

Documentazione progetto "Smart calendar for business"

Filippo Lazzari [matricola], Mauro Riva [1053644] e Marco Rodolfi [1040347]

1 Iterazione 0

1.1 Introduzione e utilizzo del sistema

L'obiettivo di questo progetto è quello di poter organizzare i turni di un'azienda, specialmente per quelli che devono gestire turni a rotazione che devono essere sempre presenti per il servizio clienti, come tutti i servizi di ristorazione, per esempio.

Questo è stato idealizzato come tre componenti distinte all'interno dell'applicativo:

1. Una parte lato server che gestirà i client tramite chiamate REST e farà girare l'algoritmo per ottimizzare la selezione dei turni.
2. Una parte web che gestisca la registrazione dell'azienda al servizio.
3. Un'applicativo per dispositivi Android che permetta agli utenti dell'azienda di gestire il proprio stato e richiedere malattie/straordinari.

1.1.1 Il server dell'applicativo

Questo deve poter gestire più di una singola azienda e verrà gestito direttamente da chi gestisce il servizio. Lo stesso quindi deve poter permettere la registrazione di più aziende sullo stesso sistema. Per permettere di gestire le decisioni dell'algoritmo, è richiesta un'interfaccia web che permetta la gestione da parte della stessa azienda.

1.1.2 L'applicativo Android

Per permettere una maggiore flessibilità di gestione da parte dei dipendenti che lavorano, abbiamo pensato di creare un'applicativo Android che permetta di gestire il proprio stato all'interno dell'azienda, richiedere malattie e verificare i propri straordinari. La comunicazione verso il server è quindi gestita tramite chiamate REST.

1.1.3 L'algoritmo

L'algoritmo deve saper gestire dinamicamente l'allocazione dei turni e poter gestire i turni scoperti verificando se persone di altre sedi della stessa azienda e che coprano ruoli identici possano sostituire le persone nella stessa, tenendo conto sia dell'eventuale distanza dalla sede in cui dovrebbero coprire il turno sia delle loro ore massime di straordinari imposti per limiti di legge.

1.2 Requisiti funzionali e casi d'uso

I casi d'uso pensati per il nostro software sono disegnati qua sotto:

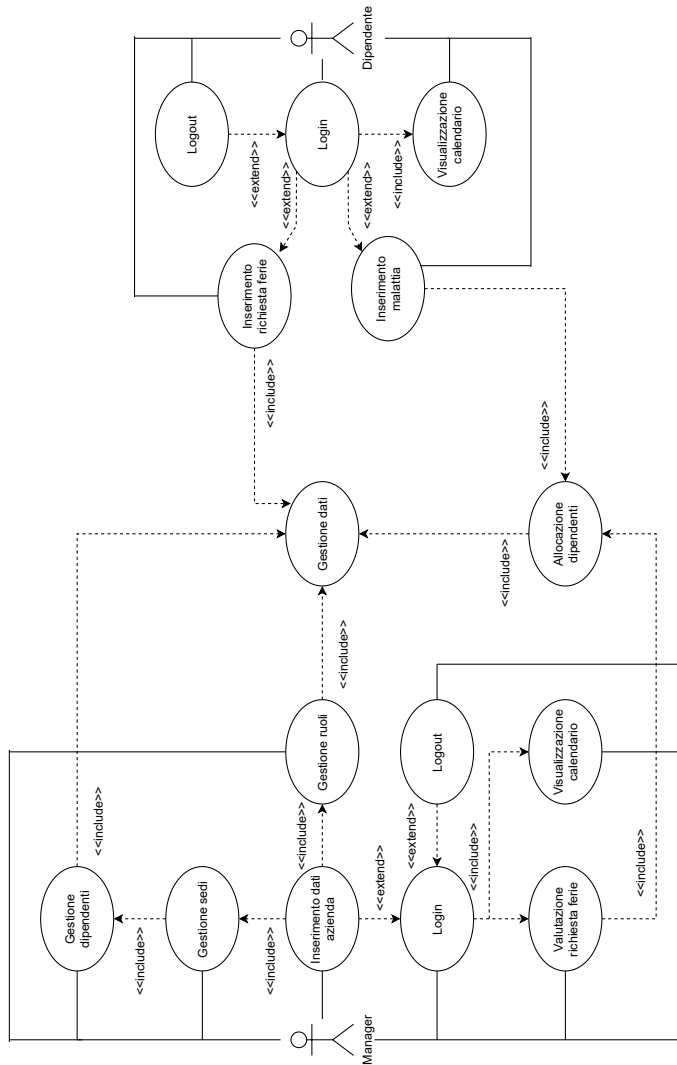


Figura 1: Casi d'uso pensati

Questa prima tabella rappresenta i casi d'uso iniziali che abbiamo ritenuto importanti da includere già all'inizio del nostro progetto:

IDENTIFICATIVO	NOME FUNZIONE
UC1	Inserimento dati azienda
UC2	Login
UC3	Logout
UC4	Gestione ruoli
UC5	Gestione sedi
UC6	Gestione dipendenti
UC7	Algoritmo di organizzazione dei dipendenti

Tabella 1: Alta priorità

Questa seconda tabella invece stava ad indicare cosa ci sarebbe piaciuto includere come idea ma non necessarie per la prima iterazione:

IDENTIFICATIVO	NOME FUNZIONE
UC8	Visualizzazione calendario
UC9	Login dipendenti
UC10	Inserimento malattia
UC11	Inserimento richiesta di ferie
UC12	Valutazione richiesta di ferie

Tabella 2: Media priorità

Infine queste sono casi d'uso interessanti da avere come aggiunta ma non necessariamente essenziali per l'applicativo stesso:

IDENTIFICATIVO	NOME FUNZIONE
UC13	Personalizzazione calendario
UC14	Visual. avanzata della situazione dei dip.
UC15	Limitazioni nell'inserimento delle ferie
UC16	Modifica credenziali di accesso

Tabella 3: Bassa priorità

1.3 Requisiti non funzionali

1.3.1 Usabilità

L'usabilità é stata considerata nella creazione dell'applicativo Android, permettendo una più facile gestione dei turni da parte degli utenti dell'azienda.

1.3.2 Manutenibilità

1.3.3 Efficienza

Algoritmo? Che altro?

1.4 Design Pattern

1.5 Toolchain utilizzati

NOME TOOLCHAIN	UTILIZZO
IntelliJ IDEA Ultimate	Utilizzato come IDE di sviluppo per backend
Android Studio	Utilizzato per lo sviluppo dell'applicativo Android
Git & Gitlab	Software e piattaforma per gestire la distribuzione del codice sorgente
Cocalc	Piattaforma utilizzata per la scrittura collaborativa della documentazione \LaTeX
Java 17	Linguaggio di programmazione usato per il server e l'applicativo Android
Draw.io	Piattaforma utilizzata per creare diagrammi UML
Spring Boot	Il framework utilizzato lato server per il suo funzionamento
Spring Data JPA	Interfaccia per l'astrazione della base di dati
JGraphT	Libreria per i grafi utilizzata per lo sviluppo dell'algoritmo
PostgreSQL	Il DB scelto per gestire la base di dati stessa

Tabella 4: Toolchain Utilizzati