

Schema del Progetto di Software di Gestione

Questo documento descrive i requisiti tecnici e il progetto di base per un software di gestione per una società B2B che si occupa di parti auto. Il software utilizzerà React, TypeScript e Bootstrap per il frontend, Node.js/Express.js per il backend, MongoDB per il database e Python per il motore statistico.

Ruoli degli Utenti e Permessi

- ****Amministratore:****

- Accesso completo a tutte le funzionalità.
- Gestione di utenti, prodotti e promozioni.
- Visualizzazione e analisi di tutte le statistiche e gli avvisi.
- Visualizzazione della progressione e delle promozioni attive per tutti i clienti.

- ****Agente di Vendita:****

- Gestione delle promozioni.
- Visualizzazione delle statistiche dei clienti e degli avvisi per i clienti associati.
- Visualizzazione delle promozioni relative ai clienti associati.

- ****Cliente:****

- Visualizzazione della propria cronologia degli acquisti e delle statistiche di spesa.
- Visualizzazione delle promozioni disponibili.
- Visualizzazione della progressione verso le promozioni.

Funzionalità Principali

- **Gestione Clienti:**

- Tracciare le abitudini di spesa e la cronologia degli acquisti.
- Generare avvisi per clienti inattivi (visualizzabili solo da amministratori e agenti).
- Assegnare i clienti agli agenti (un cliente per agente, più clienti per agente).

- **Gestione Promozioni:**

- Creare, inviare e gestire promozioni via SMS/Email.
- Tracciare la progressione verso le promozioni in base alla spesa.

- **Gestione Ordini:**

- Importare ordini da file CSV.
- Futuro: Integrare con l'API di e-commerce per ottenere gli ordini direttamente.

- **Statistiche e Avvisi:**

- Generare e visualizzare statistiche sul comportamento dei clienti e sulle vendite.
- Inviare avvisi per cambiamenti significativi nel comportamento dei clienti.

Architettura di Alto Livello

1. **Frontend:**

- **Framework:** React.js
- **Linguaggio:** TypeScript
- **Stile:** Bootstrap

2. **Backend:**

- **Framework:** Node.js/Express.js

- **Linguaggio:** TypeScript
- **Database:** MongoDB
- **Autenticazione:** JWT, OAuth2
- **Parsing CSV:** Utilizzare una libreria come `csv-parser` per gestire i file CSV.
- **Integrazione API:** Predisposizione per l'integrazione con un'API di e-commerce.

3. **Motore Statistico:**

- **Linguaggio:** Python
- **Librerie:** Pandas, NumPy, Scikit-learn
- **Comunicazione:** RabbitMQ per la coda dei messaggi

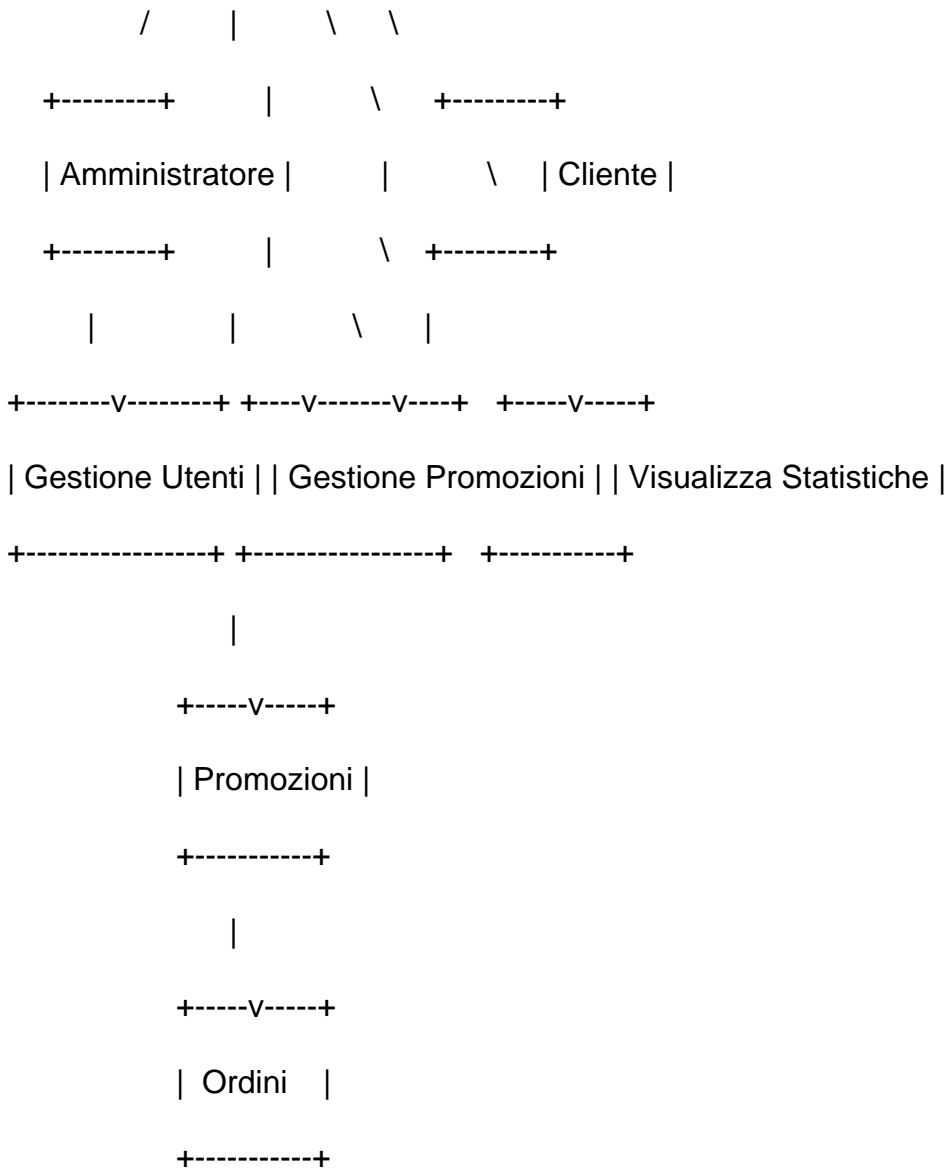
4. **Deployment:**

- **Provider Cloud:** AWS, Azure, o Google Cloud
- **CI/CD:** Jenkins, GitHub Actions, o GitLab CI

Diagrammi Dettagliati

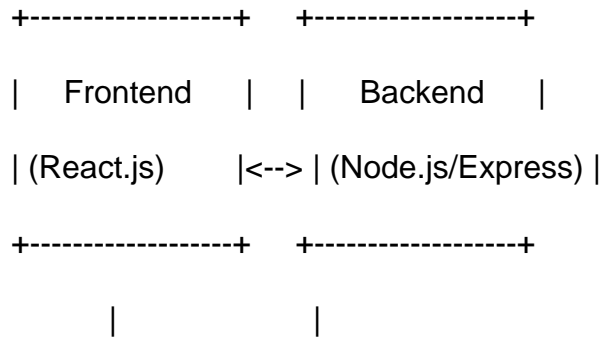
Diagramma dei Casi d'Uso

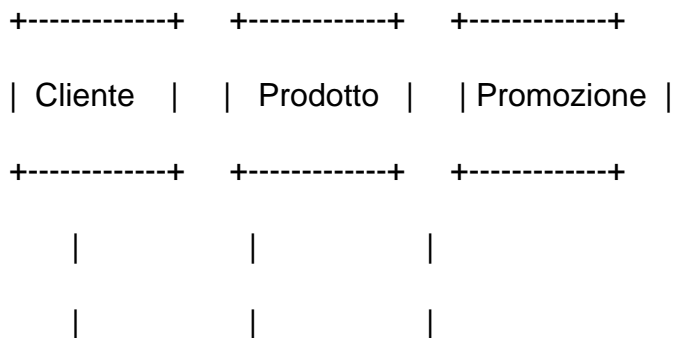
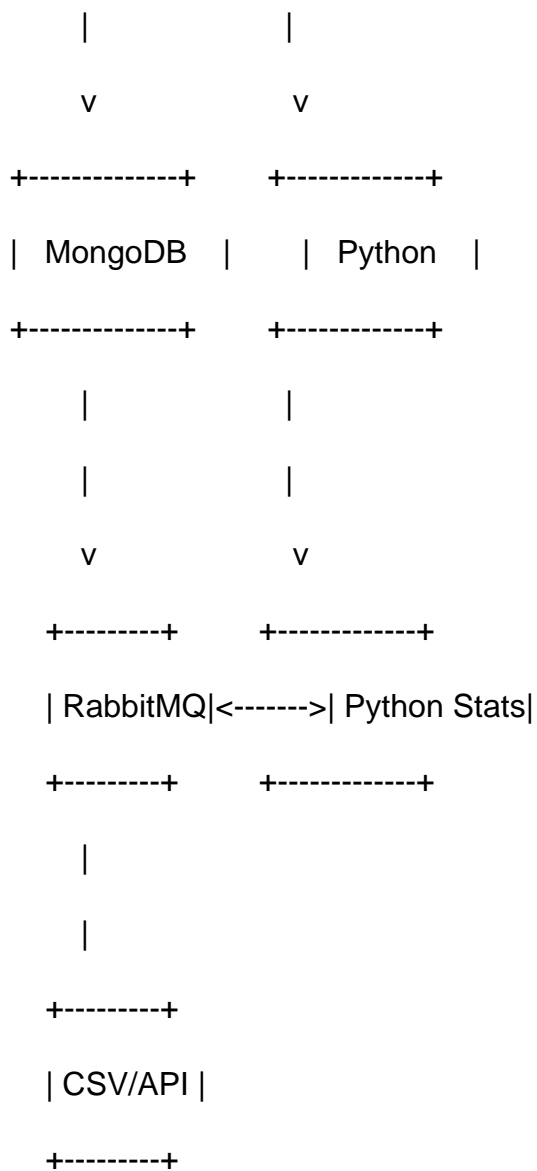


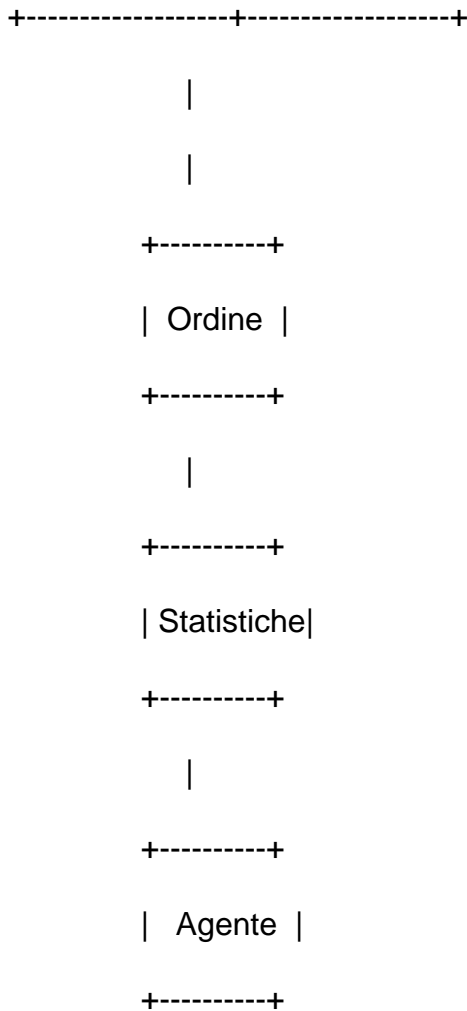


Component Diagram

Diagramma dei Componenti







Panoramica Funzionale

Componenti del Frontend

1. **Dashboard Amministratore:**

- Gestione Utenti: Aggiungere, modificare, eliminare utenti.
- Gestione Promozioni: Creare e inviare promozioni.
- Visualizzare Statistiche Clienti: Analisi e report dettagliati.
- Importazione Ordini: Caricare file CSV per importare ordini.
- Visualizzare e Gestire Avvisi.
- Visualizzare la progressione e le promozioni attive per tutti i clienti.

2. ****Dashboard Agente di Vendita:****

- Gestione Promozioni: Creare e inviare promozioni.
- Visualizzare Statistiche Clienti: Informazioni sul comportamento dei clienti associati.
 - Visualizzare Avvisi: Notifiche per cambiamenti significativi nel comportamento dei clienti associati.
- Visualizzare Promozioni: Relativi ai clienti associati.

3. ****Dashboard Cliente:****

- Visualizzare Statistiche Personali: Accesso alla cronologia degli acquisti personali e alle statistiche di spesa.
- Visualizzare Promozioni: Accesso alle promozioni disponibili.
- Visualizzare Progressione: Progressione verso le promozioni.

Servizi del Backend

1. ****Gestione Utenti:****

- Endpoint API per operazioni CRUD sugli utenti.
- Autenticazione e autorizzazione tramite JWT.
- Assegnare i clienti agli agenti.

2. ****Gestione Promozioni:****

- Endpoint API per creare e inviare promozioni via SMS/Email.
- Tracciare la progressione dei clienti verso le promozioni.

3. ****Gestione Ordini:****

- Endpoint API per caricare e analizzare file CSV.
- Servizio futuro per l'integrazione con l'API di e-commerce.

4. ****Gestione Statistiche e Avvisi:****

- Endpoint API per generare e recuperare statistiche.
- Integrazione con Python per analisi avanzate.

Prossimi Passi

1. ****Definire gli Endpoint API:****

- Elencare tutti gli endpoint necessari per la gestione di utenti, promozioni, ordini e statistiche.
- Includere endpoint per il caricamento di file CSV e l'integrazione futura con l'API.

2. ****Configurare l'Ambiente Iniziale:****

- Configurare MongoDB, Node.js/Express, e le strutture del progetto React.
- Configurare RabbitMQ e l'ambiente Python.

3. ****Sviluppare le Funzionalità Core:****

- Implementare le funzionalità principali come l'autenticazione degli utenti, la gestione delle promozioni e il tracciamento base degli ordini.
- Assegnare i clienti agli agenti e gestire le associazioni.

4. ****Integrare la Gestione dei CSV:****

- Implementare la funzionalità di caricamento e analisi dei file CSV.

5. ****Pianificare l'Integrazione Futura con l'API:****

- Progettare il sistema in modo da poter passare facilmente dai caricamenti CSV alle chiamate API.

- Implementare un servizio di placeholder per l'integrazione con l'API.

6. ****Implementare il Frontend:****

- Creare componenti React per le dashboard di amministratori, agenti di vendita e clienti.

7. ****Test e Deployment:****

- Eseguire test approfonditi di tutte le funzionalità.

- Configurare una pipeline CI/CD per il deployment.