

Design Goals

I Design Goals sono organizzati in base a diversi criteri: prestazioni, affidabilità, costi, manutenzione e criteri dell'utente finale.

Performance criteria

- **Tempo di risposta:** Car-Zone deve rispondere alle richieste dell'utente entro 5 secondi. Questo assicura un'esperienza utente rapida e reattiva, fondamentale per accessi frequenti al database e per la fruibilità del sito.
- **Throughput:** Il sistema deve essere in grado di gestire simultaneamente più richieste da diversi utenti, specialmente in momenti di traffico elevato. Sarà ottimizzato per processare molteplici richieste simultaneamente, riducendo al minimo i rallentamenti.
- **Uso della memoria:** Il sistema utilizzerà efficientemente la memoria, per bilanciare velocità e consumo di risorse.

Dependability criteria

- **Robustezza:** Car-Zone deve essere in grado di gestire input non validi senza causare interruzioni o comportamenti inaspettati. Verranno implementati controlli di validazione per prevenire crash e rispondere in modo sicuro agli input non corretti o fuori dai limiti.
- **Disponibilità:** Car-Zone deve garantire un'elevata disponibilità, assicurando che il sistema sia accessibile agli utenti per la maggior parte del tempo. L'obiettivo è mantenere una percentuale di uptime che permetta l'uso continuo per le attività quotidiane, minimizzando le interruzioni programmate per la manutenzione e riducendo al minimo i tempi di inattività non previsti.
- **Affidabilità:** Car-Zone deve mantenere un comportamento coerente e affidabile, con output che rispecchino correttamente le aspettative dell'utente anche in presenza di errori imprevisti.
- **Tolleranza ai guasti:** Il sistema deve continuare a funzionare in condizioni di errore (ad esempio, perdita temporanea di connettività).
- **Sicurezza:** Poiché Car-Zone gestirà dati sensibili degli utenti, il sistema sarà progettato per resistere ad attacchi malevoli, assicurando che le informazioni siano protette tramite autenticazione e crittografia adeguata.
- **Disponibilità:** Il sistema deve essere disponibile per gli utenti il più possibile, con una strategia di manutenzione programmata per minimizzare i tempi di inattività.

Cost criteria

- **Costo di sviluppo:** I costi iniziali di sviluppo saranno ottimizzati tramite una progettazione modulare e componenti riutilizzabili, riducendo il tempo necessario per future implementazioni e aggiornamenti.
- **Costo di distribuzione:** L'implementazione e l'installazione del sistema devono essere eseguite in modo da minimizzare i costi per gli utenti.

- **Costo di manutenzione:** Il sistema sarà progettato per facilitare correzioni di bug e aggiornamenti, garantendo nel contempo un basso costo di manutenzione a lungo termine.

Maintenance criteria

- **Estensibilità:** Car-Zone sarà progettato in modo modulare per permettere l'aggiunta di nuove funzionalità in futuro, con il minimo impatto sul sistema esistente.
- **Modificabilità:** Le funzionalità di base devono poter essere aggiornate o modificate con facilità, per poter adattare il sistema a nuovi requisiti o funzionalità richieste dal mercato.
- **Adattabilità:** Il sistema deve essere facilmente adattabile a diversi domini applicativi, permettendo il riutilizzo delle funzionalità principali in altri contesti o settori.
- **Portabilità:** Il sistema sarà sviluppato per essere compatibile con diverse piattaforme, così da facilitarne il trasferimento o l'adattamento in altre applicazioni o contesti.
- **Leggibilità:** Il codice del sistema deve essere scritto in modo chiaro e leggibile per facilitare la comprensione da parte degli sviluppatori, sia attuali che futuri.
- **Tracciabilità dei requisiti:** Ogni requisito sarà mappato al codice specifico per facilitare le verifiche e garantire che il sistema soddisfi le specifiche concordate.

End User criteria

- **Utilità:** Car-Zone deve supportare efficientemente le attività chiave dell'utente, come la ricerca e l'interazione con i veicoli disponibili, per garantire una user experience ottimale.
- **Usabilità:** L'interfaccia del sistema sarà intuitiva e facile da usare per utenti di diversi livelli di esperienza. L'accesso e la navigazione saranno ottimizzati per l'uso su desktop e dispositivi mobili, con un design responsive per migliorare l'accessibilità.

I trade-off sono necessari per bilanciare i design goals con le limitazioni di risorse, tempo, e budget.

- **Tempo di risposta vs. Consumo di memoria**
Per migliorare la velocità del sistema, verrà utilizzata una cache mirata che bilancia le prestazioni e il consumo di memoria, memorizzando solo le informazioni più utili per gli utenti.
- **Costo di sviluppo vs. Qualità**
Per ottimizzare i costi senza sacrificare troppo la qualità, si darà priorità alle funzionalità principali, rinviando quelle meno urgenti e garantendo il rilascio puntuale del sistema.
- **Sicurezza vs. Usabilità**
La sicurezza verrà progettata per essere efficace ma semplice, con soluzioni come l'autenticazione a due fattori opzionale, che proteggono i dati senza complicare eccessivamente l'esperienza utente.

- **Scalabilità vs. Costi di manutenzione**

Si adotterà un'architettura modulare dove necessario, implementando miglioramenti nelle altre aree solo in base a esigenze future, così da controllare i costi di manutenzione.

- **Tempo di consegna vs. Funzionalità**

Per rispettare le scadenze, verranno implementate prima le funzionalità essenziali, lasciando quelle secondarie per aggiornamenti successivi, garantendo così un rilascio nei tempi stabiliti.