

UNIVERSITÀ DEGLI STUDI DI VERONA



DIPARTIMENTO DI INFORMATICA

CORSO DI LAUREA TRIENNALE IN INFORMATICA

PROGETTO DI INGEGNERIA DEL SOFTWARE

GESTIONALE REAZIONI AVVERSE DA VACCINO ANTICOID E ANTINFLUENZALE

Anno Accademico 2021–2022

Danial Enayatzadeh
VR464061

Alessio Gjergji
VR455916

Michael Boadi
VR458310

Indice

1	Requisiti dell'interazione utente e sistema	2
1.1	Specifiche casi d'uso	2
1.1.1	Note generali	2
1.2	Casi d'uso relativi ai Medici	3
1.2.1	Effettuare segnalazioni	3
1.2.2	Inserimento dati del paziente	4
1.2.3	Inserimento dati della reazione avversa	4
1.2.4	Visualizzare i dati relativi ai pazienti	5
1.3	Casi d'uso relativi al farmacologo	6
1.3.1	Ricezione avvisi	6
1.3.2	Visualizzare segnalazioni	7
1.3.3	Effettua analisi	8
1.3.4	Proporre fase di controllo	9
1.4	Diagrammi di attività	9
2	Sviluppo: progetto dell'architettura e dell'implementazione del sistema	13
2.1	Note sul processo di sviluppo	13
2.2	Progettazione e pattern architetturali usati	13
2.2.1	MVC	13
3	Diagramma delle classi	14
3.1	Rappresentazione della view	15
3.2	Rappresentazione del model	15
3.2.1	Breve commento sulle interfacce	15
3.3	Classe <i>Model</i>	15
4	Diagramma delle attività a software implementato	16
4.1	Inserimento dei dati relativi al paziente	17
4.2	Inserimento di una segnalazione per un nuovo paziente	17
4.3	Visualizzazione degli avvisi non letti	18
5	Pattern di progettazione DAO	19
6	Considerazioni progettuali	20

7	Test e validazione	20
7.1	Ispezione del codice e della documentazione	20
7.2	test degli sviluppatori	21
7.3	Test con dati corretti	22
7.4	Test specifici	23
7.5	Test utente generico	24
8	Immagini relative ai test	25

1 Requisiti dell’interazione utente e sistema

1.1 Specifiche casi d’uso

1.1.1 Note generali

Il sistema proposto supporta l’utilizzo della piattaforma al personale suddiviso in Medici e Farmacologi. Entrambe le categorie hanno a disposizione delle credenziali (*pre-fornite dagli amministratori del sistema*) con cui possono effettuare l’autenticazione. In caso questa vada a buon fine il personale verrà reindirizzato alla rispettiva schermata iniziale, a seconda della professione associata alle credenziali. Esse contengono nell’identificativo come prima lettera “M” per i medici, mentre il farmacologo “F”.

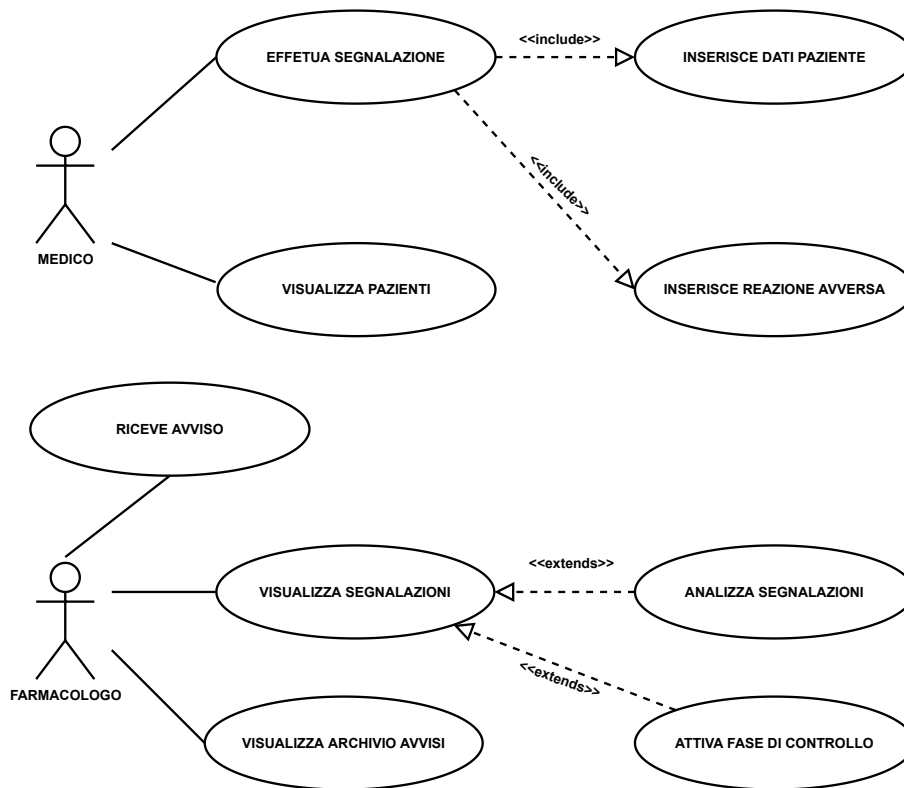


Figura 1: Casi d'uso

1.2 Casi d'uso relativi ai Medici

Dopo una opportuna autenticazione il medico viene introdotto ad una interfaccia che permette l'inserimento dei dati del paziente e la visualizzazione gli identificativi relativi ai propri pazienti, ed eventualmente visualizzare informazioni aggiuntive.

1.2.1 Effettuare segnalazioni

I medici devono poter effettuare delle segnalazioni. Per far questo è necessario **inserire i dati relativi al paziente, i dati relativi reazione avversa e i dati relativi alla vaccinazione** ed eventuali **fattori di rischio**. La data della segnalazione viene autonomamente dal sistema.

Caso d'uso: Inserimento segnalazione

Attori:

Medico

Precondizioni:

1. Il medico deve essere autenticato.

Sequenza di eventi:

1. Il medico è introdotto all'interfaccia di base.
2. (a) Il medico inserisce i dati del paziente.
(b) Il medico seleziona un paziente già esistente.
3. Il medico inserisce i dati relativi alla reazione.
5. Il medico inserisce i dati relativi alle vaccinazioni precedenti.
6. Il medico conferma i dati inseriti.

Postcondizioni:

1. La segnalazione è inserita con successo.

I dati inseriti dal medico vengono salvati al momento della conferma dei dati inseriti del paziente su un file.

1.2.2 Inserimento dati del paziente

Nella fase di inserimento dei dati, vi sono due alternative:

- Compilazione di una nuova scheda del paziente, completa delle informazioni del paziente, tra le quali: professione, provincia di residenza, fattori di rischio, anno di nascita e identificativo univoco.
- Selezione di una scheda pre-esistente, **tra quelle a carico del medico in questione**, inserendo l'identificativo.

L'identificativo univoco del paziente è gestito dal sistema, ciò significa che non dovranno esserci pazienti con stesso identificativo, anche tra medici diversi.

1.2.3 Inserimento dati della reazione avversa

L'inserimento della reazione avversa è accompagnata dell'inserimento delle vaccinazioni del paziente.

- Inserimento in sistema di una reazione avversa, immettendo un nome univoco, una gravità (*compresa tra 1 e 5*), una descrizione, la relativa data e l'inserimento delle vaccinazioni accompagnate delle tipologie di vaccino per poter segnalare che vaccinazione potrebbe aver provocato tale reazione.
- Inserimento delle informazioni relative al vaccino, immettendo tipologia, data, numero di dose e sede.

1.2.4 Visualizzare i dati relativi ai pazienti

Sarà possibile visualizzare i dati di tutti i pazienti di cui il medico ha effettuato una segnalazione. Per semplicità, il sistema mette a disposizione la possibilità di caricare i dettagli relativi al singolo paziente ricercando il suo identificativo.

Caso d'uso per la visione di tutti i pazienti a carico del medico

Caso d'uso: Visualizzazione pazienti
Attori: Medico
Precondizioni: 1. Il medico deve essere autenticato.
Sequenza di eventi: 1. Il medico è introdotto all'interfaccia di base. 2. Il medico preme il pulsante per visualizzare l'elenco degli identificativi di tutti i suoi pazienti.
Postcondizioni: 1. Il sistema fornisce le informazioni relative ai pazienti.

Caso d'uso per la visione del singolo paziente

Caso d'uso: Visualizzazione paziente
Attori: Medico
Precondizioni: 1. Il medico deve essere autenticato.
Sequenza di eventi: 1. Il medico è introdotto all'interfaccia di base. 2. Il medico preme il pulsante per visualizzare l'elenco degli identificativi di tutti i suoi pazienti. 3. Il medico ricerca l'identificativo relativo al paziente che gli interessa.
Postcondizioni: 1. Il sistema fornisce le informazioni relative al paziente indicato.

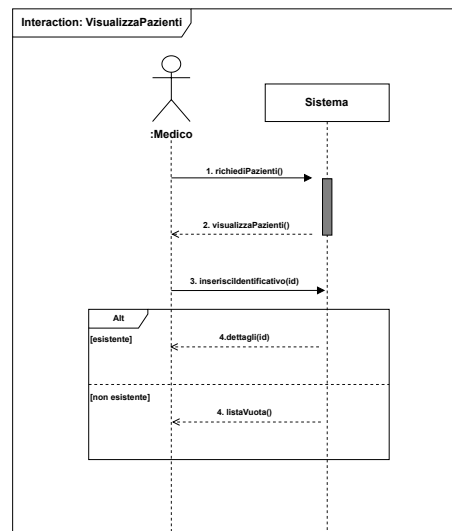


Figura 2: Visualizzazione dei pazienti a carico del medico curante.

1.3 Casi d'uso relativi al farmacologo

1.3.1 Ricezione avvisi

Il sistema deve fornire un meccanismo di gestione ed invio di avvisi verso il farmacologo responsabile, per farlo può utilizzare tre metodi:

1. Avviso settimanale (*inviato ogni fine settimana*).
2. Quando il numero di segnalazioni raggiunge la soglia di 50.
3. Quando un vaccino ha accumulato in un mese oltre 5 segnalazioni di gravità superiore a 3.

Il sistema avverte il farmacologo responsabile, ciò significa che quando il responsabile effettuerà l'autenticazione, gli avvisi ricevuti potranno essere consultabili nella scheda relativi agli avvisi non letti. Tramite un'opzione presente nel menù è possibile rivedere avvisi in precedenza letti.

Visualizzazione avvisi non ancora letti

Attori:

Farmacologo

Precondizioni:

1. Il farmacologo deve essere autenticato, devono esservi degli avvisi nel sistema.

Sequenza di eventi:

1. Il farmacologo accede all'interfaccia di base.
2. Il farmacologo seleziona l'opzione di lettura degli avvisi non letti.

Postcondizioni:

1. Gli avvisi devono essere segnati come già letti.

Visualizzazione avvisi già letti

Attori:

Farmacologo

Precondizioni:

1. Il farmacologo deve essere autenticato.

Sequenza di eventi:

1. Il farmacologo visualizza l'interfaccia di base.
2. Il farmacologo seleziona l'opzione di lettura degli avvisi già letti.

Postcondizioni:

1. Gli avvisi già letti vengono visualizzati.

1.3.2 Visualizzare segnalazioni

Il farmacologo dopo l'autenticazione può vedere la lista di segnalazioni presenti in memoria. Può vederne i dettagli ed **effettuare delle analisi**.

Visualizzare segnalazioni

Attori:

Farmacologo

Precondizioni:

1. Il farmacologo deve essere autenticato.

Sequenza di eventi:

1. Il farmacologo visualizza l'interfaccia di base.
2. Il farmacologo accede all'interfaccia per il controllo delle segnalazioni.

Postcondizioni:

1. Il farmacologo visualizza le segnalazioni. Potrà passare alla fase di analisi.

1.3.3 Effettua analisi

Il farmacologo può effettuare alcune analisi sulle segnalazioni presenti in memoria. Sarà possibile visualizzare all'interno dell'interfaccia le informazioni relative a:

- Conta segnalazioni per provincia.
- Conta vaccinazioni per sede di vaccinazione.
- Conta segnalazioni gravi in settimana (*gravità maggiore di 3*).

Selezionando la tipologia di vaccino a cui è interessato, vengono mostrati sulla schermata i dettagli relativi al numero di segnalazioni presenti per vaccino.

Con ulteriori opzioni è possibile filtrare le segnalazioni per vaccino, sede e provincia.

Effettua analisi

Attori:

Farmacologo

Precondizioni:

1. Il farmacologo deve essere autenticato e aver visualizzato le segnalazioni.

Sequenza di eventi:

1. Il farmacologo seleziona l'analisi che gli interessa

Postcondizioni:

1. Il sistema fornisce la risposta.

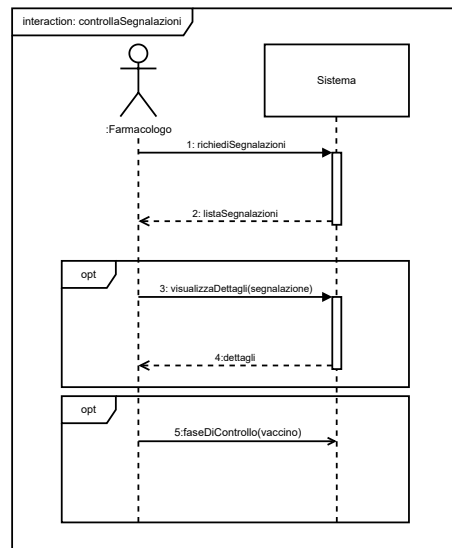


Figura 3: Controllo delle segnalazioni da parte del farmacologo.

1.3.4 Proporre fase di controllo

In base alle analisi eseguite e alla lettura delle segnalazioni, il farmacologo può proporre di effettuare un controllo su un determinato tipo di vaccino. Il sistema deve registrare e tracciare queste proposte per ogni vaccino.

Propone fase di controllo

Attori:

Farmacologo

Precondizioni:

1. Il farmacologo deve essere autenticato e aver visualizzato le segnalazioni.

Sequenza di eventi:

1. Il farmacologo seleziona il vaccino interessato.
2. Il farmacologo conferma.

Postcondizioni:

1. Il sistema memorizza la relativa proposta che verrà tracciata dal sistema.

1.4 Diagrammi di attività

Nota: *i seguenti diagrammi catturano una singola attività di un utente rispetto al sistema. Non è stata rappresentata nel diagramma la possibilità di ripetere più volte la stessa operazione in sequenza*

senza chiudere il software. Questo per ragioni di chiarezza e leggibilità; si considerano quindi le singole attività d'interazione.

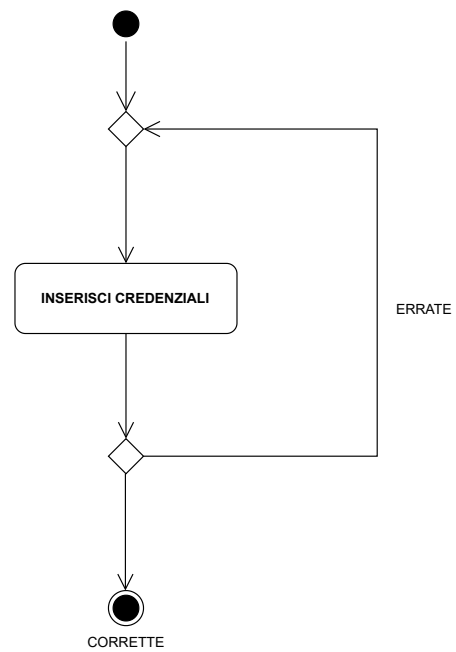


Figura 4: Autenticazione di un utente.

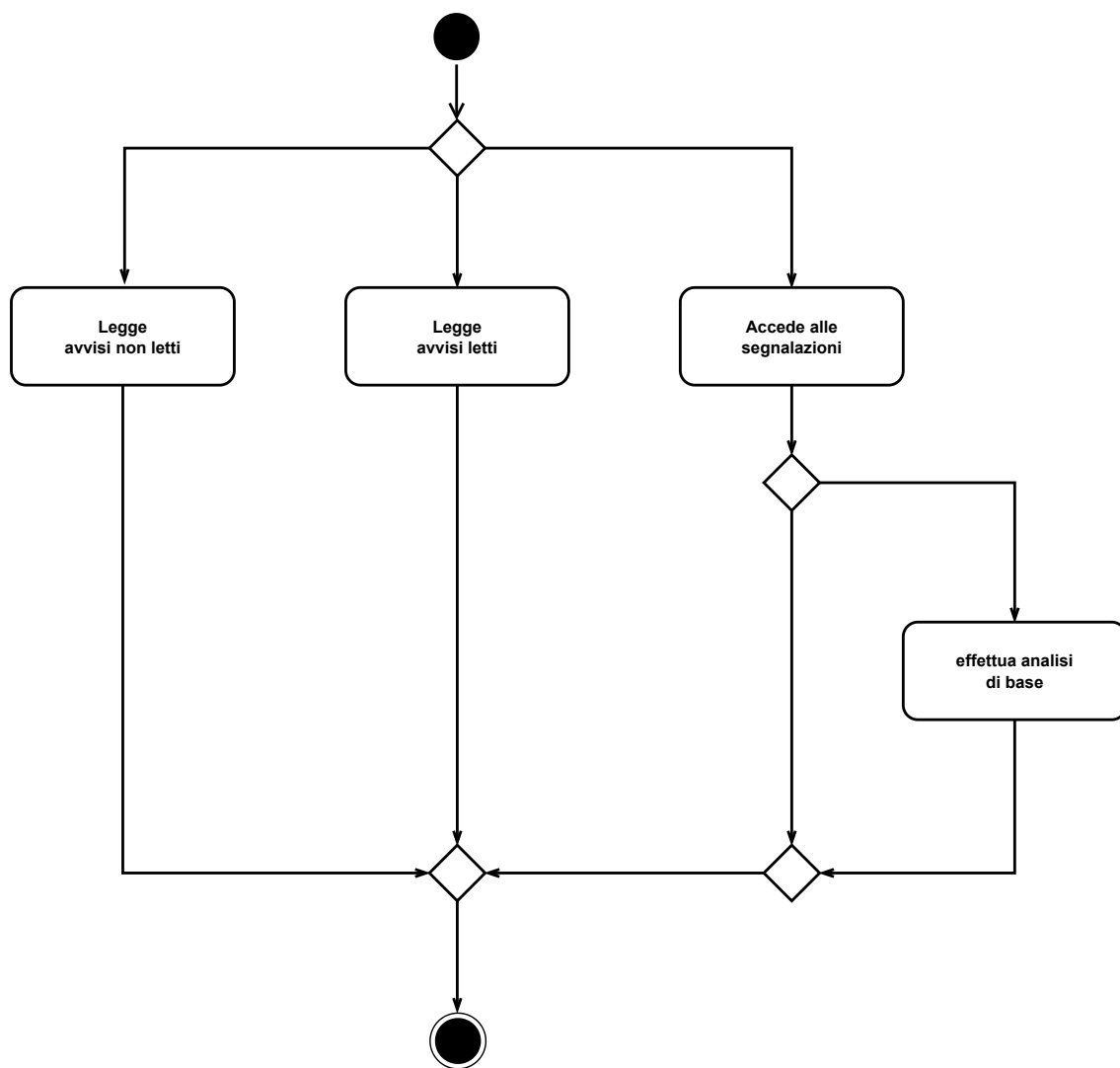


Figura 5: Farmacologo.

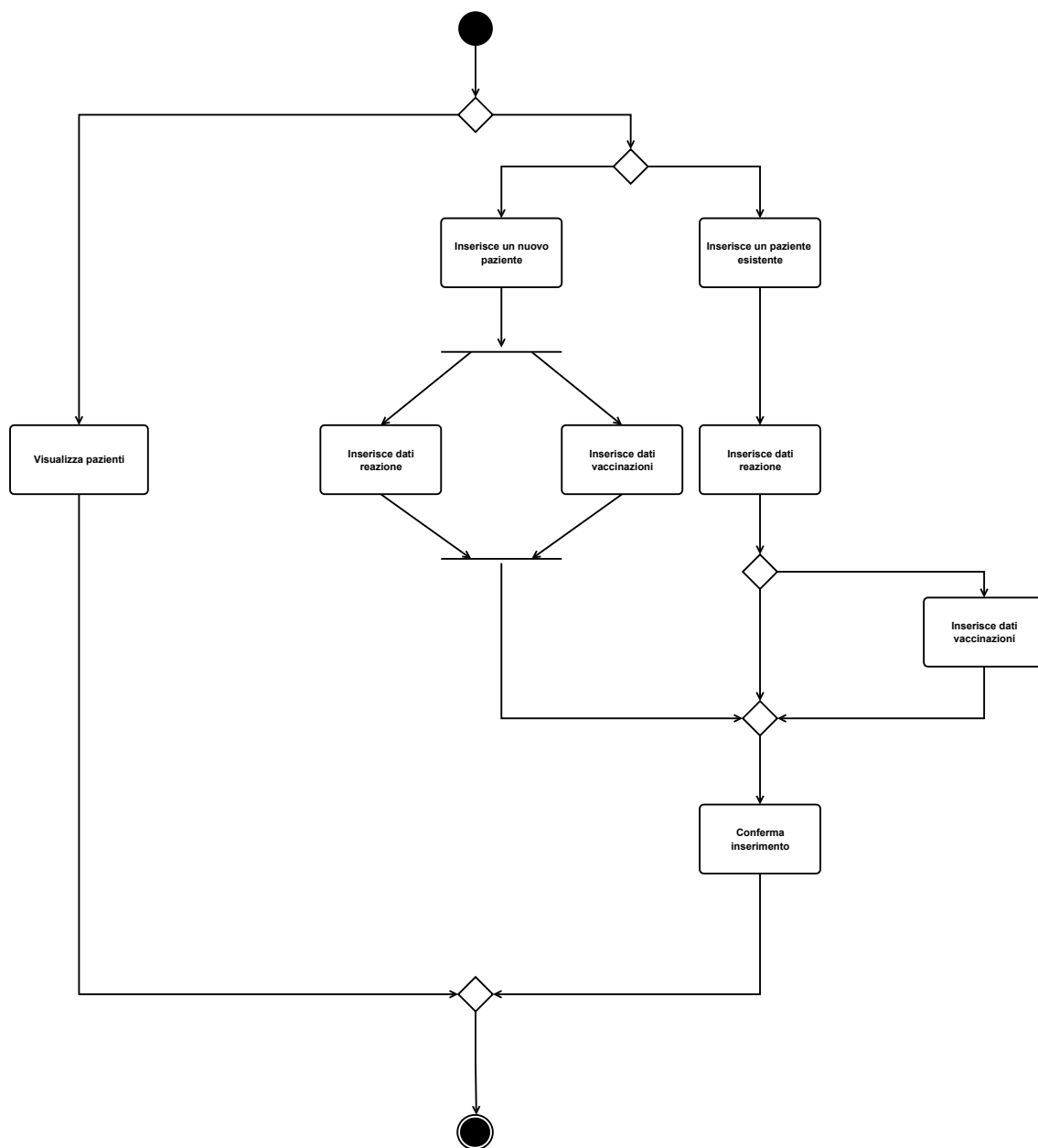


Figura 6: Medico.

2 Sviluppo: progetto dell'architettura e dell'implementazione del sistema

2.1 Note sul processo di sviluppo

Il processo di sviluppo è stato essenzialmente di tipo *agile* ed *incrementale*. Tuttavia, si è cercato quanto più possibile di mantenerlo sequenziale (*seppure inserendole all'interno di un ciclo*) le fasi di progettazione, implementazione e validazione. Questo è stato fatto semplicemente per cercare di procedere sempre prima progettando e dopo implementando, con l'obiettivo di avere sempre un percorso strutturato da seguire. Dopo ogni modifica significativa (*visione*) è stata condotta una breve attività di test. Si noti che queste fasi non sono sempre state lineari, ma hanno incluso attività di *refactoring* su componenti in precedenza sviluppati. Durante ogni ciclo, e parallelamente a queste tre attività, è costruita la documentazione relativa al progetto. In questa attività si è raccolto il materiale UML generato nelle fasi di test e di progettazione (*principalmente diagrammi delle classi*) nel presente documento complessivo, aggiungendo inoltre UML descrittivi (i *sequence diagram sono stati prodotti dopo aver implementato i relativi metodi*).

Prima di cominciare un ciclo principalmente, si è condotta la fase di analisi dei requisiti generando i relativi *use-cases* e i *diagrammi di attività*. Anche la (*semplice*) progettazione architetture è stata fatta prima di iniziare il lavoro centrale in modo da assicurare di trovarsi per lo meno in una situazione solida da quel punto di vista.

Per quanto riguarda l'implementazione, non sono state fatte grosse divisioni o piani di sviluppo programmatici. Si è seguito l'ordine prioritario di sviluppo andando ad aggiungere ciò che era ritenuto necessario man mano.

2.2 Progettazione e pattern architetturali usati

2.2.1 MVC

Il sistema è stato progettato utilizzando le tecniche di modellazione ad oggetti. Dal punto di vista architetture, si è scelto di utilizzare il **pattern MVC**. Le ragioni sono semplici: si è pensato che, essendo nativamente implementato con *JavaFX* e relativi *FXML* (*utilizzate per produrre l'interfaccia grafica del progetto*) fosse la scelta adeguata.

Ciò ha permesso di separare nettamente e comodamente le tre componenti a livello logico e di sviluppo. Ogni membro inserito in un **package** separato.

- **Model:** sottoparte del sistema che riguarda i dati e le informazioni memorizzate. Definisce la struttura dell'informazione gestita dal sistema.
- **View:** sottoparte del sistema che rappresenta visivamente il modello, e quindi, i dati del sistema. Si è utilizzato *javaFX* (*file FXML e l'ausilio del tool grafico di Scene Builder*) per realizzare questa parte.
- **Application:** dove sono presenti i controller, sottoparte del sistema che definisce la logica applicativa, ovvero il comportamento del sistema a fronte degli stimoli esterni. Consiste nel listener che ascoltano la vista.

le azioni dell'utente saranno catturate dai listener. I listener saranno istruiti a reagire di conseguenza, andando a modificare le informazioni contenute in *Model* e quindi ad aggiornare la *View*, che rappresenta il modello.

Non sono state operate ulteriori articolazioni: model, view, control sono tre blocchi monolitici. Questo perchè si è considerato che i vari elementi costituiti giacessero sullo stesso piano.

Seguono un semplice schema dell'architettura del sistema ed i diagrammi UML delle classi Model, View e Control. Le interazioni fondamentali tra Model-View-Control avranno visibili nei diagrammi di sequenza.

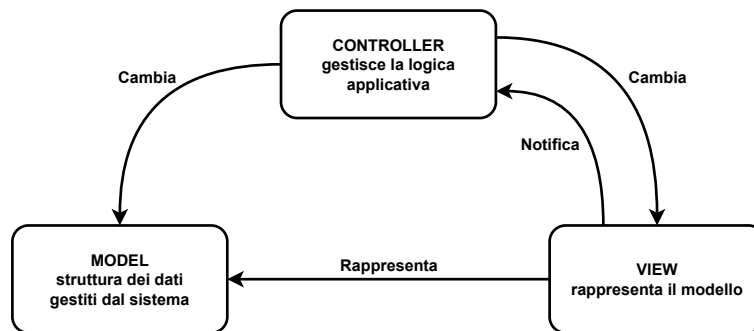


Figura 7: Pattern architetturale MVC.

3 Diagramma delle classi

Si è scelto di rappresentare i diagrammi delle classi relative al *model* e al *view* perchè le più espressive dal punto di vista architetturale.

3.1 Rappresentazione della view

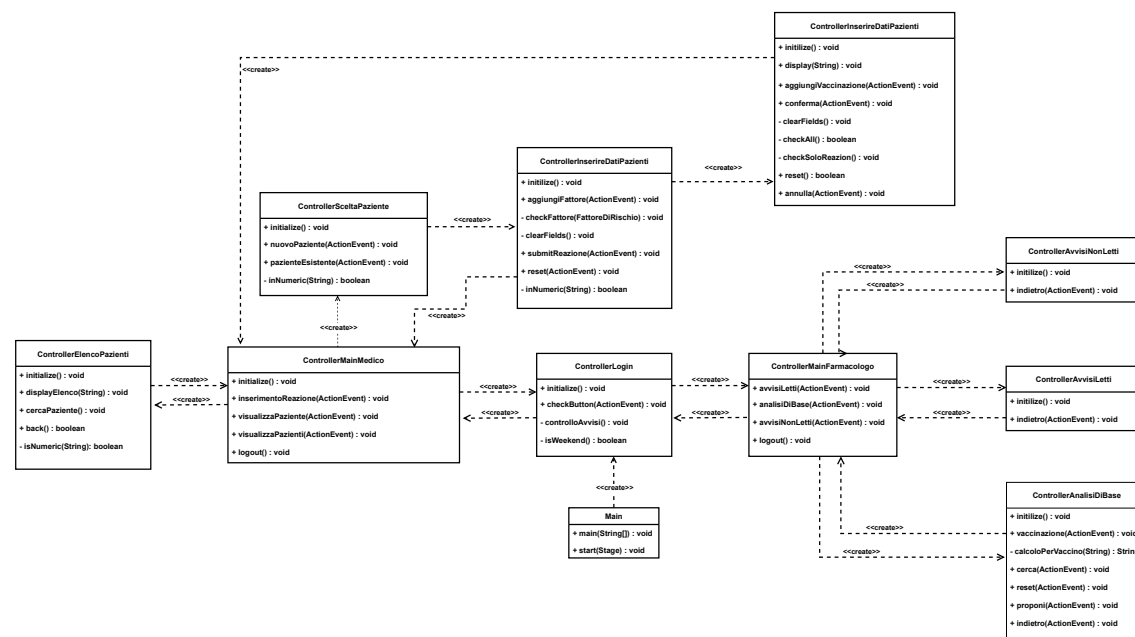


Figura 8: Rappresentazione della sequenza di invocazione delle scene.

3.2 Rappresentazione del model

Le classi contenute nel package `model` implementano l'interfaccia `Serializable` per poter ricreare gli oggetti dopo averli inseriti nei relativi file, si è scelto di non rappresentare il collegamento con tale interfaccia per facilitare la lettura di tale diagramma delle classi.

3.2.1 Breve commento sulle interfacce

Per gestire la comunicazione tra Modello, Vista e Controllore, sono state adottate le seguenti scelte. Per la comunicazione Vista-Controllore, si è banalmente utilizzato il meccanismo basato su *ActionEvent*. L'implementazione è standard, per cui è possibile sorvolare.

Per la comunicazione Modello-Vista e Modello-Controllore, si è utilizzato come punto di accesso alle informazioni le classi *PazientiDaoImpl*, *SegnalazioniDaoImpl*, *AvvisiDaoImpl*, che immagazzinano e forniscono dati. Per i dettagli riguardo la sua implementazione si rimanda al capitolo 5.

3.3 Classe *Model*

Le nostre classi che rappresentano il modello fungono da intermediari per il trasferimento di dati relativi a pazienti, segnalazioni e avvisi, in questo modo le nostre classi relative ai controllori non accedono direttamente ai dati, garantendo una netta separazione per una maggiore mantenibilità del software.

4.1 Inserimento dei dati relativi al paziente

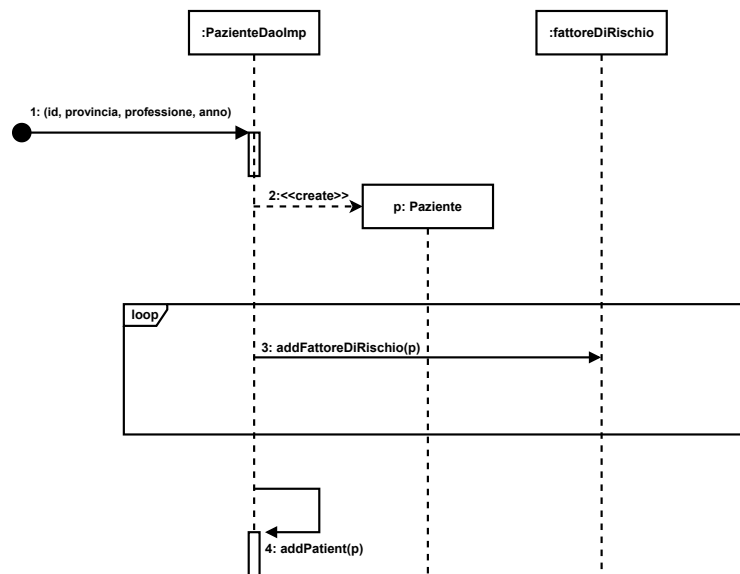


Figura 10: aggiunta di un paziente con relative informazioni nel sistema.

4.2 Inserimento di una segnalazione per un nuovo paziente

L'inserimento della segnalazione relativa ad un paziente nuovo necessita di almeno una vaccinazione presente per tale paziente, per un paziente esistente è possibile indicare solamente la reazione avversa, perchè sicuramente presenterà almeno una vaccinazione.

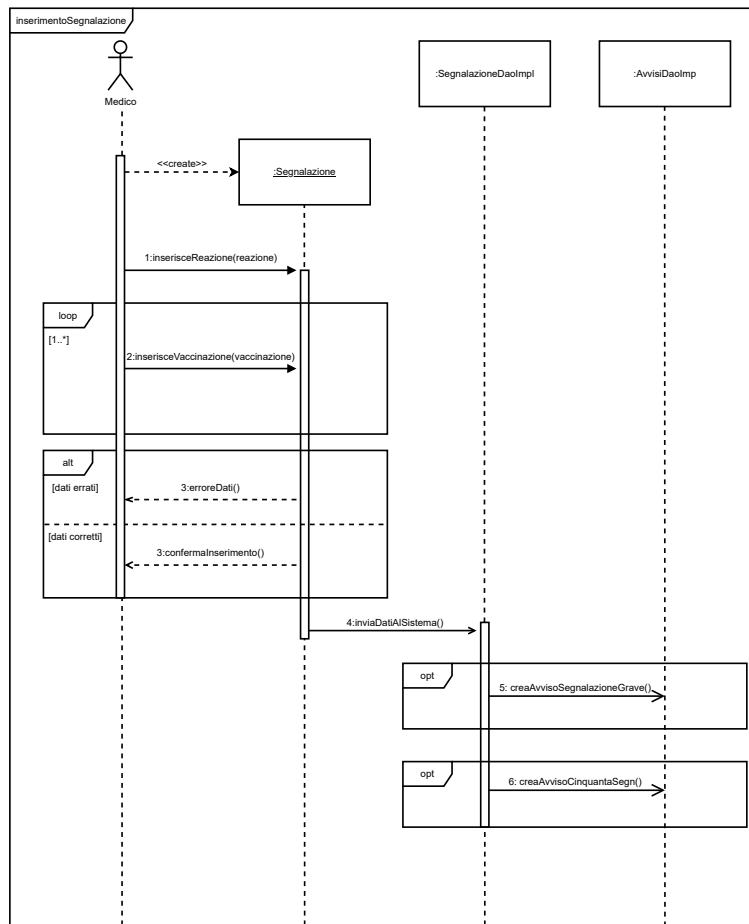


Figura 11: inserimento di una segnalazione con i relativi dati.

4.3 Visualizzazione degli avvisi non letti

La visualizzazione degli avvisi letti e non letti si differenzia solamente nel trasferimento di avvisi non letti in avvisi letti.

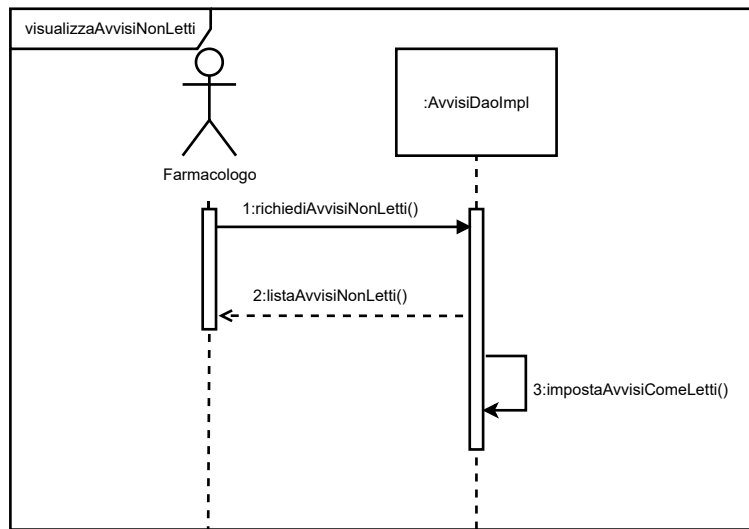


Figura 12: visualizzazione degli avvisi non letti da parte del farmacologo.

5 Pattern di progettazione DAO

Per la gestione dei dati si è ritenuto di utilizzare il pattern DAO, utilizzato per separare i dati di basso livello che accedono all'API dalle operazioni di alto livello. Per ogni classe del Model si è implementato:

- **Interfaccia DAO** - Definisce le operazioni standard da eseguire.
- **Classe concreta DAO** - Implementa l'interfaccia sopra indicata. Con tale classe si accede ai dati presenti nei file.
- **Model Object** - Questo è una semplice classe Java che contiene i metodi setter, getter e metodi ausiliari.

Le classi-modello implementate con il pattern DAO sono:

- **Avvisi** - relativa alla gestione degli avvisi del farmacologo.
- **Segnalazione** - relativa alle segnalazioni effettuate dai medici.
- **Paziente** - relativa alla gestione dei dati dei pazienti.

Nota Non è stato deciso di modellare ulteriori classi secondo il pattern DAO, nello specifico le classi *Vaccinazione*, *Reazione* e *FattoriDiRischio* perchè possiamo accedere alle relative informazioni con facilità attraverso il paziente e alle segnalazioni.

6 Considerazioni progettuali

Nelle considerazioni progettuali abbiamo pensato che nel tempo, il paziente potesse sviluppare nuove reazioni e di conseguenza abbiamo inserito la possibilità di immettere nuove eventuali vaccinazioni effettuate dal paziente, in presenza di una nuova reazione. Le reazioni dello stessa tipologia possono svilupparsi più volte in un paziente, ovviamente l'inserimento di una reazione dello stesso tipo e nella stessa data è un errore che deve essere intercettato.

Tutte le vaccinazioni devono essere in giorni diversi, e le vaccinazioni covid devono rispettare l'ordine cronologico rispetto alla dose inserita.

Nel sistema è possibile inserire anche dosi successive in assenza delle precedenti, infatti un medico potrebbe non interessarsi di determinate vaccinazioni. In presenza di dose unica per vaccinazione covid, non è possibile effettuare vaccinazioni della stessa casa farmaceutica (*es. in data 12/7/2021 è stata effettuata una vaccinazione covid con Moderna ed è stata una dose unica, non è possibile inserire vaccinazioni Moderna*).

Le vaccinazioni Antinfluenzali sono sempre dosi uniche, non vi è limite di inserimento.

Gli avvisi settimanali vengono inviati durante il fine settimana, ogni volta che il farmacologo effettua un'autenticazione. Si presuppone infatti che il farmacologo effettui solo un'autenticazione per il controllo settimanale.

Le proposte da parte del farmacologo vengono inserite su un file consulabile e sono composte da tipo di vaccino e data relativa alla proposta. Le proposte non vengono controllate dal sistema, è possibile infatti inserire ogni tipologia di proposta anche nello stesso giorno e dello stesso tipo.

I campi relativi alla professione e alla sede di vaccinazione sono inseribili dall'utente, i valori ammessi sono alfabetici.

Per utilizzare l'opzione di ricerca presente nelle analisi di base del medico se non si vuole effettuare alcuna ricerca per vaccino è fondamentale inserire l'opzione "NONE".

7 Test e validazione

Per verificare la solidità del software prodotto, si sono svolte le seguenti attività:

1. Ricognizione del documento e delle specifiche e confronto con i diagrammi prodotti.
2. Verifica della consistenza tra diagrammi e codice prodotto.
3. Ispezione del codice, verifica della correttezza dei pattern, ricerca di malepratiche varie.
4. Test degli sviluppatori software.
5. Test utente generico sul software.

7.1 Ispezione del codice e della documentazione

In questa fase si è semplicemente rivisto il documento delle specifiche e lo si è confrontato con i diagrammi UML prodotti, per verificare la correttezza degli *use case*, *activity diagram* e *diagrammi delle classi*. Una volta finita questa attività si è confrontato il codice (*staticamente*) ai diagrammi UML, per verificarne la consistenza. Infine si è data una nuova ispezione del codice per cercare punti sensibili, malepratiche e cattivi usi dei pattern.

7.2 test degli sviluppatori

In questa fase gli sviluppatori hanno immesso nel sistema degli input (*sia corretti che errati*) per vedere se la reazione del software fosse quella attesa. Alcuni dei test svolti sono:

- **Verifica del corretto funzionamento dell'autenticazione:** i dati errati vengono respinti, ed a fronte dei dati corretti all'utente viene mostrata la schermata iniziale legata alla sua professione.
- **Farmacologo in assenza di dati presenti nei file:** si è provato a premere ogni pulsante pigiabile in assenza assoluta di dati, per vedere se questo creasse eccezioni o errori.
- **Medico in assenza di dati presenti nei file:** si è provato a premere ogni pulsante pigiabile in assenza assoluta di dati, per vedere se questo creasse eccezioni o errori.
- **Inserimento dei dati del paziente ed eventuali fattori di rischio:** si è provato a premere ogni pulsante presente nella scena in assenza di dati o con dati inseriti in maniera errata, cercando di "catturare" ogni possibile inserimento errato.
- **Inserimento di una segnalazione e vaccinazioni:** La fase di verifica ha seguito due strade:
 - Inserimento di una seconda segnalazione ad un paziente precedentemente registrato nel sistema immettendo nuove vaccinazioni (*anche con dati errati, come sequenza di dosi in date non compatibili*) o non immettendole perchè precedentemente inserite.
 - Inserimento di una prima segnalazione ad un paziente non ancora presente nel sistema, tale controllo è stato effettuato come nel caso precedente, accertandosi che potesse contenere vaccinazioni precedenti alla data della reazione e che fosse stata inserita almeno una vaccinazione da associare a tale segnalazione.
- Verificato che **il medico vedesse tutti i suoi pazienti** e non altri pazienti appartenenti ad altri medici e gli eventuali dati relativi a vaccinazioni e reazioni inseriti.
- **Verificato il sistema di avvisi** (*Avviso del fine settimana, avviso della soglia dei 50 e avviso delle 5 reazioni di gravità superiore a 3*).
- **Verifica dei diversi inserimenti di input:** input errati, vuoti, stringe malformate, numeri negativi.
- Verifica della reazione del software ad inserimento di date incompatibili con vaccinazioni, reazioni e nascita.
- Inserimenti di vaccinazioni, reazioni, pazienti e segnalazioni duplicati (*abbiamo pensato che si potessero verificare reazioni della stessa natura in date diverse per lo stesso paziente*).
- Controllato che il meccanismo di filtraggio dei dati nell'analisi effettuata dal farmacologo rispettasse le sue richieste (*ricerca per tipo di vaccino, ricerca per sede di vaccinazione, ricerca per luogo di nascita*).

7.3 Test con dati corretti

Si è deciso di effettuare un test per verificare se il sistema fosse in grado di mantenere i dati in maniera corretta e per verificare la correttezza degli avvisi inviati dal sistema. Si è utilizzato un piccolo programma che generasse 500 pazienti qui di seguito riportato:

```
for (int i = 0; i < 251; i++) {
    long minDay = LocalDate.of(1970, 1, 1).toEpochDay();
    long maxDay = LocalDate.of(2000, 12, 31).toEpochDay();
    long randomBirth =
        ThreadLocalRandom.current().nextLong(minDay, maxDay);
    String job = generatingRandomAlphabeticString();
    Paziente tmp = new Paziente(i,
        LocalDate.ofEpochDay(randomBirth),
        job,
        province.get(rnd.nextInt(15)));
    pazienti.addPatient(tmp);
    pazienti.pazientiSerialize();
    minDay = LocalDate.of(2022, 6, 10).toEpochDay();
    maxDay = LocalDate.of(2022, 6, 22).toEpochDay();
    long randomVaccinazione =
        ThreadLocalRandom.current().nextLong(minDay, maxDay);
    String descrizione = generatingRandomAlphabeticString();
    //prima vaccinazione
    Vaccinazione vaccino = new Vaccinazione(i,
        LocalDate.ofEpochDay(randomVaccinazione),
        VaccinoTipo.values()[rnd.nextInt(6 - 2) + 2 ],
        DoseVaccino.IV,
        descrizione);
    List<Vaccinazione> vaccinazioni = new ArrayList<>();
    vaccinazioni.add(vaccino);
    //seconda vaccinazione
    minDay = LocalDate.of(2022, 6, 1).toEpochDay();
    maxDay = LocalDate.of(2022, 6, 9).toEpochDay();
    randomVaccinazione =
        ThreadLocalRandom.current().nextLong(minDay, maxDay);
    vaccino = new Vaccinazione(i,
        LocalDate.ofEpochDay(randomVaccinazione),
        VaccinoTipo.values()[rnd.nextInt(6 - 2) + 2 ],
        DoseVaccino.III,
        descrizione);
    vaccinazioni.add(vaccino);
    //terza vaccinazione
    minDay = LocalDate.of(2022, 5, 20).toEpochDay();
    maxDay = LocalDate.of(2022, 5, 30).toEpochDay();
    randomVaccinazione =
        ThreadLocalRandom.current().nextLong(minDay, maxDay);
```

```

        vaccino = new Vaccinazione(i,
            LocalDate.ofEpochDay(randomVaccinazione),
            VaccinoTipo.values()[rnd.nextInt(6 - 2) + 2 ],
            DoseVaccino.II,
            descrizione);
        vaccinazioni.add(vaccino);
        //quarta vaccinazione
        minDay = LocalDate.of(2022, 5, 10).toEpochDay();
        maxDay = LocalDate.of(2022, 5, 19).toEpochDay();
        randomVaccinazione =
            ThreadLocalRandom.current().nextLong(minDay, maxDay);
        vaccino = new Vaccinazione(i,
            LocalDate.ofEpochDay(randomVaccinazione),
            VaccinoTipo.values()[rnd.nextInt(6 - 2) + 2 ],
            DoseVaccino.I,
            descrizione);
        vaccinazioni.add(vaccino);
        pazienti.updateListaVaccinazione(i, vaccinazioni);
        pazienti.pazientiSerialize();
        minDay = LocalDate.of(2022, 6, 22).toEpochDay();
        maxDay = LocalDate.of(2022, 7, 22).toEpochDay();
        long randomReazione =
            ThreadLocalRandom.current().nextLong(minDay, maxDay);
        Reazione reazione = new Reazione(i,
            TipoReazione.values()[rnd.nextInt(6 - 2) + 2 ],
            LocalDate.ofEpochDay(randomReazione),
            RiskLevel.values()[rnd.nextInt(4)], descrizione);
        pazienti.updateReazione(i, reazione);
        Segnalazione segn = new Segnalazione(tmp,
            reazione, LocalDate.ofEpochDay(randomReazione));
        segnalazioni.addSignal(segn);
        segnalazioni.addSegnalazioneToReazione(i, segn, reazione);
        segnalazioni.addSegnalazioneToVaccinazione(i, segn,
            vaccinazioni);
        segnalazioni.addSignalToPatient(i, segn);
        System.out.println("Aggiunto il paziente: " + i);
        segnalazioni.senalazioniSerialize();
        pazienti.pazientiSerialize();
        avvisi.avvisiSerialize();
    }
}

```

7.4 Test specifici

- Login: il login con dati errati mostra una *label* di colore rosso che avvisa l'utente dell'incorrettezza dei dati, il login con i dati corretti ci porta alla schermata specifica legata alla professione.

- Nel'elenco dei pazienti si è provato ad inserire stringhe malformate per cercare l'identificativo interessato, verificando che i dati immessi fossero respinti.
- Nella scelta del paziente, si è provato ad inserire stringhe malformate, dati di pazienti di altri medici o pazienti inesistenti verificando che il sistema non li accettasse.
- Nell'inserimento dei dati del paziente si è verificato che immettendo dati errati, dati non completi il sistema li respingesse in qualunque caso. Per dati non corretti si intende anche dati di pazienti esistenti.
- Nell'inserimento di un nuovo paziente, si è provato ad inserire segnalazioni in assenza di vaccini, con stringhe malformate e con errori di diverso tipo, tra cui: vaccinazioni successive alle reazioni, vaccinazioni in date errate, vaccinazioni in ordine di dosi errate, più di quattro vaccinazioni anticovid, vaccinazioni con date uguali e vaccinazioni antinfluenzali che non fossero dosi uniche.
- Nell'inserimento di un paziente esistente, si è verificato che durante l'annullamento dell'inserimento di tale paziente non si eliminassero i dati precedentemente inseriti e che si potesse inserire segnalazioni senza compilare campi relativi alle vaccinazioni precedenti (*perchè precedentemente inserite*).
- Si è verificato che gli avvisi generati fossero corretti con il *tool* di generazione di segnalazioni specificato in precedenza e che, se letti, fossero inseriti in avvisi non letti.
- Si è verificato che al selezionamento di un determinato vaccino o della sede di vaccinazione o del comune relativo al paziente presente nelle analisi di base del farmacologo rispecchiassero il numero effettivo indicato.
- Si è verificato che inviando determinate proposte non fossero proposte vuote, bisogna segnalare il tipo di vaccinazione.

Le immagini relative al testing sono presenti alla sezione 8

7.5 Test utente generico

Come utile scaglione il software è stato sottoposto ad un test da parte di alcuni utenti individui con limitata dimestichezza informatica e assoluto distacco dallo sviluppo. In questa fase non si è cercato in nessun modo di guidare o strutturare l'esperienza, per non influenzare in alcun modo il risultato, piuttosto si è lasciato che l'utente sperimentasse la navigabilità del progetto. Non è stata data nessuna spiegazione sull'utilizzo del software, se non una generale indicazione dei suoi fini e delle credenziali per l'autenticazione.

8 Immagini relative ai test

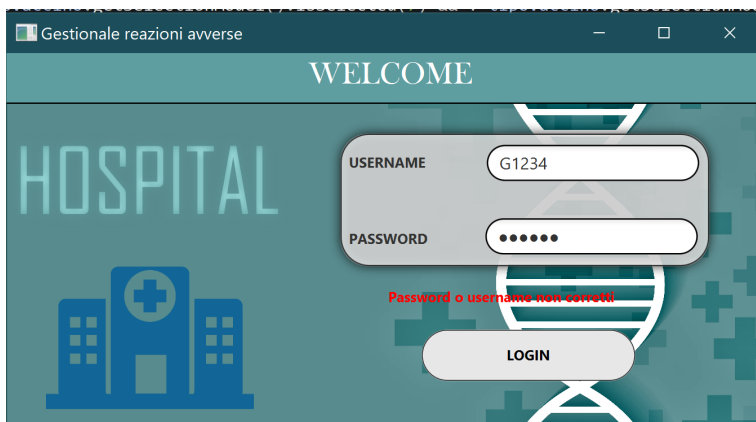


Figura 13: Dati errati nel login.

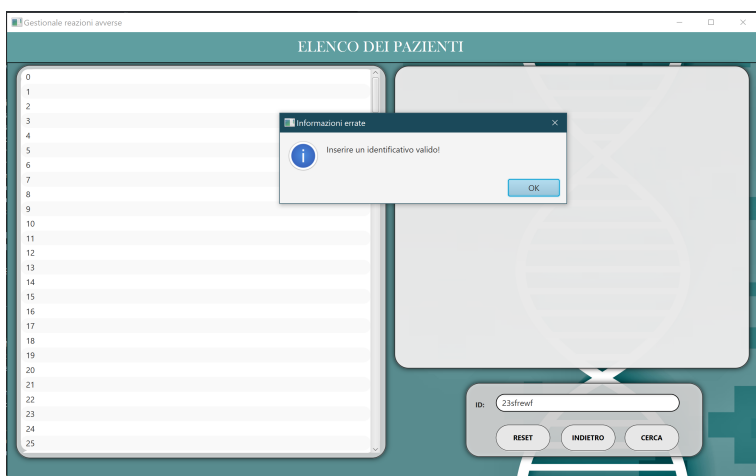


Figura 14: Stringhe malformate.

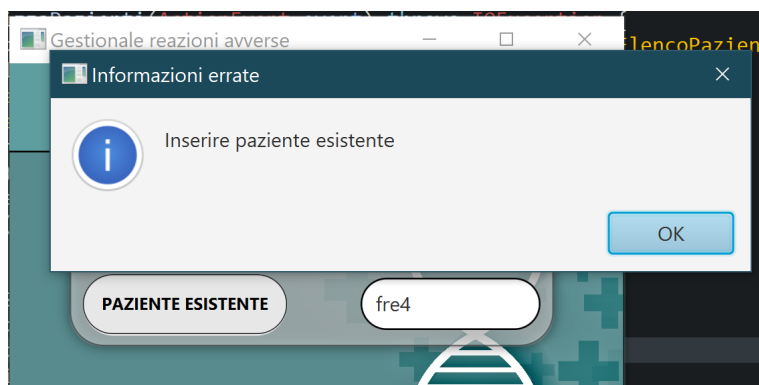


Figura 15: Srtinghe malformate nella ricerca di pazienti esistenti.

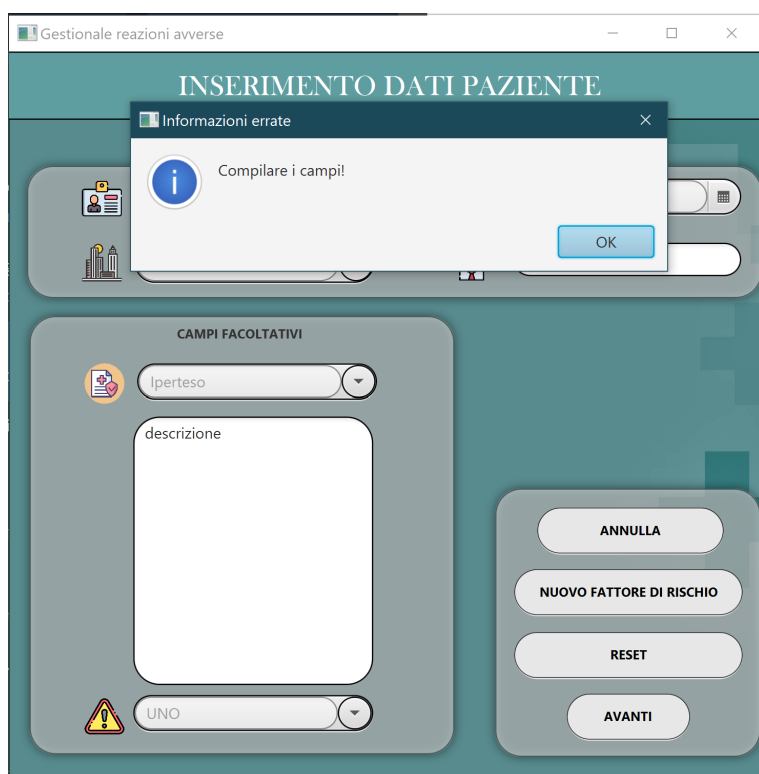








Figura 16: Dati non corretti nell'inserimento dei pazienti.

Gestionale reazioni avverse


SEGNALAZIONE RELATIVA AL PAZIENTE


 15 7/14/2020 

 ASMA 



 DUE 



Descrizione



 Informazioni errate


 Il paziente deve avere almeno una vaccinazione

OK

 15 Data vaccinazione 

 vaccino 

 Dose 

 Sede

ANNULLA

RESET


AGGIUNGI VACCINAZIONE


CONFERMA


Figura 17: Segnalazioni in assenza di vaccinazioni.

Home Farmacologo

SEGNALAZIONE RELATIVA AL PAZIENTE


**15**





desc

Informazioni errate

 Il vaccino antinfluenzale è sempre una dose unica

OK

**15**







ANNULLA

RESET

AGGIUNGI VACCINAZIONE

CONFERMA

Figura 18: Le vaccinazioni antinfluenzali sono dosi uniche.

Gestionale reazioni avverse

ANALISI

Segnalazione numero: 2, del paziente 1
residente a: Alessandria, AL
ha avuto la seguente reazione: DERMATITI, 2022-07-23, 2022-07-07, UNO,
nella data: 2022-07-07,
segnalata il: 2022-07-23
LE VACCINAZIONI NEI MESI PRECEDENTI SONO STATE:
Vaccinazione del 2022-06-19 fatta a: ghtmrhimdx Dose n° IV, SPUTNIK
Vaccinazione del 2022-06-01 fatta a: ghtmrhimdx Dose n° III, ANTINFLUENZALE
Vaccinazione del 2022-05-24 fatta a: ghtmrhimdx Dose n° II, MODERNA
Vaccinazione del 2022-05-18 fatta a: ghtmrhimdx Dose n° I, MODERNA

Segnalazione numero: 8, del paziente 7
residente a: Alessandria, AL
ha avuto la seguente reazione: ALTRO, 2022-07-23, 2022-07-12, TRE, xdgyxmnny
nella data: 2022-07-12,
segnalata il: 2022-07-23
LE VACCINAZIONI NEI MESI PRECEDENTI SONO STATE:
Vaccinazione del 2022-06-20 fatta a: xdgyxmnny Dose n° IV, SINOVAC
Vaccinazione del 2022-06-03 fatta a: xdgyxmnny Dose n° III, SPUTNIK
Vaccinazione del 2022-05-26 fatta a: xdgyxmnny Dose n° II, ANTINFLUENZALE
Vaccinazione del 2022-05-15 fatta a: xdgyxmnny Dose n° I, MODERNA

Segnalazione numero: 12, del paziente 11
residente a: Alessandria, AL
ha avuto la seguente reazione: ALTRO, 2022-07-23, 2022-07-13, DUE, wumdluxpi
nella data: 2022-07-13,

Informazioni errate

Selezionare il tipo di vaccino

OK

PROPONI FASE DI CONTROLLO

NONE

Alessandria, AL

Sede

INDIETRO RESET CERCA

Figura 19: Proposte vuote.