Immagine che contiene testo, Carattere, logo, design

Descrizione generata automaticamente

**Pagina riservata alla copertina del documento.**

1. **INTRODUZIONE**

**Scopo del Sistema**

Il sistema Focus Project si pone l'obiettivo di fornire alle aziende uno strumento avanzato e versatile per ottimizzare la gestione delle risorse umane e dei progetti aziendali. Lo scopo è semplificare dinamiche complesse, riducendo i tempi operativi e migliorando l'efficienza complessiva dell'organizzazione.

**Obiettivi di design**

* Performance Ottimali

Il sistema deve garantire tempi di risposta rapidi per operazioni critiche come l'assegnazione di progetti e la gestione delle risorse umane, con un obiettivo di tempo di risposta inferiore a 2s. Mentre per operazioni di generazione di statistiche, tempi inferiori a 30s.

* Sicurezza

Ogni password associata a un account deve rispettare standard di sicurezza, includendo almeno 8 caratteri, una combinazione di almeno un carattere speciale e una lettera maiuscola. Le password devono essere crittografate.

* Facilità di Manutenzione

Il codice sorgente e la struttura del sistema devono essere organizzati in modo da agevolare la manutenzione e gli aggiornamenti futuri.

* Interfaccia Utente Intuitiva

L'interfaccia utente deve essere progettata con un approccio intuitivo, consentendo agli utenti di interagire con il sistema senza la necessità di istruzioni complesse.

* Compatibilità Cross-Platform

Il software deve essere progettato per essere compatibile con diverse piattaforme e dispositivi.

**Definizioni, acronimi e abbreviazioni**

|  |  |  |
| --- | --- | --- |
| **Acronimo** | **Abbreviazione** | **Definizione** |
| UI | User Interface | L’area visuale con cui gli utenti interagiscono per utilizzare il software. |
| DMBS | Database Management System | Un software che gestisce l'organizzazione, lo storage e l'accesso ai dati nel database. |
| HTTPS | HyperText Transfer Protocol over Secure Socket Layer | Un protocollo di comunicazione sicuro utilizzato su reti informatiche. |
| API | Application Programming Interface | Un insieme di regole e strumenti che consente a software diversi di comunicare tra loro. |
| MVC | Model View Controller | Un pattern architetturale che separa il sistema in tre componenti principali: Modello (logica e dati), Vista (presentazione) e Controller (gestione degli input). |
| JSP | Java Server Pages | Una tecnologia Java utilizzata per creare pagine web dinamiche. |
| JDBC | Java Database Connectivity | Una API Java che consente l’accesso e la gestione dei database relazionali. |
| URL | Uniform Resource Locator | L'indirizzo web che identifica una risorsa su Internet. |
| RAD | Requirements Analaysis Document | documento che raccoglie e analizza i requisiti di un progetto software |

**Riferimenti**

In questo testo sono contenuti riferimenti ad altri documenti:

Requirements Analysis Document (RAD): sono descritte le funzionalità individuate in fase di analisi dei requisiti.

Persistent data: è descritta la gestione dei dati persistenti.

**Panoramica**

Il presente documento si propone di fornire una visione dettagliata dell'architettura software, mediante:

* Decomposizione del sistema in sottosistemi;
* Mapping Hardware/Software;
* Gestione dei dati persistenti;
* Controllo degli accessi e della sicurezza;
* Controllo del flusso globale;
* Boundary condition;
* infine, verranno illustrati i servizi forniti da ogni singolo sottosistema.

1. **ARCHITETTURA SOFTWARE PROPOSTA**

**Overview**

Il sistema Focus Project sarà strutturato seguendo il modello architetturale Model/View/Controller (MVC), suddividendo i sottosistemi in tre categorie principali:

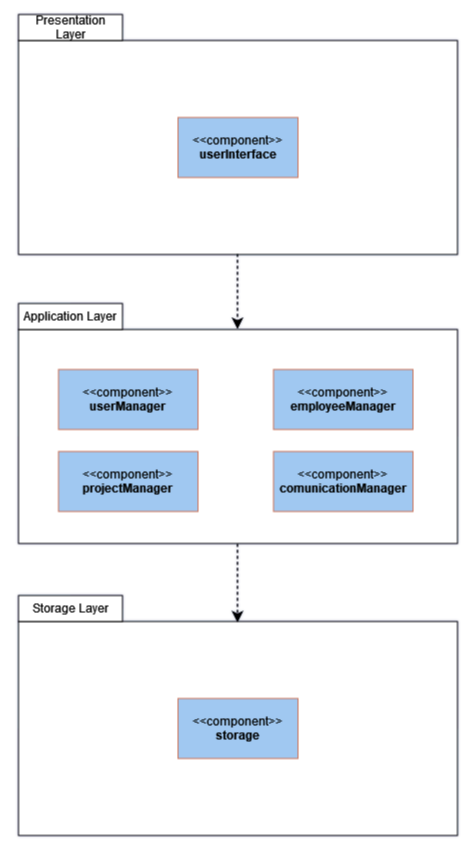
**Model (Storage)**: qui risiederanno le logiche di business, la gestione dei dati e le operazioni sul database necessarie per il corretto funzionamento del sistema.

**View (Presentation Layer)**: i sottosistemi di questa categoria avranno il compito di presentare all'utente i dati provenienti dal dominio applicativo. Saranno responsabili della creazione dell'interfaccia utente e di mostrare in modo efficace le informazioni necessarie agli utenti del sistema.

**Controller (Application Layer)**: questi sottosistemi gestiranno la logica applicativa. Saranno responsabili di ricevere e gestire le richieste in arrivo dall'utente attraverso le interfacce utente create dalla View. In base a queste richieste, il Controller interagirà con il Model e la View per garantire un corretto flusso di informazioni e operazioni.

Il sistema Focus Project sarà implementato utilizzando Servlet e pagine JSP, che favoriranno l'adozione del modello MVC e garantiranno un'organizzazione chiara e modulare dei sottosistemi.

**Decomposizione in sottosistemi**



|  |  |
| --- | --- |
| **Component** | **Descrizione** |
| userInterface | gestisce l'interfaccia utente, fornendo un'esperienza interattiva e intuitiva per gli utenti. |
| userManager | Gestisce le operazioni di registrazione, accesso e gestione degli account. |
| employeeManager | Gestisce le operazioni relative alla gestione dei dipendenti. Si occupa di aggiungere e rimuovere dipendenti, nonché delle operazioni legate ai turni. |
| projectManager | Gestisce tutte le attività legate alla gestione dei progetti. Include l'aggiunta di nuovi progetti, l'assegnazione di responsabili e subordinati, nonché la gestione delle fasi di un progetto. |
| comunicationManager | Gestisce le operazioni di comunicazione. Si occupa di inserire news e gestire le richieste di permesso da parte dei dipendenti. |

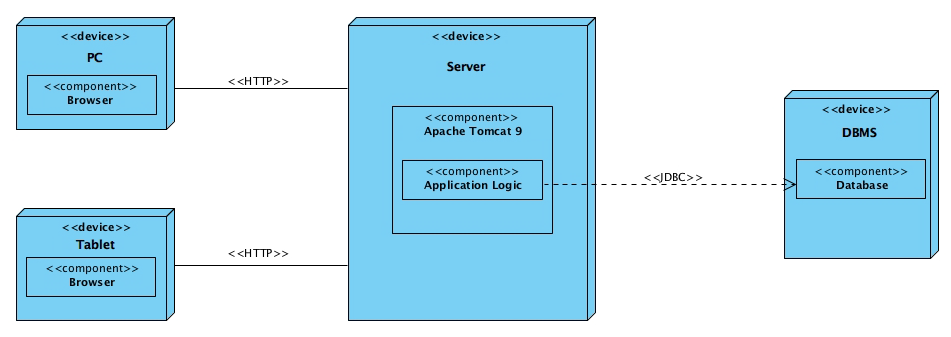
**Hardware/software mapping**

Focus Project adotta un'architettura Client/Server per gestire le interazioni tra gli utenti e il sistema.

Sul lato server, sarà utilizzato Apache Tomcat 9.0 come Web Server, impiegando Java Servlet per gestire la logica applicativa e JSP (Java Servlet Pages) per implementare l'interfaccia utente.

Dal lato client, l'interazione avverrà attraverso il web browser del cliente, facilitando la comunicazione diretta tra l'utente e il sistema. Le componenti interagiranno mediante richieste al server e risposte al client tramite il protocollo HTTP.

Per la gestione dei dati persistenti, il server utilizzerà l’API JDBC per comunicare con il sistema di gestione del database.



**Gestione dei dati persistenti**

Per la gestione dei dati persistenti si sceglie il DBMS MySQL. Fare riferimento al documento “Persistent\_data”.

**Controllo degli accessi e della sicurezza**

Per garantire un ambiente sicuro e prevenire accessi non autorizzati, sarà implementato un efficace controllo degli accessi basato sull'autenticazione mediante inserimento di credenziali, comprese di username e password. Al fine di assicurare un elevato grado di protezione, le informazioni sensibili degli utenti saranno crittografate prima di essere memorizzate nel database.

La tabella di controllo degli accessi, conforme al diagramma degli attori presente nel RAD, delineerà in dettaglio le varie funzionalità a cui gli attori avranno accesso attraverso i diversi sottosistemi.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **Utente** | **Account** | **Progetto** | **Turno** | **Task** | **Permesso** | **Comunicazione** |
| Subordinato | Accesso  Disconnessione | Visualizza | Visualizza | Visualizza  Completa | Richiede | Visualizza |
| Responsabile | Accesso  Disconnessione | Visualizza  Conclude  Modifica team  Crea avviso | Visualizza  Assegna  Cancella | Visualizza  Assegna  Modifica  Cancella | Richiede  Gestisce | Visualizza |
| Dirigente | Registrazione  Accesso  Disconnessione | Crea  Visualizza  Cancella  Modifica  Crea avviso | Visualizza  Assegna  Cancella | - | Gestisce | Visualizza  Crea |

**Controllo del flusso globale**

…

**Boundary conditions**

…

1. **SERVIZI DEI SOTTOSISTEMI**