anche i gruppi 4.12.

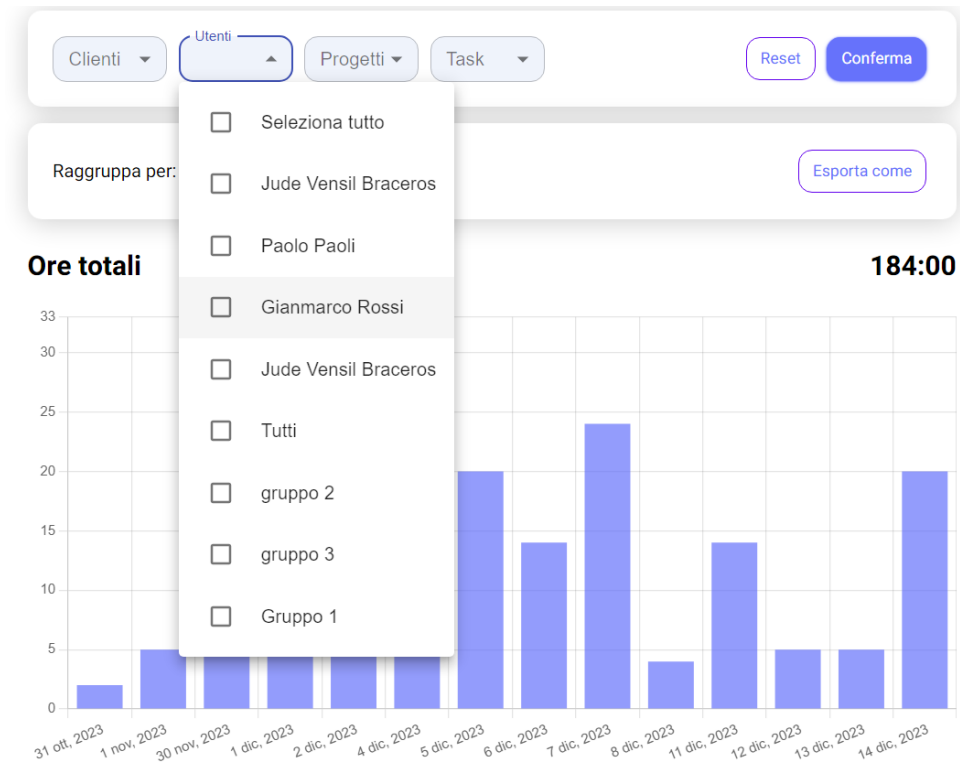


Figura 4.12: Filtro utenti e gruppi nella pagina report

4.2.5 Aggiunta di task

Una feature che ho sviluppato per famigliarizzarmi con le tecnologie usate è stata quella di permettere di aggiungere dei task ai progetti 4.13. Prima dello sviluppo di questa feature, le task erano predefinite e non potevano essere personalizzate o adattate alle esigenze specifiche del progetto.

Task

Aggiungi Task

Aggiungi

Task	Azioni
Analisi dei requisiti	
Documentazione	
Sviluppo back-end	
Analisi	
Sviluppo	
Deploy	
Ferie	
Malattia	
Permesso	

Figura 4.13: Form per l'aggiunta di task

4.3 Verifica e Validazione

Per la parte di verifica e validazione ho seguito due processi diversi per il codice che ho prodotto per il frontend e per il codice prodotto per il backend.

4.3.1 Frontend

Durante lo sviluppo del frontend ho applicato solo l'analisi statica del codice sorgente per mantenere il codice scritto da me qualitativamente in pari con quello scritto precedentemente da altri. Per fare questo ho utilizzato un linter, ovvero uno strumento che analizza il codice sorgente per rintracciare errori, inconsistenze stilistiche e imporre standard di codifica, best practices, in modo che il codice abbia una formattazione uniforme e che sia coerente nello stile di scrittura.

Il linter utilizzato tipicamente è ESLint [4] ed è stato configurato dall'azienda che precedentemente ha stabilito le regole per il linter e installato plugin personalizzati a supporto di ESLint. Grazie all'analisi statica ho potuto rilevare errori e problemi di codice mentre scrivevo, permettendomi di risolverli prima che diventassero problemi più gravi e questo ha reso lo sviluppo è più rapido ed efficiente.

4.3.2 Backend

Invece per lo sviluppo del backend oltre a impiegare l'uso dell'analisi statica del codice sorgente, ho applicato anche l'analisi dinamica attraverso la scrittura di test per verificare il comportamento del software rispetto ai requisiti specificati. Per i test ho utilizzato il framework di test Jest [4], che fornisce un'ampia gamma di funzionalità per scrivere, eseguire e analizzare test in modo efficace e efficiente.

```
1   it("Should return a valid group", async () => {
2     const res = await exec();
3     expect(res.status).toBe(200);
4     expect(res.body.status).toBe("ok");
5     expect(res.body.item).toMatchObject(group);
6   });
7
```

Listing 4.3 esempio di un test che controlla che l'operazione di POST vada a buon fine

Grazie ai test ho potuto constatare che le API che sviluppavo avevano un comportamento corretto e soprattutto se dovesse capitare a qualcun altro di lavorare al backend, i test garantirebbero che continui a funzionare correttamente e fornirebbe loro anche del codice che funge come documentazione perchè in essi descrivono come l'applicazione dovrebbe funzionare e quali siano i suoi requisiti.

4.3.3 Formatter

Durante lo sviluppo sia del frontend che del backend per mantenere il codice leggibile e quindi più manutenibile ho utilizzato *Prettier*, uno strumento che mi ha permesso di formattare il codice senza che io lo facessi manualmente. Oltre al fatto che supporta TypeScript ho deciso di usare Prettier perchè non c'è stato bisogno di configurarlo perchè avente già delle impostazioni predefinite.

Capitolo 5

Conclusioni

5.1 Consuntivo delle attività

La tabella seguente 5.1 confronta la durata pianificata all'inizio del progetto con la durata effettivamente impiegata per ciascuna attività chiave.

Tabella 5.1: Tempi previsti ed effettivi dello svolgimento del progetto.

<i>Descrizione attività</i>	<i>Durata prevista</i>	<i>Durata effettiva</i>
Comprensione sistema e obiettivi	20	20
Analisi dei requisiti	40	40
Progettazione	28	28
Studio e setup ambiente di sviluppo	8	4
Implementazione	138	172
Test e validazione	68	38
Documentazione	10	10
Totale	320	320

Durante l'esecuzione delle attività lo studio e il setup dell'ambiente di sviluppo si sono rivelati più semplici del previsto e le ore sono state riallocate per l'implementazione. Un'altra attività che è stata sovrastimata è stata quella di test e validazione che come descritto a 4.3 per la parte del frontend si è trattato solo di analisi statica, mentre per il backend, grazie al progetto del corso di *Ingegneria del software* la scrittura dei test è stato più rapido del previsto.

5.2 Raggiungimento degli obiettivi

La tabella 5.2 elenca gli obiettivi definiti in collaborazione con l'azienda, come indicato nella tabella 2.1, che sono stati conseguiti.

Tabella 5.2: Stato di completamento degli obiettivi.

<i>Obiettivo</i>	<i>Definizione</i>	<i>Stato</i>
O01	Apprendimento delle tecnologie di sviluppo come React e Node.js e per il versionamento del progetto come git.	Raggiunto
O02	Creazione di gruppi di utenti.	Raggiunto
O03	Visualizzazione dei gruppi di utenti con relativi dati.	Raggiunto
O04	Assegnazione e rimozione di un utente da un gruppo.	Raggiunto
O05	Visualizzazione dei progetti per assegnazione diretta o per gruppo.	Raggiunto
D01	Definizione di uno o più manager per gruppo.	Raggiunto
D02	Visualizzazione report con capability da Manager.	Raggiunto
F01	Implementare il multilingua.	Non raggiunto
F02	Visualizzare progetti abilitati.	Raggiunto
F03	Effettuare il refactoring dei form.	Non raggiunto

Gli unici obiettivi non rispettati sono due facoltativi, e non per una questione di tempo, visto che avevo concluso tutti requisiti obbligatori e desiderabili a una settimana dalla conclusione dello stage, ma l'azienda ha preferito che io lavorassi a dei bug fix e a pensare a come rendere la versione mobile della webapp responsive.

5.3 Sviluppi futuri

Il progetto a questo punto è pronto per essere utilizzato dal team interno ed esiste anche la possibilità di rilasciarlo al pubblico, ma vista la presenza di una notevole concorrenza, il prodotto potrebbe essere risultare solo ridondante nel mercato. Le lacune che il prodotto ancora presenta sono il fatto che la versione mobile non sia affatto responsive e quindi risulta inutilizzabile per un numero crescente di utenti che preferiscono accedere ai servizi tramite dispositivi mobili; Inoltre, abbiamo riscontrato che la libreria attualmente utilizzata per implementare i form presenta un grave bug che limita la loro funzionalità. In particolare, ho notato che non è possibile avere più di un form in una pagina, il che è diventato un problema durante l'aggiunta di nuove

funzionalità come l'aggiunta di task a un progetto. Questo è un problema critico che non può essere risolto semplicemente attraverso un refactor del codice esistente; è necessario esaminare attentamente le alternative e valutare l'implementazione di una nuova libreria.

5.4 Conoscenze acquisite

Dato che ho svolto il tirocinio in un'azienda molto piccola e ho lavorato in un ufficio open space ho potuto assimilare conoscenze utili per la mia crescita personale non solo discutendo con i miei colleghi ma anche semplicemente ascoltando passivamente quello che succedeva. Quindi in questa sezione oltre a elencare le varie conoscenze tecniche che ho appreso o approfondito, elencherò anche alcune delle *soft skills*_[G] che ho acquisito.

Sviluppo di una webapp con un fullstack JavaScript/TypeScript

Durante lo stage, ho avuto l'opportunità di approfondire significativamente la mia conoscenza di JavaScript. Sebbene durante il mio percorso universitario abbia avuto solo due esperienze di utilizzo di JavaScript, principalmente per semplici controlli sugli input dei form, e durante il corso "Ingegneria del software" abbia sviluppato una webapp utilizzando un'intera tecnologia stack simile a quello impiegato per questo progetto, devo ammettere che non ho avuto un'esperienza completa con tali tecnologie in precedenza. Durante lo stage, tuttavia, ho avuto l'opportunità di immergermi completamente in queste tecnologie. In particolare, ho acquisito una conoscenza approfondita di TypeScript, un linguaggio che non solo ho imparato ad utilizzare, ma che ho adottato come una parte essenziale del mio processo di sviluppo da qui in avanti. La comprensione di TypeScript mi ha permesso di scrivere codice più robusto, mantenibile e sicuro, migliorando notevolmente le mie capacità di sviluppo software e preparandomi per sfide più avanzate nel mondo professionale.

Virtualizzazione dei container

Grazie a Docker, anche se l'ho utilizzato in minima parte, ho scoperto la grande utilità dei container e la differenza con la virtualizzazione tradizionale. I container offrono un ambiente isolato e leggero per l'esecuzione di applicazioni, fornendo una soluzione efficace per la distribuzione di software in modo coerente e affidabile su diverse piattaforme e ambienti di esecuzione, e questo mi ha permesso di risparmiare tempo nel capire come installare tutte le dipendenze che servivano per l'avvio del database, visto che richiedeva anche che io capissi come Redis, un'altra tipo di database usato insieme a MongoDB.

Acquisito una migliore comprensione di React

Durante il mio stage, ho avuto l'opportunità di immergermi nel mondo di React e di approfondire significativamente la mia comprensione di questa libreria JavaScript per la creazione di interfacce utente. Sebbene avessi già avuto qualche esperienza limitata con React in passato, durante il corso universitario e in progetti personali, è stato durante lo stage che ho veramente potuto esplorare appieno le sue potenzialità e affinare le mie competenze nell'utilizzarle. Attraverso la progettazione e lo sviluppo di applicazioni web complesse, ho affrontato sfide pratiche che mi hanno permesso di comprendere a

fondo i concetti fondamentali di React, come il concetto di componenti, lo stato e le proprietà, il ciclo di vita dei componenti e la gestione degli eventi. Grazie a questa esperienza, sono ora in grado di sviluppare applicazioni più sofisticate e performanti utilizzando React e di adattarmi in modo più efficace alle esigenze del mondo dello sviluppo web moderno.

Internizzato la metodologia di sviluppo agile Durante lo stage, ho avuto l'opportunità di immergermi nella metodologia di sviluppo Agile e di applicarla in contesti reali. Collaborando con il team e partecipando alle attività quotidiane, ho interiorizzato i principi agili e ho imparato ad adattare il mio approccio lavorativo per soddisfare le esigenze del progetto. Attraverso sprint planning, daily stand-up, sprint review e retrospective, ho contribuito attivamente al processo di sviluppo, lavorando in modo collaborativo e iterativo per consegnare valore ai clienti in modo tempestivo. La metodologia Agile mi ha insegnato l'importanza della comunicazione chiara, della flessibilità e dell'auto-organizzazione, preparandomi per affrontare sfide complesse e adattarmi a un ambiente di lavoro dinamico.

5.5 Valutazione personale

Il mio stage è stato un'esperienza estremamente istruttiva e appagante, consentendomi di applicare le conoscenze accademiche acquisite durante il mio percorso di studi in modo pratico e concreto. Tuttavia, durante questo periodo ho anche riscontrato alcune discrepanze rispetto al mio percorso accademico. Ad esempio, ho notato che alcune metodologie o tecnologie utilizzate nell'ambiente lavorativo differivano dalle pratiche insegnate a scuola. Nonostante queste discrepanze, trovo che l'esperienza di stage sia estremamente preziosa, in quanto offre una visione diretta del mondo professionale, consentendo agli studenti di familiarizzare con le aspettative, le norme e le sfide che affrontano i professionisti del settore. Infine, lo stage mi ha personalmente permesso di avere un'opportunità lavorativa subito dopo la laurea, in un ambiente giovane e molto flessibile.

Glossario

DBMS Un Database Management System (DBMS) è un software che permette agli utenti di interagire con i database. Facilita la creazione, manutenzione, e uso dei database provvedendo un'interfaccia per gli utenti e applicazioni per memorizzare, recuperare, e gestire dati. .

JSON JavaScript Object Notation è un formato di dati leggero e basato su testo che viene comunemente utilizzato per rappresentare dati strutturati. [5].

NoSQL Il termine "NoSQL" fa riferimento a tipi di database non relazionali e questi database archiviano i dati in un formato diverso dalle tabelle relazionali. [6].

Product Backlog Una lista prioritizzata di tutte le funzionalità, miglioramenti e correzioni di bug desiderati per il prodotto..

Refactoring Processo in cui si ristrutturava il codice senza cambiare la propria funzionalità..

Scrum Scrum è un framework di project management agile che aiuta i team a strutturare e gestire il loro lavoro organizzandolo in cicli di sviluppo chiamati sprint.

soft skills Le "soft skills" sono un insieme di abilità personali, sociali e comportamentali che consentono a una persona di interagire efficacemente con gli altri e di affrontare situazioni diverse nella vita personale e professionale..

Sprint Lasso di tempo in cui il team di sviluppo si impegna per produrre un incremento potenziato e funzionante del prodotto.

Sprint backlog L'insieme di compiti e attività che il team si impegna a completare durante lo sprint.

SSO SSO è un metodo di autenticazione che consente agli utenti di accedere a più sistemi o applicazioni utilizzando le stesse credenziali di accesso. Invece di dover inserire le credenziali di accesso separatamente per ogni sistema, l'utente effettua l'accesso una sola volta e ottiene l'accesso automatico a tutte le risorse autorizzate senza dover reinserire le credenziali..

Stack tecnologico Insieme di tecnologie, software e strumenti che vengono utilizzati per creare e gestire una soluzione digitale, come un'applicazione web o mobile. [7].

Use cases Uno use-case è una descrizione di una specifica interazione tra un sistema (tipicamente un'applicazione software) e uno o più attori (utenti o altri sistemi) per raggiungere un obiettivo particolare. Esso delinea i passaggi o le azioni che gli attori compiono e descrive la risposta del sistema a tali azioni. I casi d'uso sono utilizzati nello sviluppo del software e nell'ingegneria dei sistemi per catturare e documentare i requisiti funzionali. .

User stories Le user stories sono una descrizione concisa e informale di una feature o funzionalità dalla prospettiva di un utente. Tipicamente usata nelle metodologie di sviluppo agile per raccogliere i requisiti e Per comunicarli efficacemente tra il team di sviluppo e gli stakeholders..

Riferimenti Bibliografici

Siti web consultati

- [1] «What are sprints in project management?» (), indirizzo: <https://www.atlassian.com/agile/scrum/sprints> (cit. a p. 3).
- [2] «Cos'è MERN.» (), indirizzo: <https://www.geeksforgeeks.org/mern-stack/> (cit. a p. 9).
- [3] «What's MUI.» (), indirizzo: <https://mui.com/> (cit. a p. 27).
- [4] «Eslint.» (), indirizzo: <https://eslint.org/> (cit. a p. 30).
- [5] «Che cos'è JSON.» (), indirizzo: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON> (cit. a p. 35).
- [6] «Che cos'è NoSQL.» (), indirizzo: <https://www.oracle.com/it/database/nosql/what-is-nosql/> (cit. a p. 35).
- [7] «Che cos'è uno stack tecnologico.» (), indirizzo: <https://www.economyup.it/innovazione/cose-il-tech-stack-e-perche-e-importante-per-le-aziende/> (cit. a p. 35).