



UNIVERSITÀ
di **VERONA**

Ingegneria del Software
A.A. 2019/2020

***Documentazione progetto
Indagini Epidemiologiche***

Cavaliere Matteo	VR437065
Di Bari Cristiano	VR429675
Lonardi Enrico	VR429679

Sommario:

Introduzione	3
Requisiti ed interazioni utente-sistema	4
Casi d'uso	4
Casi d'uso relativi al personale a Contratto	5
Casi d'uso relativi al personale Autorizzato dell'Ente	6
Casi d'uso relativi al personale Analista	7
Diagrammi di attività	9
Interfaccia del personale a contratto	9
Interfaccia del personale autorizzato dell'ente	10
Interfaccia del ricercatore analista	11
Sviluppo del prototipo e scelte progettuali	12
Processo di sviluppo	12
Progettazione e pattern architetturali usati	12
Database	13
Diagramma delle classi	14
Organizzazione della GUI	15
Diagrammi di sequenza	16
Design pattern utilizzati	18
Attività di Test e Validazione	19

Introduzione

Obiettivo dell'elaborato è la progettazione di un prototipo software per la gestione delle informazioni socio-sanitarie di monitoraggio della popolazione italiana per la prevenzione di contagi e pandemie.

Il sistema dovrà essere usato da diverse categorie di personale dell'ente incaricato del monitoraggio: il personale autorizzato, il personale a contratto ed i ricercatori analisti, ognuno di questi ruoli avrà un interfaccia dedicata che permetterà di svolgere determinate azioni.

Il software memorizza i dati relativi alle località geografiche monitorate dall'ente incaricato, i dati di ogni comune relativi al numero di potenziali casi di contagio ed il numero annuale di decessi per ogni provincia con la relativa causa. Deve essere inoltre possibile analizzare i dati raccolti in modo da poter osservare le varie informazioni storicizzate aggregate a livello di comune, provincia, regione e nazione.

Per realizzare il prototipo si è scelto di sviluppare un'applicazione desktop in Java fruibile grazie all'implementazione di un interfaccia grafica sviluppata in JavaFX tramite il tool Scene Builder.

In questa relazione ci proponiamo di raccogliere la documentazione sviluppata e di fornire spiegazioni e dettagli sulle scelte progettuali ed implementative.

Requisiti ed interazioni utente-sistema

Casi d'uso

Il sistema supporta l'utilizzo da parte di 4 categorie diverse di utenti identificati da un ruolo: il personale autorizzato dell'ente, il personale a contratto, il personale analista e l'amministratore di sistema. Ogni utente ha a disposizione delle credenziali che gli vengono assegnate dall'amministratore con cui può effettuare l'autenticazione. Ad accesso effettuato, verrà caricata un'interfaccia grafica differente in base al ruolo dell'utente autenticato.

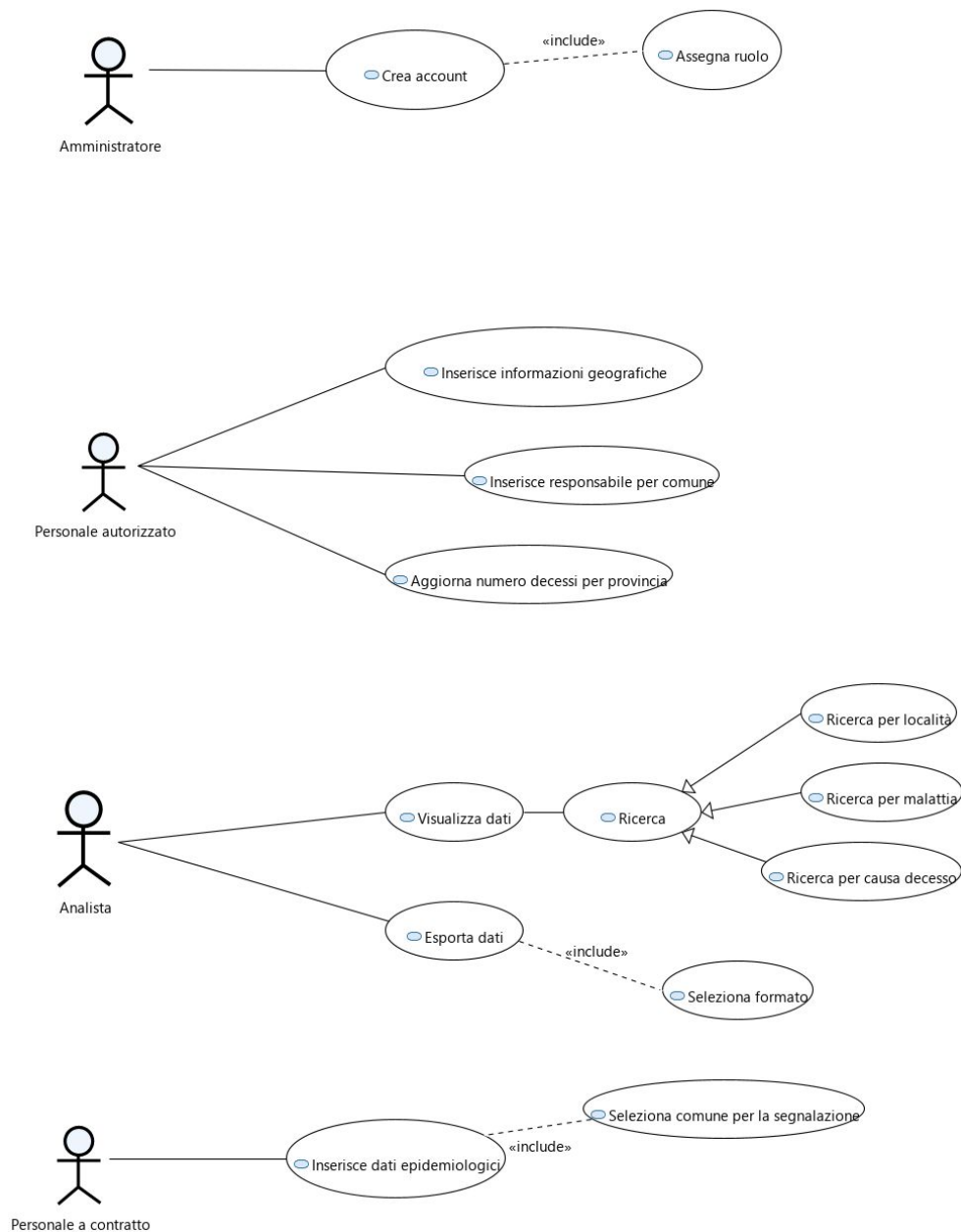


Figura 1: casi d'uso

Casi d'uso relativi al personale a Contratto

Il personale a contratto ha il compito di registrare settimanalmente i dati relativi ai casi di contagio in un comune italiano specificando il tipo di malattia contagiosa e distinguendo le persone in cura presso il medico di base da quelle ricoverate in terapia intensiva. Ad ognuno di essi vengono assegnati uno o più comuni di competenza che determinano la possibilità di inviare le segnalazioni sui contagi avvenuti in quel comune.

Inviare una nuova segnalazione di contagio

Attori: Personale a contratto

Precondizioni: L'utente deve essersi autenticato con un account associato al ruolo di personale a contratto

Passi:

1. L'utente a contratto accede al sistema
2. L'utente accede all'interfaccia per le segnalazioni di contagio
3. L'utente seleziona il comune scegliendolo tra quelli di sua competenza
4. L'utente inserisce i dati epidemiologici relativi alle diverse cause di contagio per il comune prescelto distinguendo per numero di ricoveri in terapia intensiva e numero di persone in cura presso il medico.
5. L'utente conferma la segnalazione di contagio

Postcondizioni: la segnalazione è inserita

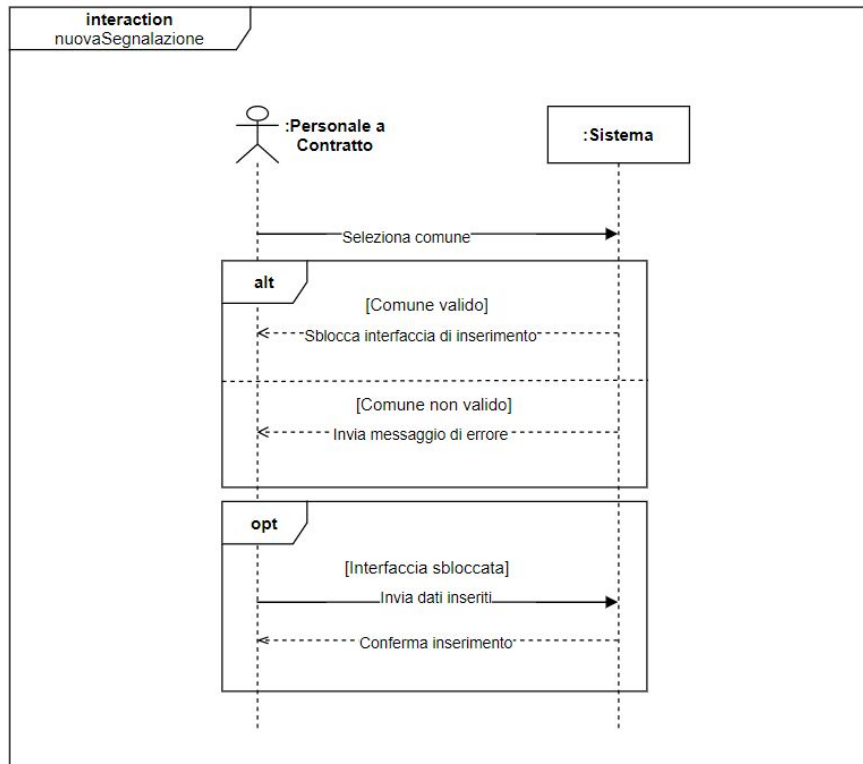


Figura 2: Sequence Diagram inserimento segnalazione di contagio

Casi d'uso relativi al personale Autorizzato dell'Ente

Il personale dell'ente incaricato del monitoraggio registra nel sistema le località geografiche di interesse, suddivise in regioni, province e comuni del territorio italiano con le relative informazioni.

Annualmente deve registrare per ogni provincia le informazioni relative ai decessi causati da incidenti stradali, malattie tumorali, malattie cardiovascolari, e malattie contagiose in modo da poter analizzare i dati confrontandoli con quelli considerati nel monitoraggio dei contagi.

Il personale autorizzato ha inoltre il compito di inserire, per ogni persona a contratto, i comuni di cui è responsabile.

Inserire dati geografici di località italiane

Attori: Personale autorizzato dell'ente

Precondizioni: L'utente deve essersi autenticato con un account associato al ruolo di personale autorizzato

Passi:

1. L'utente accede al sistema
2. L'utente accede all'interfaccia per l'inserimento dei dati geografici
3.
 - a. Crea una nuova regione inserendo i relativi dati
 - b. Seleziona una regione tra quelle già presenti nel database
4.
 - a. Crea una nuova provincia con i relativi dati
 - b. Seleziona una provincia e per quella crea un nuovo comune specificando i dati richiesti
5. Modifica un comune assegnandogli un responsabile scelto tra gli utenti a contratto

Postcondizioni: i dati geografici sono inseriti nel database

Inviare una nuova segnalazione di decesso

Attori: Personale autorizzato dell'ente

Precondizioni: L'utente deve essersi autenticato con un account associato al ruolo di personale autorizzato

Passi:

1. L'utente accede al sistema
2. L'utente accede all'interfaccia per le segnalazioni di decesso
3. L'utente seleziona una regione e una provincia fra quelle presenti nel database
4. L'utente inserisce i dati relativi ai decessi avvenuti nella provincia selezionata distinguendo le diverse cause.
5. L'utente conferma la segnalazione di decesso

Postcondizioni: la segnalazione è inserita

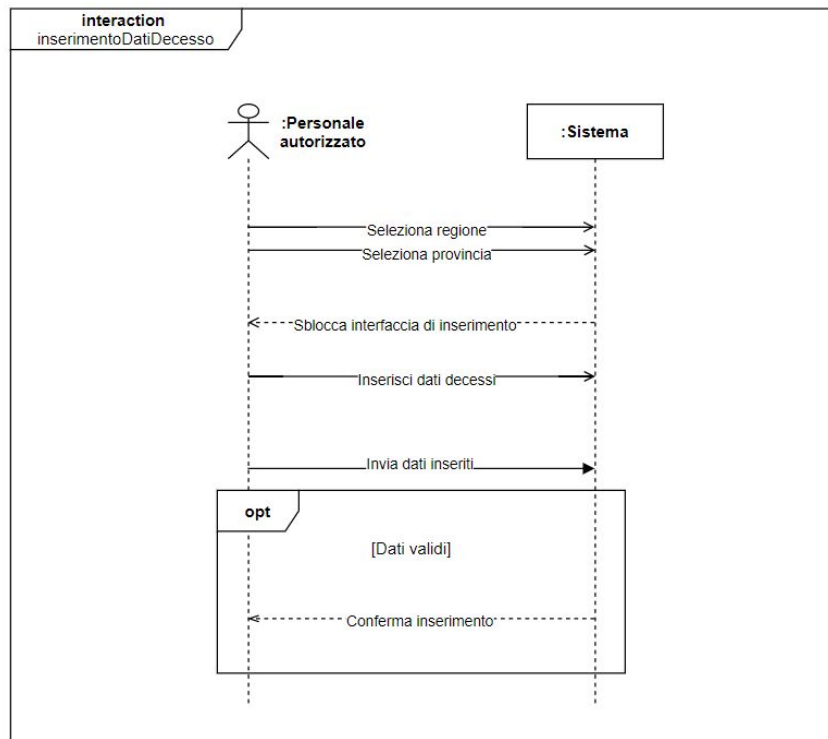


Figura 3: Sequence Diagram inserimento dati relativi ai decessi

Casi d'uso relativi al personale Analista

Il personale analista è il personale incaricato delle analisi dei dati epidemiologici, esso ha a disposizione la possibilità di analizzare anno per anno i dati acquisiti e ottenere rappresentazioni tabulari e grafiche.

Analisi dei dati relativi ai decessi ed ai contagi

Attori: Personale analista

Precondizioni: L'utente deve essersi autenticato con un account associato al ruolo di personale analista

Passi:

1. L'utente accede al sistema
2. L'utente accede all'interfaccia per l'analisi di dati epidemiologici
3. L'utente seleziona una o più località geografiche da esaminare
4. L'utente seleziona la malattia
5. L'utente seleziona la causa di decesso

Postcondizioni: i dati vengono visualizzati in forma tabellare o grafica

Esportazione dei dati relativi ai decessi ed ai contagi

Attori: Personale analista

Precondizioni: L'utente deve essersi autenticato con un account associato al ruolo di personale analista

Passi:

1. L'utente accede al sistema
2. L'utente accede all'interfaccia per l'analisi di dati epidemiologici
3. L'utente inserisce l'anno di cui visualizzare i dati
4. L'utente seleziona i dati da esportare
5. L'utente seleziona il formato di esportazione

Postcondizioni: viene creato il file con i dati esportati

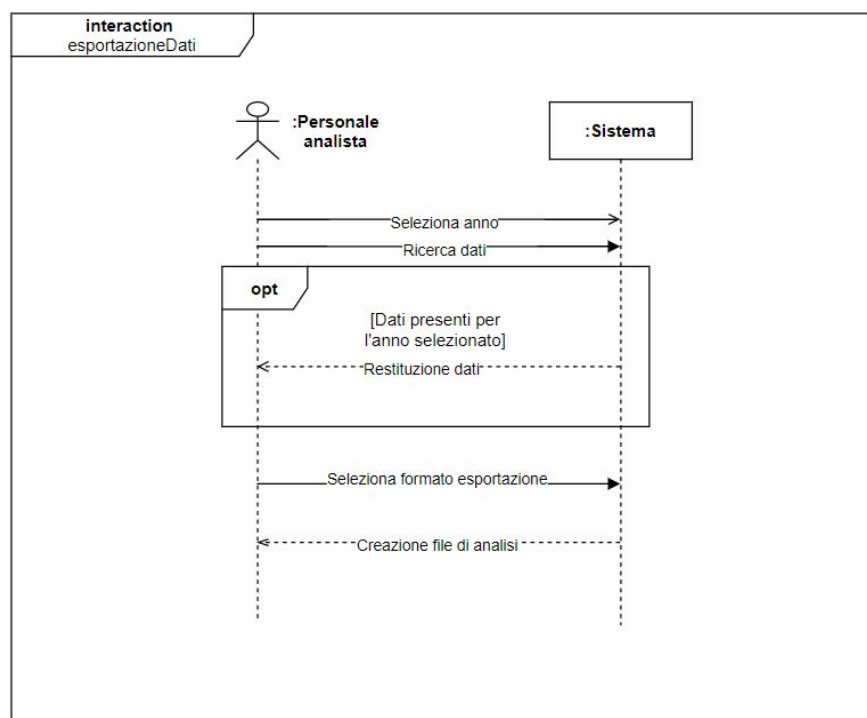


Figura 4: Sequence Diagram esportazione dati epidemiologici

Diagrammi di attività

I diagrammi di attività sono un utile strumento che guida alla comprensione dei possibili comportamenti che si possono riscontrare nell'utilizzo di un sistema, in particolare servono ad arricchire i casi d'uso specificando il flusso delle azioni. Vengono riportati di seguito alcuni esempi che aiutando a capire meglio il funzionamento del sistema in questione.

Interfaccia del personale a contratto

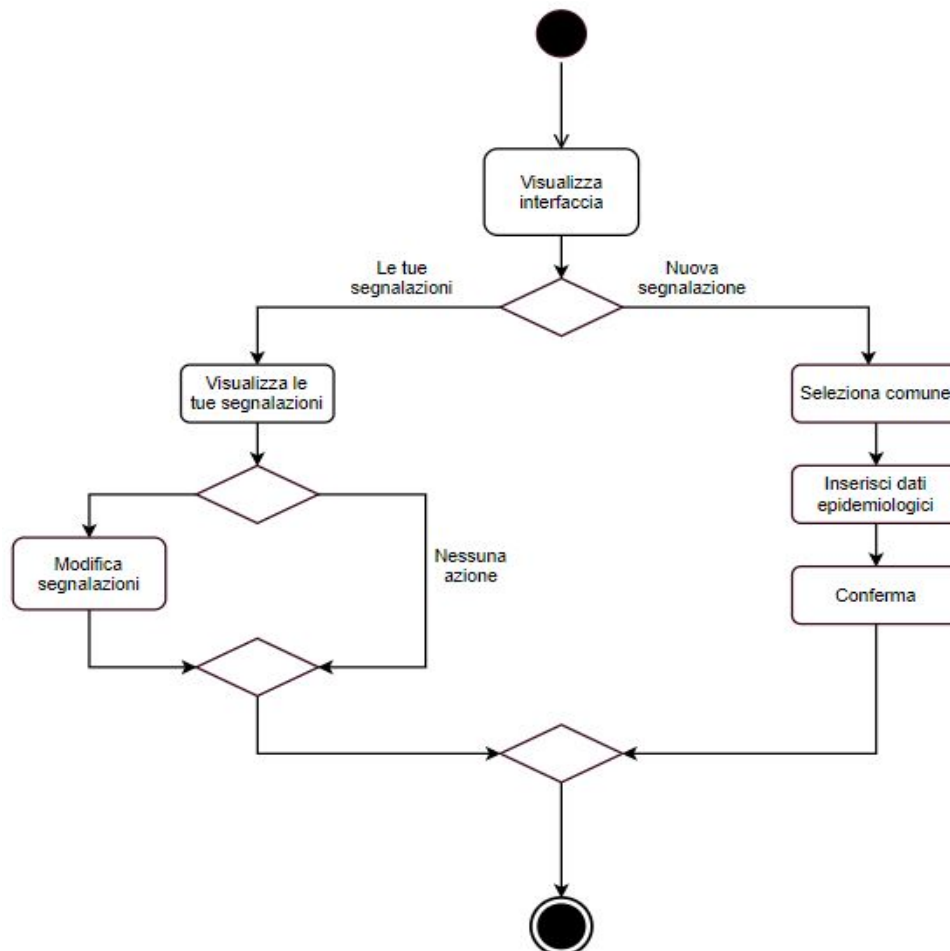


Figura 5: Activity Diagram personale a contratto

Interfaccia del personale autorizzato dell'ente

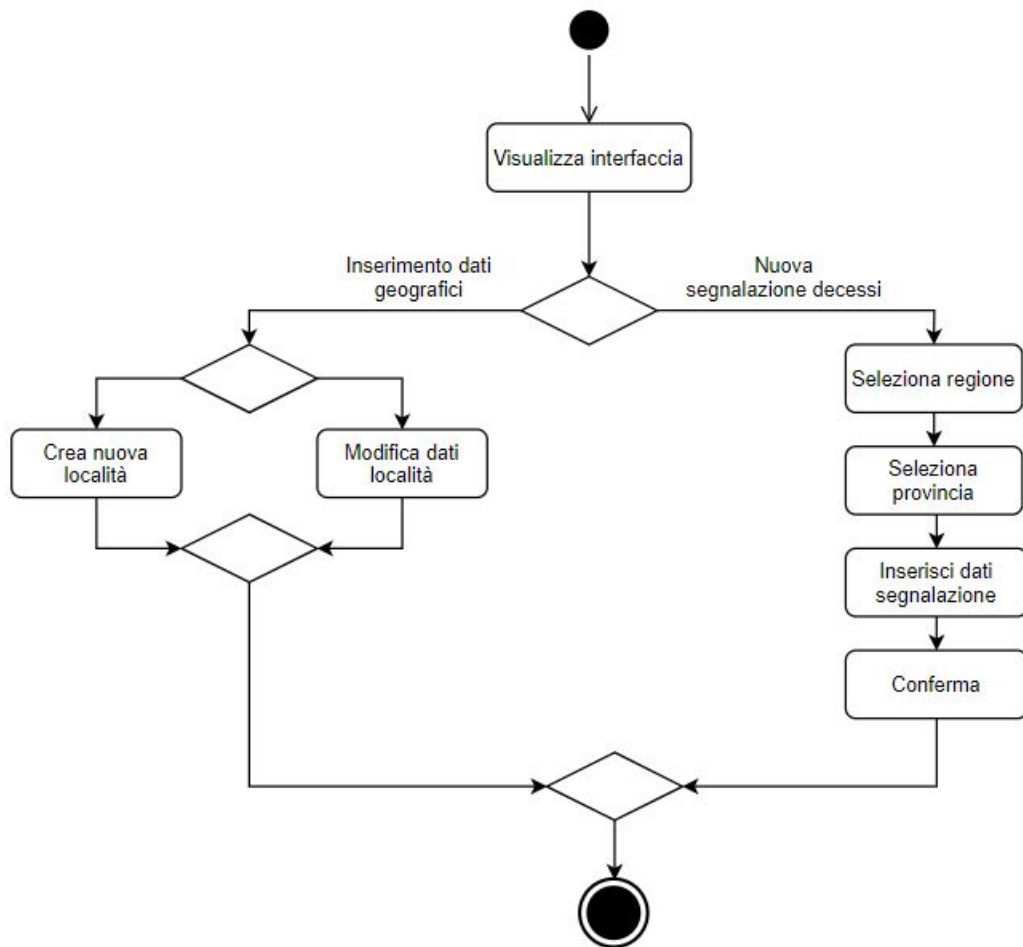


Figura 6: Activity Diagram personale a autorizzato

Interfaccia del ricercatore analista

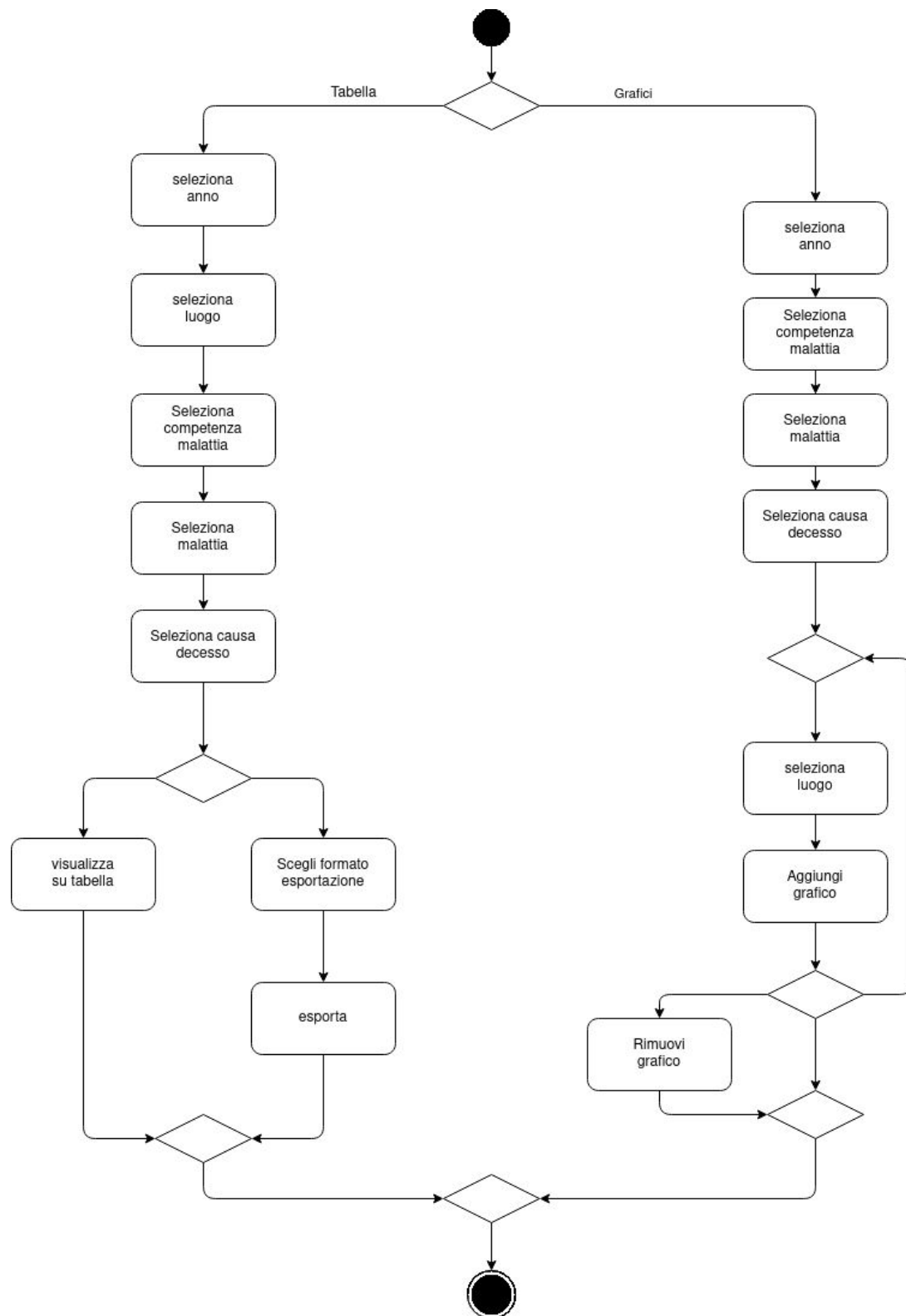


Figura 7: Activity Diagram personale analista

Sviluppo del prototipo e scelte progettuali

Processo di sviluppo

Per lo sviluppo del sistema è stato scelto di adottare un modello incrementale basato su un approccio *agile* che ha permesso di migliorare la rapidità di progettazione e implementazione del prototipo.

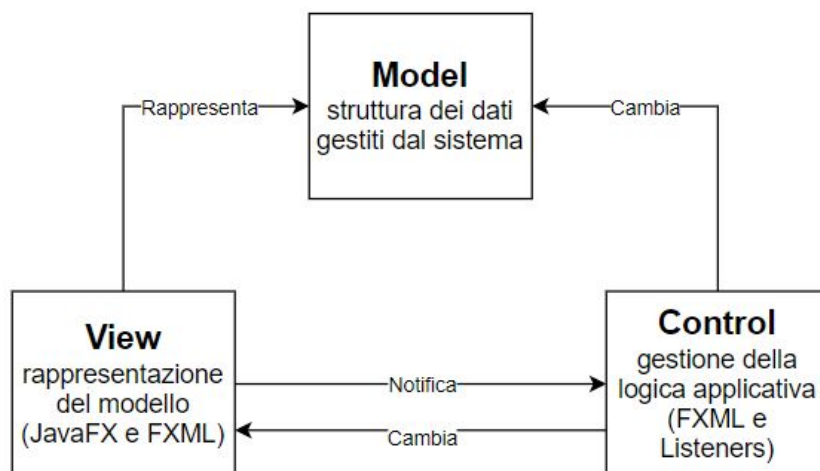
Il metodo lavorativo attuato è stato essenzialmente di tipo “*Scrum*”, basato sullo sviluppo iterativo, infatti prima di ogni incremento il gruppo si riuniva tramite conferenze online per definire gli obiettivi generali e le specifiche da completare entro una scadenza definita.

A completamento di un modulo del software veniva condotta un'attività di test mirata a risolvere eventuali errori o a migliorare il codice. La fase finale dello sprint si concludeva con la stesura della documentazione per l'incremento.

Prima di cominciare il ciclo principale, si è condotta una fase di *analisi dei requisiti*, che ha portato alla generazione dei casi d'uso e dei diagrammi di attività.

La maggior parte del lavoro è stata svolta a distanza ricorrendo al metodo del *pair programming*; non potendo lavorare fianco a fianco sono stati usati strumenti di controllo di versione come Git e piattaforme per le videoconferenze.

Progettazione e pattern architetturali usati



Per lo sviluppo del progetto è stato deciso di basarsi sul design architetturale *MVC* (Model View Controller) in quanto l'organizzazione e la gestione del sistema risultano particolarmente comode lavorando con *JavaFX* e *FXML* che supportano naturalmente questo modello. Le tre parti del modello vengono divise a livello logico e di sviluppo in

packages all'interno del progetto dando un senso di ordine e permettendo una più semplice gestione dei file sorgenti.

Viene spiegato di seguito, brevemente, il ruolo di ogni componente dell'architettura:

- **Modello:** è la sottoparte del sistema che fornisce la rappresentazione dei dati e delle informazioni necessarie al funzionamento del sistema, in particolare qui vengono definite tutte le classi, le interfacce e le strutture dati che il sistema gestisce.
- **Vista:** è la sottoparte del sistema che riguarda la rappresentazione grafica del modello, quindi riguarda tutte le interfacce utente che permettono con le quali si interagisce col sistema.
- **Controllore:** è la sottoparte del sistema che gestisce la logica applicativa, quindi definisce i comportamenti del sistema sulla base dell'interazione da parte dell'utente.

Le azioni a livello utente verranno catturate dai *Listener* e gestite dai relativi *Event Handlers* e in base alla specifica attività la vista verrà aggiornata conseguentemente, segnalando eventuali errori commessi dall'utente.

Database

Vista la necessità di mantenere in memoria un insieme di dati persistenti abbiamo deciso di appoggiarci al sistema di database *MySQL*. Per ogni classe della categoria Model è stata creata una corrispondente tabella nel database dalla quale i controller recuperano i dati salvati ed istanziano gli oggetti necessari. È stata inoltre realizzata una classe di interfacciamento che provvede a nascondere la complessità delle operazioni di comunicazione con il *DBMS* (si veda DAO Pattern).

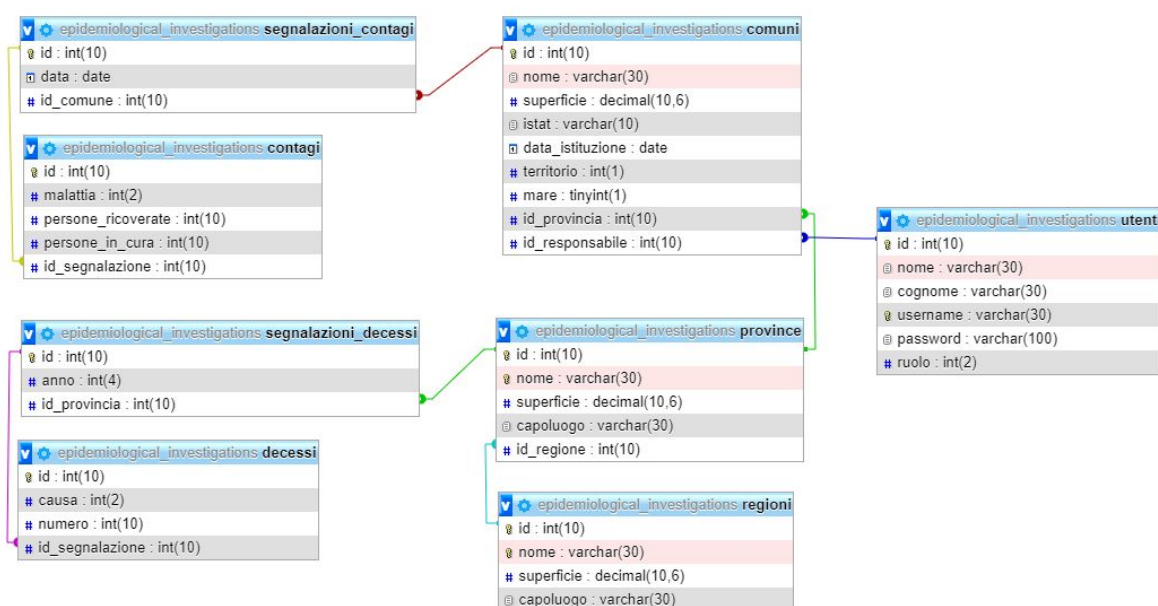


Figura 8: Schema del Database

Diagramma delle classi

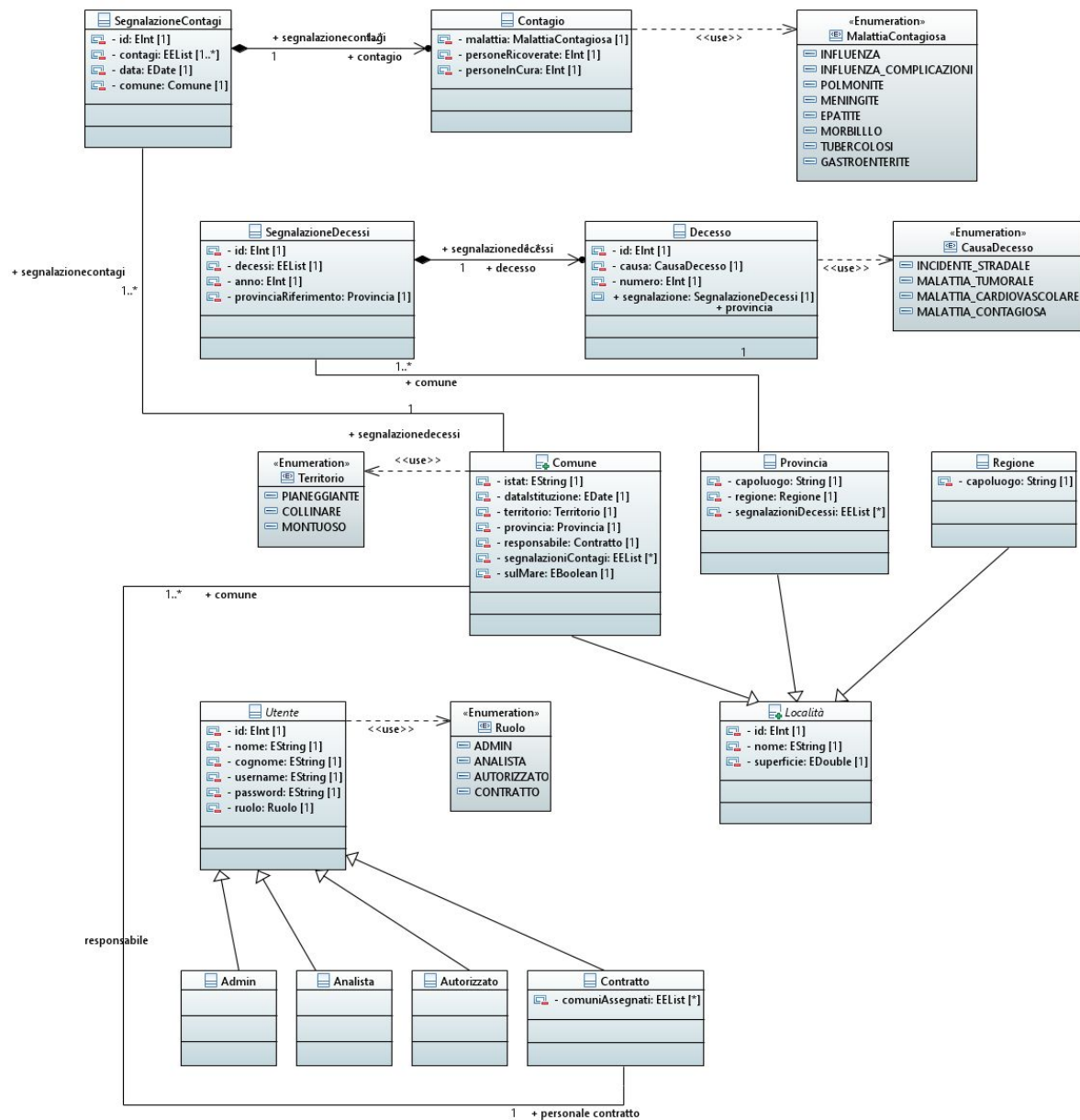


Figura 9: Diagramma delle classi

Organizzazione della GUI

Di seguito sono mostrate le classi controller che formano l'interfaccia del sistema con i principali metodi attivabili dagli utenti che interagiscono con gli oggetti grafici.

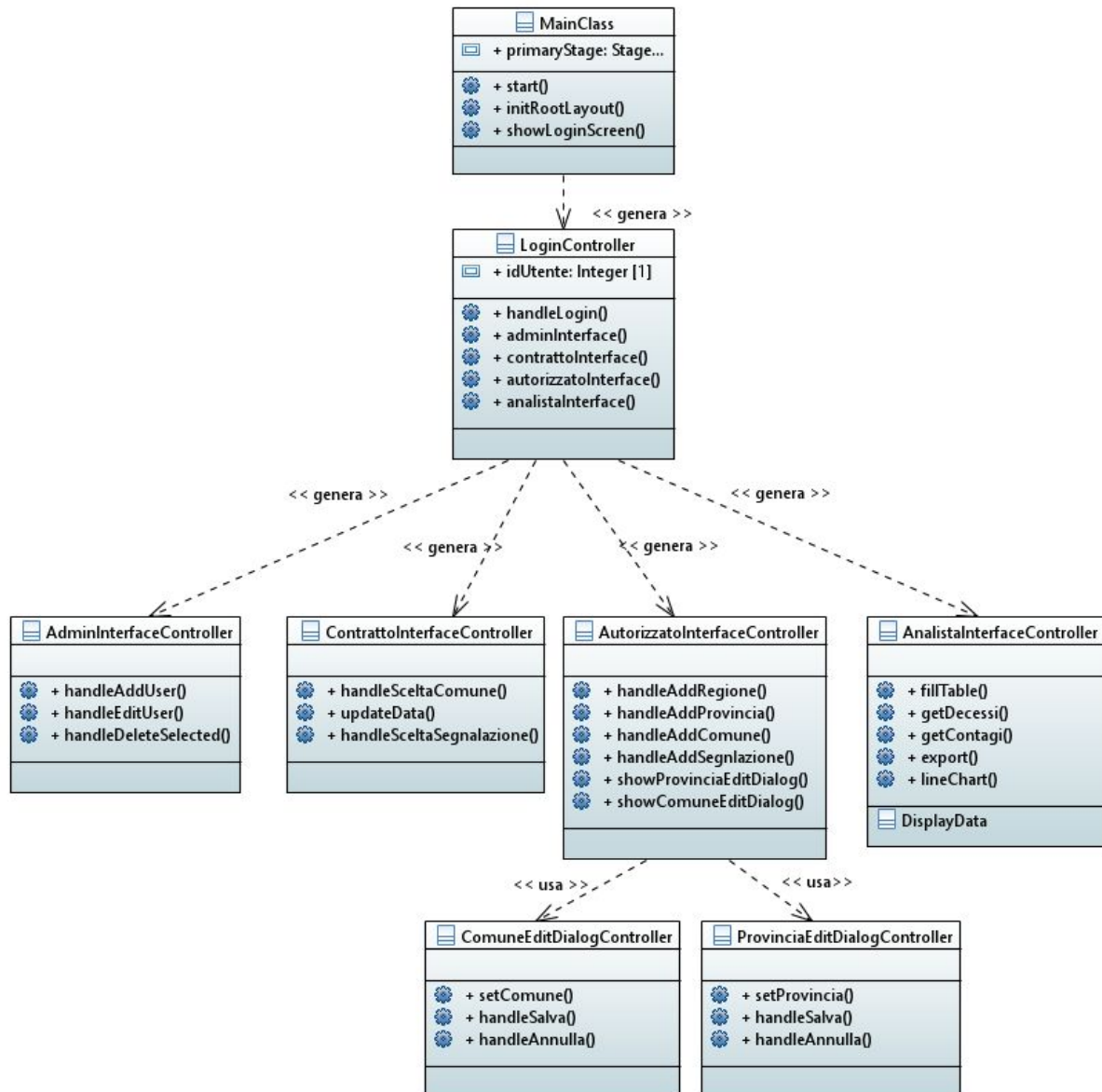


Figura 10: Diagramma delle classi della GUI

Diagrammi di sequenza

I *Diagrammi di sequenza* mostrano le dinamiche di alcune interazioni tra le classi del progetto particolarmente complesse o interessanti.

I diagrammi presentati di seguito semplificano la logica del programma in quanto non sono state inseriti direttamente i metodi associati ad ogni classe per una maggiore chiarezza esplicativa.

Inserimento di una nuova segnalazione relativa ai contagi riscontrati in un comune

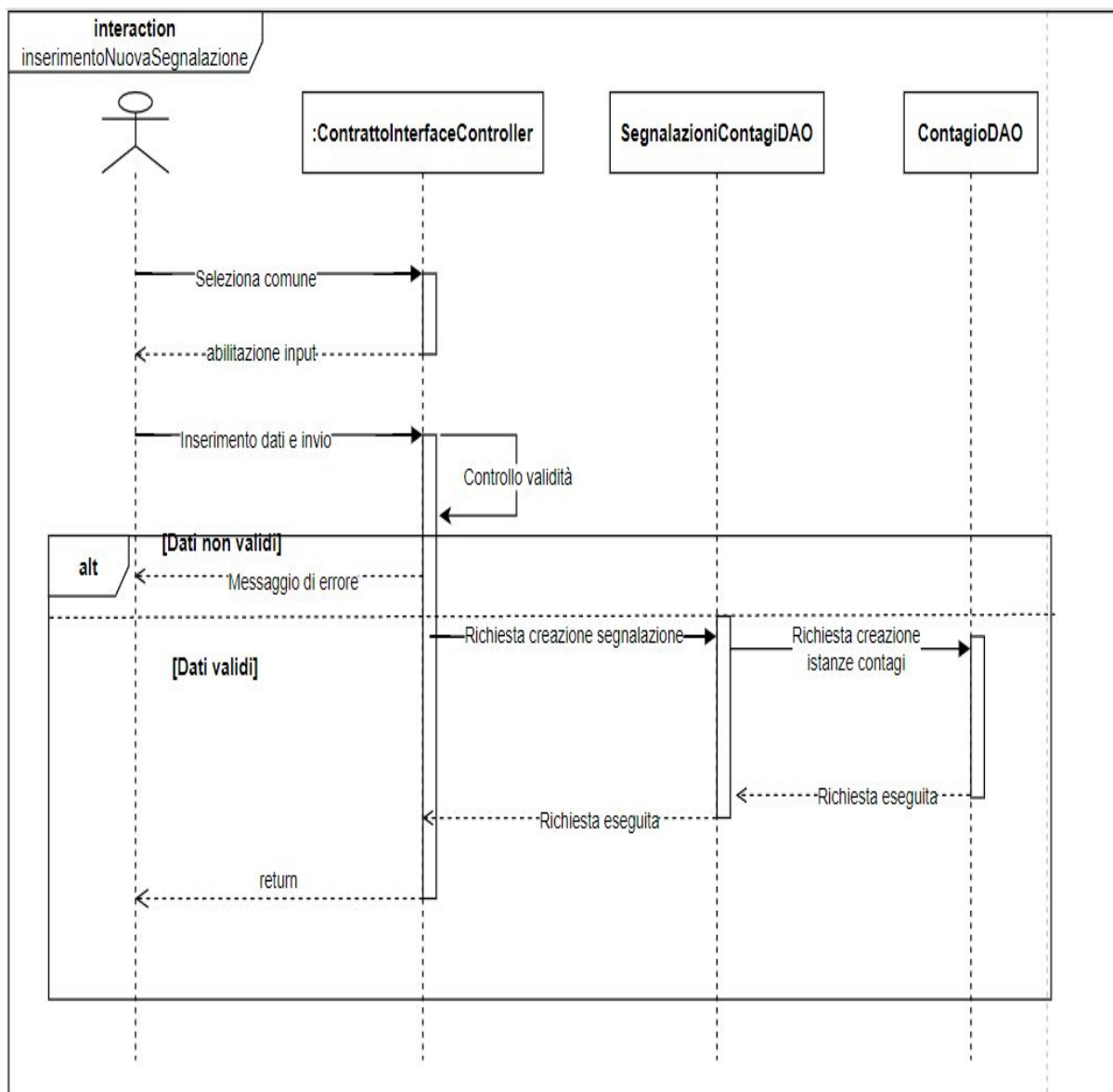


Figura 11: Diagramma di sequenza Inserimento nuova segnalazione di Contagio

Inserimento / modifica dei dati di Regioni e Comuni

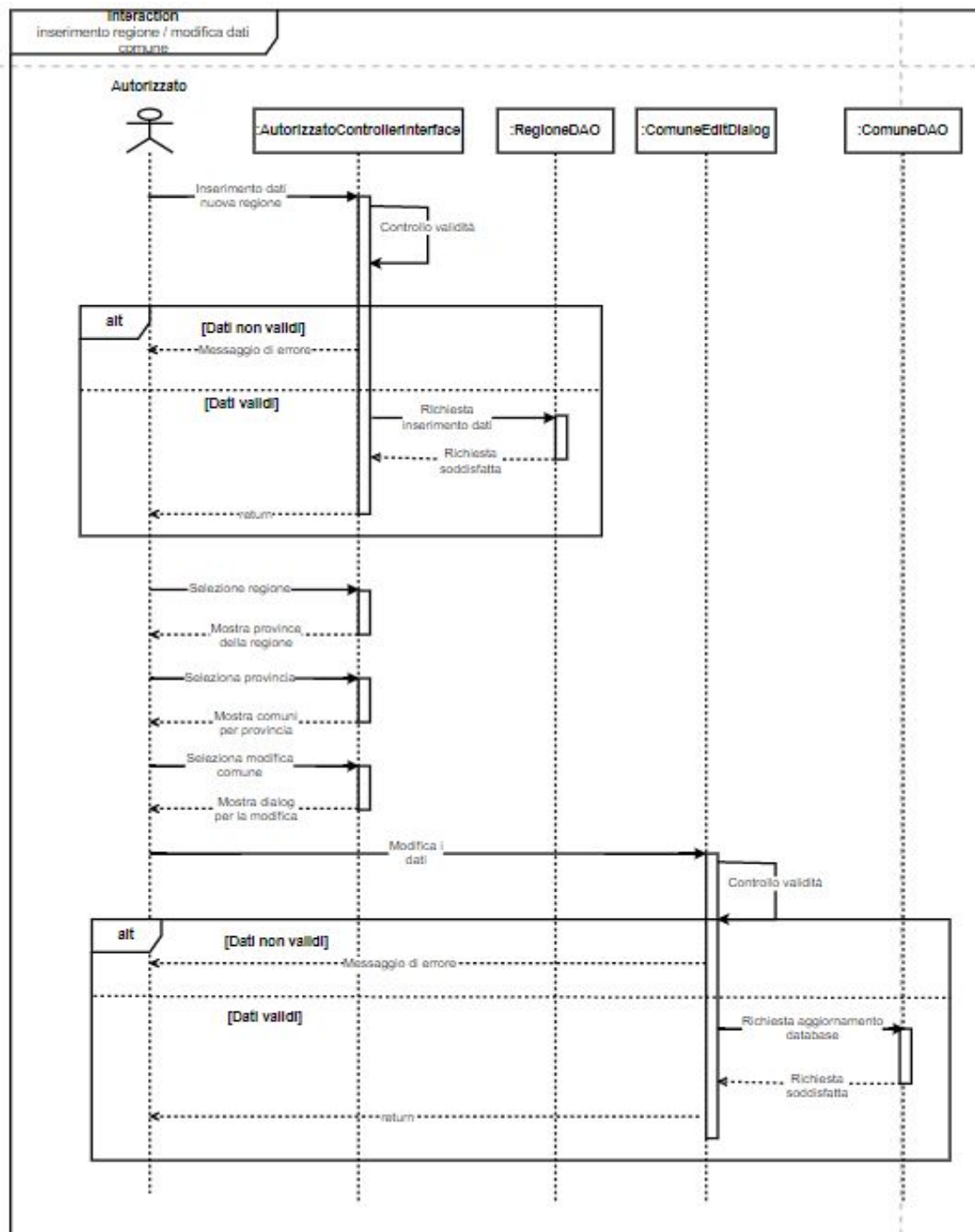


Figura 12: Diagramma di sequenza inserimento/modifica dati geografici

Design pattern utilizzati

Il sistema implementa alcuni dei più noti design pattern, di seguito vengono descritti quelli utilizzati:

- **Pattern DAO (Data Access Object):** è il pattern utilizzato per l'interazione con il database, in particolare fornisce una separazione tra il livello di accesso ai dati e la costruzione degli oggetti corrispondenti. Ogni classe della sezione Model è rappresentata da una tabella del database e dispone quindi della relativa classe DAO per l'interrogazione e il salvataggio dei dati.
- **Pattern Singleton:** è il pattern che permette un minor utilizzo di risorse andando a creare una singola istanza di una classe. Questo pattern viene combinato con le classi DAO in modo da creare un singolo oggetto per le query sul database.
- **Pattern Factory:** è il pattern che nasconde all'esterno la logica di classi DAO necessarie per istanziare un oggetto che comunica con il database. Infatti, la classe MySQLDAOFactory si occupa di instaurare la connessione con il database e restituire l'oggetto DAO relativo alla tabella da interrogare / modificare.

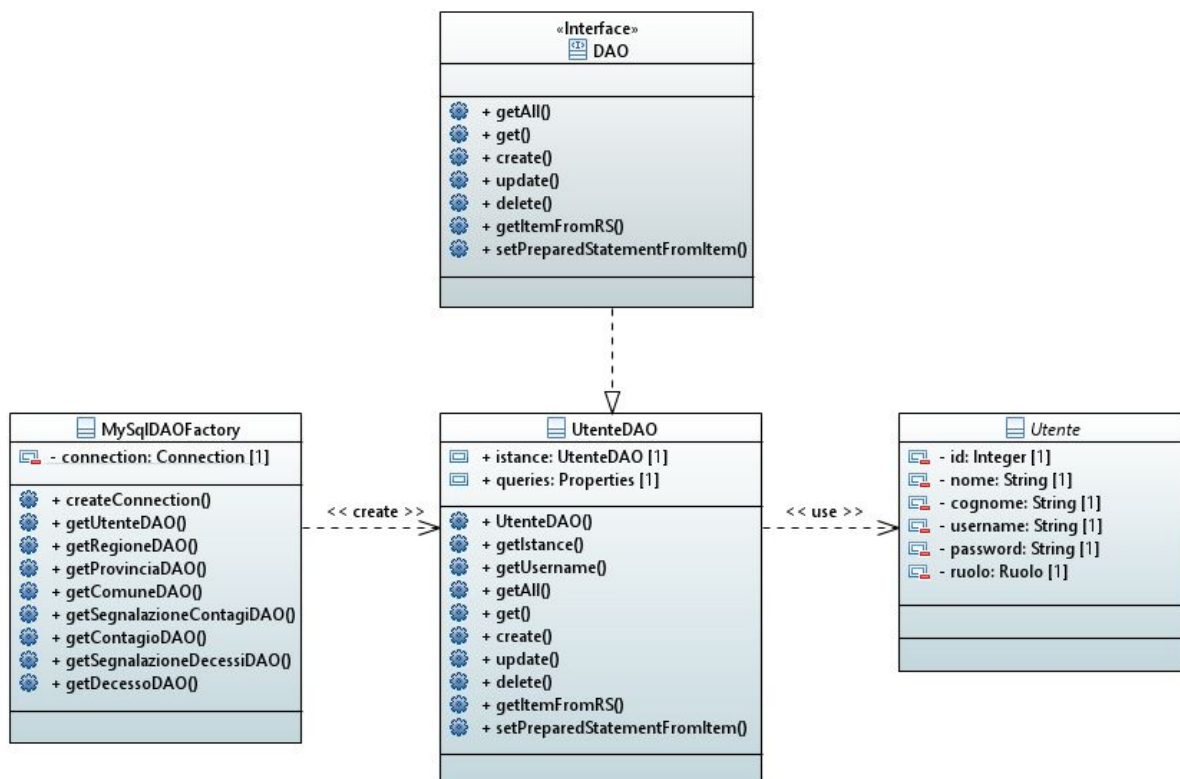


Figura 13: Diagramma delle classi di alcuni pattern utilizzati

Attività di Test e Validazione

La progettazione di ogni elemento del software è stata fortemente legata ad una fase di testing. Ogni volta che una nuova funzionalità o interfaccia grafica veniva implementata vi era subito un test da parte degli sviluppatori per assicurarsi che funzionasse in modo appropriato.

Per ogni componente sono stati testati anche i casi limite, inserendo dati errati, per assicurarsi che il programma rispondesse adeguatamente fermando le operazioni di modifica e notificando l'utente dell'errore.

Quando il software è diventato operativo sono stati eseguiti dei test sull'intero sistema, in ogni sua componente; infine dopo un refactoring del codice per uniformarlo e limare alcune imprecisioni questi test sono stati ripetuti.

Abbiamo poi verificato che il software riflettesse quanto scritto sulla documentazione, in particolare abbiamo controllato staticamente il codice per accertare la correttezza dei diagrammi UML.

Gli autori del test sono stati in primis gli sviluppatori, che hanno svolto test sia durante la scrittura del software, che una volta finito.

La conoscenza del funzionamento del codice ed il conseguente condizionamento ci ha portati a rivolgerci ad altri utenti estranei al progetto ma comunque inseriti nel contesto universitario e con un background informatico. Questi ci hanno permesso di rilevare velocemente errori riguardanti principalmente la gestione di alcuni casi limite.

Infine hanno eseguito dei test anche altri utenti completamente estranei al software e mediamente inesperti in ambito informatico; questi ultimi sono stati di fondamentale importanza per avere un feedback sull'usabilità del software e sulla gradevolezza dell'interfaccia grafica e ci hanno permesso di limare alcune imperfezioni al fine di rendere il programma più user-friendly.