

#swEngComo

Traffic Monitor

A.A. 2018/2019



POLITECNICO
MILANO 1863

Altieri Matteo	869296
Genesi Lorenzo	868898
Marforio Simone	867995

Tema di progetto	5
Studio di fattibilità	6
Committente	6
Obiettivo	6
Potenziali benefici	6
Risorse necessarie	6
Consegne	7
Potenziali ostacoli	7
Rischi	7
Ipotesi e assunzioni - specifica requisiti	8
Sintesi dei goal - I*	9
Diagramma: modulo SDM	9
Descrizione	9
Modulo SRM - Utente e Applicazione	10
Modulo SRM - Sistema Centrale	11
Modulo SRM - Centraline automobilistica e centralina stradale	12
Data Dictionary	13
Ipotesi e assunzioni - design	15
Class Diagram	17
Immagine	17
Descrizione	17
Object Diagram	20
Immagine	20
Descrizione	20
Activity Diagram	22
Activity centralina	22
Diagramma	22
Descrizione	22
Activity applicazione	24
Diagramma	24
Descrizione	24
State Diagram	25
State Centralina	25
Diagramma	25

Descrizione	25
State Applicazione	26
Diagramma	26
Descrizione	26
Sequence Diagram	27
Sequence centralina stradale	27
Diagramma	27
Descrizione	27
Sequence evento-notifica	28
Diagramma	28
Descrizione	28
Collaboration Diagram	28
Collaboration centralina stradale	29
Diagramma	29
Descrizione	29
Component Diagram	30
Diagramma	30
Deployment Diagram	31
Diagramma	31
Descrizione	31
Use Case Diagram	32
Use case utente	32
Diagramma	32
Descrizione	32
Struttura sistema	33
Componenti	33
Metodo di comunicazione	33
Database	33
IDE	33
Risorse esterne	33
Funzionamento	34
Monitoraggio del traffico con centraline stradali	35
Progetto centralina stradale	35
Progetto sistema-centrale	35
Progetto applicazione	37
Progetto sistema-centrale	37
Avviso cambiamento situazione traffico per gli utenti interessati tramite notifica	39

Progetto sistema-centrale	39
Progetto applicazione	39
Monitoraggio dello stato di funzionamento delle centraline e diagnostica	40
Progetto centralina stradale	40
Progetto sistema-centrale	40
Visualizzazione mappa con stato di traffico di ogni segmento, stato di funzionamento centraline e storico dati e segnalazioni	41
Progetto interfaccia-utente	41
Casi d'uso	42
Introduzione	42
Setup	42
Avvio	42
Funzionamento	43
Centralina - simulazione traffico e guasti	43
Notifica	44
Interfaccia utente - visualizzazione traffico, guasti e storico	45
Applicazione - segnalazione	46
Sistema centrale - feedback	48
Cambiamenti rispetto alle consegne	49

Requisiti

Tema di progetto

Realizzare un sistema per il monitoraggio e il controllo integrato del traffico cittadino, composto dai seguenti sotto-sistemi che operano in modo distribuito:

-**Sistema centrale:** incaricato di memorizzare tutte le informazioni di stato, inviare notifiche a sistemi esterni in caso di specifici eventi, mostrare lo stato dell'intero sistema e sottosistemi. Il sistema quindi include una interfaccia utente che consente di esplorare le varie informazioni attuali.
Opzionale: è possibile decidere di mostrare i dati anche in un qualche tipo di forma grafica (diagrammi, mappe. ecc.).

-**Centraline stradali:** incaricate di monitorare il flusso di traffico del segmento stradale in cui collocate e inviarlo al sistema centrale con periodicità proporzionale all'ammontare di traffico

-**Centraline automobilistiche:** incaricate di inviare con periodicità fissa il dato di velocità (e posizione) del veicolo su cui sono installate

-**Applicazioni mobili:** installate su telefono cellulare e incaricate di inviare al sistema centrale esplicite segnalazioni di traffico (coda, con posizione GPS) da parte degli utenti / guidatori. Le applicazioni inoltre ricevono notifiche dal sistema centrale per qualsiasi evento di traffico (coda, velocità lenta, traffico elevato) in un raggio fisso dalla posizione (ultima registrata) del telefono

Specificare, progettare e implementare il sistema distribuito necessario, coprendo: sistema centrale, applicazione mobile, e una a scelta tra centralina stradale e centralina automobilistica. Definire esplicitamente tutti i formati dei dati scambiati e le modalità di scambio (protocollo). E' possibile raffinare i requisiti ed aggiungere ipotesi e assunzioni sul contesto, sensate e in linea con quanto indicato nei requisiti. Tali estensioni devono essere esplicitamente riportate nella documentazione di progetto (sezione specifica requisiti).

Studio di fattibilità

Committente

Il professore/committente, colui che ci ha fornito il progetto con i relativi requisiti e le scadenze da rispettare.

Contestualizzato nel mondo reale, il committente potrebbe essere un comune che vuole monitorare il traffico sul territorio e fornire un servizio aggiuntivo ai suoi cittadini, o un ente privato a cui è stato affidato il compito di gestire le strade.

Obiettivo

L'obiettivo è progettare, sviluppare e dimostrare il funzionamento di un sistema di monitoraggio e controllo del traffico con relativi sottosistemi, documentazione allegata e rispettando le deadline imposte dal committente.

Potenziali benefici

- Capacità di controllare lo stato di traffico sulle strade.
- Ottenere di conseguenza informazioni e dati statistici su cui costruire soluzioni per migliorare il traffico sulle strade.
- Benefici per gli utenti del sistema che possono ad esempio evitare strade saturate.
- Acquisizione dei metodi di progettazione e sviluppo da parte degli studenti incaricati del lavoro.
- Ottenimento dei 3 CFU associati alla consegna del progetto con il voto migliore possibile.

Risorse necessarie

Progettisti e sviluppatori: 3 studenti che si occupano dell'intera fase di analisi, progettazione, sviluppo, testing e stesura della documentazione. Gli studenti dovranno applicare le conoscenze acquisite durante gli anni di studi, in particolare quelle fornite dal corso di "Ingegneria del software" (non ancora terminato al momento di inizio del progetto).

Dispositivi e strumenti: computer con i software appositi utilizzati per completare l'obiettivo sopra indicato.

Personale: disponibilità dei tutor per revisionare le fasi di lavoro ed avere un riscontro costante su quanto fatto.

Tempo di lavoro: bisogna rispettare le date delle scadenze (riportate sotto). Per questo è necessario un impegno stimato di almeno 12 ore settimanali. In fase di implementazione sarà probabilmente necessario aumentare il tempo settimanale da dedicare allo sviluppo.

Consegne

1. 30/10/2018 - Consegna requisiti
2. 27/11/2018 - Consegna UML
3. 18/12/2018 - Consegna Java

Potenziali ostacoli

- Poca esperienza in materia di sviluppo e progettazione da parte degli studenti
- Problemi dovuti all'utilizzo di particolari software per la prima volta
- Impegni universitari non prorogabili dovuti alla partecipazione ad altri corsi

Rischi

- Non rispettare i tempi di consegna
- Non riuscire a rispettare i requisiti richiesti

In particolare, il primo può essere una conseguenza del secondo. Per minimizzare i rischi è necessario lavorare in gruppo in modo coordinato, per poter supervisionare costantemente quanto fatto. Inoltre, si deve fare affidamento alle sessioni di tutoraggio per avere dei feedback costanti ed apportare subito eventuali modifiche prima di proseguire con il progetto.

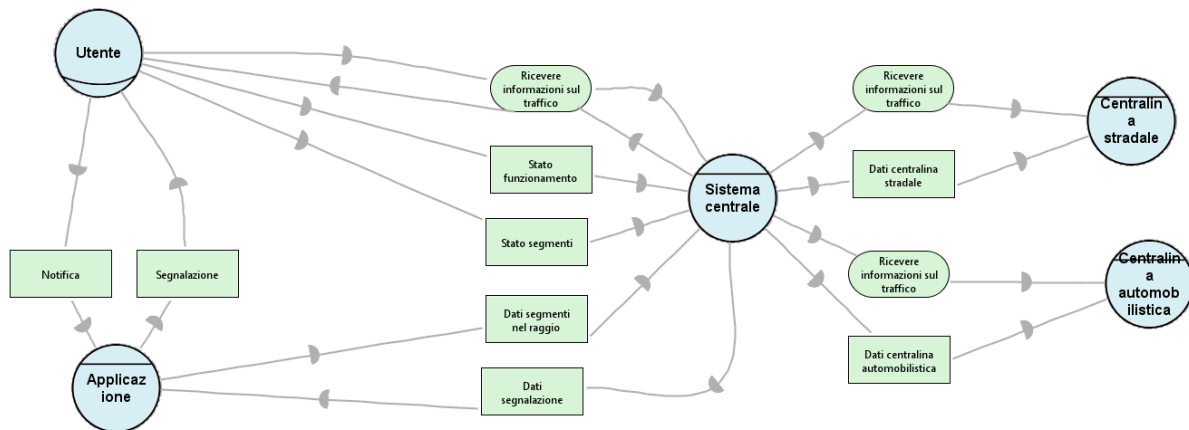
Tutti i rischi derivanti da una contestualizzazione nel mondo reale (come perdite finanziarie, questioni burocratiche, assistenza programmata...) sono stati ignorati, a causa della natura scolastica del progetto.

Ipotesi e assunzioni - specifica requisiti

- I requisiti esplicitano un invio di notifiche verso **sistemi esterni**. Considerato che la comunicazione tra centraline stradali e automobilistiche è unilaterale, l'applicazione è l'unico sottosistema che oltre ad inviare dati ne riceve.
- Il **cambiamento di stato** di un segmento stradale è definito come l'evoluzione del flusso di traffico su una strada monitorata (ad esempio se su un segmento stradale si passa da una situazione traffico scorrevole a una di coda ferma, si ha un cambiamento di stato).
- Il **raggio di notifica** è la distanza massima dal segmento che subisce un cambiamento di stato, entro la quale si deve trovare un utente per ricevere una notifica sull'applicazione mobile.
- Nelle specifiche viene richiesto che l'utente possa consultare lo **stato del sistema**. Assumiamo quindi che possa visualizzare lo **stato del traffico** sul territorio monitorato e il corretto funzionamento dell'intero sistema (definito come **stato di funzionamento**).

Sintesi dei goal - I*

Diagramma: modulo SDM



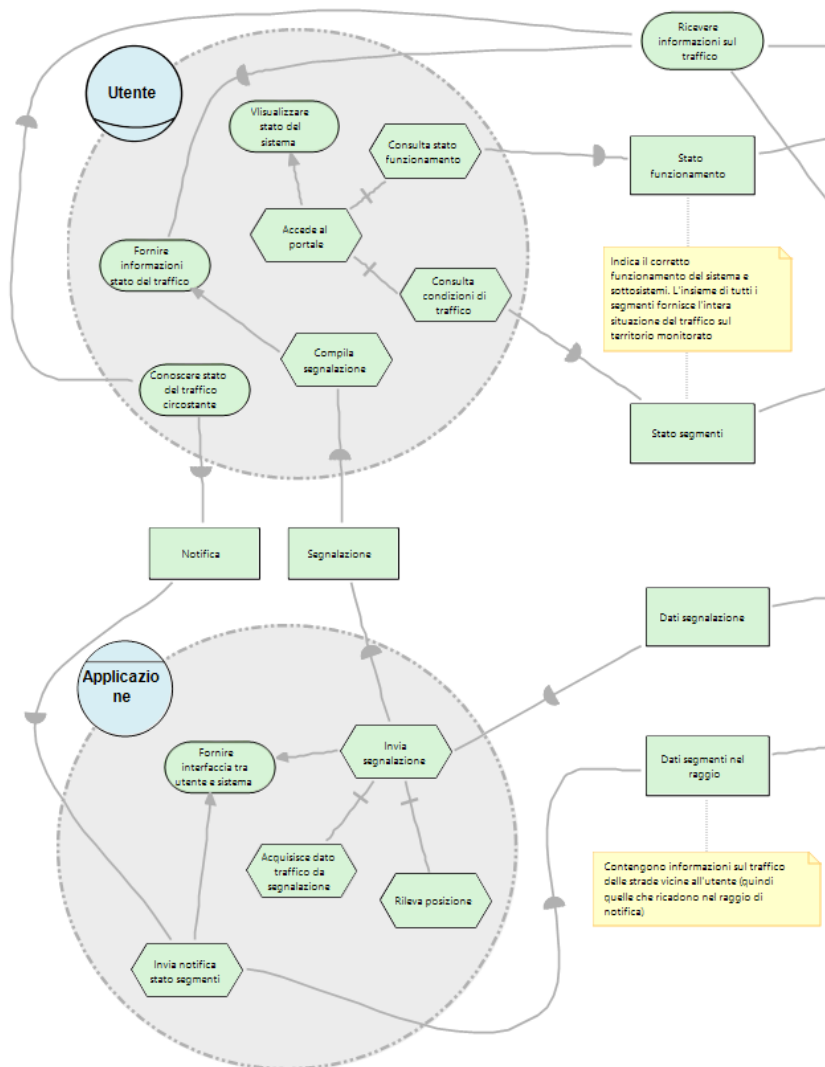
Descrizione

Analizzando la specifica di progetto che ci è stata fornita abbiamo identificato come prima cosa gli attori e successivamente ne abbiamo evidenziato gli obiettivi comuni e le dipendenze. Come si nota dal diagramma SDM, il sistema centrale ha come compito principale quello di raccogliere i dati del traffico, rappresentato dal goal "Ricevere informazioni sul traffico":

- Tramite le centraline stradali e automobilistiche
- Tramite le segnalazioni dell'utente. In particolare, questo obiettivo è bilaterale (appartiene anche all'utente), in quanto le persone sono il target del servizio offerto.

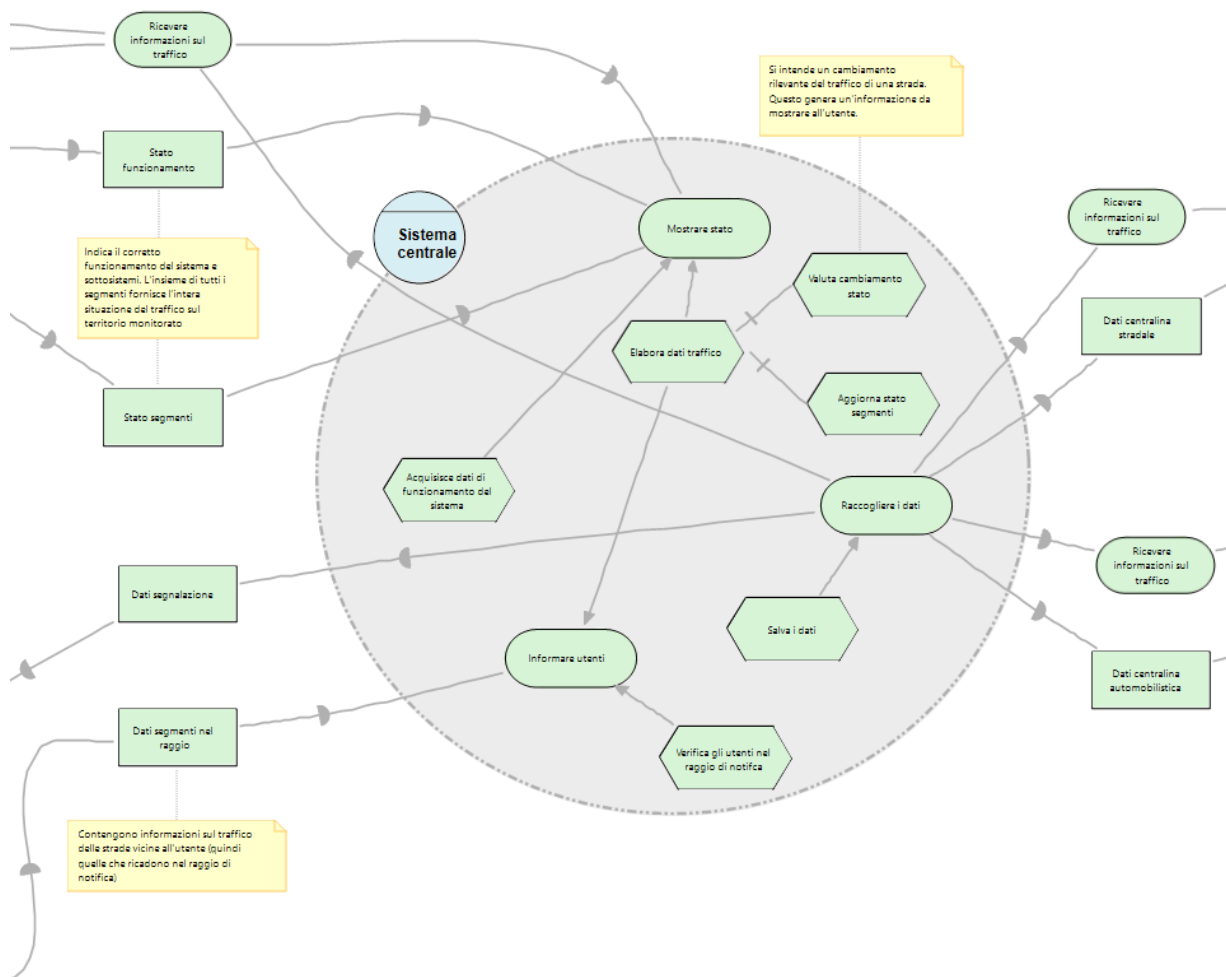
Lo scambio di informazioni avviene tramite un'applicazione mobile, che permette la ricezione di notifiche e la segnalazione dello stato del traffico. Inoltre, l'utente, attraverso un'interfaccia grafica, può verificare lo stato di funzionamento del sistema e dei sottosistemi, e visualizzare tutti i dati sul traffico.

Modulo SRM - Utente e Applicazione



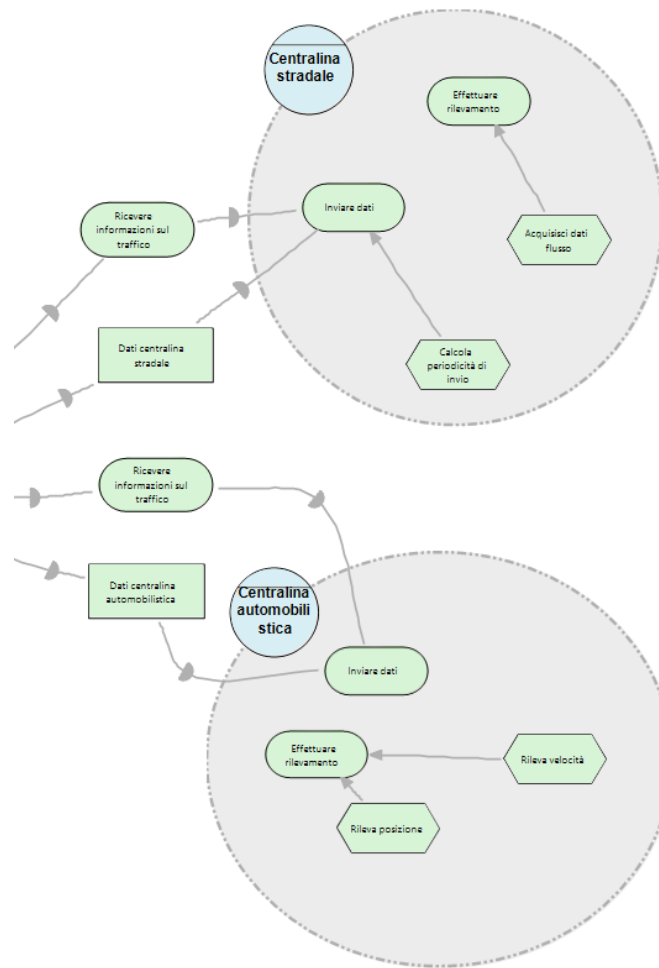
L'utente ha come obiettivi quelli di fornire informazioni sul traffico e di ricevere informazioni riguardanti la viabilità circostante. Tutto questo avviene per mezzo di un'applicazione che ha come scopo quello di fare da tramite tra la persona e il sistema centrale. La comunicazione tra utente e sistema avviene attraverso la compilazione di apposite segnalazioni, mentre il sistema comunica verso l'utente mediante l'invio di notifiche, che vengono recapitate all'applicazione. Un altro goal dell'utente è quello di visualizzare lo stato dell'intero sistema.

Modulo SRM - Sistema Centrale



Il sistema centrale ha come obiettivi quelli di raccogliere e mettere a disposizione dell'utente i dati forniti dalle centraline e dall'applicazione mobile. Inoltre ha il compito di avvisare gli utenti interessati di un cambiamento della viabilità, in seguito all'elaborazione dei dati acquisiti.

Modulo SRM - Centraline automobilistica e centralina stradale



Gli obiettivi delle centraline sono comuni ad entrambi i sistemi, devono effettuare il rilevamento dei dati di loro competenza ed inviarli al sistema centrale. Si differenziano l'una dall'altra per il tipo di dato inviato: la centralina automobilistica rileva posizione e velocità del veicolo mentre la centralina stradale acquisisce il flusso del traffico del segmento stradale che sta monitorando. Un'altra differenza è data dalla frequenza di invio dei dati: con periodicità fissa per quanto riguarda la centralina automobilistica, proporzionale alla quantità di traffico per la quella stradale.

Data Dictionary

NOME	DESCRIZIONE	SINONIMI	ESEMPI	ATTRIBUTI	RELAZIONI
Sistema centrale	sistema informatico che si occupa di ricevere/inviare/mostrare/analizzare i dati del traffico				-Interfaccia amministrativa -Centralina stradale -Applicazione mobile -Centralina automobilistica
Centralina stradale	sistema in posizione fissa che raccoglie/ elabora/ invia i dati del traffico su un segmento stradale	sensore stradale		id, dato, periodicità	-Sistema centrale -Segmento stradale
Centralina automobilistica	sistema installato su autoveicoli che raccoglie/ trasmette dati (velocità e posizione) con periodicità fissa	sensore auto		id, dato, periodicità	-Sistema centrale
Applicazione mobile	sistema installato su dispositivi mobile permette invio/ricezione di informazioni sul traffico	app, applicazione			-Sistema centrale -Utente -Notifica -Segnalazione
Utente	persona che utilizza l'app mobile oppure l'interfaccia utente	persona, user	pippo97	username, mail, password, posizione	-Applicazione mobile -Segnalazione -Notifica -Interfaccia utente -Sistema centrale
Interfaccia utente	portale informativo che permette di consultare lo stato del sistema centrale, dei sottosistemi e del traffico cittadino	interfaccia, schermata, portale			-Sistema centrale -Utente

NOME	DESCRIZIONE	SINONIMI	ESEMPI	ATTRIBUTI	RELAZIONI
Segmento stradale	parte di strada soggetta alla misurazione del flusso di traffico	segmento, tratto	segmento1 → via Valleggio: dalla rotonda alla caserma dei pompieri	identificativo	-Stato
Stato (del segmento)	descrive l'andamento del traffico del segmento stradale	viabilità, stato del traffico	coda, traffico lento	stato	-Segmento stradale
Raggio di notifica	raggio di dimensione prestabilita che rappresenta la distanza entro la quale si devono trovare gli utenti per ricevere un aggiornamento sul traffico	raggio, distanza	10 km 5 km 1 m	centro, raggio	-Notifica -Sistema centrale -Utente
Segnalazione (utente)	informazione inviata dall'utente tramite applicazione al sistema centrale	aggiornamento, avviso, update	45.763662 9.096483 04/02/2018 08.00	latitudine longitudine data ora stato	-Applicazione mobile -Sistema centrale -Utente
Notifica	informazione inviata dal sistema centrale all'applicazione, nel caso in cui l'utente ricada nel raggio di notifica	alert, avviso	Via Napoleona: traffico lento	titolo, descrizione	-Raggio di notifica -Sistema centrale -Applicazione mobile -Utente

Design

Ipotesi e assunzioni - design

- Ogni centralina sarà dotata di una **diagnostica** interna che rileverà possibili guasti; questa parte verrà simulata con l'attributo "funzionante" della classe Centralina. Sarà poi il metodo **diagnostica** a modificare l'attributo "attivo" della centralina in modo tale che venga disabilitato l'invio dei dati e il GestoreCentralina possa, attraverso il suo metodo di diagnostica, gestire le centraline guaste.
- Il **GestoreCentralina** tra i suoi compiti ha quello di verificare eventuali guasti alle centraline. Questa funzionalità viene svolta dalla funzione di diagnostica che, nel caso di esito negativo, modifica lo stato della centralina sul database da attivo a inattivo. La diagnosi avviene in modo automatico, tutti i dati inviati dalle centraline hanno un attributo "AttesaMassima", che è uguale al doppio del tempo della periodicità di invio della centralina ($\text{AttesaMassima} = 2 * \text{periodicità}$). Il GestoreCentralina può quindi tenere traccia dei tempi di invio dei dati. Se una centralina non rispetta il tempo di consegna il gestore la valuta come guasta.
- Per indicare se e quanto un **segmento stradale** è trafficato viene calcolato dalle centraline stradali o dal sistema centrale (basandosi su centraline automobilistiche e segnalazioni) un **indice di flusso**, ossia un numero in percentuale, tramite velocità media dei veicoli e limite di velocità sul segmento. L'indice di flusso viene discretizzato in tre fasce per essere il più intuitivo possibile. Il sistema centrale lavora con l'indice di flusso numerico, l'utente sull'applicazione vedrà una breve descrizione testuale, mentre sulla mappa verranno utilizzati i colori.

INDICE DI FLUSSO	DESCRIZIONE SEGMENTO	APPLICAZIONE UTENTE	COLORE SEGMENTO
0 - 30	libero con un basso livello di traffico	"Strada libera"	 Verde
30 - 70	presenta dei rallentamenti con un medio livello di traffico	"Traffico scorrevole"	 Giallo
70 - 100	presenta code con un elevato livello di traffico	"Traffico elevato"	 Rosso

- Un **evento** è il passaggio di un segmento stradale da una fascia di traffico all'altra. Questa valutazione viene effettuata dal GestoreSegmentiStradali e comporta l'invio di una notifica verso tutti gli utenti che si trovano nei pressi del segmento interessato.

[illegible]

In seguito ad una fase di analisi, siamo arrivati a definire tutti i gestori che di fatto compongono il sistema centrale e che si specializzano in ambiti diversi, comunicando tra di loro attraverso le interfacce messe a disposizione.

Ogni classe gestore è caratterizzata dall'utilizzo di un pattern **singleton**.

#swEng Traffic Monitor

- centralina (stradale o automobilistica)
- applicazione
- interfaccia utente

Centralina

Esistono due tipi di centralina, che di fatto sono figlie della classe astratta *"Centralina"*. La differenza tra le due risiede principalmente nel tipo di dati elaborati e nelle operazioni che verranno eseguite su di essi. Infatti, le centraline produrranno due tipi diversi di dato (che derivano entrambi dalla classe astratta *"DatoCentralina"*).

Il dato prodotto dalla centralina automobilistica è infatti relativo ad una sola macchina, e il sistema centrale si occuperà di elaborare l'indice di traffico partendo da un insieme di informazioni ricevute dalle centraline automobilistiche.

Il dato prodotto dalla centralina stradale invece viene costruito elaborando le informazioni di traffico sul segmento stradale a cui è associata la centralina. Conterrà quindi già un indice di flusso calcolato dalla centralina stessa.

La classe centralina non ha un'interfaccia in quanto la comunicazione con il sistema è unilaterale.

Abbiamo utilizzato il pattern **factory** per la costruzione del dato da parte delle centraline.

Applicazione

L'applicazione (che deve essere associata ad un utente) permette di comunicare con il sistema centrale attraverso una segnalazione o ricevere informazioni sul traffico attraverso una notifica.

La segnalazione si suddivide in tre livelli di traffico, e contiene anche la posizione dell'utente.

NB: tutti i dati che vengono inviati al sistema centrale (da parte delle centraline o da parte dell'applicazione) derivano dalla classe astratta *"Dato"*, che contiene di default un timestamp e delle coordinate.

Interfaccia utente

L'interfaccia utente è quella che permette di visualizzare lo stato di traffico dell'intera area monitorata attraverso una mappa. La mappa è quindi formata da un insieme di segmenti stradali a cui sono associati degli indici di flusso del traffico. Inoltre, è possibile visualizzare lo

stato di funzionamento del sistema, e controllare dunque quali sono le centraline guaste. Infine, è possibile visualizzare uno storico.

[illegible]

L'object diagram rappresenta un'istanza dell'intero sistema, con i valori degli attributi di ogni classe in quel determinato istante.

#swEng Traffic Monitor

A sinistra si trovano le istanze delle centraline automobilistiche con i valori rilevati al momento. Solamente una centralina non è al momento funzionante e questo può essere causato dal malfunzionamento della centralina stessa.

Sia le centraline stradali sia quelle automobilistiche sono collegate alle rispettive istanze di Dato che devono inviare al sistema centrale.

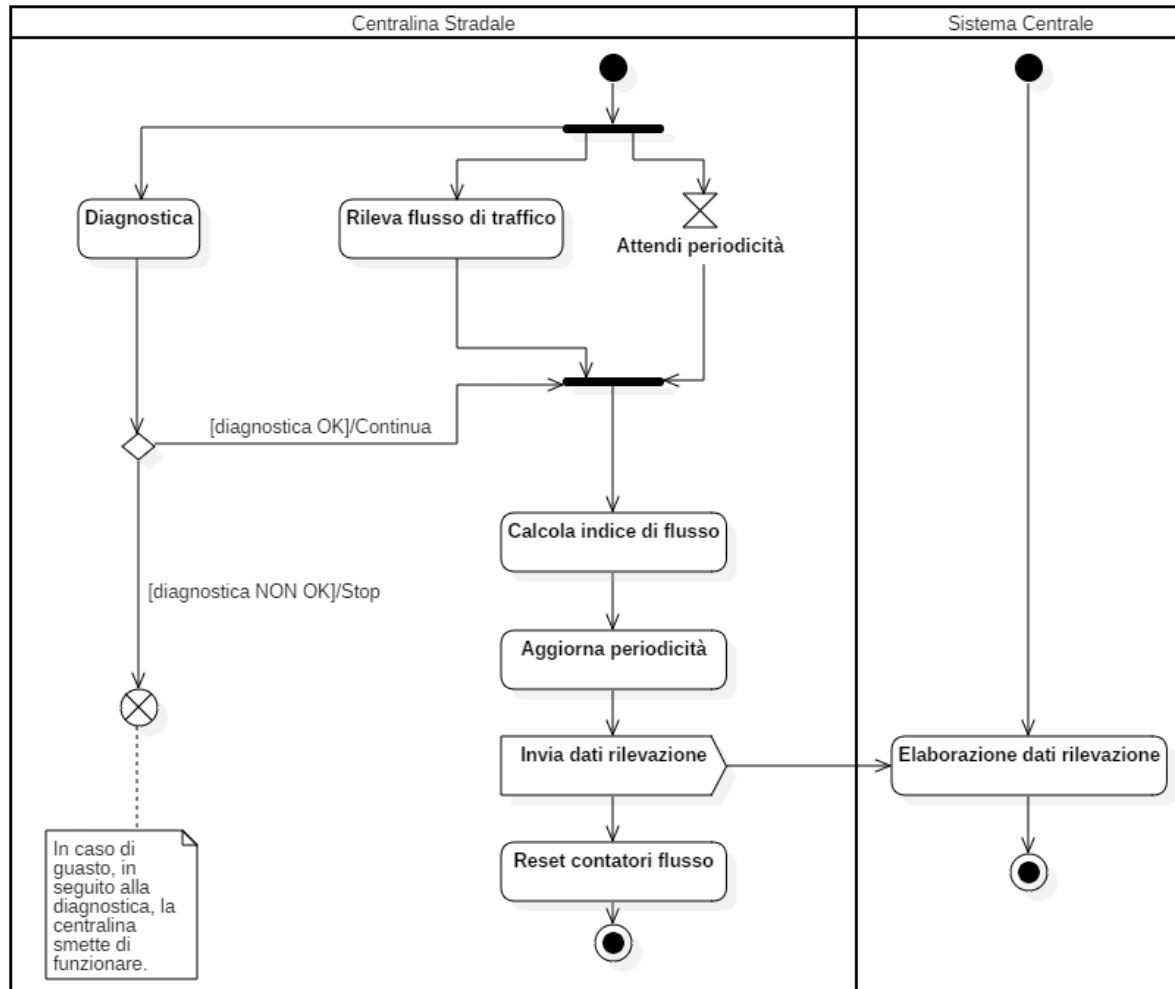
Nella parte inferiore del diagramma troviamo le istanze degli utenti, delle applicazioni, delle segnalazioni e delle notifiche che vengono scambiate con il sistema.

Il sistema centrale è composto invece dai vari gestori che tengono traccia di tutti le istanze con cui si interfacciano e scambiano informazioni tra di loro per garantire il corretto funzionamento del servizio.

Activity Diagram

Activity centralina

Diagramma



Descrizione

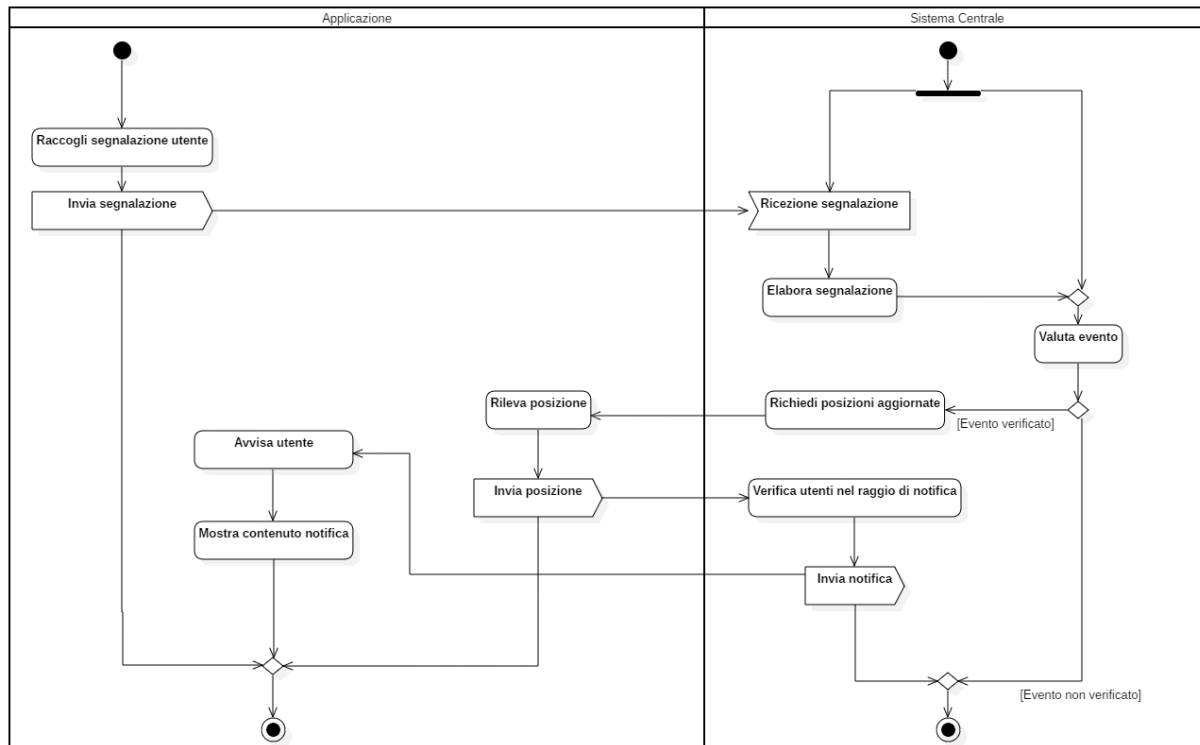
La centralina continua ad operare in un ciclo continuo, a meno di entrare in uno stato di non funzionamento a causa di un guasto. Inizia il suo ciclo con una fase di rilevazione del flusso di traffico per un tempo definito dalla periodicità (che può variare di volta in volta).

Successivamente, si passa a calcolare l'indice con i dati raccolti, si aggiorna la periodicità, si inviano i dati al sistema centrale che potrà elaborarli ed infine si resettano i contatori per ripartire con la prossima rilevazione. Notare come si può assumere che tutte le rilevazioni siano

fatte in modo continuo: la parte di elaborazione ed invio dati ha una durata estremamente limitata.

L'unico caso particolare è la rilevazione di un guasto da parte della diagnostica, che porta la centralina ad uno stato passivo nella quale smetterà di adempiere il suo compito.

Diagramma



Descrizione

Le attività che riguardano l'applicazione e il sistema centrale sono multiple, e possono partire da una o dall'altro.

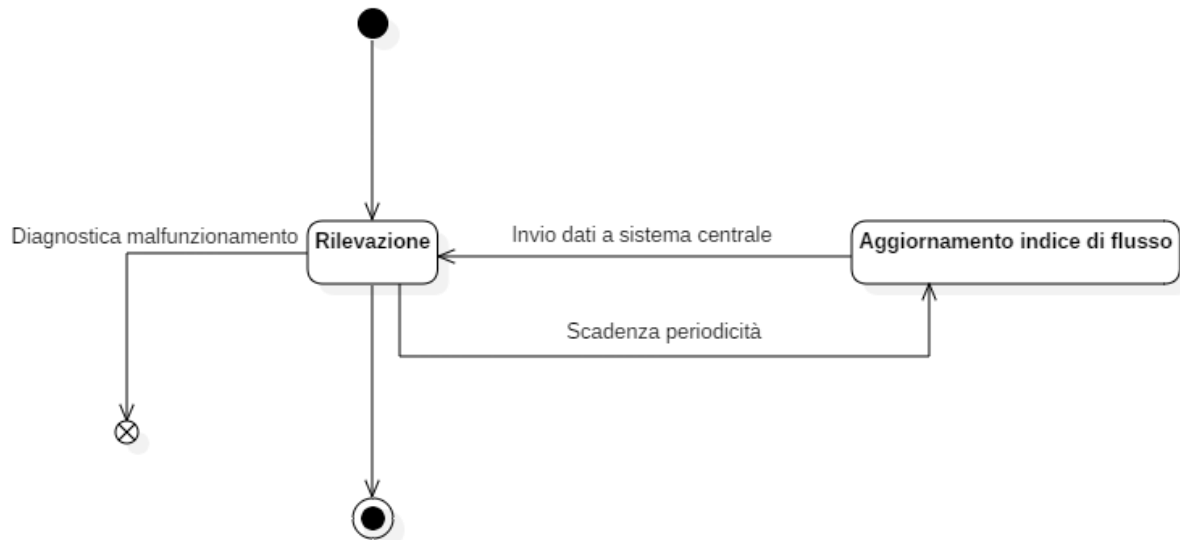
L'applicazione deve poter dare la possibilità di fare una segnalazione, dopo aver raccolto i dati necessari.

Il sistema centrale invece, deve valutare la presenza di un evento (in seguito ad una segnalazione oppure in seguito alle altre informazioni in suo possesso che non dipendono dall'applicazione). Nel caso in cui si verifichi un evento, allora richiede all'applicazione la posizione attuale in modo da verificare quali sono gli utenti interessati dalla notifica. Infine l'app dovrà mostrare il contenuto della notifica.

State Diagram

State Centralina

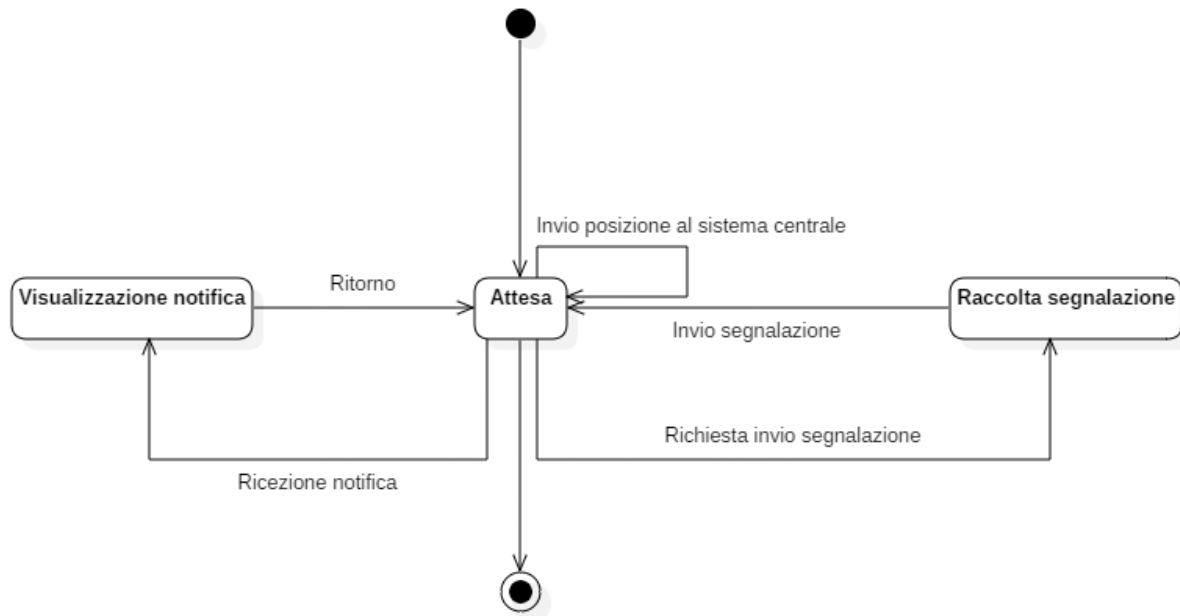
Diagramma



Descrizione

La centralina rimane stabile in uno stato di rilevazione del flusso di traffico e, tra una rilevazione e l'altra, si occupa di un breve stato di elaborazione che, come specificato nell'Activity Diagram, ha una durata estremamente limitata.

L'altro possibile stato è quello passivo nella quale smetterà di adempiere il suo compito, in seguito alla rilevazione di un guasto.

Diagramma**Descrizione**

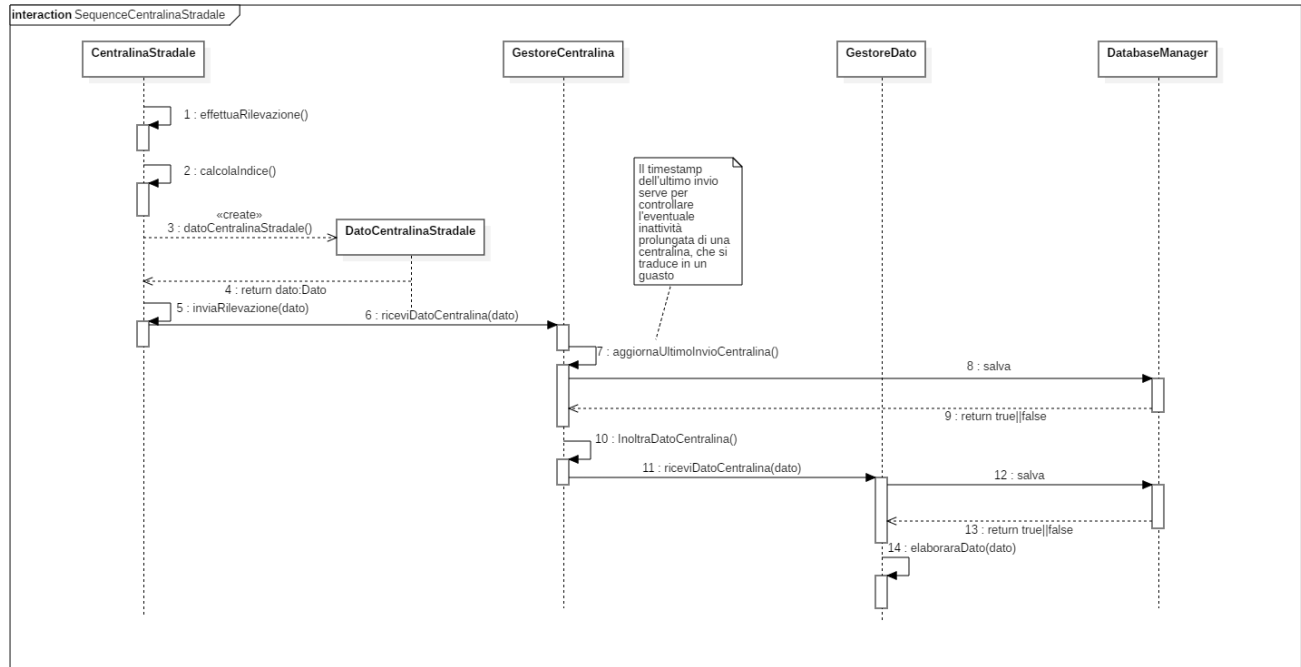
Il funzionamento dell'applicazione si costruisce intorno ad uno stato principale di attesa, dal quale si può passare alla visualizzazione di una notifica (in seguito alla ricezione della stessa), o all'invio di segnalazione dopo aver raccolto i dati necessari.

L'invio della posizione periodico viene sempre effettuato, e riporta l'applicazione nello stato principale di attesa.

Sequence Diagram

Sequence centralina stradale

Diagramma



Descrizione

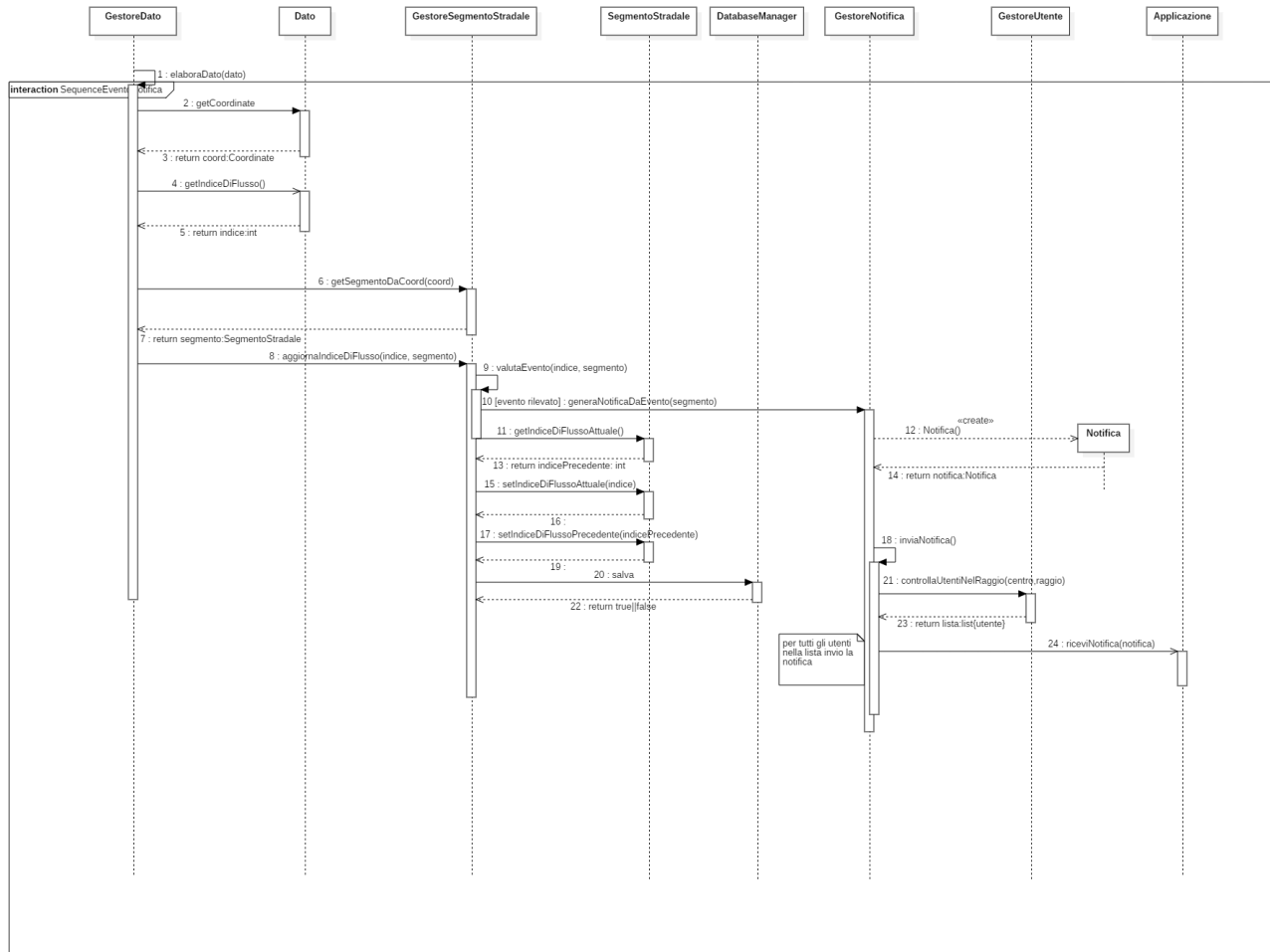
Questo sequence diagram mostra la sequenza di funzioni che vengono invocate durante la raccolta e lo scambio dei dati tra la CentralinaStradale e il sistema centrale.

Dopo aver effettuato la rilevazione e calcolato l'indice di flusso, la centralina impacchetta le informazioni raccolte e le invia al GestoreCentralina. Questo componente funge da intermediario tra le centraline e il GestoreDato. Inoltre, tiene traccia degli orari degli ultimi dati inviati di ogni centralina, per poter rilevare eventuali ritardi di trasmissione e quindi possibili guasti.

Dopo aver annotato l'orario di ricezione, il pacchetto viene inoltrato al GestoreDato che si occuperà di salvarlo nel database e processarlo.

Sequence evento-notifica

Diagramma



Descrizione

Come si vede dal diagramma, il GestoreDato processa il dato ricevuto dalle centraline, estraendo l'indice di flusso e le coordinate di provenienza.

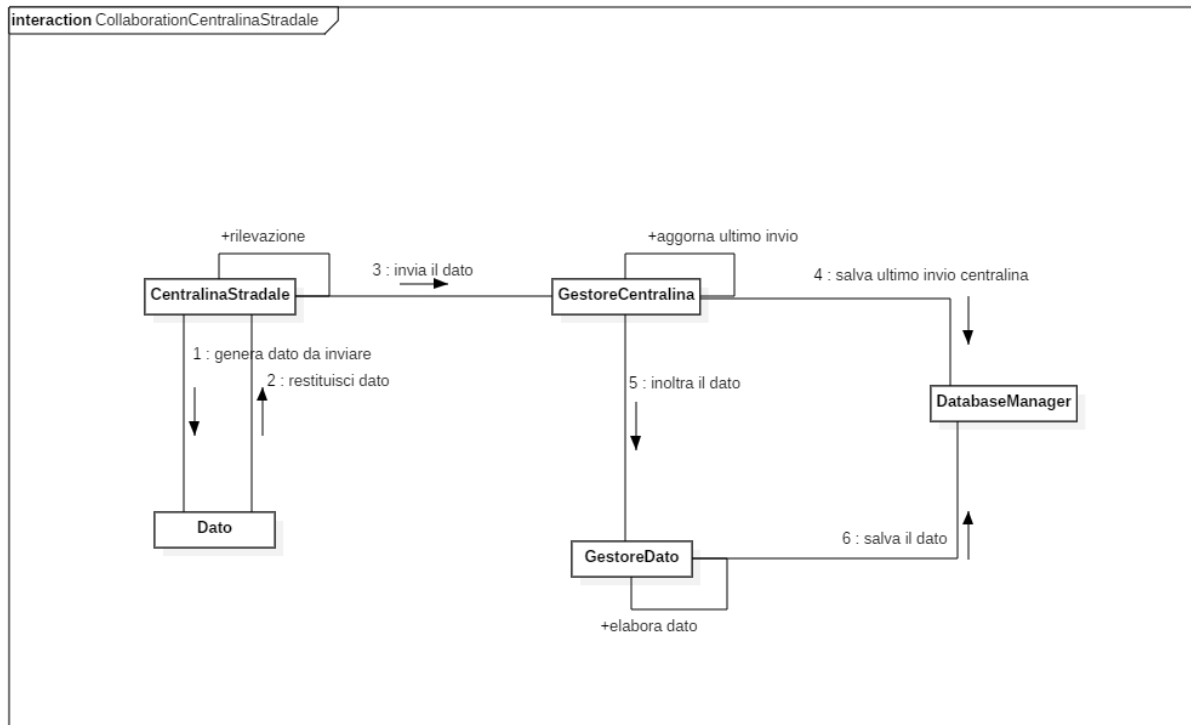
In seguito, questi dati vengono forniti al GestoreSegmentoStradale valuta se c'è stato un significativo cambiamento del traffico (evento), ed in caso affermativo si rivolge al GestoreNotifica per avvisare gli utenti interessati. Successivamente salva temporaneamente il vecchio indice di flusso del segmento e, lo aggiorna con quello nuovo e lo salva sul database.

Il gestore delle notifiche richiede al GestoreUtente una lista di tutte le persone che si trovano nelle vicinanze della strada interessata e invia alla loro applicazione una notifica.

Collaboration Diagram

Collaboration centralina stradale

Diagramma

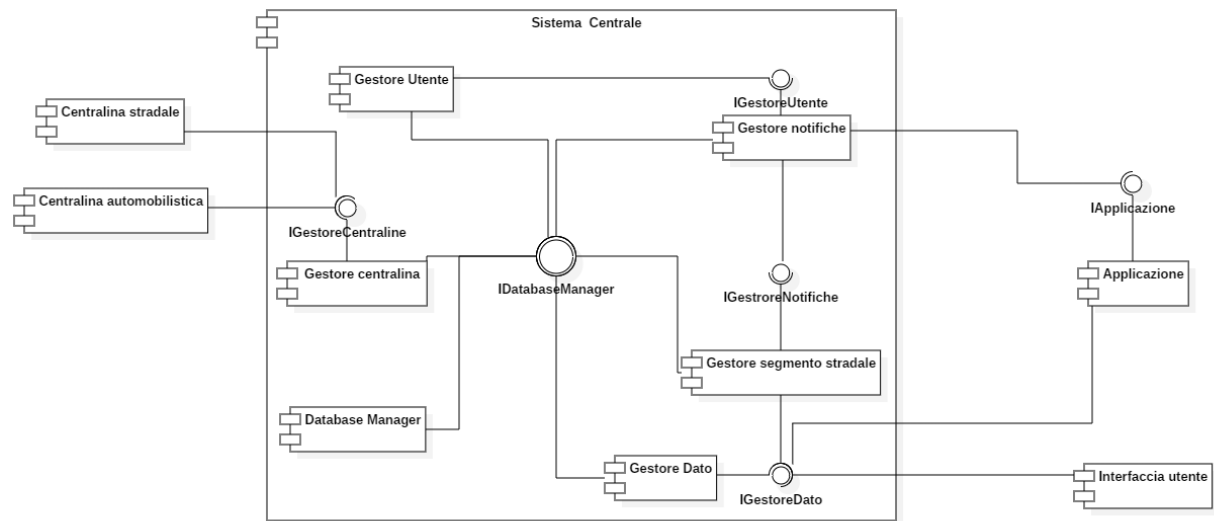


Descrizione

Come si nota dal diagramma, la centralina stradale, dopo aver effettuato una rilevazione, richiede alla classe Dato di costruire il pacchetto da inviare e lo spedisce al GestoreCentralina, il quale aggiorna i suoi riferimenti all'ultimo invio della centralina e inoltra il dato al GestoreDato che si occupa di elaborarlo e salvarlo nel database.

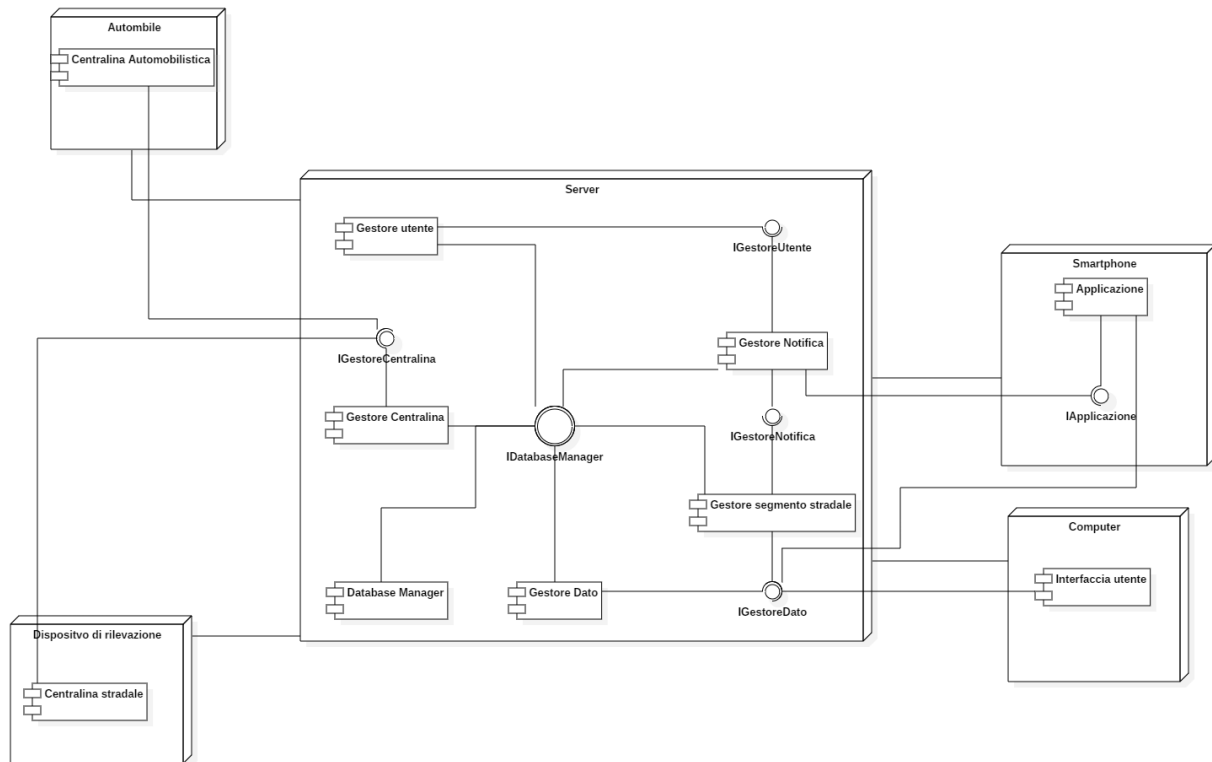
Component Diagram

Diagramma



Deployment Diagram

Diagramma



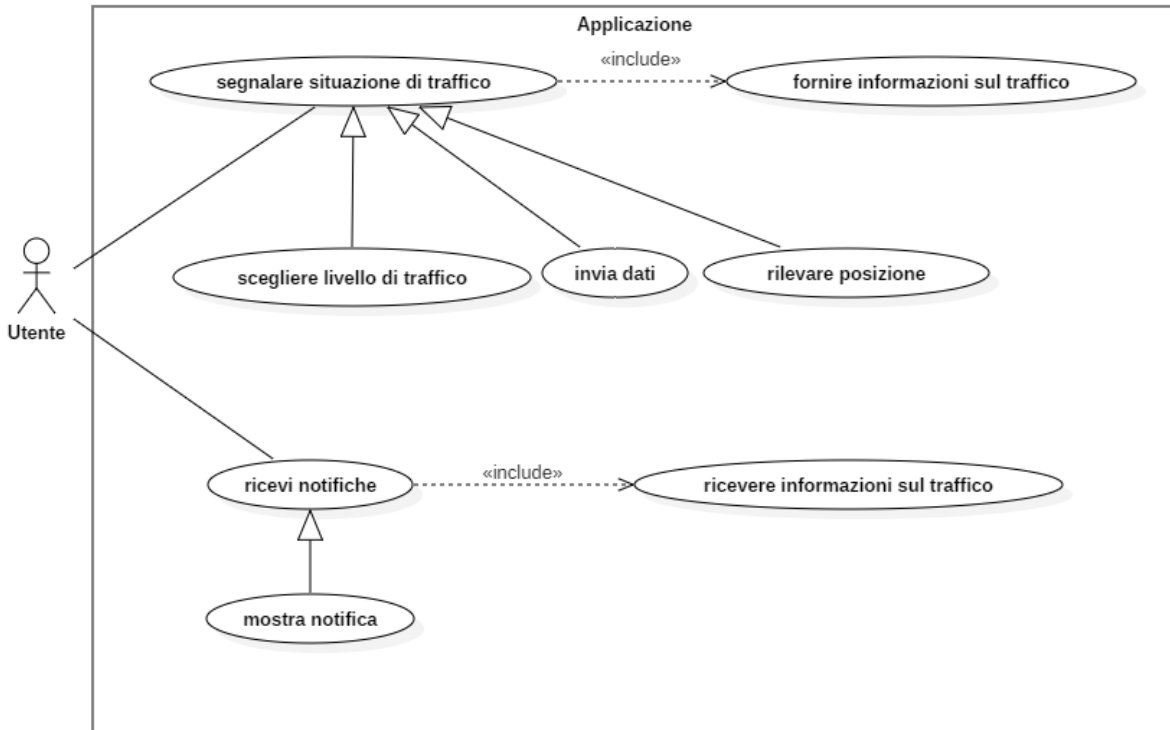
Descrizione

Il component Sistema Centrale si trova su un Server di sistema e comunicherà con gli altri node per far sì che i gestori possano raggiungere centraline, applicazioni e Interfaccia Utente. In particolare, le centraline si troveranno su Automobili e Dispositivi di rilevazione dati, le applicazioni su Smartphone e le interfacce utenti su un Computer.

Use Case Diagram

Use case utente

Diagramma



Descrizione

Il diagramma mostra come l'utente si rapporta all'applicazione.

L'utente può ricevere notifiche dal sistema, riguardanti lo stato di traffico che si trovano vicino a lui.

Può anche inviare una segnalazione scegliendo un livello di traffico, tra "strada libera", "traffico scorrevole", "traffico elevato".

Java

Struttura sistema

Componenti

Per ogni componente è stato creato un progetto apposito.

- Sistema centrale: si occupa di elaborare tutte le informazioni ricevute dai sistemi esterni e della comunicazione con essi
- Centralina stradale: si occupa del monitoraggio del segmento a cui è associata, elaborando un indice di traffico
- Applicazione: si occupa di comunicare all'utente le notifiche contenenti gli aggiornamenti sul traffico e permette di inviare segnalazioni sullo stato del traffico
- Interfaccia utente: permette di avere una visione generale sullo stato del sistema, attraverso una mappa che mostra i segmenti stradali, una lista delle centraline ed il loro stato di funzionamento e uno storico di tutti i dati ricevuti (dati centraline e segnalazioni)

Metodo di comunicazione

RMI

Database

SQLite

IDE

IntelliJ Idea 2018.3

Risorse esterne

Abbiamo utilizzato delle librerie e delle API esterne, le cui dipendenze sono già incluse nel file ".iml" di configurazione di ogni progetto.

- Testing JUnit: junit-4.12.jar, hamcrest-core-1.3.jar - su ogni componente
- Libreria database: lib\sqlite-jdbc-3.23.1.jar - su sistema centrale
- File database: db\traffic_monitor.db - su sistema centrale
- API Google maps: tramite script Javascript - su interfaccia utente

Funzionamento

Per documentare il funzionamento del progetto, prendiamo in esame ogni sua funzionalità e mostriamo come viene svolta all'interno dell'intero sistema, passando tra i componenti, le classi e i metodi.

Di fatto, le funzionalità sono:

- Monitoraggio del traffico con centraline stradali, con cambio della periodicità di invio dei dati
- Monitoraggio del traffico con segnalazioni utente
- Avviso cambiamento situazione traffico per gli utenti interessati tramite notifica
- Monitoraggio dello stato di funzionamento delle centraline e diagnostica
- Visualizzazione mappa con stato di traffico di ogni segmento, stato di funzionamento centraline e storico dati e segnalazioni

Monitoraggio del traffico con centraline stradali

Progetto centralina stradale

Il progetto della centralina stradale utilizza un'interfaccia grafica per vedere in tempo reale il suo funzionamento e che permette volendo di inserire manualmente i dati di un rilevamento. La grafica viene gestita attraverso un pattern MVC dove:

- Model: CentralinaStradale.java
- View: CentralinaStradaleGUI.java
- Controller: ControllerCentralinaStradale.java

Il funzionamento della centralina è scandito da un timer che si occupa, allo scadere del valore di tempo indicato dalla periodicità di lanciare una diagnostica (il cui funzionamento sarà spiegato successivamente nello stato di funzionamento delle centraline), di effettuare la rilevazione e infine di inviare la rilevazione effettuata.

Il metodo che ha il compito di raccogliere i dati simulati ha due tipi di funzionamento: quello automatico che, tramite una funzione, simula in modo pseudocasuale un traffico che aumenta sempre di più con il passare del tempo, o quello manuale, nel quale vengono presi i dati forniti nella GUI. I dati utili al calcolo della situazione del traffico e che vengono quindi raccolti dalle centraline sono numero di macchine e velocità media.

Al momento dell'invio della rilevazione viene calcolato un nuovo indice di traffico, basato sulle informazioni rilevate e quelle relative al segmento monitorato. L'indice di flusso viene calcolato come il rapporto tra la velocità media registrata sul segmento e la velocità massima della strada. Questo valore viene espresso in percentuale e trasformato nel suo reciproco ($100 - \text{valoreCalcolato}$) per indicare il livello di traffico, dove 100 corrisponde al massimo livello di traffico.

Basandosi su questo dato, viene aggiornata la periodicità di invio del prossimo dato (sapendo che se il livello di traffico sta aumentando, il tempo di rilevazione e invio dovrà diminuire). Viene infine inviato il dato tramite RMI al sistema centrale.

A questo punto, vengono resettati i valori utili alla rilevazione e viene fatto ripartire il timer con la nuova periodicità.

Progetto sistema-centrale

Il sistema centrale riceve il dato della centralina stradale con RMI e il *GestoreCentralina* si occupa di salvare le informazioni relative all'ultimo invio e l'attesa massima. Il dato viene inoltrato al *GestoreDato* che si occuperà di salvarlo nel database e poi di elaborarlo.

Tramite le coordinate presenti all'interno del dato che arriva dalla centralina si ricava il segmento stradale a cui deve essere associato il nuovo indice di flusso; il *GestoreSegmentiStradali* si occupa quindi di aggiornare lo stato di traffico del segmento interessato.

A questo punto si valuta l'evento (questa parte verrà approfondita in seguito) e si salva il nuovo indice nel database.

Monitoraggio del traffico con segnalazioni utente

Progetto applicazione

Il progetto dell'applicazione utilizza un'interfaccia grafica che permette di inviare le segnalazioni. La grafica viene gestita attraverso un pattern MVC dove:

- Model: Applicazione.java
- View: ApplicazioneGUI.java
- Controller: ControllerApplicazione.java

Al momento della creazione di un'applicazione viene settato l'RMI in entrata, necessario per la ricezione delle notifiche, comunicando al sistema centrale l'indirizzo IP e il numero di porta su cui sarà in ascolto l'applicazione.

Per inviare una segnalazione, sulla GUI è possibile scegliere uno dei 3 livelli di traffico disponibili (La strada presenta molti rallentamenti - La strada ha qualche rallentamento - La strada è libera) e utilizzare l'apposito bottone. È possibile inserire manualmente la posizione (latitudine e longitudine).

Alla pressione del bottone, vengono prelevati i dati dalla GUI grazie ai quali si crea una nuova segnalazione, a cui è associato un indice di traffico calcolato come valore medio della fascia di appartenenza (15 - 50 - 85). La segnalazione viene poi inviata tramite RMI al sistema centrale, precisamente al *GestoreDato*.

Progetto sistema-centrale

Il *GestoreDato* riceve la segnalazione tramite RMI, si occupa quindi di salvarla nel database e di chiamare il metodo che la elabora.

Per elaborare una segnalazione ci si appoggia su una struttura dinamica composta da una lista di liste. Su questa struttura sono salvate tutte le segnalazioni ricevute, raccogliendole per riga in base al segmento a cui sono associate.

ES: (dove s1, s2... sono le segnalazioni)

Segmento 1 Via Anzani -> s1 s4 s5

Segmento 2 Via Valleggio -> s2

Segmento 3 Via dei Mille -> s3 s6

La prima operazione è quella di associare la segnalazione ad un segmento. La segnalazione viene elaborata solo se è relativa ad un segmento non monitorato da una centralina e questa risulta attiva, altrimenti fornirebbe solo informazioni superflue.

Si controlla quindi scorrendo la struttura se esistono già segnalazioni associate al segmento relativo alla nuova segnalazione. Se sì, allora si aggiunge la segnalazione alla riga, altrimenti si aggiunge una nuova riga alla struttura. Ogni segnalazione ha una validità temporale, dopo la quale viene eliminata. Se su una riga si hanno almeno $n_segnalazioni$ allora si procede a calcolare un indice medio tra tutte le segnalazioni relative al segmento a cui sono associate e si aggiorna il suo indice. In questo modo si evita di fare affidamento su poche segnalazioni che potrebbero essere imprecise.

Il funzionamento del metodo che aggiorna l'indice, chiamato sul *GestoreSegmentoStradale*, è lo stesso del caso precedente con il dato della centralina, con l'accorgimento che, dal momento in cui il segmento non è monitorato da una centralina, l'informazione riguardo il suo indice di flusso ha una validità temporale.

Avviso cambiamento situazione traffico per gli utenti interessati tramite notifica

Progetto sistema-centrale

Tutto parte dopo un aggiornamento di indice di flusso relativo ad un segmento. Qui il *GestoreSegmentoStradale* confronta l'indice di flusso precedente, l'indice di flusso attuale e il nuovo indice di flusso del segmento. Ogni indice viene associato ad una fascia di traffico:

- fascia 1: 0 - 30
- fascia 2: 31 - 70
- fascia 3: 71 - 100

Se dall'indice precedente a quello attuale si è passati ad una fascia superiore, e questo cambiamento viene confermato con il nuovo indice, allora solo a questo punto il *GestoreNotifica* si occupa di inviare la notifica. In questo modo si evita di generare delle notifiche ad ogni minimo cambiamento anche temporaneo dello stato di traffico.

Nel *GestoreNotifica* viene quindi creata una nuova notifica con una descrizione dello stato di traffico basata sull'indice di flusso.

Qui viene preso il punto medio del segmento e si richiede al *GestoreUtente* quali sono gli utenti che cadono nel raggio di notifica. Un utente cade nel raggio di notifica se la differenza tra il centro del segmento su cui è avvenuto l'evento e la sua posizione è minore del raggio di notifica. In questo momento quindi il *GestoreUtente* chiede a tutte le applicazioni tramite RMI di inviare la posizione attuale dell'utente che sta utilizzando l'applicazione.

La posizione viene ricevuta dal *GestoreDato* attraverso RMI, viene salvata nel database e inoltrata al *GestoreUtente*.

Dal *GestoreUtente* si hanno quindi i dati aggiornati per verificare quali utenti cadono nel raggio di notifica, e si può tornare al *GestoreNotifica*, dove si completa l'invio delle notifiche agli utenti interessati tramite RMI.

Progetto applicazione

L'*Applicazione* riceve la notifica tramite RMI e si occupa di mostrarne il contenuto nella GUI.

Monitoraggio dello stato di funzionamento delle centraline e diagnostica

Progetto centralina stradale

Nella centralina stradale sono presenti due attributi di tipo booleano: *attivo* e *funzionante*. L'attributo *funzionante* può essere modificato manualmente attraverso la GUI e simula un malfunzionamento della centralina. In modo automatico, ogni volta che termina il periodo di rilevazione, attraverso una funzione di diagnostica, in caso di guasto viene modificato il valore dell'attributo *attivo* e viene bloccato il funzionamento della centralina, facendola smettere di rilevare e inviare i dati di traffico.

Progetto sistema-centrale

Anche il sistema centrale prevede una funzione di diagnostica delle centraline, che permette di tenere sotto controllo il loro funzionamento. Questo metodo viene invocato periodicamente e, scorrendo l'intera lista delle centraline, controlla se è già passato il periodo di attesa massima che era stato fornito dall'ultimo dato ricevuto. Nel caso in cui il tempo sia scaduto il *GestoreDato* interpreta questo ritardo come un problema della centralina, la segna come non funzionante e riporta questa modifica anche sul database.

Visualizzazione mappa con stato di traffico di ogni segmento, stato di funzionamento centraline e storico dati e segnalazioni

Progetto interfaccia-utente

Il progetto dell'interfaccia utente utilizza un'interfaccia grafica che permette di visualizzare tutte le informazioni richieste. La grafica viene gestita attraverso un pattern MVC dove:

- Model: `InterfacciaUtente.java`
- View: `InterfacciaUtenteGUI.java`
- Controller: `ControllerInterfacciaUtente.java`

All'avvio, tramite RMI l'*InterfacciaUtente* richiede i seguenti dati al sistema centrale:

- Lo storico al *GestoreDato* che contiene tutte le segnalazioni degli utenti e le rilevazioni effettuate dalle centraline
- Lo stato delle centraline stradali al *GestoreCentralina*
- Lo stato dei segmenti stradali al *GestoreSegmentiStradali*

Sul sistema centrale ognuno di questi metodi non fa altro che ritornare una lista di dati:

- Il *GestoreDato* richiede al database tutti i dati raccolti dalle centraline e con le segnalazioni
- Il *GestoreCentralina* ritorna la lista delle centraline in suo possesso
- Il *GestoreSegmentoStradale* ritorna la lista dei segmenti in suo possesso

I dati dello storico vengono divisi in base alla loro provenienza (centraline o segnalazioni) e vengono mostrati in due sezioni diverse all'interno della GUI.

In un'altra sezione viene mostrato lo stato attuale di funzionamento delle centraline.

Infine, viene creato il codice necessario alla visualizzazione di una mappa tramite la classe *Code*. Questa contiene uno script in Javascript, a cui vengono aggiunti i dati appena ricevuti, che utilizza le API di Google Maps e che contiene le informazioni dei segmenti e delle centraline da mostrare. Di fatto viene generata una mappa che evidenzia lo stato di traffico dei segmenti e le posizioni delle centraline sulle strade. Lo script verrà eseguito dal browser integrato di Java, a cui viene riservato una sezione nella GUI.

Casi d'uso

Introduzione

In questa sezione viene spiegato come fare il setup del progetto e viene mostrato il suo funzionamento attraverso i casi d'uso che riguardano le features che ci è stato chiesto di sviluppare.

Setup

Utilizzando IntelliJ come IDE è possibile importare i progetti e riutilizzare il file.iml in modo da mantenere automaticamente tutte le dipendenze e i riferimenti alle librerie utilizzate.

Di default tutti i moduli del sistema sono impostati per lavorare con RMI sulla stessa postazione.

Altrimenti è necessario modificare gli indirizzi IP all'interno della classe RMConfig nel package dati. Qui si trova la stringa `private static String IP = "localhost:"` dove si deve modificare l'indirizzo in base all'applicativo che dobbiamo eseguire.

- Sistema centrale: lasciare `"localhost:"`
- Applicazione, centralina stradale e interfaccia utente: immettere l'indirizzo IP del dispositivo su cui si trova il sistema centrale, per esempio `"192.168.1.5:"`

Avvio

Possiamo ora avviare i software prestando attenzione ad attivare prima il sistema centrale e successivamente i sistemi esterni, mandando in esecuzione il Main di ogni progetto.

- Il main del sistema centrale manda in esecuzione il software senza GUI, ma con dei feedback visibili dall'output della console
- Il main delle centraline stradali manda in esecuzione 7 centraline con relative GUI
- Il main dell'applicazione manda in esecuzione 3 applicazioni con relative GUI
- Il main dell'interfaccia utente manda in esecuzione 1 interfaccia

Funzionamento

I seguenti passi vi guideranno attraverso il programma e permetteranno di dimostrarne il funzionamento.

Centralina - simulazione traffico e guasti

Dall'interfaccia della centralina è possibile visualizzare le informazioni sulla centralina stessa, disattivare la simulazione automatica del traffico per poterne inserire i valori manualmente, e cambiare lo stato di funzionamento di una centralina simulando un guasto.

Procediamo disattivando le centraline cs000001 e cs000007 (togliendo la spunta dal flag "funzionante") e impostando una rilevazione manuale sulla cs000003, inserendo i valori "20" per numero di macchine e "5" per velocità media (di fatto simuliamo una situazione molto trafficata). Tutte le altre lavoreranno in automatico.

The screenshot shows a software window titled 'cs000003'. The main display area is titled 'Informazioni Centralina' and contains the text 'Centralina cs000003'. Below this, it shows 'Centralina ATTIVA' in green, 'Segmento monitorato: Via Viganò', and 'Velocità massima: 50.0'. To the left, under 'Gestione traffico', there is a checkbox for 'Traffico automatico' which is unchecked, and two input fields: 'Numero macchine' with the value '20' and 'Velocità media' with the value '5'. Below that, under 'Stato Centralina', there is a checkbox for 'Centralina funzionante' which is checked. To the right of these controls is a large empty box labeled 'Ultimo dato inviato'.

Terminato il primo periodo di rilevazioni, impostato su 60 s, noteremo i primi cambiamenti. Le due centraline rotte cambieranno il loro stato da ATTIVO a INATTIVO, mentre le altre invieranno al sistema centrale i dati della misurazione.

cs000001

Informazioni Centralina

Centralina cs000001

Centralina INATTIVA

Segmento monitorato: Via Leone Leoni

Velocità massima: 30.0

Gestione traffico

☒ Traffico automatico

Numero macchine

Velocità media

Stato Centralina

☐ Centralina funzionante

Ultimo dato inviato

Notifica

Dopo qualche secondo, nell'applicazione con l'utente "Tyler Durden", che si trova in prossimità di via Viganò, monitorata dalla centralina cs000003, apparirà una notifica che avvisa di un cambiamento di traffico sul segmento stradale in questione.

tyler_durden

Segnalazioni Notifiche

16:24

Aggiornamento traffico in Via Viganò: La strada presenta molti rallentament

Interfaccia utente - visualizzazione traffico, guasti e storico

La variazione della situazione di traffico può essere notata anche premendo il tasto “aggiorna” dell’interfaccia utente: il segmento di via Viganò avrà un colore differente, che evidenzia la situazione di alto traffico.

Si noti che nel frattempo le altre centraline stanno simulando in automatico una condizione di traffico sempre crescente, motivo per cui anche altri segmenti stradali potranno apparire di colori diversi ed altri utente potranno ricevere notifiche.

Per quanto riguarda lo stato delle centraline, visualizzabile nella parte sinistra della finestra, potrebbe volerci qualche minuto prima di vedere segnate come non funzionanti le centraline cs00007 e cs00001. Questo ritardo è dovuto alla politica di diagnostica del sistema centrale, che non potendo comunicare con le centraline, esegue periodicamente (ogni 150 s) una verifica del tempo di inattività di una centralina, e in caso di ritardi prolungati la segnala come non funzionante.

Centraline	
NOME CENTRALINA	ATTIVA
cs000001	false
cs000002	true
cs000003	true
cs000004	true
cs000005	true
cs000006	true
cs000007	false

Nella parte inferiore dell’interfaccia utente è possibile visualizzare lo storico di tutti i dati delle centraline e di tutte le segnalazioni.

The screenshot displays the #swEng Traffic Monitor interface. It features a map of a city area with several red pins indicating traffic monitoring points labeled with IDs like cs000001, cs000002, cs000003, cs000004, cs000005, cs000006, and cs000007. The interface includes a sidebar with a table of centraline status, a main map area, and a bottom section for historical data and user reports.

Centraline Status Table:

NOME CENTRALINA	ATTIVA
cs000001	false
cs000002	true
cs000003	true
cs000004	true
cs000005	true
cs000006	true
cs000007	false

Dati centraline:

- 2018-12-20 | Centralina: cs000005 | Indice: 75 | Vel. media: 7.0 | N. macchine: 24
- 2018-12-20 | Centralina: cs000007 | Indice: 76 | Vel. media: 6.0 | N. macchine: 25
- 2018-12-20 | Centralina: cs000003 | Indice: 81 | Vel. media: 9.0 | N. macchine: 40
- 2018-12-20 | Centralina: cs000004 | Indice: 76 | Vel. media: 11.0 | N. macchine: 41
- 2018-12-20 | Centralina: cs000001 | Indice: 78 | Vel. media: 6.0 | N. macchine: 25
- 2018-12-20 | Centralina: cs000006 | Indice: 83 | Vel. media: 4.0 | N. macchine: 24
- 2018-12-20 | Centralina: cs000002 | Indice: 77 | Vel. media: 6.0 | N. macchine: 23
- 2018-12-20 | Centralina: cs000004 | Indice: 77 | Vel. media: 6.0 | N. macchine: 23

Segnalazioni utenti:

- 2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
- 2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
- 2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
- 2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
- 2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoElevato | Coordinate: 45.798268, 9.092142
- 2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoElevato | Coordinate: 45.798268, 9.092142
- 2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoElevato | Coordinate: 45.798268, 9.092142

Applicazione - segnalazione

Passiamo ora all'utilizzo dell'applicazione. Aprendo quella associata all'utente "Doctor Who", che si trova su un segmento di via Anzani, cominciamo ad inviare in modo consecutivo, tre segnalazioni "La strada ha qualche rallentamento" e una segnalazione "La strada presenta molti rallentamenti". Queste segnalazioni portano a registrare un cambiamento dell'indice di traffico sul segmento, e la conseguente generazione di una notifica da parte del sistema centrale e inviata a tutti gli utenti nelle vicinanze. L'avviso di cambiamento deve quindi essere ricevuto sull'applicazione con "Arya Stark" come utente, che si trova su un segmento vicino, e su quella del "Doctor Who".



The screenshot shows a web application window titled "doctor_who". It has two tabs: "Segnalazioni" (active) and "Notifiche". The main content area displays "Stato di traffico attuale :" followed by a dropdown menu showing "La strada presenta molti rallentamenti" with a downward arrow. Below this is a button labeled "Invia segnalazione". At the bottom, there is a section titled "Posizione" containing two input fields: "Latitudine Attuale" with the value "45.801233" and "Longitudine Attuale" with the value "9.091384".

Le variazioni possono essere visualizzate aggiornando la mappa dell'interfaccia utente. Essendo segnalazioni effettuate da persone, queste hanno una scadenza prefissata, aspettando quindi il tempo di decadenza delle segnalazioni (impostato su 5 minuti), il sistema centrale azzererà automaticamente il valore dell'indice di traffico sul segmento.

The screenshot shows the 'InterfacciaUtenteGUI' application window. It features a map of a city area with several traffic segments highlighted in red and green. The segments are labeled with IDs like cs000001, cs000002, cs000003, cs000004, cs000005, cs000006, and cs000007. The map includes various landmarks and street names.

On the left side, there is a table titled 'Centraline' with columns 'NOME CENTRALINA' and 'ATTIVA'. The table contains the following data:

NOME CENTRALINA	ATTIVA
cs000001	false
cs000002	true
cs000003	true
cs000004	true
cs000005	true
cs000006	true
cs000007	false

Below the table, there is a section titled 'Dati centraline' which displays a list of data points for each centraline, including the date, centraline ID, index, average speed, and number of vehicles. For example:

```
2018-12-20 | Centralina: cs000005 | Indice: 75 | Vel. media: 7.0 | N. macchine: 24
2018-12-20 | Centralina: cs000007 | Indice: 76 | Vel. media: 6.0 | N. macchine: 25
2018-12-20 | Centralina: cs000003 | Indice: 81 | Vel. media: 9.0 | N. macchine: 40
2018-12-20 | Centralina: cs000004 | Indice: 76 | Vel. media: 11.0 | N. macchine: 41
2018-12-20 | Centralina: cs000001 | Indice: 78 | Vel. media: 6.0 | N. macchine: 25
2018-12-20 | Centralina: cs000006 | Indice: 83 | Vel. media: 4.0 | N. macchine: 24
2018-12-20 | Centralina: cs000002 | Indice: 77 | Vel. media: 6.0 | N. macchine: 23
```

At the bottom, there is a section titled 'Segnalazioni utenti' which displays a list of user reports, including the date, user name, signal type, and coordinates. For example:

```
2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
2018-12-20 | Utente: arya_stark | Tipo segnalazione: TrafficoElevato | Coordinate: 45.801906, 9.092918
2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoElevato | Coordinate: 45.798268, 9.092142
2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoScorrevole | Coordinate: 45.798268, 9.092142
2018-12-20 | Utente: tyler_durden | Tipo segnalazione: TrafficoScorrevole | Coordinate: 45.798268, 9.092142
```

Per generare un evento attraverso un utente abbiamo dovuto inviare tre segnalazioni consecutive dal momento che il sistema centrale definisce un evento generato da segnalazioni solamente se ne ha almeno tre associate allo stesso segmento, calcolandone la media. In questo modo si evita di considerare segnalazioni da parte di singoli utenti, che potrebbero non essere completamente affidabili. Inoltre il sistema non tiene in considerazione le segnalazioni associate ad un segmento monitorato da una centralina funzionante (fornirebbe solo informazioni superflue). Per questo motivo inviando qualsiasi numero di segnalazioni con l'applicazione di "Tyler Durden" non si noterà mai una variazione di traffico, dal momento in cui si trova sul segmento monitorato dalla cs000003.

Sistema centrale - feedback

Per avere una visione completa anche su ciò che accade sul sistema centrale, è possibile leggere, attraverso il terminale, tutti i metodi che vengono chiamati dai gestori durante tutta la fase di elaborazione dati.

Variazioni

Cambiamenti rispetto alle consegne

Di seguito tutti i cambiamenti fatti nella documentazione rispetto alle precedenti consegne:

- Correzione eventuali errori di battitura
- Riscrittura di alcuni concetti in modo più semplice e chiaro
- Cancellazione del package Sistema Centrale dal Class Diagram in seguito a consiglio del tutor
- Revisione del procedimento di richiesta delle posizioni agli utenti (prevalentemente nel Sequence Diagram)
- Revisione del procedimento di diagnostica dopo la consegna Java
- Revisione della spiegazione del funzionamento per renderla più discorsiva e meno legata al codice, ma alla logica del programma (informazioni e spiegazioni riferite ai singoli metodi si trovano come commento direttamente nei file sorgente)
- Aggiunta dei casi d'uso