



UNIVERSITÀ DEGLI STUDI DI BERGAMO

Report Modulo NCC per Odoo

Cristiano Gambirasio Matr. 1066866

ANNO ACCADEMICO 2023/2024

1 Analisi dei requisiti

Il modulo è pensato per supportare un'azienda di NCC, ovvero Noleggio Con Conducente. Questa tipologia di azienda si occupa di fornire servizi di trasporto in modo simile a un taxi, ma dando priorità a fattori come l'eleganza del veicolo e un servizio di lusso che deve garantire professionalità e puntualità.

A differenza di un servizio di taxi, lo use case più frequente è quello in cui i servizi sono prenotati con largo anticipo, è quindi fondamentale avere un metodo organizzativo veloce e di facile utilizzo per l'inserimento di nuove prenotazioni e la gestione della flotta e degli autisti.

Prima di iniziare lo sviluppo del modulo è stata fatta un'analisi dei requisiti in modo da dare priorità alle funzionalità più importanti.

1.1 Modello MoSCoW

Tramite il modello MoSCoW è stato possibile dividere i requisiti in quattro livelli di priorità.

Mo	S	Co	W
1. Inserimento nuove prenotazioni 2. Visualizzazione delle prenotazioni inserite 3. Funzione di assegnazione servizio ad un dipendente	1. Integrazione con modulo Parco Auto e selezione del veicolo usato per il trasporto 2. Calendario dei servizi da svolgere	1. Integrazione con Modulo di Contabilità 2. Grafico di resoconto degli incassi 3. Invio automatico email di conferma	1. Interfaccia mobile e design personalizzato 2. Configurazione personalizzabile dei parametri di prenotazione

Figure 1: Modello MoSCoW

- Mo (must-have): In questa classe troviamo le funzionalità essenziali all'utilizzo del componente. Senza di queste funzionalità il componente non riuscirebbe a svolgere i compiti base per cui è pensato.
- S (should-have): Contiene i requisiti che forniscono funzionalità secondarie, ma ancora piuttosto importanti per il risultato finale. Anche se non fondamentali queste funzioni dovrebbero essere implementate.
- Co (could-have): Raggruppa tutte le funzionalità che avrebbero un'utilità durante l'utilizzo e sarebbe bello averle implementate. La mancata realizzazione di queste funzioni non compromette l'esperienza di utilizzo.
- W (won't-have): Contiene le funzioni che, pur essendo compatibili con l'idea del progetto, non hanno molta importanza e probabilmente hanno benefici che non valgono il tempo e costi di sviluppo.

2 Casi d'uso

Per lo sviluppo del componente è stato fondamentale capire e studiare i modi con cui diversi attori si interfacciano con il sistema.

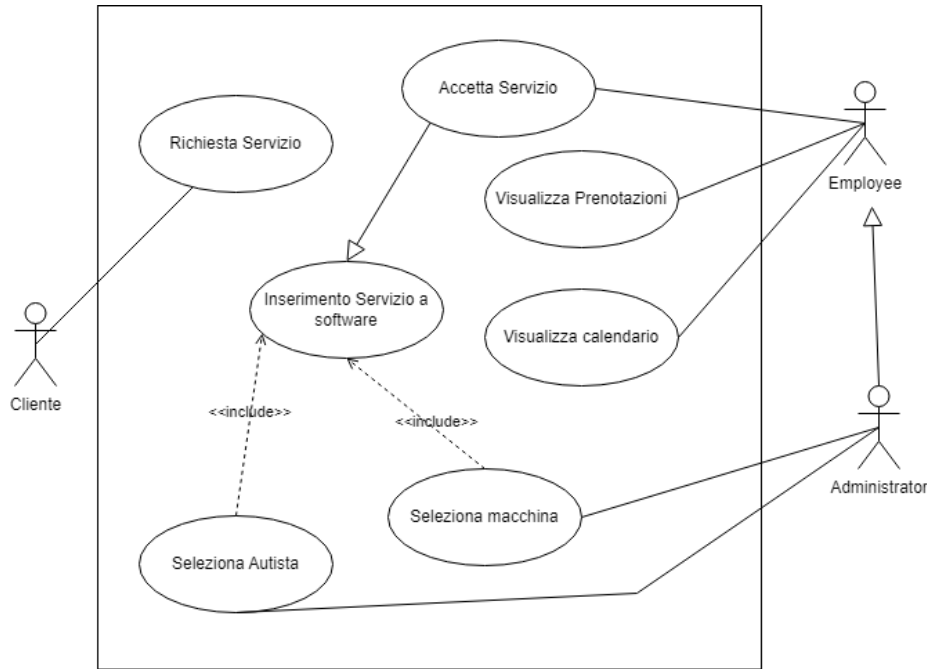


Figure 2: Use Case Diagram

- Il cliente momentaneamente richiede un servizio ancora tramite email, ma uno sviluppo futuro potrebbe far utilizzare il sistema direttamente all'utente in modo da avere le richieste con i relativi dati già integrati nell'ERP.
- L' Employee è un impiegato "semplice" dell'azienda, il suo compito è quello di prendere le richieste di servizio e accettarle inserendo i dati nel software. Oltre ad inserire i servizi, ha la possibilità di visualizzare l'elenco delle prenotazioni, o un calendario contenente come eventi le varie prenotazioni.
- L' Administrator è il gestore dell'azienda, che si occupa anche di assegnare i servizi ad una coppia di pilota e auto. Questa fase non avviene durante l'inserimento della prenotazione, ma in un secondo momento in cui l' Administrator può avere un'idea più complessiva delle prenotazioni. In questo modo è possibile effettuare assegnazioni più efficienti. Oltre a questo l'amministratore ha accesso a tutte le funzioni disponibili agli impiegati.

3 Diagramma delle classi

Il diagramma delle classi è utile per visualizzare le entità presenti nel sistema.

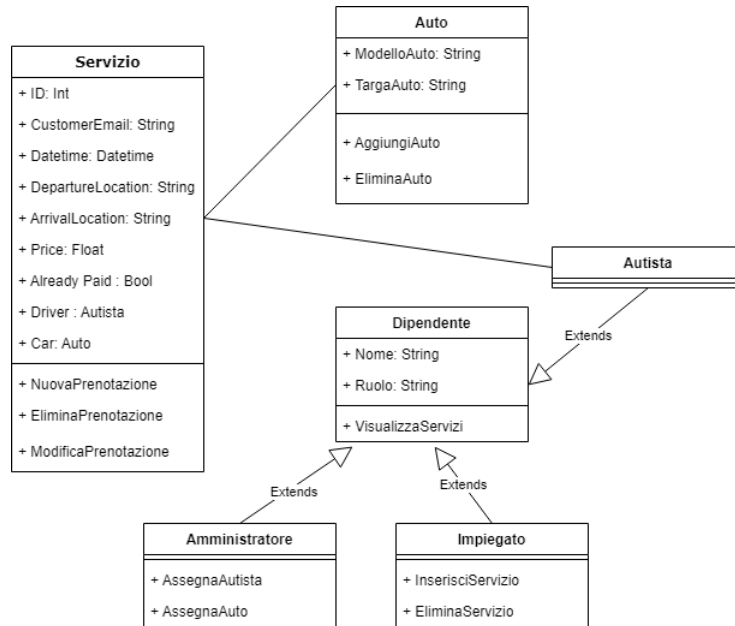


Figure 3: Class Diagram

Le entità principali nel sistema sono tre:

- **Auto**: Descrive un'auto posseduta dalla azienda di NCC. Viene descritta da modello e targa. Il sistema permette l'inserimento e l'eliminazione di un veicolo.
- **Dipendente**: Sono gli attori nella azienda. In particolare abbiamo:
 1. **Autisti**: Si limitano alla visualizzazione dei servizi presenti nel sistema in modo da svolgerli.
 2. **Impiegati**: Possono interagire con il sistema per inserire ed eliminare servizi.
 3. **Amministratore**: Interagisce con il sistema assegnando ad ogni servizio un autista e un'auto.
- **Servizi**: Contengono una serie di dati che descrivono il servizio, possono essere aggiunti, rimossi, o modificati (in modo da permettere di inserire successivamente i campi Driver e Car).

Ogni Servizio è associato con un Dipendente (in particolare un Autista) e un Auto

4 Diagramma delle attività

Per comprendere il ciclo di vita di una prenotazione è stato realizzato un activity diagram.

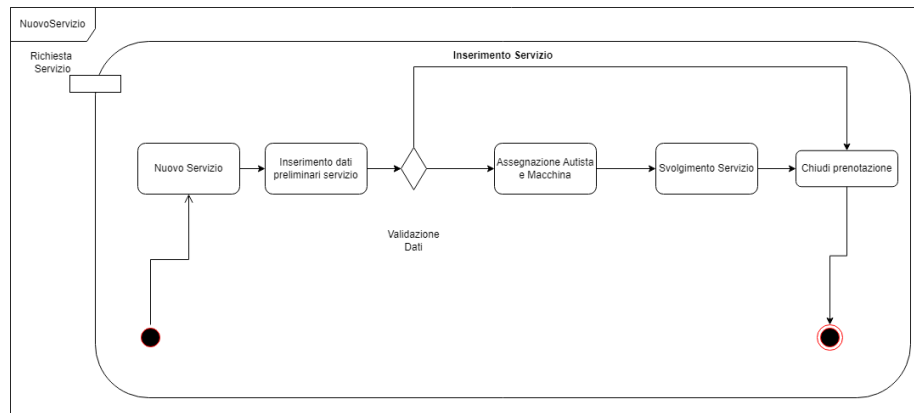


Figure 4: Activity Diagram

1. L'interfaccia all'inizio del processo è una nuova richiesta di servizio.
2. Iniziato il processo, un impiegato inserisce i dati relativi alla prenotazione nel sistema.
3. Dopodiché il sistema esegue un controllo sulla validità degli input (in particolare sulla data inserita). Se l'input non è valido annulla l'inserimento e richiede che l'attività ricominci dall'inserimento dei dati.
4. Se l'impiegato inserisce dati validi, la prenotazione rimane inserita nel sistema ed è visualizzabile nelle sezioni dedicate all'amministratore. Quest'ultimo, nel momento più opportuno, assegnerà al servizio un autista e un'auto.
5. Una volta assegnato il servizio, questo sarà visualizzabile sul sistema dai piloti, che quindi avranno un elenco di trasporti da eseguire.
6. Una volta che il servizio è stato eseguito si può dire che la prenotazione è conclusa e si può terminare il suo ciclo di vita.