



UNIVERSITÀ DI VERONA

DIPARTIMENTO DI INFORMATICA

Corso di Laurea Triennale in Informatica

CAUTO

Configuratore d'auto

Candidato:
Gennari Alessia
Moretto Mattia
Pasetto Michele

ANNO ACCADEMICO 2023/2024

Indice

	Pagina
1 Desiderata	2
2 Specifiche Tecniche	4
2.1 Tecnologie utilizzate	4
2.1.1 C++	4
2.1.2 Angular	4
2.1.3 Json	4
2.1.4 Pistache	5
2.2 Formule applicate	5
3 Casi d'uso	6
3.1 Diagramma dei casi d'uso	6
3.2 Schede di specifica	8
4 Diagramma della classi	12
5 Diagrammi d'interazione	13

Capitolo 1

Desiderata

É richiesto lo sviluppo di un sistema informatico per la vendita online di automobili di un gruppo multi-concessionaria con la possibilità di configurare la versione di automobile desiderata.

Il gruppo vende auto di diversi modelli, raggruppati per marca. Per ogni modello di auto il sistema deve poter memorizzare:

- Un nome univoco
- Una descrizione
- Le dimensioni (altezza, lunghezza, larghezza, peso e volume del bagaglio)
- Possibili immagini che permettano di vedere l'auto da vari punti di vista con (potenzialmente diversi colori)

Il sistema dovrà registrare tutte le ruote a catalogo e definire possibili optional:

- Il colore
- Possibili aggiunte
 - Ruotino di scorta
 - Ruota di scorta
 - Vetro oscurato
 - Interni in pelle
 - Ruote con diametri maggiori di quello standard

Per ogni optional deve essere definito il prezzo e se applicabile o meno al modello scelto.

Per alcuni modelli deve poter essere applicato uno sconto che può variare da un mese all'altro e viene applicato in fase di costruzione del preventivo ed opportunamente indicato. Il cliente deve anche indicare, in caso di acquisto, dove intende ritirare l'auto.

Un generico utente anche senza autenticarsi deve poter configurare l'auto per la quale pensa di chiedere un preventivo. Successivamente nel momento in cui decide di richiedere un preventivo al gruppo è richiesto che l'utente si registri (nel caso in cui non lo fosse già) e poi effettuare l'accesso al sistema. Solo a seguito di ciò avrà quindi la possibilità di richiedere la valutazione della sua auto usata allegando delle fotografie.

Nel caso di valutazione dell'usato, il preventivo verrà finalizzato da una persona del negozio, che provvederà a valutare l'usato. Il preventivo potrà essere prodotto anche come file pdf:

Il sistema memorizza i preventivi effettuati. Entro 20 giorni il potenziale acquirente può confermare il preventivo e pagare un acconto, per ordinare l'auto nella configurazione indicata. Il sistema provvederà a proporre una data di consegna (a un mese più 10 giorni per ogni optional richiesto) entro la quale avere l'auto ordinata.

Il gruppo ha diversi sedi in Italia dov'è possibile ritirare l'auto ordinata. Per ogni sede il sistema deve memorizzare nome, indirizzo completo e tutti gli ordini relativi a quella sede.

Il sistema deve permettere agli impiegati del gruppo di accedere tramite autenticazione al sistema e gestire i preventivi che richiedono la valutazione dell'usato. Gli impiegati potranno poi gestire gli ordini e avvisare i clienti quando l'auto ordinata è pronta per la consegna (dopo il pagamento dell'importo dovuto).

La segreteria amministrativa del gruppo è responsabile dell'inserimento delle informazioni su modelli e optional di ogni marca. Essa può accedere al sistema e visualizzare i preventivi per cliente, per marca e per negozio di consegna richiesto.

Il sistema deve inoltre permettere all'utente di configurare l'auto avendo un controllo del prezzo finale dell'auto ad ogni momento.

Capitolo 2

Specifiche Tecniche

2.1 Tecnologie utilizzate

Di seguito vengono elencate e descritte le principali tecnologie utilizzate per lo sviluppo del software per dare un'idea generale di come è composto quanto richiesto. Queste particolari scelte progettuali sono dettate dalla confidenza e praticità nell'utilizzo da parte dei componenti del gruppo nell'utilizzarli anche in ambito lavorativo.

2.1.1 C++

Si tratta di un linguaggio di programmazione di alto livello orientato agli oggetti e fortemente tipizzato, è stato sviluppato come estensione del linguaggio C ed introdotto negli anni 80. C++ combina le caratteristiche della programmazione procedurale con quelle della programmazione orientata agli oggetti, permettendo la creazione di software più complessi, efficienti e modulari. È ampiamente utilizzato nello sviluppo di sistemi operativi, applicazioni desktop, giochi, software per dispositivi embedded e applicazioni ad alte prestazioni. Grazie alla sua flessibilità e potenza, C++ rimane uno dei linguaggi più influenti e utilizzati nel mondo della programmazione.

2.1.2 Angular

Angular è un framework open-source per lo sviluppo di applicazioni web, mantenuto da Google e basato su TypeScript, Angular facilita la creazione di applicazioni web dinamiche e reattive grazie alla sua architettura a componenti, che consente la suddivisione del codice in moduli riutilizzabili e facilmente gestibili. Tra le sue caratteristiche principali, Angular offre strumenti per il data binding, la gestione delle forme, l'iniezione di dipendenze e il routing, semplificando lo sviluppo di applicazioni complesse e scalabili. Grazie alla sua versatilità e potenza Angular è una scelta popolare per la creazione di applicazioni web moderne, sia in ambito enterprise che consumer.

2.1.3 Json

JSON (JavaScript Object Notation) è un formato leggero per lo scambio di dati, facile da leggere e scrivere sia per le persone che per le macchine. Utilizzato principalmente per trasmettere dati tra un server e un client web, JSON rappresenta le informazioni in un formato testuale strutturato basato su coppia chiave-valore, simile ad un oggetto JavaScript. Grazie alla sua semplicità e compatibilità con la maggior parte dei linguaggi di programmazione, JSON è diventato uno standard ampiamente adottato per l'interscambio di dati nelle applicazioni web e nelle API. Nel nostro progetto è stato utilizzato come database.

2.1.4 Pistache

Pistache è un framework leggero e open-source scritto in C++ per lo sviluppo di server HTTP. Progettato per essere semplice ed efficiente, Pistache offre un architettura basata su eventi e supporta la creazione di server web ad alte prestazioni, in grado di gestire numerose connessioni simultanee. È particolarmente adatto per lo sviluppo di API RESTfull, grazie alla sua capacità di gestire richieste e risposte HTTP in modo intuitivo e veloce. Pistache è una scelta popolare per gli sviluppatori in C++ che cercano un framework minimalista ma potente per costruire applicazioni web scalabili e performanti.

2.2 Formule applicate

Nello sviluppo di questo software ci siamo soffermati su due punti in particolare che richiedevano precise formule per soddisfare la desiderata.

La prima riguarda la fase di configurazione dell'auto dove il prezzo finale deve sempre essere controllabile in ogni momento pertanto il calcolo che viene fatto è il seguente:

$$p_f = p_0 + \sum_0^i p_i$$

$p_f = \text{Prezzo finale} \quad p_0 = \text{Prezzo base}$
 $i = \text{Numero componenti scelti} \quad p_i = \text{Prezzo componente}$

Viene omesso il fatto che al prezzo finale ottenuto può essere applicato uno sconto in quanto avverrebbe soltanto in fase di creazione del preventivo e dipende fortemente dal modello e dal periodo dell'ordine

La seconda invece riguarda la data di consegna entro la quale l'utente può ritirare l'auto ordinata che deve soddisfare il seguente presupposto:

$$d_c > date(now) + (30 + i * 10)$$

$d_c = \text{data consegna} \quad i = \text{numero componenti}$

Capitolo 3

Casi d'uso

I casi d'uso permettono di modellare i requisiti funzionali ossia quei requisiti che specificano cosa deve essere fatto. Essi sono indipendenti dalla tecnologia, dall'architettura, dalla piattaforma e dal linguaggio di programmazione.

I casi d'uso specificano cosa ci si aspetta da un sistema e allo stesso tempo nascondono come il sistema lo implementa. Sono una sequenza di azioni che producono un risultato osservabile da un attore e vengono utilizzati per descrivere i requisiti iniziali (analisi) e a convalidare l'architettura. Un ottimo elemento per rappresentarli è il diagramma dei casi d'uso che procediamo quindi a sviluppare.

3.1 Diagramma dei casi d'uso

Si tratta di un diagramma che esprime un comportamento desiderato o offerto e, solitamente, l'oggetto esaminato è un sistema o una sua parte. Permette di individuare chi o che cosa ha a che fare con il sistema (attore) e cosa esso può fare.

Tipicamente il diagramma dei casi d'uso è la prima cosa ad essere creata in un processo o ciclo di sviluppo nell'ambito dell'analisi dei requisiti

Per comporre correttamente il diagramma dobbiamo prima fare un passo indietro e individuare tutti i casi d'uso che compongono il sistema.

Per sviluppare i casi d'uso è necessario prima analizzare quella che è la desiderata, ossia, in questo caso, la consegna fornita per lo sviluppo del progetto che è stata riportata nel capitolo 1. Formalizzare la desiderata individuando i requisiti è un elemento fondamentale in quanto una specifica errata o incompleta è una delle cause principali del fallimento dei progetti software.

Procedendo quindi per step andiamo ad analizzare tutto quello che ci viene richiesto, che possiamo riassumere nel seguente modo:

- Software per la vendita di auto (configurabili) online
- Un catalogo che prevede diversi modelli raggruppati per marca e con i relativi optional e immagini illustrative
- Sconti sui sigoli modelli applicabili e opportunamente indicati in fase di creazione del preventivo
- La possibilità per gli utenti che utilizzano la piattaforma di configurare in ogni momento l'auto per cui pensano di richiedere un preventivo anche senza autenticarsi e, previa autenticazione, la possibilità di richiedere il preventivo e la valutazione del suo usato

Lo step successivo è quindi quello di individuare gli attori ossia ruoli che vengono assunti dagli utenti o altre entità che interagiscono col sistema. Un attore deve essere per forza esterno al sistema inoltre non necessariamente deve essere un umano. Nello sviluppo del nostro progetto gli attori individuati sono i seguenti:

- Utente/Cliente
- Impiegati
- Segreteria

Per ognuno di questi attori è ora possibile individuare i casi d'uso ossia l'obiettivo prefissato all'inizio del capitolo e sono gli elementi che permettono la creazione del diagramma

- Il cliente consulta il catalogo, si registra e può effettuare acquisti
- Il cliente personalizza l'auto
- Il cliente richiede un preventivo e sceglie la sede per il ritiro
- Il cliente può richiedere la valutazione dell'usato
- Gli impiegati si autenticano e gestiscono i preventivi che richiedono la valutazione dell'usato
- Gli impiegati gestiscono gli ordini e avvisano quando l'auto è pronta per la consegna
- La segreteria si occupa dell'inserimento delle informazioni dei modelli e optional per ogni marca
- La segreteria può visualizzare ogni preventivo per cliente, marca e per negozio di consegna
- Il cliente paga l'acconto
- Il cliente ritira l'auto

Arrivati a questo punto ci è possibile procedere allo sviluppo del diagramma del caso d'uso collegando innanzitutto i diversi attori ai relativi casi d'uso attraverso le associazioni ossia il primo elemento di collegamento.

Successivamente ci siamo occupati delle diverse dipendenze tra i casi d'uso, nel nostro caso abbiamo deciso di utilizzare:

- **Include:** dipendenza dove il caso incluso fa parte del comportamento di quello che lo include, si tratta di un comportamento obbligatorio
- **extend:** dipendenza dove il caso d'uso che estende specifica un incremento di comportamento a quello esteso, si tratta di un comportamento opzionale che gestisce casi particolari

Una volta esserci soffermati su tutti i punti siamo arrivati a creare il diagramma nella sua interezza che si presenta nel seguente modo:

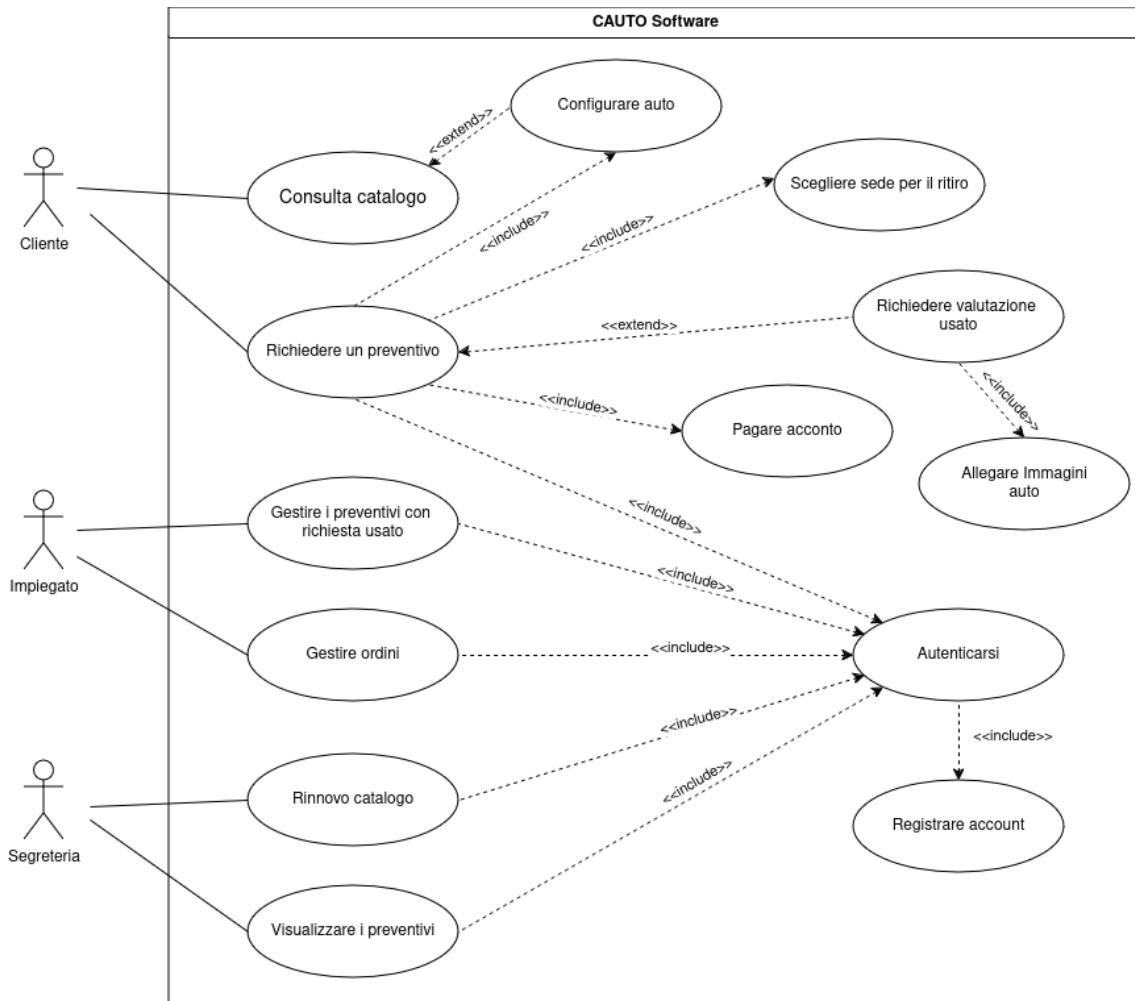


Figura 3.1: Diagramma Casi d'uso

Quanto riportato è il diagramma d'uso inteso come punto di partenza per lo sviluppo del progetto. Si ha però l'esigenza di abbinare a questo diagramma delle specifiche testuali, più formali, in quanto non sono adatti a mostrare la sequenza temporale delle operazioni oppure lo stato del sistema e degli attori prima e dopo l'esecuzione. È necessario quindi procedere con la descrizione (specifica) dei casi d'uso

3.2 Schede di specifica

Ogni caso d'uso ha un nome e una specifica, quest'ultima è composta da:

- **precondizioni:** condizioni che devono essere vere prima che il caso d'uso di possa eseguire
- **sequenza degli eventi:** i passi che compongono il caso d'uso.
- **postcondizioni:** condizioni che devono essere vere quando il caso d'uso termina l'esecuzione

Partendo quindi dagli utenti/clienti si procede ad analizzare quelle che sono le schede di specifica che lo riguardano

Consulta catalogo
ID: CU1
Attori: Utente
Precondizioni: Nessuna
Sequenza: <ol style="list-style-type: none"> 1. Entrare nella pagina dedicata al catalogo delle auto 2. Osservare e configurare auto
Postcondizioni: Nessuna

È importante evidenziare il fatto che dal consultare il catalogo non è stata specificata alcuna post-condizione in quanto per come il software è stato predisposto un utente generico ha la possibilità di consultarlo senza alcun vincolo. Il caso d'uso appena analizzato lo si può quindi definire un'azione "base" prevista dal software.

Si passa quindi ad analizzare il successivo caso d'uso che lo si può considerare quello principale all'interno di quello che è il funzionamento dell'applicazione.

Richiedere un preventivo
ID: CU2
Attori: Utente
Precondizioni: Essersi Registrato sulla piattaforma
Sequenza: <ol style="list-style-type: none"> 1. Effettuare l'accesso alla piattaforma 2. Configurare l'auto che ha intenzione di acquistare 3. (a) Richiedere direttamente preventivo (b) Richiedere la valutazione del proprio usato 4. Scegliere la sede per effettuare il ritiro dell'auto 5. Attendere finalizzazione del preventivo 6. Entro 20 giorni confermare il preventivo
Postcondizioni: Pagare acconto

Elaborate le schede di specifica legate agli utenti si procede ad analizzare quelle relative agli impiegati

Gestire i preventivi con richiesta usato
ID: CU3
Attori: Impiegato
Precondizioni: <ol style="list-style-type: none"> 1. Un cliente ha richiesto la valutazione dell'auto usata durante la richiesta di un preventivo 2. Essere registrato alla piattaforma
Sequenza: <ol style="list-style-type: none"> 1. Effettuare l'accesso alla piattaforma 2. Accedere alla pagina dedicata con la lista dei preventivi da finalizzare 3. Procedere a valutare l'usato
Postcondizioni: Finalizzare il preventivo

Gestire ordini
ID: CU4
Attori: Impiegato
Precondizioni: <ol style="list-style-type: none"> 1. Essere registrato alla piattaforma 2. Acconto del preventivo pagato dall'utente 3. Terminato tempo attesa per la data di consegna
Sequenza: <ol style="list-style-type: none"> 1. Effettuare l'accesso alla piattaforma 2. Accedere alla pagina dedicata con la lista dei preventivi 3. Procedere ad avvisare l'utente che l'auto è pronta
Postcondizioni: Consegna l'auto nella sede indicata

Terminate le schede di specifiche che riguardano gli impiegati si procede con quelle legate alla segreteria

Rinnovo catalogo
ID: CU5
Attori: Segreteria
Precondizioni: Essere registrato alla piattaforma
Sequenza: <ol style="list-style-type: none"> 1. Effettuare l'accesso alla piattaforma 2. Accedere alla pagina dedicata con il catalogo delle auto 3. Inserimento di informazioni su modelli e optional di ogni marca
Postcondizioni: Nessuna

Il caso d'uso che riguarda il rinnovo del catalogo è un'azione fine a se stessa, nel senso che si tratta di qualcosa che viene svolto dalla segreteria come parte del proprio lavoro, da questa azione non ne consegue nessun'altra ma è importante per avere il catalogo sempre aggiornato e influisce sui casi d'uso degli altri attori. Lo stesso vale per la prossima scheda, si potrebbe definire la segreteria come un attore passivo.

Visualizzare i preventivi
ID: CU6
Attori: Segreteria
Precondizioni: Essere registrato alla piattaforma
Sequenza: <ol style="list-style-type: none"> 1. Effettuare l'accesso alla piattaforma 2. Accedere alla pagina dedicata con la lista dei preventivi 3. Visualizzare i preventivi per cliente, marca e negozio di consegna richiesto
Postcondizioni: Nessuna

Capitolo 4

Diagramma della classi

Un diagramma delle classi è un tipo di diagramma utilizzato nella modellazione ad oggetti. Serve per rappresentare la struttura statica di un sistema mostrando le classi, gli attributi, i metodi e le relazioni tra di esse come l'ereditarietà, l'associazione e la composizione. Si tratta di uno strumento fondamentale per la progettazione del software poichè consente di visualizzare e definire chiaramente l'architettura del sistema, facilitando la comprensione, la manutenzione e l'implementazione del codice.

Si tratta del secondo step che abbiamo deciso di sviluppare in quanto, una volta ultimato, permette una stesura iniziale del codice avendo già ideato quella che doveva essere a grandi linee la struttura dal software.

All'interno del diagramma ogni classe viene rappresentata in forma di un rettangolo, che può avere fino a 3 slot:

- nome e l'eventuale stereotipo (in UpperCamelCase, obbligatorio)
- attributi (in lowerCamelCase, opzionali)
- operazioni (in lowerCamelCase, opzionali)

Ogni classe a sua volta si divide in:

- **classi di analisi:** solitamente contengono solo quelli più importanti e spesso specificano solo il nome
- **classi di progettazione:** forniscono una specifica completa (implementabile) della classe e dei suoi attributi

Per rappresentare invece le istanze delle classi si utilizza una notazione molto simile, quindi avranno anch'esse una forma rettangolare. Le differenze sostanziali stanno:

- il titolo degli oggetti è sottolineato e nella forma "nome:classe" con nome opzionale
- Non hanno uno slot per le operazioni, possono invece definire valori per gli attributi

Capitolo 5

Diagrammi d'interazione

Si tratta di diagrammi di comportamento che modellano le interazioni tra varie entità di un sistema inoltre permettono di visualizzare lo scambio di messaggi tra entità nel tempo. Il loro scopo principale è quello di mostrare come un certo comportamento viene realizzato dalla collaborazione delle entità in gioco.

Sostanzialmente permette di illustrare il comportamento del sistema da una prospettiva interna inoltre possono avere diversi livelli di astrazione.

Per comporre un diagramma di interazione sono necessarie due cose:

- Un comportamento da realizzare tratto da un classificatore di contesto come per esempio:
 - Un caso d'uso
 - im'operazione di classe
- Una serie di elementi che realizzano il comportamento come:
 - attori
 - istanze di classe

Tutti gli elementi che ci occorrono vengono estrapolati dai vari diagrammi e schede dei capitoli precedenti