3.3.4 Supportability

I requisiti di sopportabilità per il sistema EWMS si concentrano su:

* **Manutenibilità:** La manutenzione sarà esclusivamente dal team di sviluppo originale. Per facilitare questo processo, il codice sorgente dovrà aderire rigorosamente agli standard di codifica e alla guida di stile concordati internamente al team. Tutta la documentazione tecnica (es. commenti nel codice, file README) dovrà essere mantenuta aggiornata per facilitare la diagnosi e la correzione di bug.
* **Evolvibilità:** La struttura dati e i componenti dell'interfaccia utente relativi ai "Task" devono essere implementati in modo modulare. Questo garantirà che la futura introduzione di un'entità "Progetto", intesa come meccanismo di raggruppamento dei task, possa essere integrata con modifiche minime.
* **Installabilità e Portabilità:** Non sono previsti requisiti di portabilità. Il sistema è progettato per essere eseguito esclusivamente sull'ambiente hardware e software di destinazione definito (come specificato nella sezione 3.3.5 Implementation) e non è richiesta la capacità di migrazione o installazione su piattaforme differenti.

3.3.5 Implementation

L'intero sistema EWMS sarà progettato seguendo un'architettura a tre livelli (three-tier) e aderirà al design pattern Model-View-Controller (MVC). Questo vincolo fondamentale definisce la struttura dell'intero software e guida le decisioni implementative.

Riguardo ai vincoli sulla piattaforma hardware, non ne esistono di specifici per il Livello Client (Presentation), se non la necessità di un dispositivo in grado di eseguire un browser web moderno.

/\* Il sistema dovrà semplicemente essere ospitato su un ambiente capace di eseguire lo stack tecnologico scelto per il backend e il DBMS. \*/

Per quanto riguarda i vincoli imposti dal team di manutenzione, coincidendo questo con il team di sviluppo, si stabiliscono requisiti auto-imposti per garantire la manutenibilità (coerentemente con la sez. 3.3.4).   
L'uso di un sistema di controllo di versione Git è obbligatorio per tracciare tutte le modifiche.

3.3.6 Interface

**Interfacce utente**

Il sistema fornirà un'interfaccia utente (UI) basata sul web. Questa UI sarà accessibile tramite i browser web moderni più comuni (es. Google Chrome, Mozilla Firefox, Microsoft Edge, Safari).

Interfacce Software

* Interfaccia database: il sistema si interfaccia con il database aziendale (RBMS) utilizzando lo standard JDBC (Java Database Connectivity) per leggere e scrivere tutti i dati relativi a account, task, notifiche di sistema.
* Interfaccia di autenticazione: l’autenticazione sarà gestita internamente dal sistema.

Interfaccia di notifica:

(Il sistema implementerà un'interfaccia basata su **WebSocket** per permettere al server di inviare notifiche in tempo reale (es. 'nuovo task assegnato') ai client (browser) connessi. Questo garantirà l'aggiornamento immediato dell'interfaccia utente senza la necessità per l'utente di ricaricare la pagina.)

Interfaccia di rete:

L'applicazione web sarà accessibile esclusivamente tramite la rete intranet aziendale. L'accesso dalla rete internet pubblica non sarà consentito.

3.3.7 Packaging

L'applicazione sarà consegnata come un singolo file **JAR eseguibile** (EasyWorkManagement.jar), costruito utilizzando [Nome del tuo build tool, es. Apache Maven]. Il file JAR includerà un server web Tomcat embedded, rendendo l'applicazione autonoma.

Il server di destinazione deve avere installato un **Java Runtime Environment (JRE) versione 17** o superiore.

La connessione al database aziendale e la porta del server saranno specificate in un file di proprietà esterno (application.properties) posizionato nella stessa directory del file JAR.

L'applicazione verrà avviata eseguendo il comando (java -jar EasyWorkManagement.jar) sul server.

3.3.8 Legal

**Privacy**

Il sistema tratterà dati personali dei dipendenti dell’azienda (nome, cognome, e-mail, ruolo). Le password non verranno memorizzate in chiaro ma saranno salvate utilizzando un opportuno algoritmo di cifratura (AES, scrypt, bcrypt). L'accesso alle informazioni sarà profilato in base al ruolo (Supervisore, Dipendente).

**Licenze**

Il software è sviluppato in linguaggio Java e si basa sul framework Spring Boot (licenza Apache 2.0). La gestione delle dipendenze di terze parti è affidata ad **Apache Maven**; tutte le librerie incluse nel progetto saranno verificate per essere conformi a licenze open-source permissive.

**Copyright**

Il codice sorgente prodotto per questo progetto è sviluppato a fini didattici per il corso di Ingegneria del Software presso l’Università degli Studi di Salerno.