Università
Ca'Foscari
Venezia

Progetto di ingegneria del software 2023/2024

Documento di progettazione
Versione 2.0

Gruppo Argenta

15 gennaio 2024

Document informations

Deliverable	Documento di progettazione
Data di consegna	15/01/2024
Team leader	Donald Gera - 892604@stud.unive.it
Team members	Daniel Andrei Bercu - 891470@stud.unive.it Mattia Schiavon - 890993@stud.unive.it Marco Netti - 892399@stud.unive.it Sebastiano Sartor - 891825@stud.unive.it

Document history

Version	Issue Date	Stage	Changes	Contributors
1.0	28/11/2023	Draft	Definizione del documento di progettazione secondo le indicazioni fornite nel corso	Mattia Schiavon, Marco Netti, Sebastiano Sartor, Daniel Andrei Bercu, Donald Gera
2.0	15/01/2024	Final	Allineamento della documentazione	Daniel Andrei Bercu

Contents

1 Introduzione

1.1 Executive Summary

Il documento illustra la concezione del sistema per la nostra applicazione. Contiene una definizione dettagliata dell'architettura, presentando il modello e la sua struttura, analizzando l'approccio alla gestione dei dati. Successivamente, esamina il modello di dati e di controllo, includendo la decisione per il modello client-server e il modello manageriale. Una parte significativa del testo è poi dedicata all'utilizzo di modelli UML.

1.2 Struttura del documento

Come da indice il documento è stato strutturato secondo il modello proposto, sono presenti sette sezioni:

- Introduzione
- Glossario
- Architettura del sistema
- Modello dei dati e del controllo
- Modelli UML
- Progettazione dell'interfaccia utente
- Riferimenti

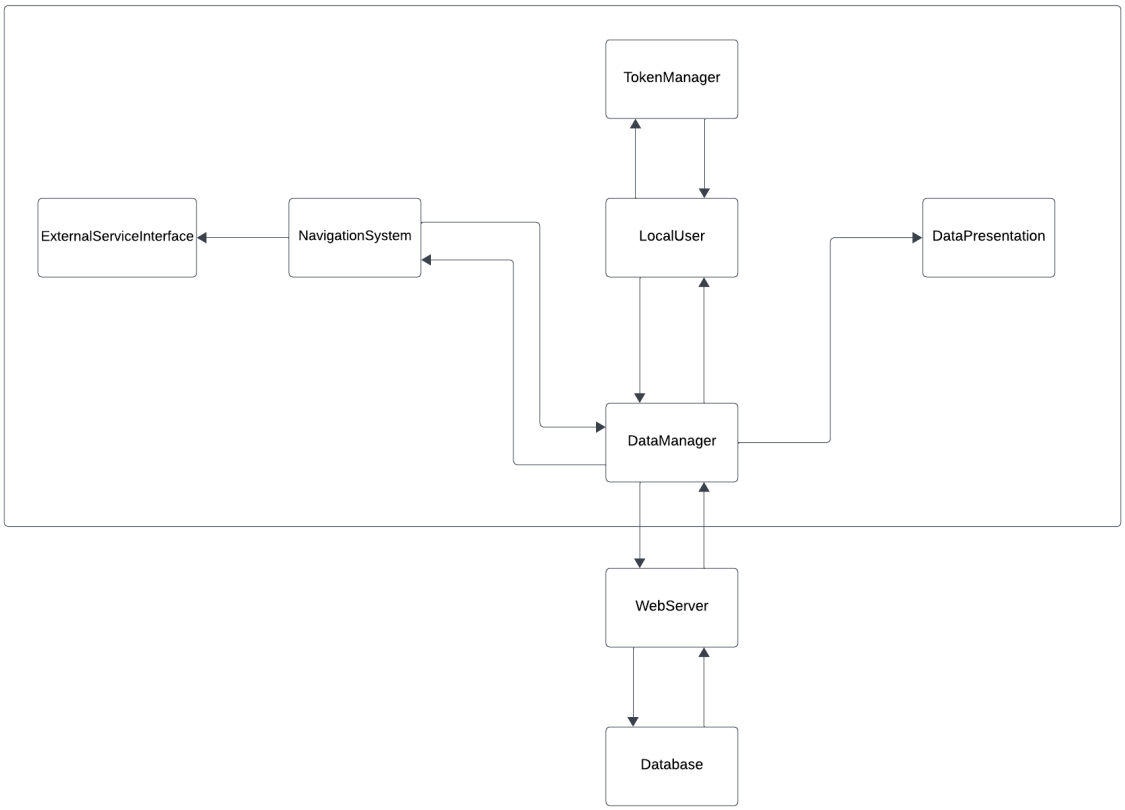
2 Glossario

- **App:** in informatica un'app è un'applicazione dedicata specificatamente a dispositivi mobili, tipicamente è realizzata con un occhio di riguardo al consumo di risorse. Fonte
- **Google Maps:** è un servizio internet geografico sviluppato da Google che consente la ricerca e la visualizzazione di carte geografiche di buona parte della Terra. Fonte
- **API:** acronimo di Application Programming Interface, si indica un insieme di procedure atte a risolvere uno specifico problema di comunicazione tra diversi computer o tra diversi software o tra diversi componenti di software. Fonte
- **GPS:** sistema di geolocalizzazione situato all'interno del dispositivo per il tracciamento dei movimenti.
- **Android:** è un sistema operativo per dispositivi mobili sviluppato da Google, ad oggi è presente in una vasta gamma di dispositivi tra i quali smartphone (principalmente), televisori, automobili e orologi. Fonte
- **Material Design:** è un design sviluppato da Google. Le regole di progettazione del Material Design si concentrano su un maggiore uso di layout basati su una griglia, animazioni e transizioni ed effetti di profondità come l'illuminazione e le ombre. Fonte
- **UML:** linguaggio di modellazione e specifica, basato sul paradigma orientato ad oggetti. Fonte
- **SQLite:** è una libreria software scritta in linguaggio C che implementa un DBMS SQL di tipo ACID incorporabile all'interno di applicazioni. Fonte

3 Architettura del sistema

3.1 Modello e struttura del sistema

Nel seguente diagramma a blocchi andiamo a rappresentare la struttura del nostro sistema.
I sottosistemi che compongono l'applicazione sono rappresentati all'interno del riquadro.



3.2 Gestione dei dati

Il sistema GaSmart adotta una gestione diversificata dei dati, tenendo conto del contesto specifico in cui si trova.
Questi contesti sono distinti nelle due successive sottosezioni.

3.2.1 Dati in locale

Per la gestione e la catalogazione dei dati locali, l'applicazione fa ampio uso della libreria SQLite. Questa libreria permette di memorizzare in modo strutturato e organizzato i dati relativi agli utenti, alle preferenze di visualizzazione, e alle configurazioni dei veicoli. Ogni utente ha i propri dati memorizzati in un file locale per garantire una risposta rapida e personalizzata.

I dati sono organizzati e memorizzati localmente attraverso SQLite in un formato ottimizzato per le operazioni di lettura e scrittura in ambiente mobile.

3.2.2 Comunicazione tra app e server

La comunicazione delle informazioni tra l'applicazione e il server avviene mediante lo scambio di messaggi in formato JSON. Questo approccio garantisce una trasmissione efficiente e strutturata dei dati tra l'app e il server remoto.

Ogni messaggio tra app e server è strutturato in formato JSON per garantire coerenza e facilitare l'interpretazione dei dati da entrambe le parti.

4 Modello dei dati e del controllo

Nello sviluppo dell'applicativo, si è presentata la necessità di definire delle regole di gestione dei dati e del flusso d'esecuzione dell'applicazione. Vengono di seguito presentati i modelli che delineano rispettivamente la gestione di tali aspetti.

4.1 Modello Client-Server

Per quanto riguarda l'ottenimento delle informazioni di distributori e veicoli, è risultata decisamente chiara la scelta di adottare un modello di condivisione dei dati che adempiesse al paradigma Client-Server.

Una singola istanza dell'applicazione, che in tale modello prende il nome di Client, effettua quotidianamente una richiesta HTTP ad un Server che, prendendo i dati dall'ente che li fornisce, elabora il CSV che li contiene e li inoltra a tutti i Client che ne fanno richiesta.

L'architettura della nostra implementazione di tale modello non si discosta dallo standard della gerarchia del modello Client-Server:

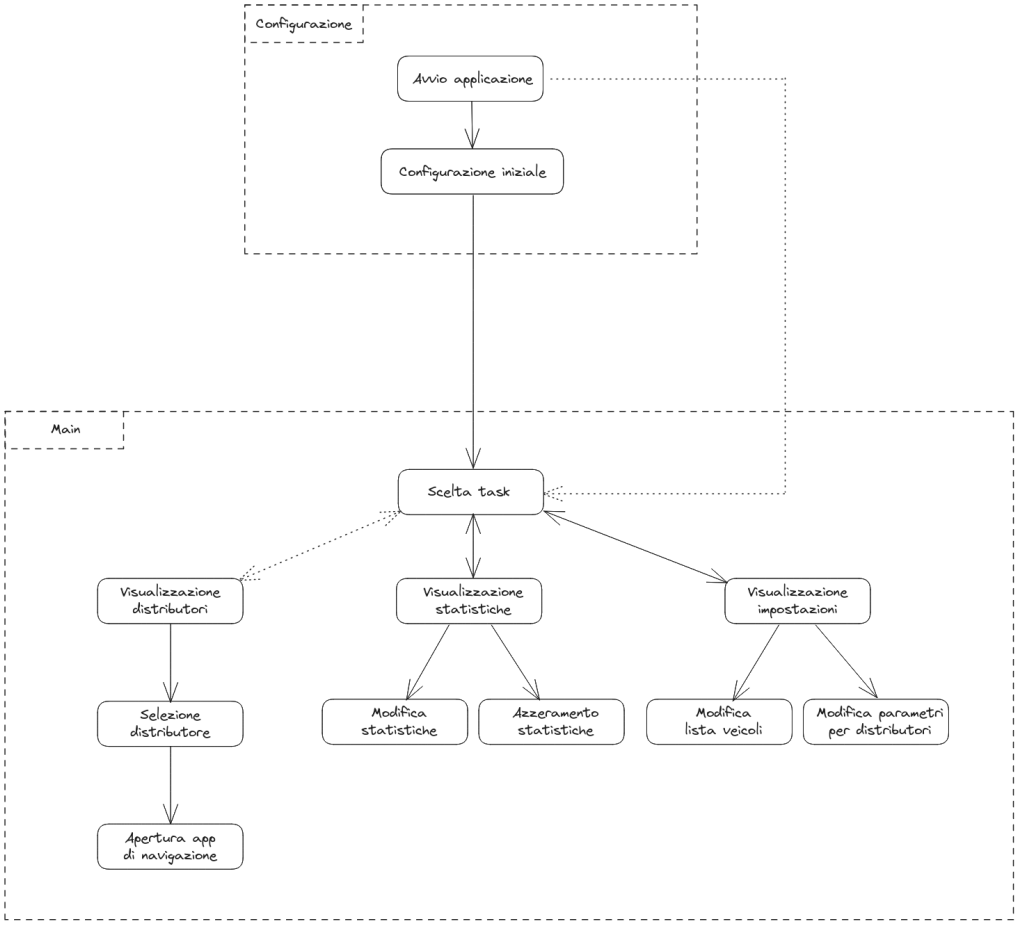
- **Presentation Layer**: il primo strato funzionale ha il compito di fornire un'interpretazione grafica e *user-friendly* dei dati all'utente, tramite un'interfaccia grafica che mostra i distributori ottenuti dal Server, ordinati secondo i parametri impostati dall'utente.
- **Application Processing Layer**: in questo livello avviene il *parsing* dei dati ricevuti dal Server, nonché la loro rielaborazione, filtraggio e trasformazione in istanze di oggetti utili alla comprensione dell'utente.
- **Data Management Layer**: tale livello gestisce lo scambio fisico e il salvataggio delle informazioni tra applicazione e Server.

Lo spunto del salvataggio dei dati accennato nella descrizione del Data Management Layer ci permette di fare una digressione sulla natura dei dati salvati dall'applicazione: GaSmart raccoglie quotidianamente informazioni su carburanti, distributori e veicoli, dati questi che non vengono sostituiti dalla richiesta dei dati del giorno successivo: l'applicazione infatti, in ogni istante, memorizza le informazioni del giorno corrente e di quello precedente, per delineare un andamento di crescita o decrescita dei prezzi nel lasso di tempo di una giornata. Ulteriori dati, come le configurazioni delle impostazioni e le statistiche dell'utente, sono memorizzate in un database realizzato appositamente.

Si aggiunge a ciò la questione della posizione dell'utente. Essa non viene memorizzata dall'applicazione: una volta richiesta, essa viene utilizzata per ordinare i distributori da mostrare ed effettuare una chiamata alle API di Google Maps allo scopo di ottenere il tragitto per il distributore selezionato. Le informazioni sulla posizione vengono richieste all'apertura dell'applicazione e vengono eliminate alla sua chiusura.

4.2 Modello *call-return*

Il controllo del sistema è centralizzato e organizzato secondo il modello *call-return*, che definisce un flusso di controllo gerarchico che parte dalla radice del sistema e prosegue sequenzialmente, procedura per procedura, attraverso un approccio top-down.



Tale modello prevede l'esistenza di due *activity* principali.

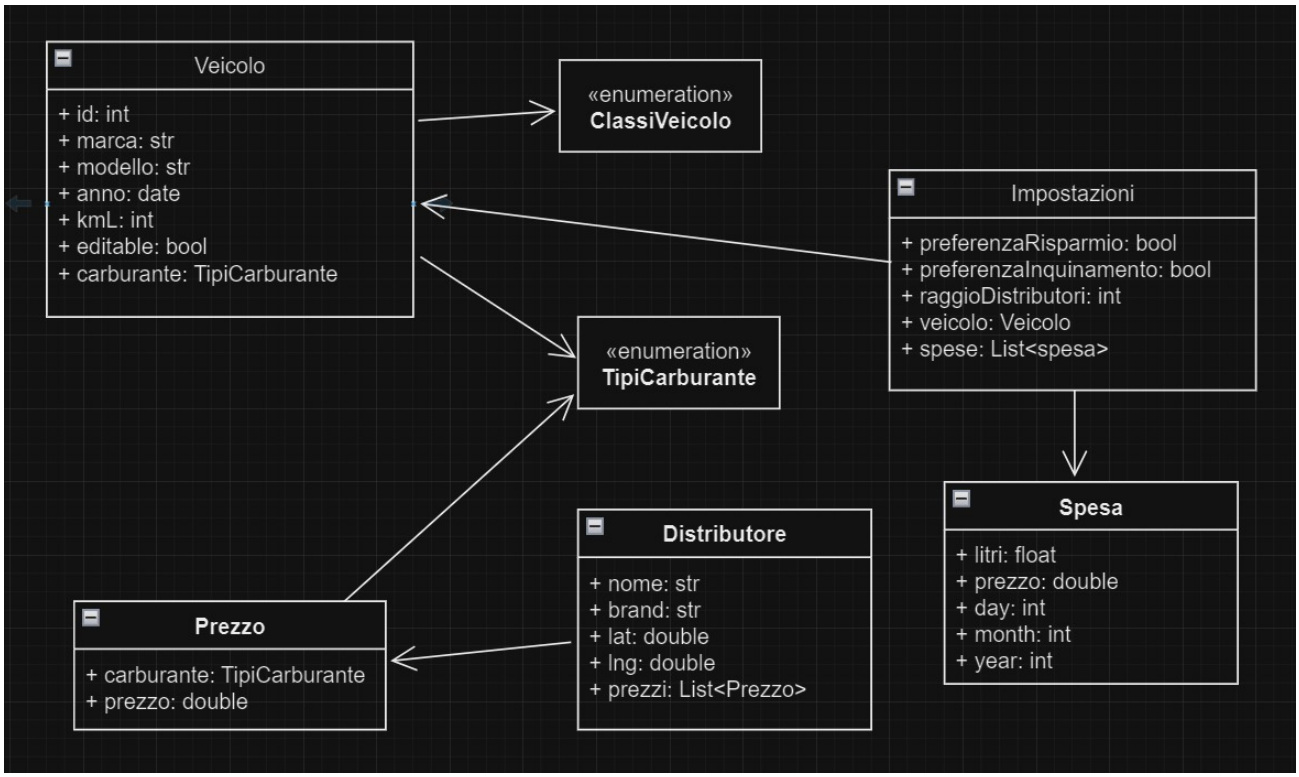
- **Configurazione**: l'applicazione viene configurata sulla base delle informazioni salvate, altrimenti porta l'utente a compiere la configurazione iniziale.
- **Main**: il corpo principale dell'applicazione, con tutte le funzionalità che essa offre.

Le punte delle frecce indicano il verso in cui è permessa l'esecuzione delle procedure, mentre le frecce tratteggiate rappresentano delle operazioni che l'app svolge in automatico: notiamo l'accesso automatico all'*activity* Main, qualora l'utente abbia già effettuato la configurazione iniziale, e la visualizzazione dei distributori, che consiste nella *home page* dell'applicazione.

5 Modelli UML

5.1 Diagramma delle classi

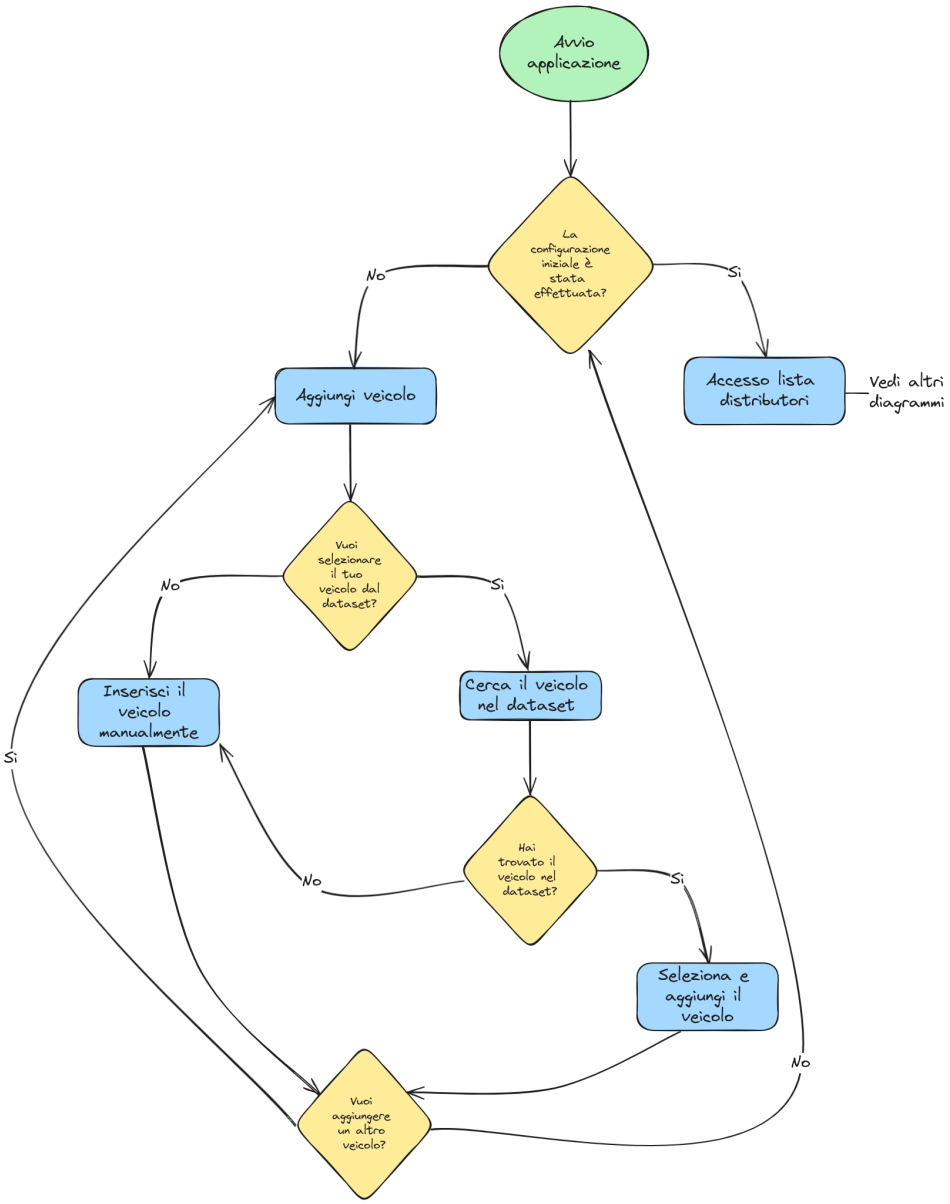
Nel diagramma a seguire andiamo ad illustrare le classi di oggetti della nostra applicazione, ogni classe contiene inoltre i relativi attributi.



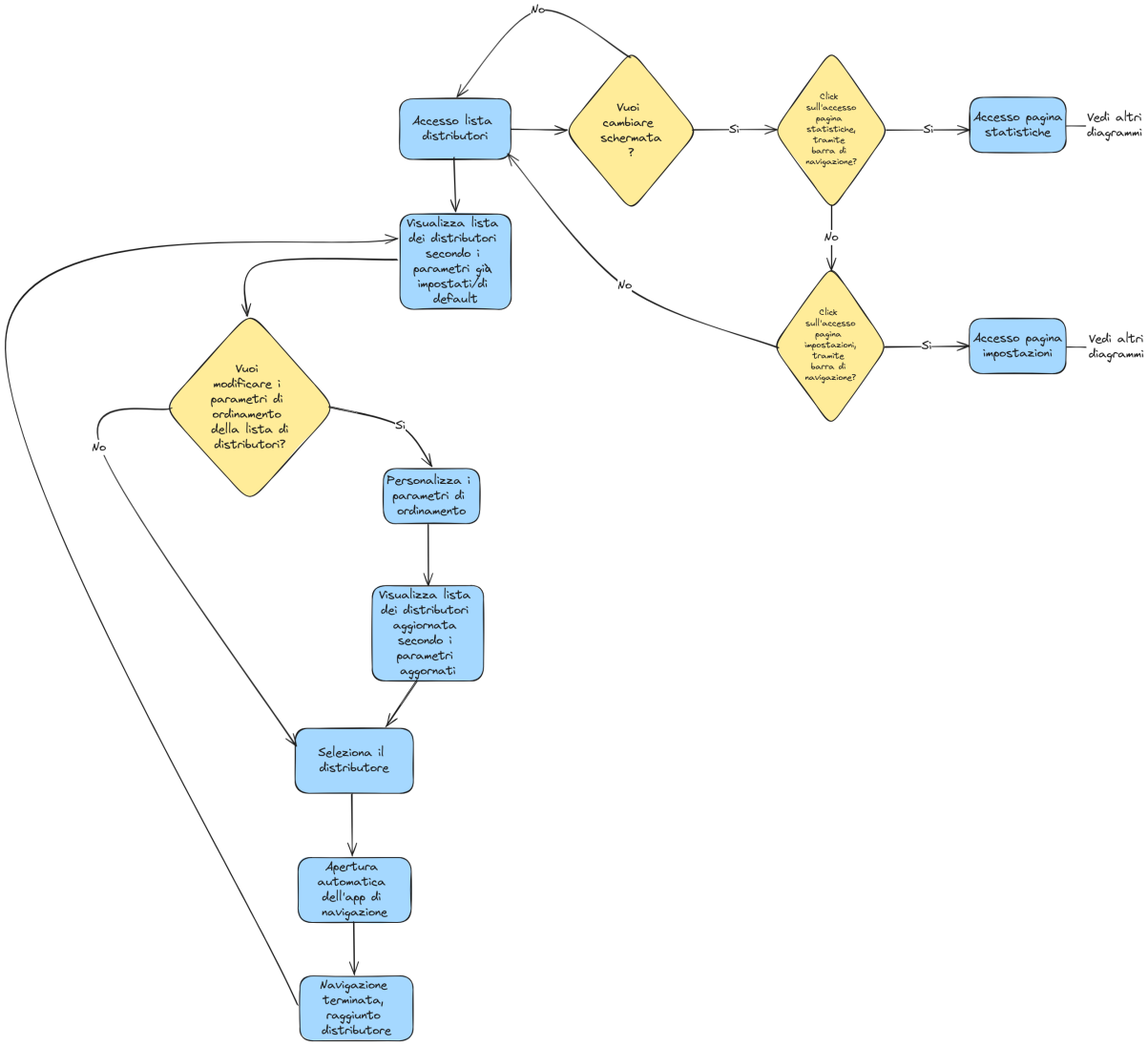
5.2 Diagramma delle attività

Il diagramma delle attività UML è uno strumento essenziale per rappresentare il flusso di processo all'interno di un sistema. Il diagramma serve a illustrare dettagliatamente come varie attività e azioni si interconnettono e si influenzano reciprocamente nel ciclo di funzionamento dell'applicazione. Sebbene per motivi di chiarezza, questi diagrammi possano essere presentati in segmenti separati, è importante sottolineare che, nella realtà, essi costituiscono parti di un unico schema integrato e circolare. Questa suddivisione aiuta a focalizzare l'attenzione su specifiche parti del processo, rendendo più semplice la comprensione e l'analisi di ciascun componente e la sua integrazione con gli altri, ma non deve trarre in inganno riguardo alla loro naturale interconnessione nell'ambito dell'intero sistema.

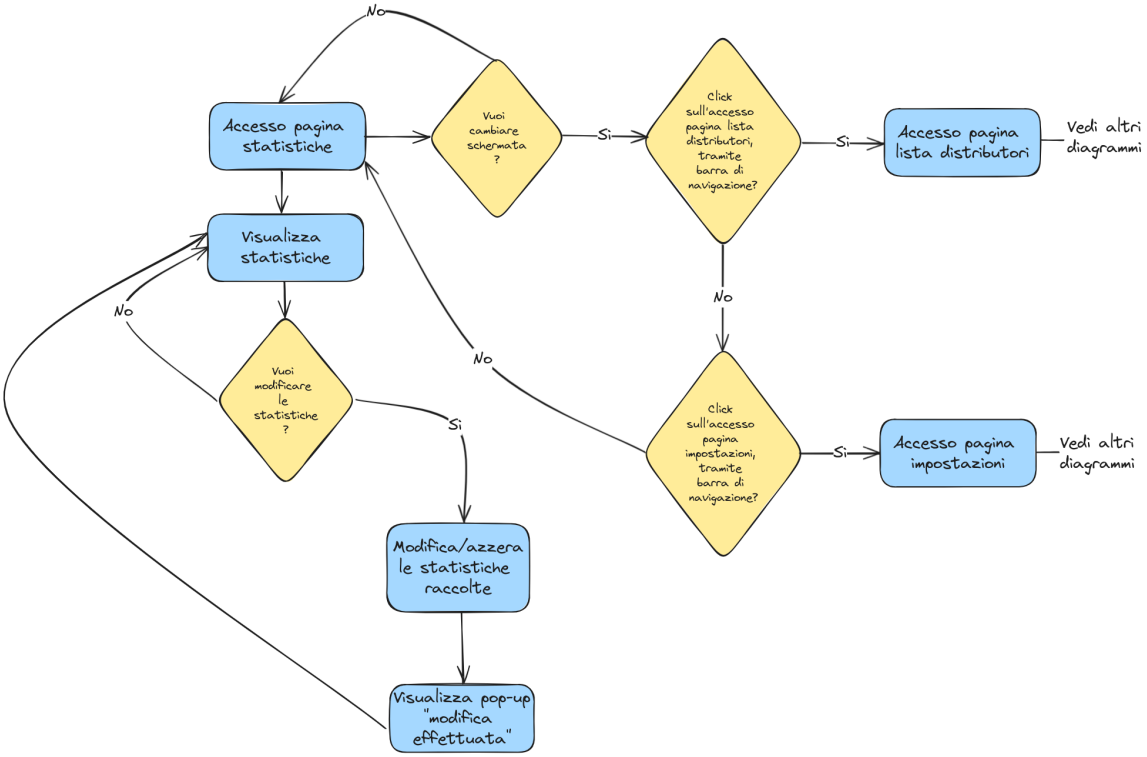
5.2.1 Configurazione iniziale: aggiunta veicolo/i



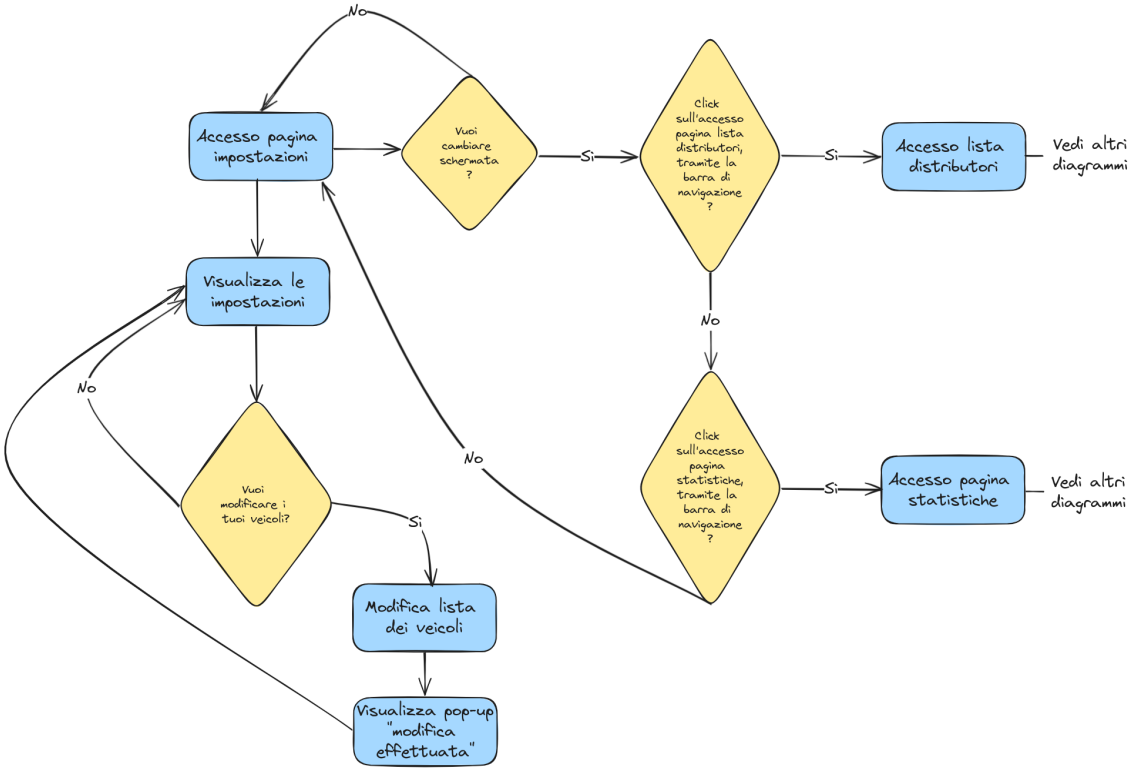
5.2.2 Accesso pagina distributori (schermata principale)



5.2.3 Accesso pagina statistiche



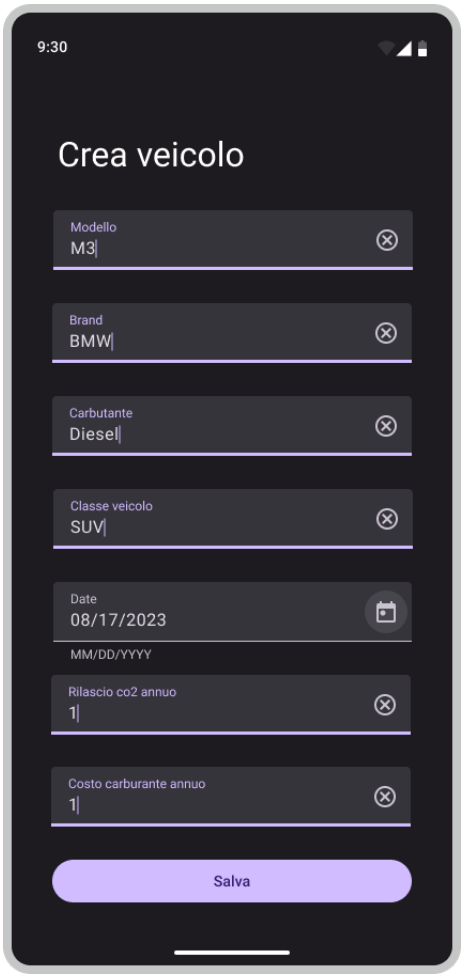
5.2.4 Accesso pagina impostazioni



6 Progettazione dell'interfaccia utente

6.1 Pagina della configurazione iniziale: aggiunta veicolo/i

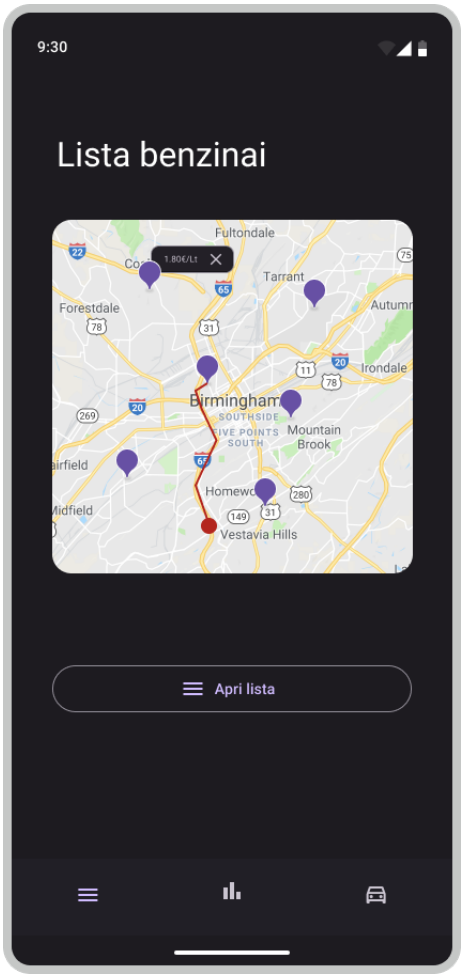
La prima volta che viene avviata l'applicazione è richiesta l'aggiunta di uno o più veicoli, questi possono essere recuperati tramite autocompletamento dal dataset oppure inseriti manualmente.



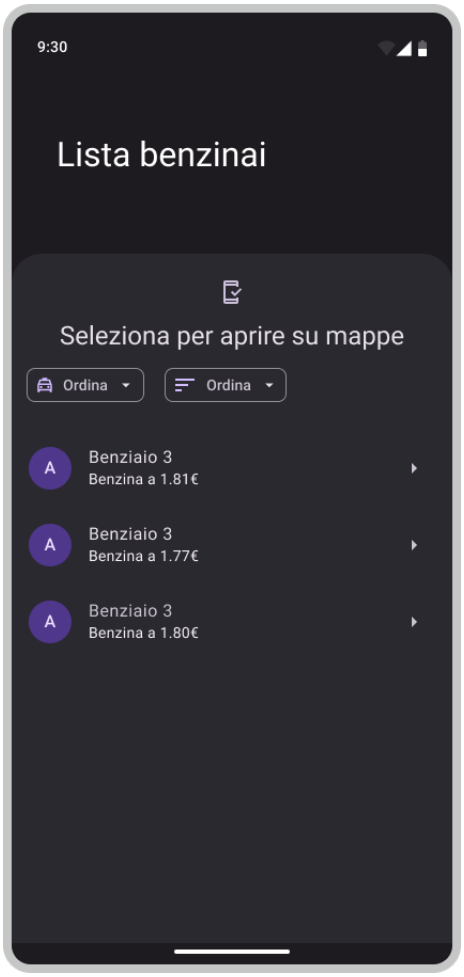
6.2 Pagina dei distributori (schermata principale)

Viene prima visualizzata una mappa che mostra i distributori vicini, poi l'utente può visualizzare l'intera lista.

6.2.1 Visualizzazione veloce

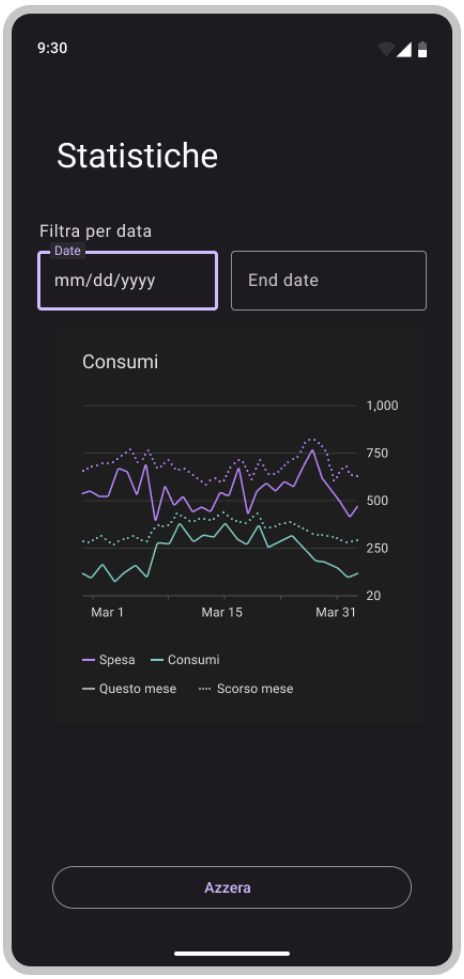


6.2.2 Visualizzazione lista distributori



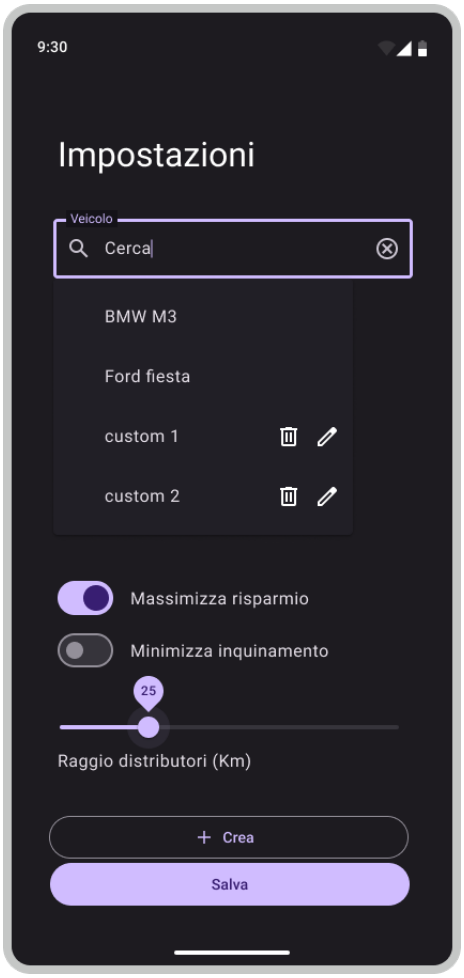
6.3 Pagina delle statistiche

Nella pagina delle statistiche l'utente può visualizzare delle informazioni chiave sui distributori scelti, è inoltre possibile specificare un lasso di tempo per modificare la visualizzazione.



6.4 Pagina delle impostazioni

Nella pagina delle impostazioni l'utente può modificare la lista dei veicoli salvati e i parametri per la visualizzazione dei distributori.



7 Riferimenti

Per la realizzazione di questo documento abbiamo utilizzato:

- le slide del corso, presenti nel rispettivo spazio Moodle
- l'esempio di documento di progettazione presente su Moodle