



Traffic monitor

Ingegneria del software 2018/2019

Gruppo 6:
Mazzoleni Sara
Megler Federico
Paone Angelo

Indice

Uml

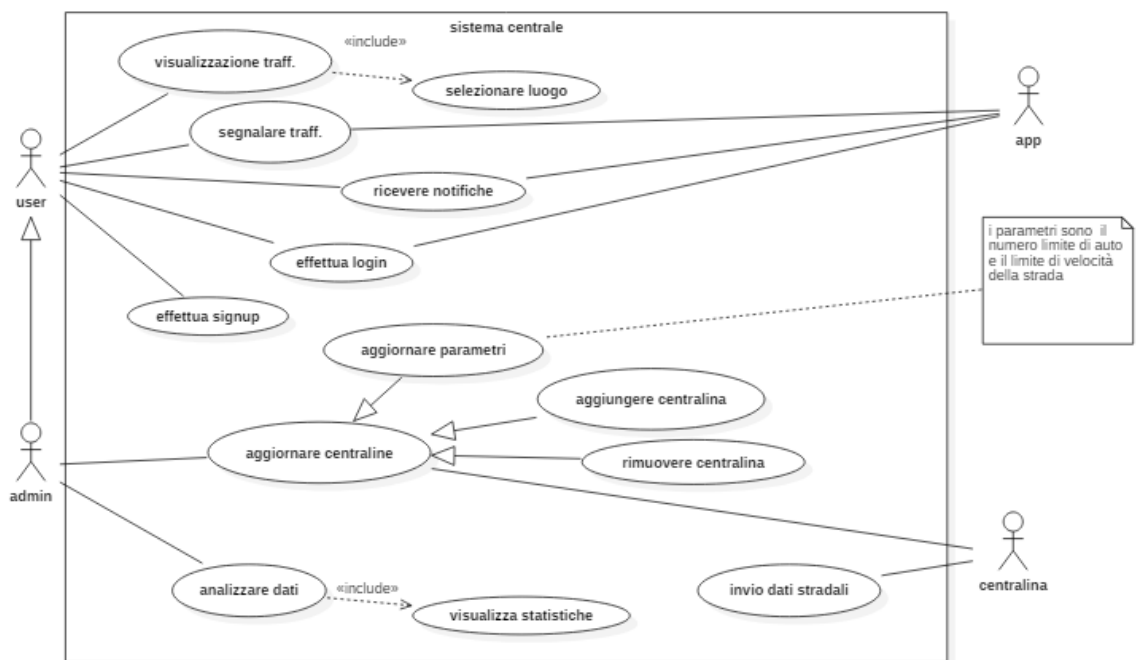
• Use Case Diagram	pag. 3
• Class Diagram	pag. 4
• Object Diagram – Aggiornamento Centralina	pag. 5
• Object Diagram – Applicazione	pag. 6
• State Diagram – Centralina	pag. 7
• State Diagram – Segnalazione	pag. 8
• Activity Diagram – Accesso Sistema Utente	pag. 9
• Activity Diagram – Accesso Sistema Admin	pag. 10
• Activity Diagram – Inserimento Segnalazione	pag. 11
• Activity Diagram – Invio Notifica	pag. 12
• Sequence Diagram – Accesso Sistema Utente	pag. 13
• Sequence Diagram – Funzionamento Centralina	pag. 14
• Sequence Diagram – Invio segnalazione	pag. 15
• Collaboration Diagram – Interazione Notifica	pag. 16
• Component Diagram	pag. 17
• Deployment Diagram	pag. 18

Uml

UseCase Diagram

Per iniziare, abbiamo identificato i quattro attori ed individuato tutte le attività ad essi collegate.

L'**utente**, dopo essersi registrato e aver effettuato il login, deve poter visualizzare lo stato del flusso automobilistico, inoltre, attraverso l'app mobile, deve poter fare eventuali segnalazioni sul traffico e ricevere notifiche in base alla sua posizione. L'**admin** ha, invece, il compito di aggiornare le centraline e di analizzare i dati relativi al traffico, visualizzando le statistiche. La **centralina** deve installare gli aggiornamenti ricevuti ed inviare dati stradali al sistema centrale. L'**app mobile** deve permettere all'utente di effettuare segnalazioni sul traffico e ricevere notifiche ad esso relative.



Class Diagram

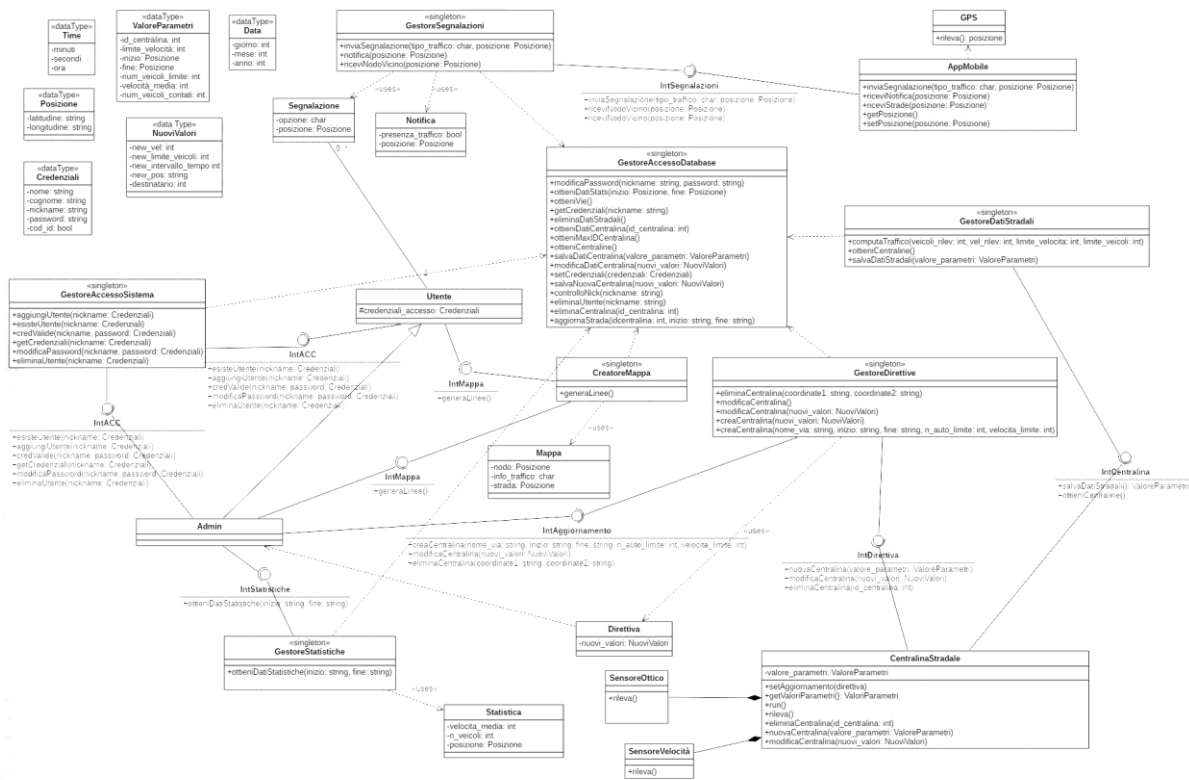
Partendo dallo use case diagram abbiamo creato il class diagram generale del sistema. Sono state individuate le seguenti classi fondamentali:

- **Utente, Admin** che rappresentano gli utilizzatori del sistema;
- **Segnalazione, Notifica** che rappresentano lo scambio di informazioni tra **AppMobile** e il sistema centrale;
- **Mappa** che rappresenta lo strumento grafico su cui si potrà analizzare il traffico cittadino
- **Statistica**, accessibile solo all'admin, fornisce in forma grafica gli andamenti giornalieri e settimanali del traffico;
- **Direttiva** che rappresenta le indicazioni sul nuovo funzionamento delle centraline deciso dall'admin;
- **CentralinaStradale** che rappresenta il mezzo fisico che si occupa di raccogliere i dati.

Ad esse correlate abbiamo inoltre creato delle classi gestori singleton, ognuna connessa con le opportune interfacce, i gestori sono:

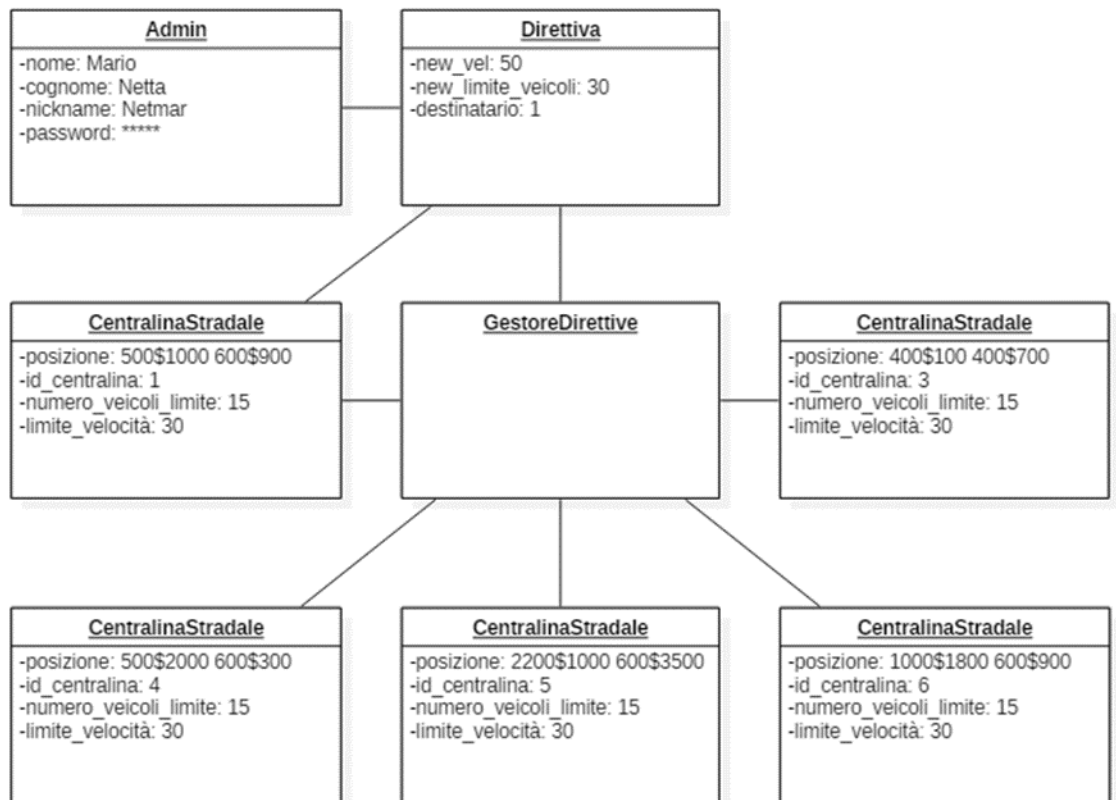
GestoreAccessoSistema, GestoreSegnalazioni, CreatoreNotifica, CreatoreMappa, GestoreAccessoDatabase, CreatoreStatistiche, GestoreDirettiva, e GestoreDatiStradali.

Abbiamo anche creato i seguenti datatype che fungono da strutture di supporto per la descrizione del sistema: **Time**, **Data**, **ValoreParametri**, **Posizione**, **Credenziali** e **NuoviValori**.



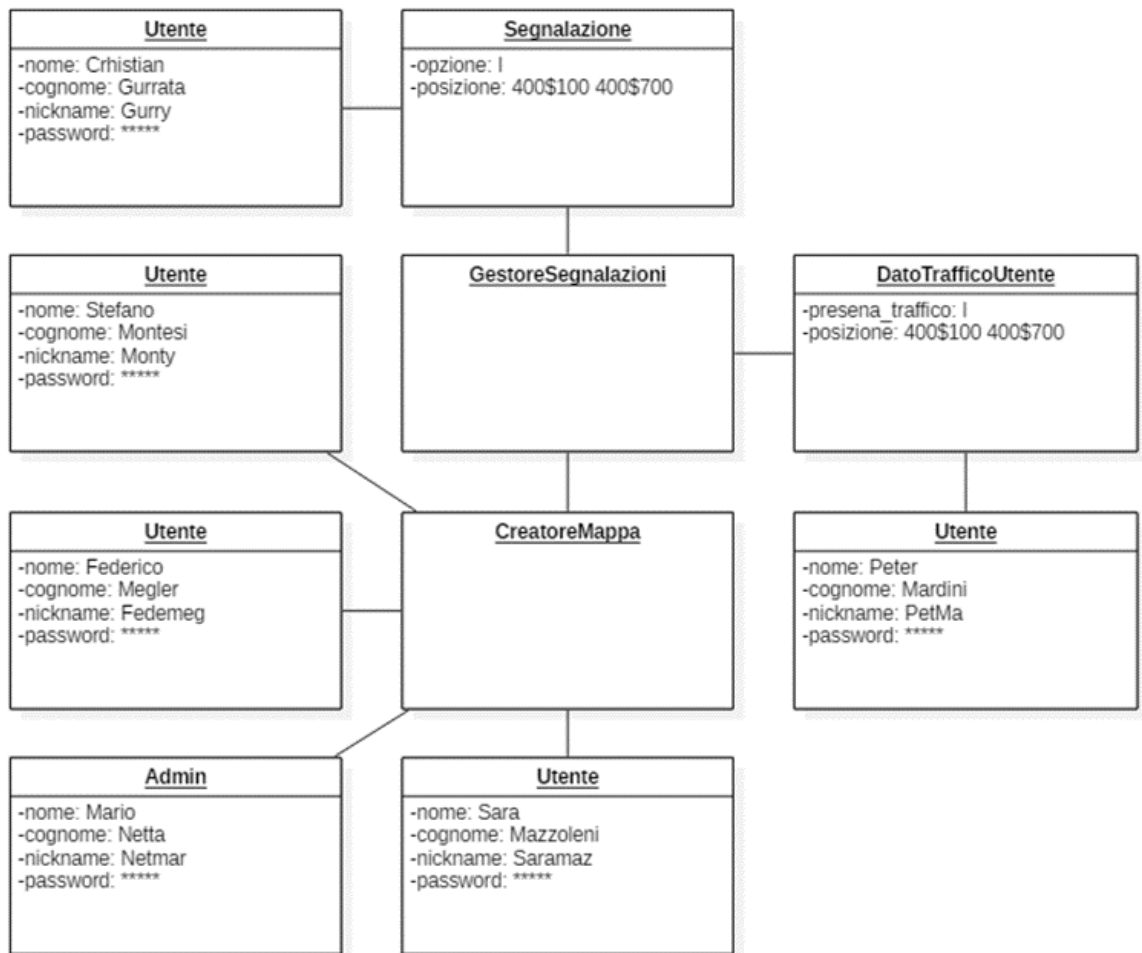
Object Diagram – Aggiornamento Centralina

Per quanto riguarda l'aggiornamento della centralina, sono stati identificati i seguenti oggetti: **Direttiva**, **Admin**, **Centralina**, **GestoreDirettive**, **CentralinaStradale**.



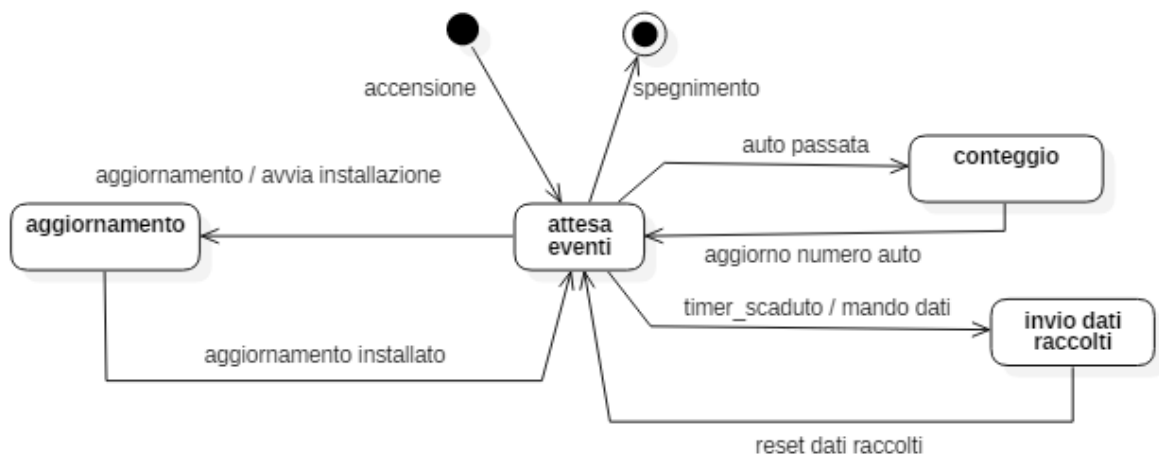
Object Diagram – Applicazione

Per quanto riguarda l'applicazione, sono stati identificati i seguenti oggetti: **Utente**, **Segnalazione**, **GestoreSegnalazioni**, **Utente**, **CreatoreMappa**, **DatoTrafficoUtente**, **Admin**.



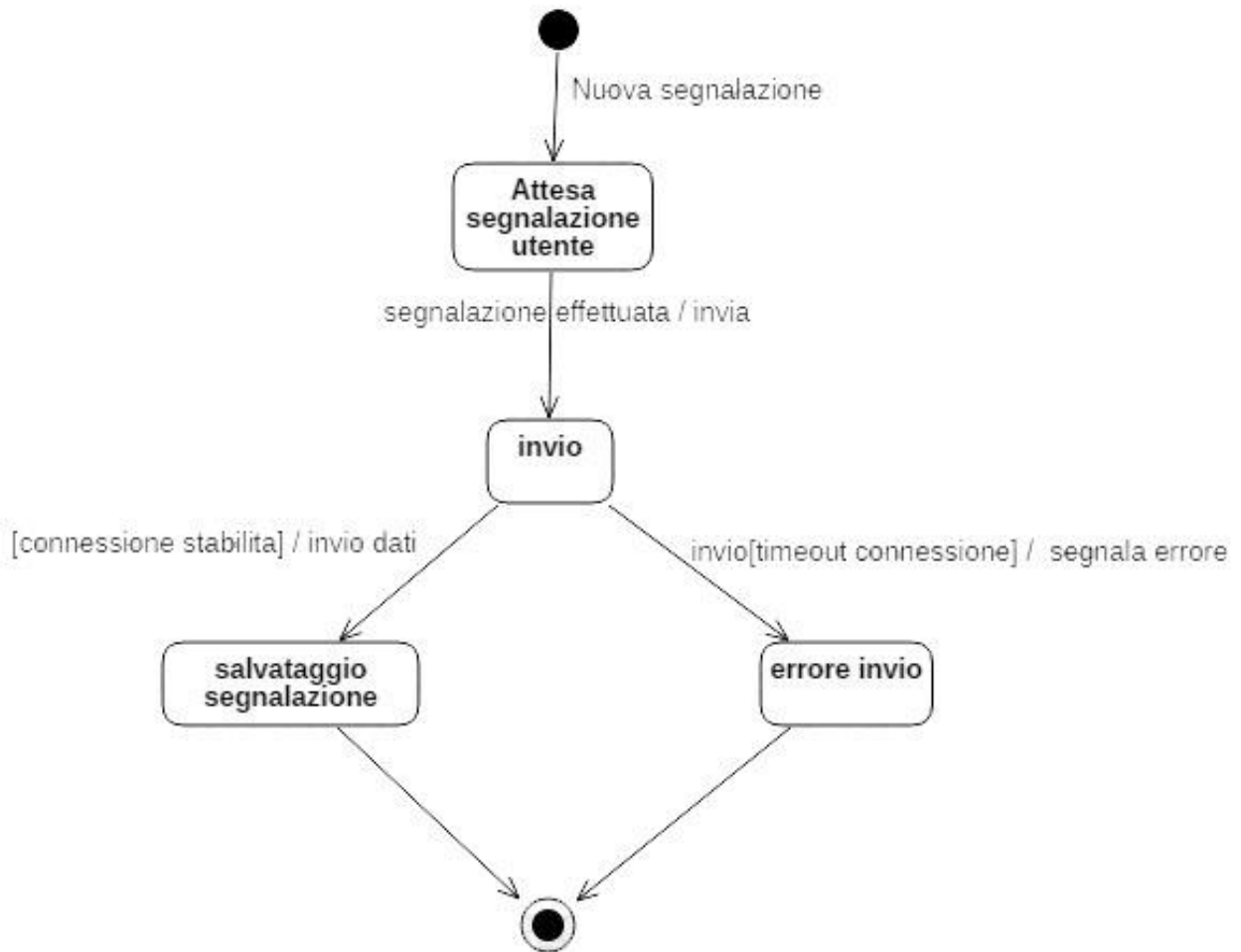
State Diagram – Centralina

La centralina si trova inizialmente in uno stato di attesa, che può essere interrotto in base alla necessità di installare un aggiornamento, di incrementare il conteggio delle auto, o di inviare dei dati al sistema in base alla scadenza di un timer preimpostato.



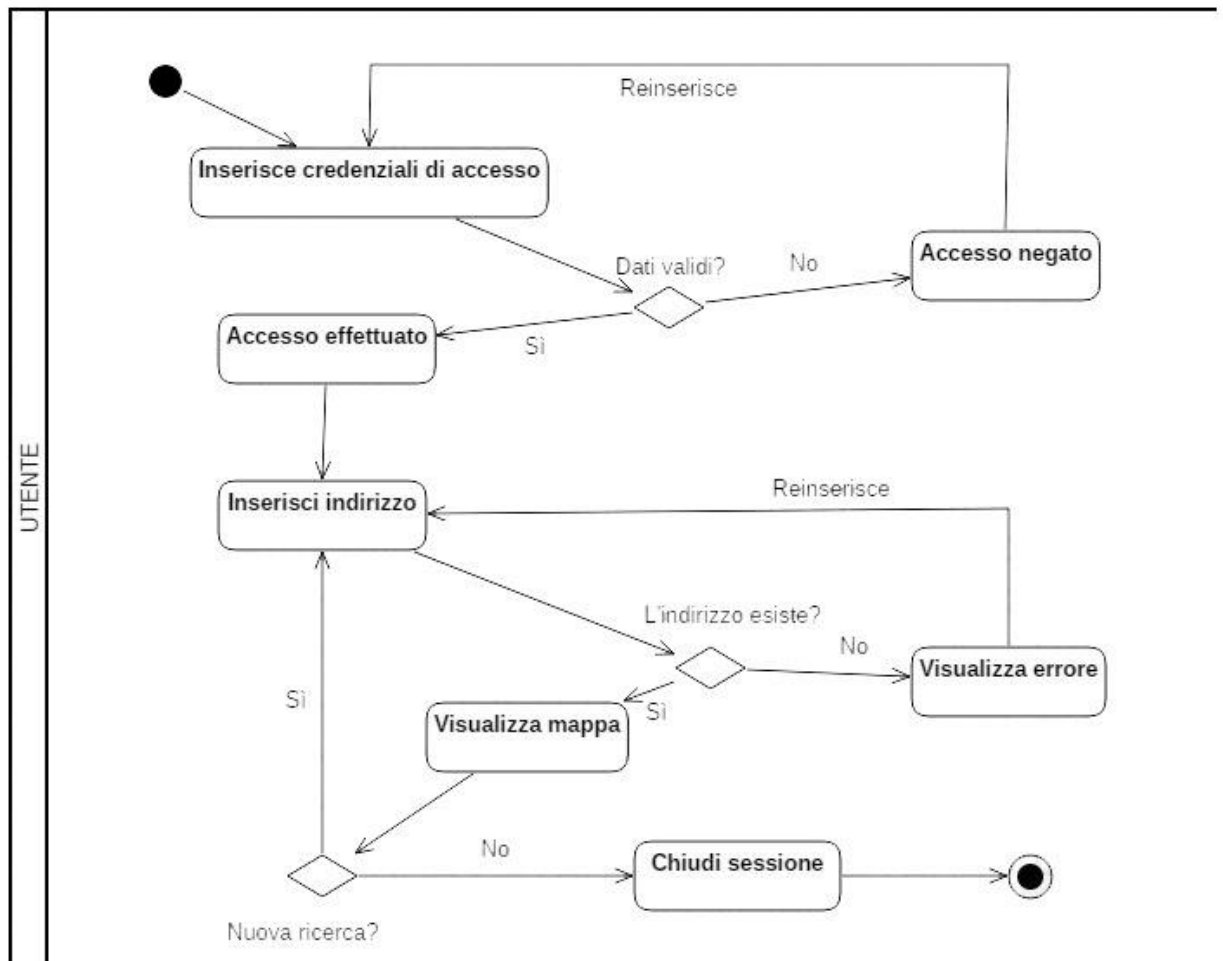
State Diagram – Segnalazione

Alla creazione di una segnalazione, ci si trova inizialmente in stato di attesa, da cui si esce nel momento in cui essa viene effettuata, per poi essere inviata. Si può ora verificare un errore di connessione che comporta un errore dell'invio, o un invio dei dati con una connessione stabilita correttamente, che porta al salvataggio della segnalazione.



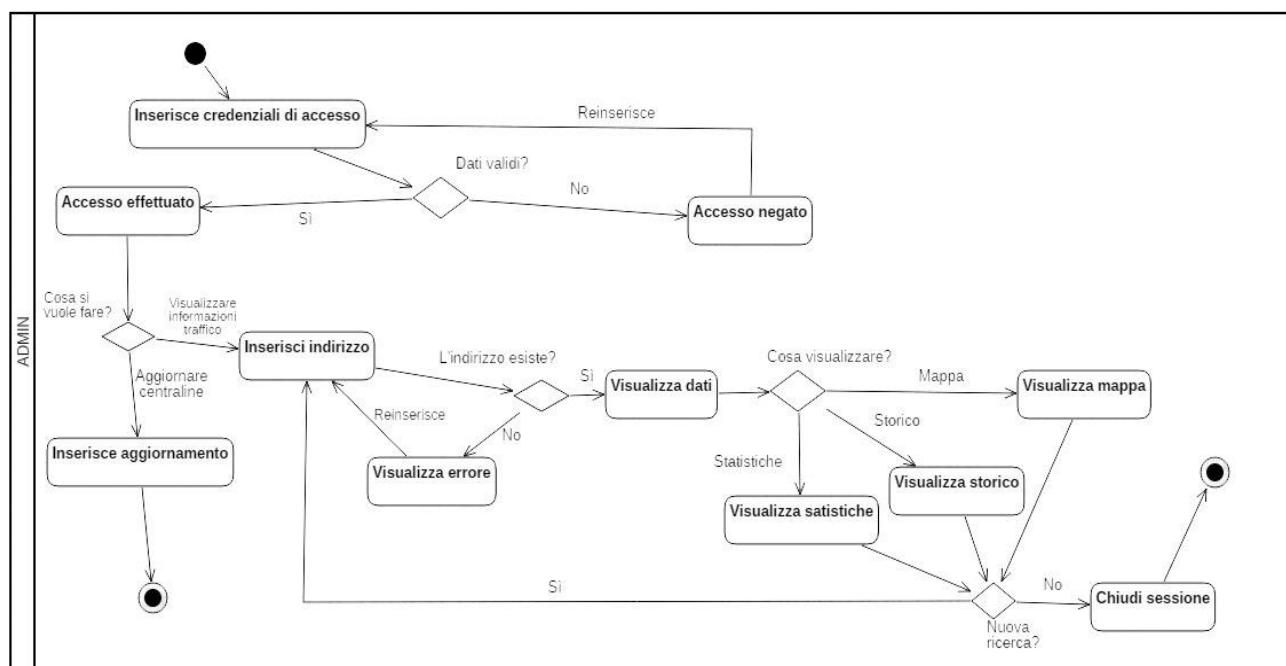
Activity Diagram – Accesso Sistema Utente

L'utente effettua l'accesso al sistema e, inserendo un indirizzo, può ottenere le informazioni relative al traffico nella via interessata tramite una mappa, in seguito si disconnette.



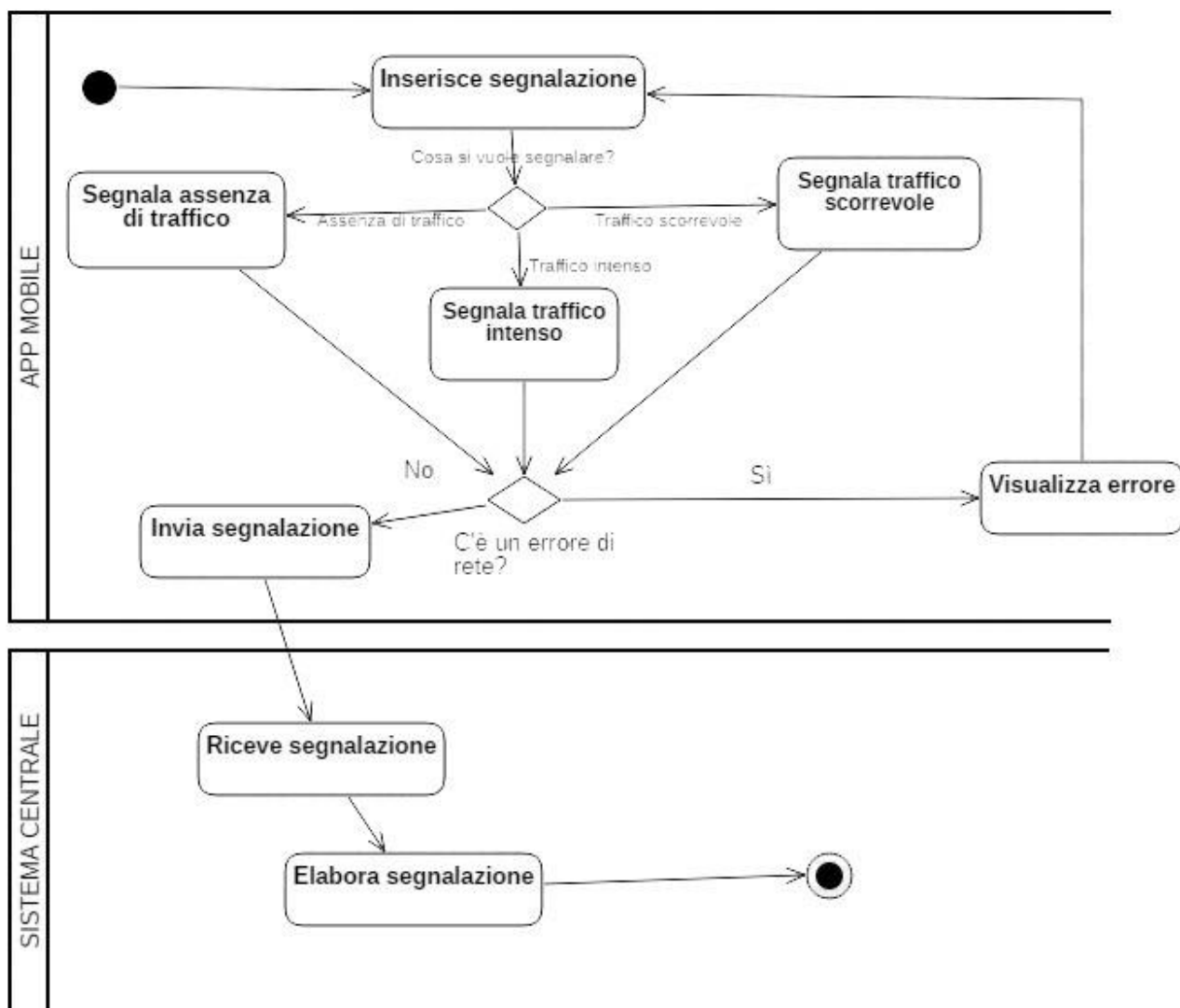
Activity Diagram – Accesso Sistema Admin

L'admin effettua l'accesso al sistema e può vedere i dati a lui riservati oppure inserire un aggiornamento per le centraline, in seguito si disconnette.



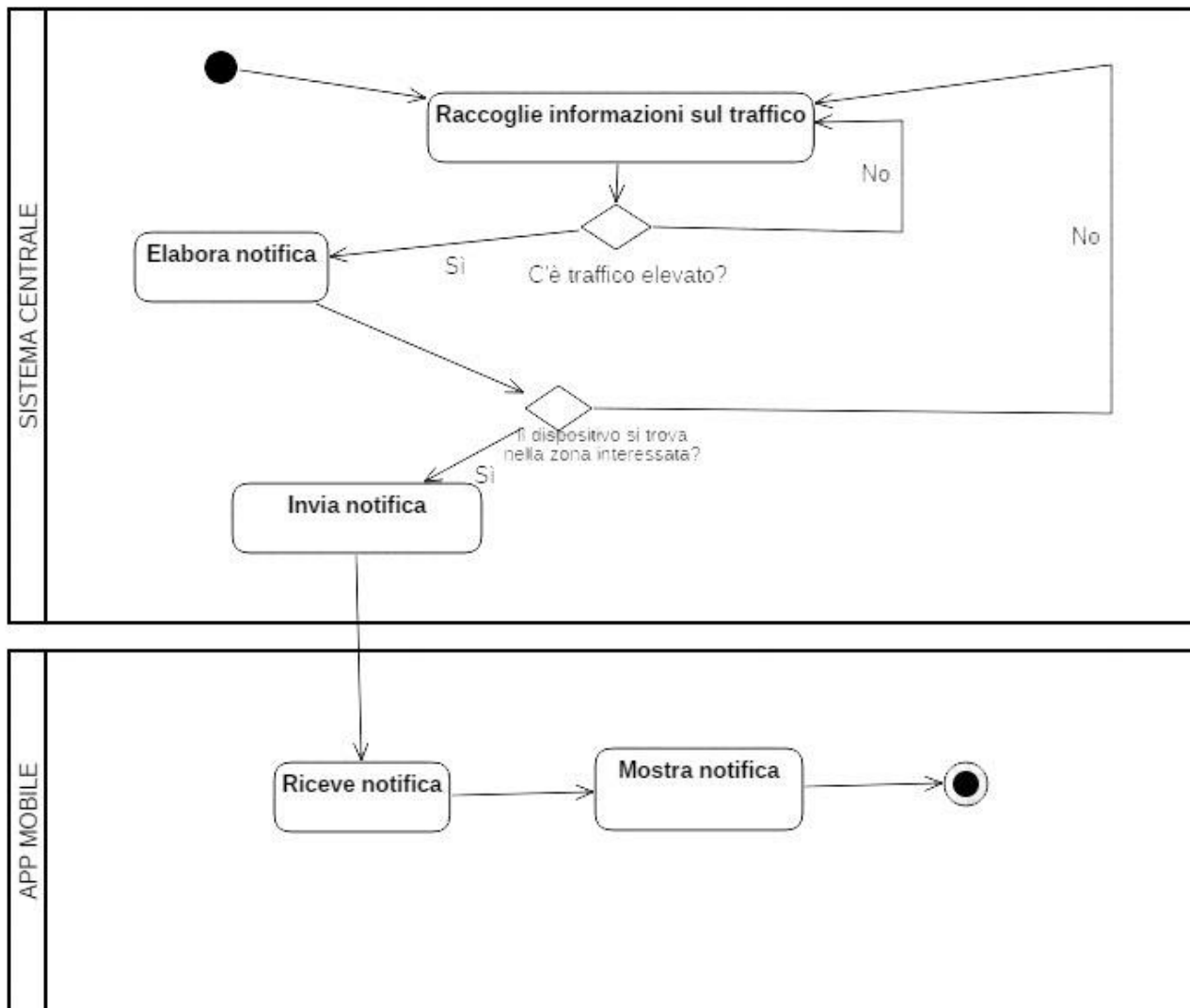
Activity Diagram – Inserimento segnalazione

L'applicazione riceve dall'utente una segnalazione sulla situazione del traffico (assente, scorrevole o intenso) e la invia al sistema centrale, che la riceve e la elabora.



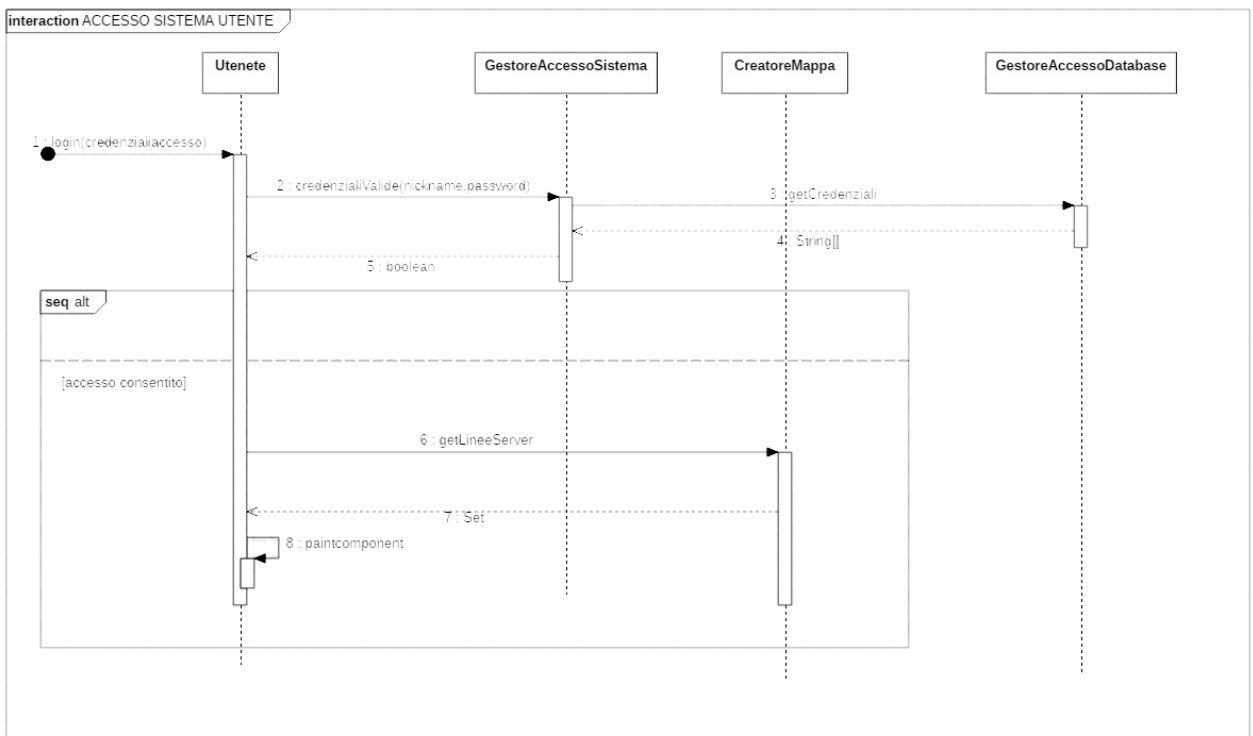
Activity Diagram – Invio notifica

Ricevuti tutti i dati stradali, il sistema centrale li elabora e, se rileva situazioni critiche di traffico, manda una notifica agli utenti che utilizzano l'applicazione mobile nella zona interessata, questi la ricevono e la visualizzano.



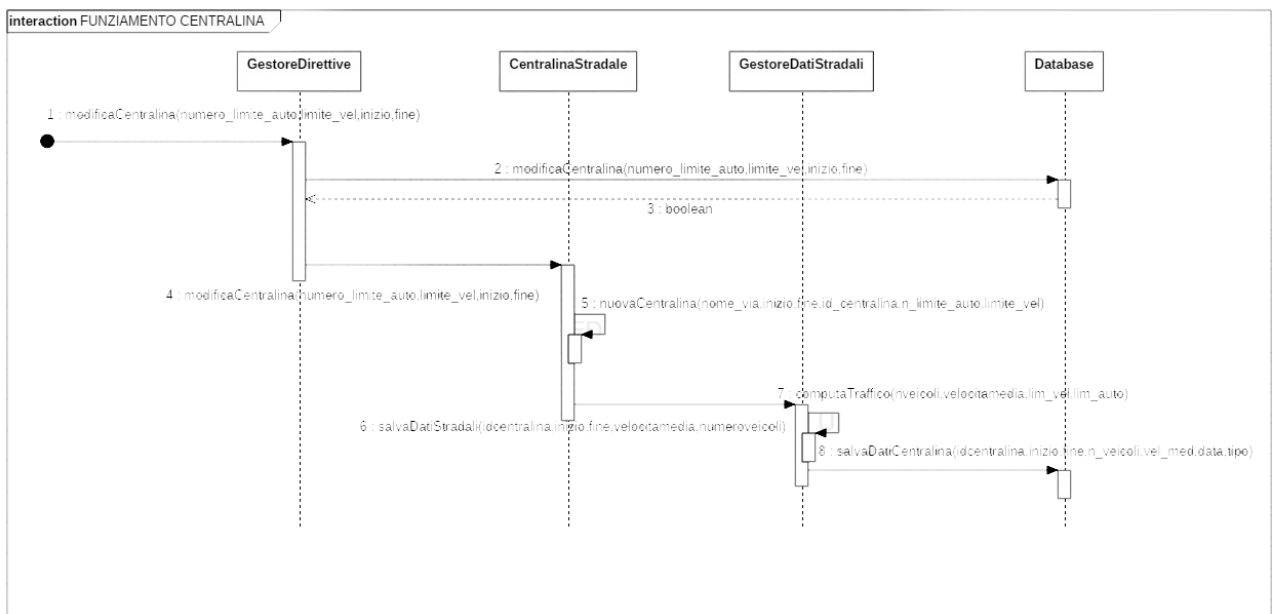
Sequence Diagram – Accesso Sistema Utente

In questo diagramma descriviamo il comportamento del sistema quando un utente si collega ad esso. In particolare, grazie al login, l'utente può effettuare più ricerche in un'unica sessione.



Sequence Diagram – Funzionamento Centralina

In questo caso vengono spiegati i passaggi dal rilevamento dei dati da parte della centralina, fino al salvataggio delle informazioni raccolte.

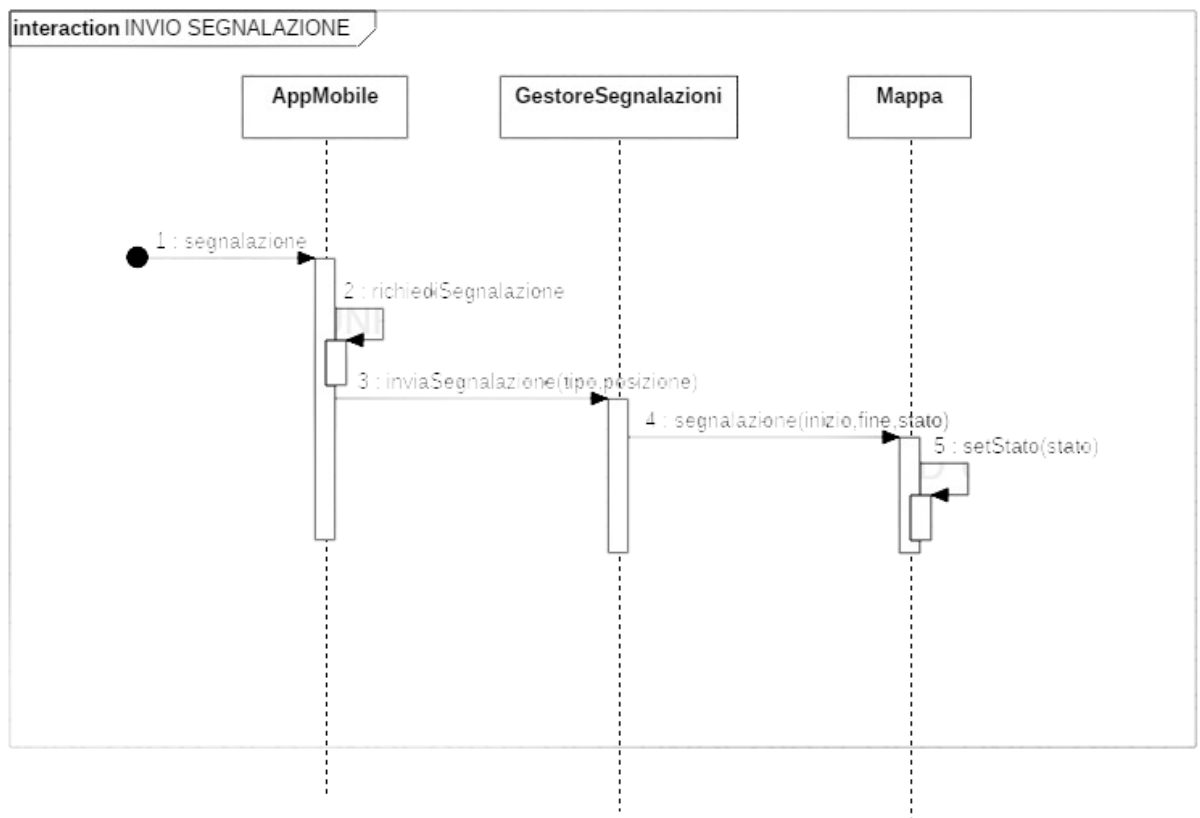


Sequence Diagram – Invio segnalazione

Gli oggetti coinvolti in questo processo sono:

- AppMobile
- Gestore Segnalazioni
- Database

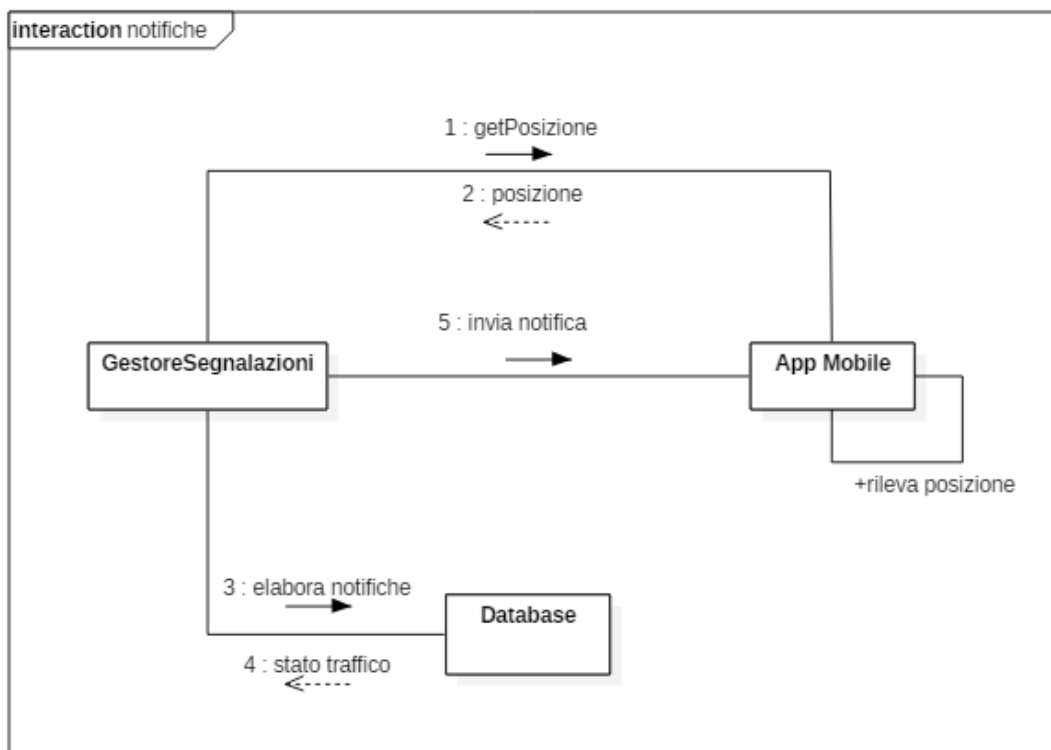
L'utente effettua una segnalazione che viene elaborata dal **Gestore segnalazioni**, il quale si occupa infine di salvare tutti i dati nel **Database**.



Collaboration Diagram – Interazione notifica

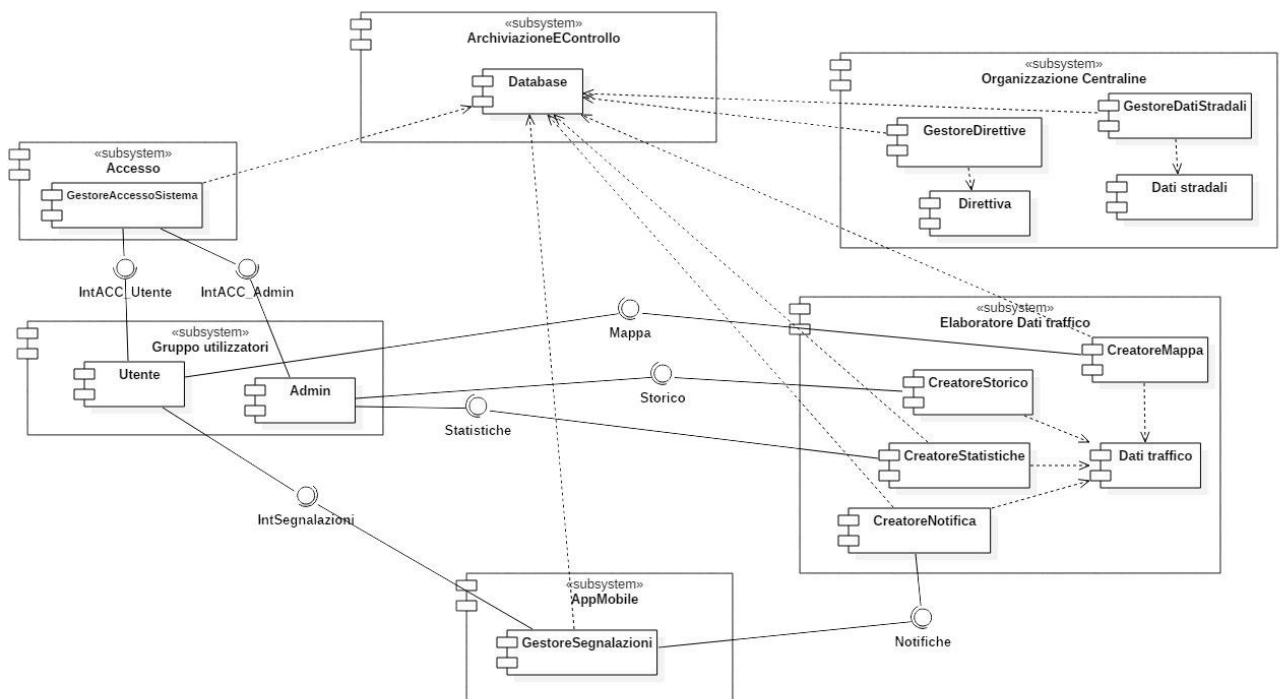
Nel diagramma sono riportate le varie cooperazioni del sistema al fine di creare una notifica.

In particolare, l'**AppMobile** rileva la propria posizione, la invia al **GestoreSegnalazioni**, il quale elabora la notifica ricevendo le relative informazioni sul traffico dal **Database**, e la invia all'app.



Component Diagram

Abbiamo diviso il sistema centrale in più sottosistemi che si dividono i compiti per la gestione dell'ecosistema. Tramite questo diagramma delle componenti descriviamo le relazioni tra i vari sottosistemi e le relative interfacce.



Deployment Diagram

Nel seguente diagramma mostriamo la distribuzione del programma nelle macchine fisiche. Esse fanno riferimento al sistema centrale utilizzando connessioni TCP/IP.

