TITOLO RELAZIONE

### A cura di

## INDICE:

## Requisiti ed interazioni utente-sistema 2

## Specifiche casi d’uso . . . . . . . . . . . . . . . . . . . . . . . . . . . . 2

## Casi d’uso relativi ai Medici . . . . . . . . . . . . . . . . . . . . . 3

## Casi d’uso relativi ai Farmacologi . . . . . . . . . . . . . . . . . . 6

## Diagrammi di attività . . . . . . . . . . . . . . . . . . . . . . . . . . . 10

## Sviluppo: progetto dell’architettura ed implementazione del sistema 13

## Note sul processo di sviluppo . . . . . . . . . . . . . . . . . . . . . . . 13

## Progettazione e pattern architetturali usati . . . . . . . . . . . . . . . 14

## Breve commento sulle interfacce . . . . . . . . . . . . . . . . . . . 18

## Implementazione e design pattern usati . . . . . . . . . . . . . . . . . 19

## Generalità . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 19

## Classe Model . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20

## Diagrammi di sequenza del software implementato . . . . . . . . 20

## Attività di test e validazione 27

## Ispezione codice e documentazione . . . . . . . . . . . . . . . . . . . . 27

## Unit test . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 27

## Test degli sviluppatori . . . . . . . . . . . . . . . . . . . . . . . . . . . 29

## Test utente generico . . . . . . . . . . . . . . . . . . . . . . . . . . . . 30

## Appendice: creazione di oggetti mediante la classe Model 31Requisiti ed interazioni utente-sistema

## Specifiche casi d’uso

## Il sistema proposto supporta la gestione per la prenotazione di passaporti da parte del personale addetto e le richieste di prenotazione per il cittadino, inizialmente identificati come utente generico. Solo successivamente, in base al percorso scelto vengono differenzializzati. L’applicazione Il cittadino è diviso in registrato(possibilità di fare login) e non registrato (registrazione necessaria).

## Nel primo caso, effettuato il login, il cittadino verrà indirizzato alla pagina iniziale che gli permetterà di fare richiesta per rinnovo/smarrimento/nuovo passaporto ecc…, in base alle disponibilità delle sedi gestite dal personale. Nel secondo caso è necessario che il cittadino si registri nel sistema altrimenti non può fare richiesta.

Il personale ha 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.

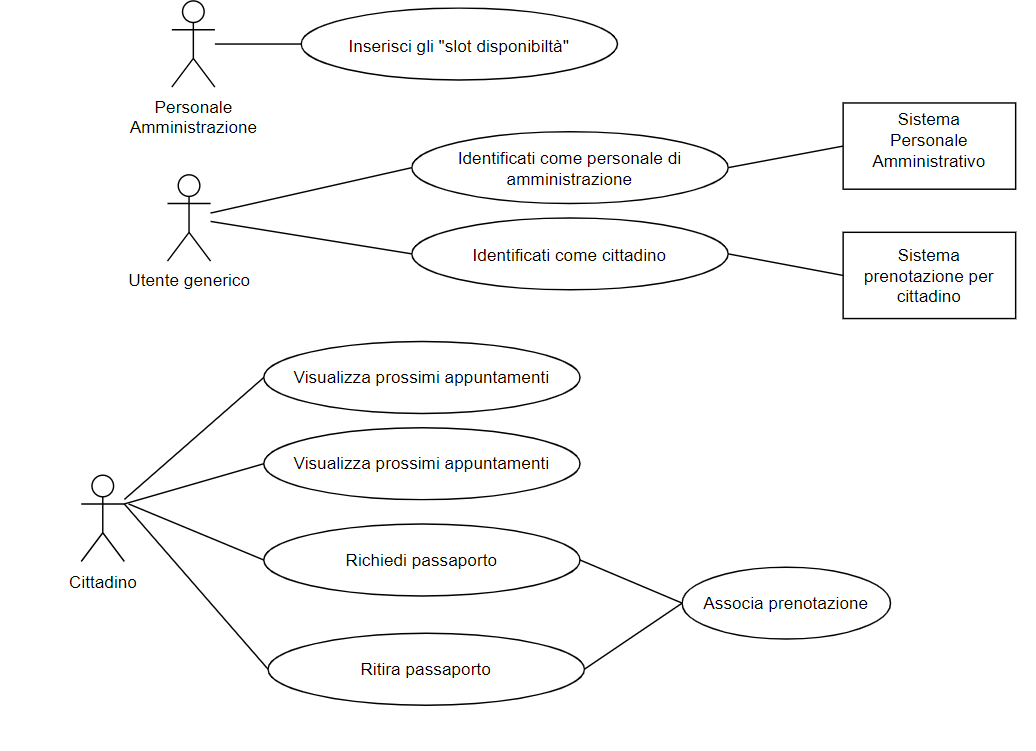


Figura: Diagramma casi d’uso

## Casi d’uso relativi a utente generico

L’utente si interfaccia con il sistema, dove viene subito invitato a scegliere il suo ruolo con conseguente autenticazione o registrazione (se non è registrato).

Nella home sono presenti due pulsanti: “Accesso personale amministrativo” e “Accesso per il cittadino”.

c

**Scelta “accesso personale amministrativo”**

L’utente generico che si identifica come personale amministrativo sarà proprietario di

un utente ed una password (pre-forniti dall’amministrazione) per poter accedere al sistema amministrativo.

**Scelta “accesso per il cittadino”**

L’utente generico che si identifica come cittadino, se privo di registrazione, deve necessariamente registrarsi nel sistema affiche quest’ultimo possa verificare la sua identità.

L’utente già registrato, può effettuare il login per entrare nel sistema.

|  |
| --- |
| *Attori: Utente generico**Precondizioni: Nessuna**Passi:**1. L’utente si interfaccia al sistema**2. L’utente* *(a). si identifica come “personale amministrativo” ed accede al sistema di amministrazione*  *(b). si identifica come “cittadino” ed accede al sistema di prenotazione* *Postcondizioni: l’utente generico diventa specifico.* |

## Casi d’uso relativi al personale di amministrazione

Dopo opportuna autenticazione tramite la pagina dedicata, il personale d’A. viene introdotto nella home dell’amministrazione che permette di inserire il numero di “slot appuntamenti” per le varie fasce orarie nelle varie sedi.

**Inserire gli “slot disponibilità”**

Il personale d’A. deve poter inserire il numero di disponibilità per gli appuntamenti. Per fare questo, è necessario scegliere la sede,il giorno, la relativa fascia oraria e il numero di posti disponibili.

|  |
| --- |
| *Attori: Personale Amministrazione**Precondizioni: Il personale deve essersi autenticato.**Passi:**1. Il personale accede al sistema**2. Il personale è introdotto all’interfaccia di base* *3. Il personale sceglie il giorno da “modificare”* *4. Il personale inserisce gli “slot appuntamenti” per la sede scelta tra le disponibili.**5. Il personale conferma il numero di slot per quella fascia oraria giornaliera in quella sede* *Postcondizioni: la disponibilità è inserita.* |

## Casi d’uso relativi al cittadino

**Note generali:**

Il cittadino per poter accedere al sistema deve essere registrato.

Se non è registrato deve effettuare necessariamente la registrazione ( per verificare l’identità del cittadino tramite CODICE FISCALE).

Dopo opportuna autenticazione tramite la pagina dedicata, il cittadino viene introdotto nella home nella quale può scegliere tra i pulsanti “richiesta passaporto” o “ritiro passaporto” per cui sucessivamente potrà scegliere uno “slot disponibilità” tra quelli resi disponibili dall’amministrazione. Presenti anche due finestra nella quale il cittadino può vedere i suoi prossimi appuntamenti e i documenti necessari per quest’ultimi. Ha la possibilità una volta finite le azioni, di eseguire il logout.

**Richiesta passaporto:**

Nell’eventualità in cui il cittadino prema il pulsate “richiesta passaporto”, il cittadino verrà introdotto in una schermata in cui deve selezionare il motivo della richiesta tra:

- Rilascio passaporto per la prima volta

- Furto

- Rilascio passaporto per scadenza del precedente

- Smarrimento

- Deterioramento

Successivamente verrà introdotto nella pagina delle disponibiltà in cui gli sarà presentato un calendario del mese corrente con il numero degli slot disponibili delle varie fasce orarie nei giorni resi disponibili dal personale amministrativo nella sede scelta; il quale dovrà scegliere per prendere appuntamento.

**Ritiro passaporto:**

Nell’eventualità in cui il cittadino prema il pulsate “richiesta passaporto”, il cittadino verrà introdotto in una schermata similare alla precedente per la richiesta in cui gli sarà reso disponibile un calendario del mese corrente con il numero degli slot disponibili delle varie fasce orarie nei giorni resi disponibili dal personale amministrativo; il quale dovrà scegliere per prendere appuntamento.

|  |
| --- |
| *Attori: Cittadino**Precondizioni: Il cittadino deve registrarsi**Passi:**1. Il cittadino accede al sistema dedicato**2. Il cittadino è introdotto all’interfaccia di base* *3. Il cittadino*  *(a).se è la prima volta, si registra*  *(b).se è gia registrato, effettua l’autenticazione* *4. Il cittadino sceglie la sede dove vuole prendere appuntamento**5. Il cittadino sceglie il giorno tra quelli disponibili per l’appuntamento**6. Il cittadino sceglie la fascia oraria del giorno scelto**7. Il cittadino conferma l’appuntamento**Postcondizioni: l’appuntamento è stato inserito.* |
|  |

## 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 semplici ragioni di chiarezza e leggerezza; si considerano quindi le singole attività d’interazione*

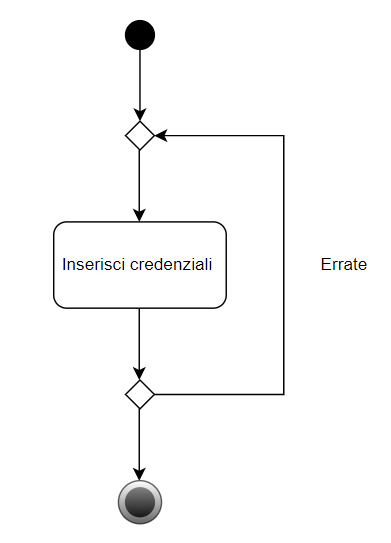


Figura:Autenticazione

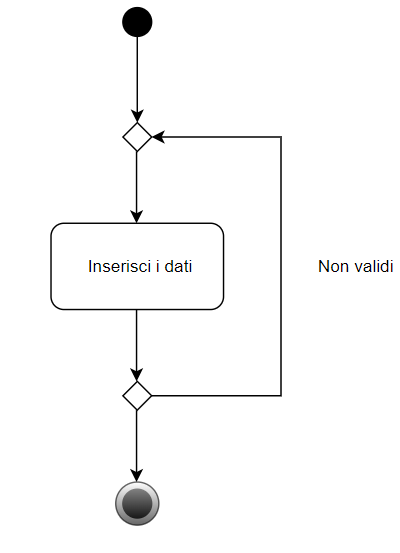


Figura:Registrazione

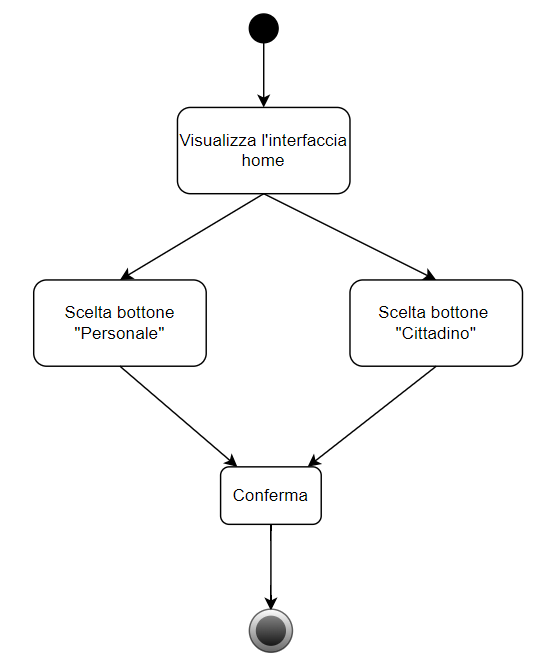


Figura:Attività Utente generico

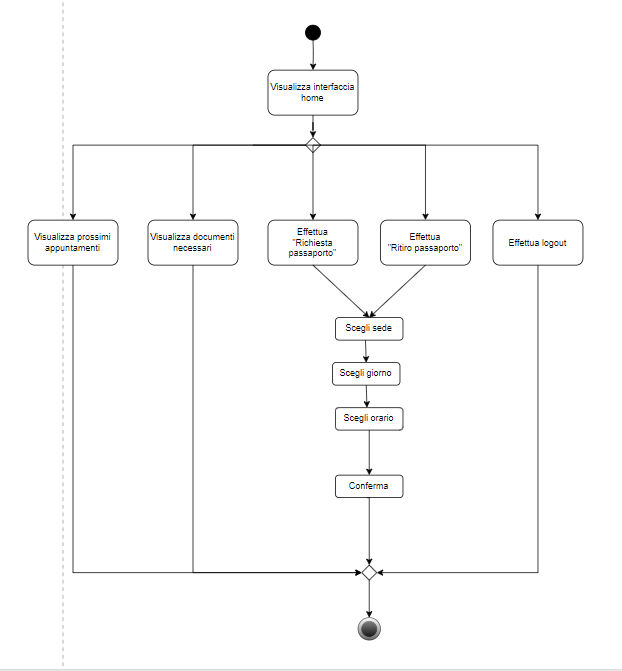
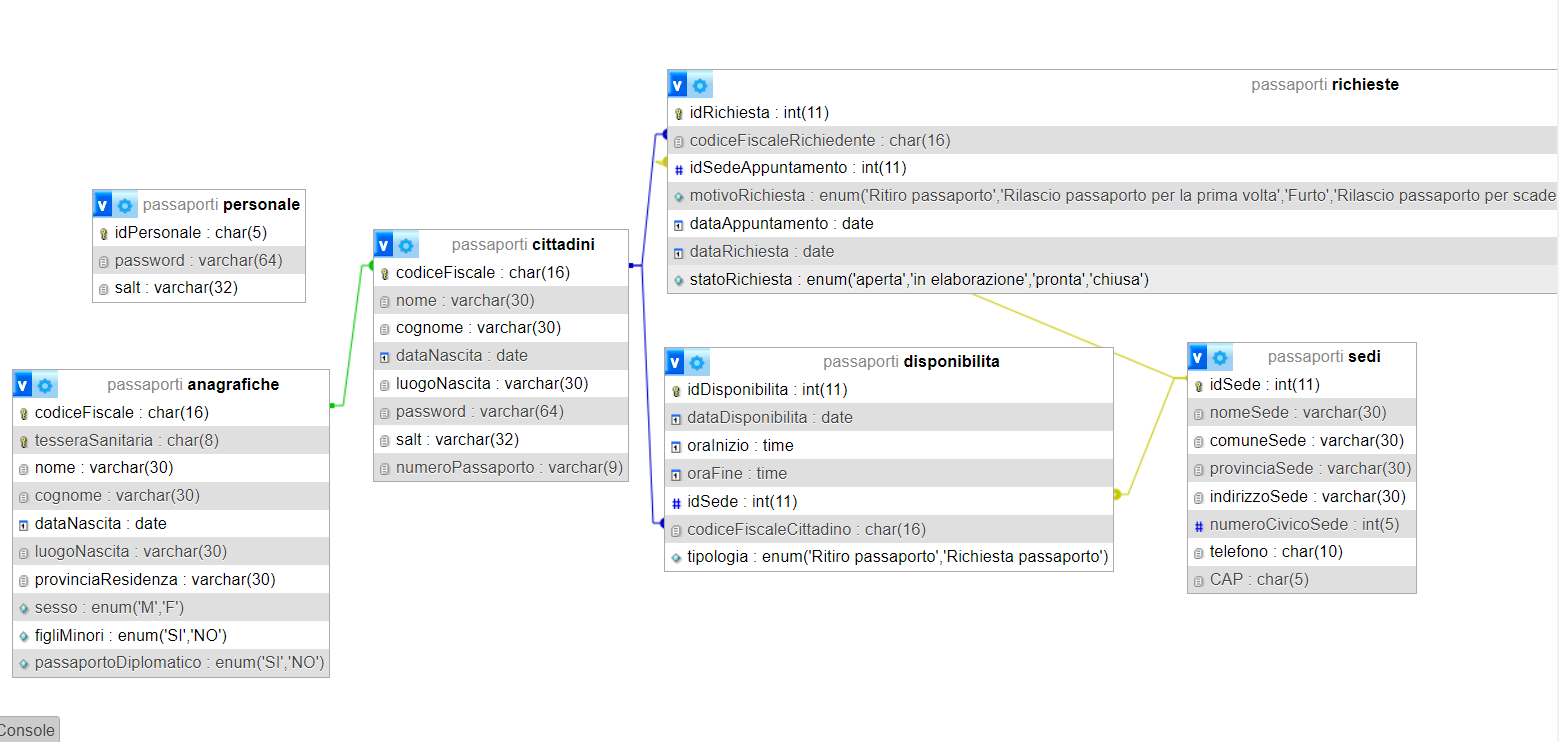


Figura: Attività Cittadino

**MANCA FIGURA AMMINISTRAZIONE**

**Descrizione database**



Il database è formato dalle tabelle:

-Personale: [idPersonale, password, salt ]

Rappresenta l’elenco del personale di amministrazione. IdPersonale è univoco per ogni impiegato. Il campo password è criptato.

-Anagrafiche:[codiceFiscale, tesseraSanitaria, nome, cognome, dataNascita, provinciaResidenza, sesso, figliMinori, passaportoDiplomatico] codiceFiscale è univoco e chiave primaria per ogni tupla che rappresenta un “cittadino” non ancora iscritto al sistema. tesseraSanitaria è univoco per ogni tupla che rappresenta un “cittadino”.

-Cittadini:[codiceFiscale, nome, cognome, dataNascita, luogoNascita, password, salt, numeroPassaporto]

Rappresenta l’elenco dei cittadini iscritti al sistema. codiceFiscale è univoco e chiave primaria per ogni cittadino. Il campo password è criptato.

-Richieste: [idRichiesta, codiceFiscaleRichiedente, idSedeAppuntamento, motivoRichiesta, dataAppuntamento, dataRichiesta, statoRichiesta]

Rappresenta l’elenco delle richieste fatte dai cittadini richiedenti il passaporto. IdRichiesta è univoco e chiave primaria per ogni tupla/richiesta.

-Disponibiltà: [idDisponibilità, dataDisponibilità, orainzio, orafine, idSede, codiceFiscaleCittadino, tipologia]

Rappresenta l’elenco delle disponibilità inserite dal personale in base alla tipologia scelta dal cittadino(Ritiro passaporto','Rilascio passaporto per la prima volta','Furto','Rilascio passaporto per scadenza del precedente','Smarrimento','Deterioramento'). IdDisponbilità è univoco e chiave primaria per ogni tupla/disponiblità.

-Sedi:[idSede, nomeSede, comuneSede, provinciaSede, indirizzoSede, nomeroCivicoSede, telefono, CAP]

Rappresenta l’elenco delle sedi diponibili per poter richiere e/o ritirare il passaporto.

idSede è univoco e chiave primaria per ogni sede.

# Progettazione e pattern architetturali usati

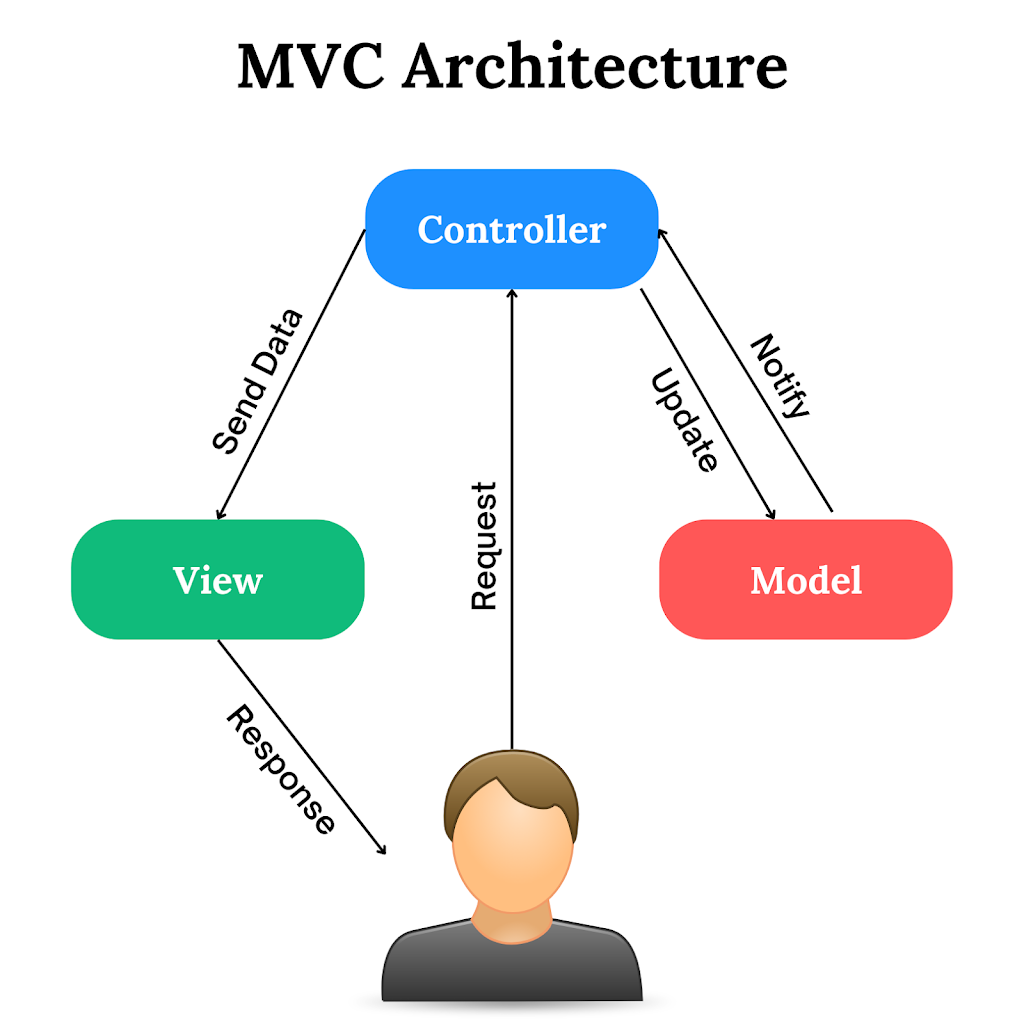
Il sistema è stato progettato utilizzando le tecniche di modellazione ad oggetti. Dal punto di vista architetturale, ci siamo ispirati al **pattern MVC**. Durante la fase di sviluppo del codice abbiamo optato poi ad una variante di quest’ultimo, aggregando model e controllore in un’unica classe. La struttura dei dati è definita e gestita dalla tecnologia MySQL e la manipolazione dei dati viene gestita all’interno del controllore tramite le query. Ciò ha permesso di separare nettamente e comodamente la componente grafica dalla componente strutturale di gestione dei dati grazie anche all’implementazione nativa di Java FX e il suo relativo formato FXML.

• **Modello-Controllore**: parte del sistema che riguarda i dati e le informazioni memorizzate. I dati vengono mossi all’interno del sistema grazie alle query sql. Questa parte del sistema definisce anche la logica applicativa, ovvero il comportamento del sistema a fronte degli stimoli esterni governati dai listener implementati nativamente da Java FX.

• **Vista**: parte del sistema che rappresenta visivamente il modello-controllore, e quindi, i dati del sistema. Si è utilizzata la tecnologia di Java FX, basata sulla piattaforma Java.

Le azioni dell’utente saranno catturate dai listener, implementati nativamente da Java FX e controllati da relative funzioni ad hoc. Ogni listener, e quindi funzione, è istruito a reagire in determinati modi in base alle esigenze e alle richieste viste a inizio progetto, andando a modificare le informazioni contenute nella base di dati e quindi ad aggiornare la vista, che rappresenta il modello.

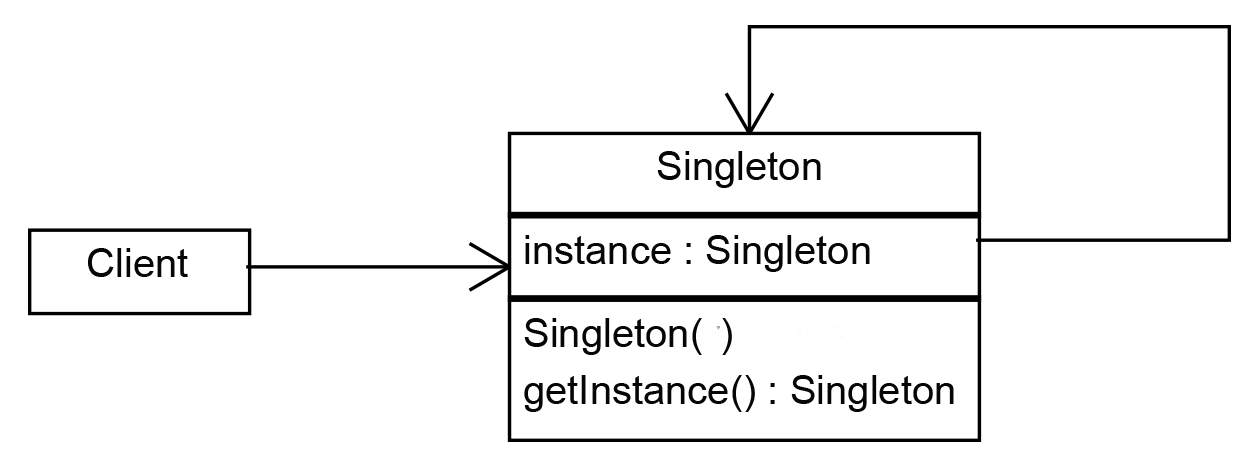
Seguono un semplice schema dell’architettura del sistema ed i diagrammi UML delle classi del Modello e della Vista-Controllore. Di seguito una rappresentazione grafica del pattern MVC, nel nostro caso Controller e Model sono uniti in una stessa classe che si interfaccia alla struttura dei dati tramite le query Sql.



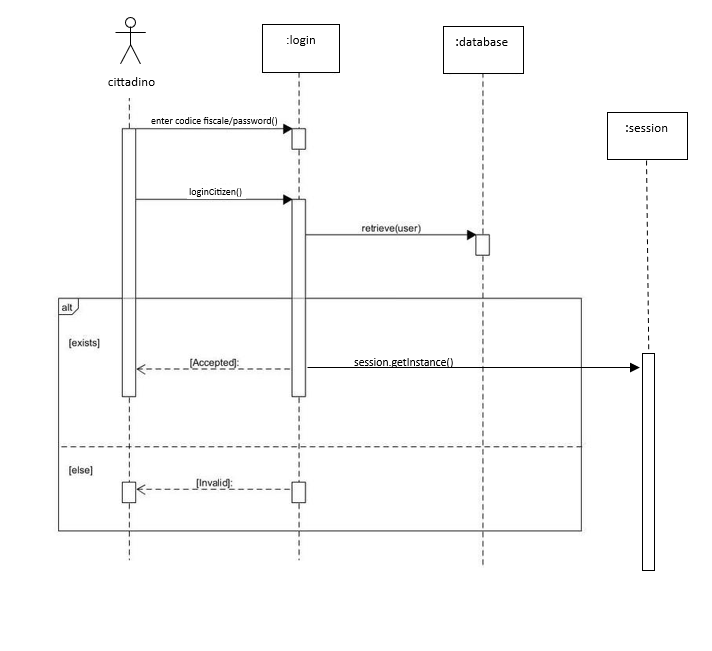
# Implementazione e design pattern usati

Per l’implementazione dell’applicazione ci siamo serviti del Design Pattern Singleton. Questo pattern è stato utilizzato all’interno della classe “UserSession”, per gestire le sessioni dell’utente. La classe mantiene all’interno il riferimento all’unica istanza Singleton della classe. Questa classe contiene poi tutti i metodi, le proprietà e gli attributi tipici dell’astrazione per cui è stata concepita. Le classi Singleton implementano un metodo statico per assicurare che nessuna altra istanza venga creata oltre la prima, restituendo contemporaneamente un riferimento all’unica esistente. Di seguito una rappresentazione del funzionamento della classe Singleton, si è deciso per comodità di rappresentare come “Client” tutte le classi con le relative funzioni che generano un’istanza Singleton.

# Diagrammi di sequenza



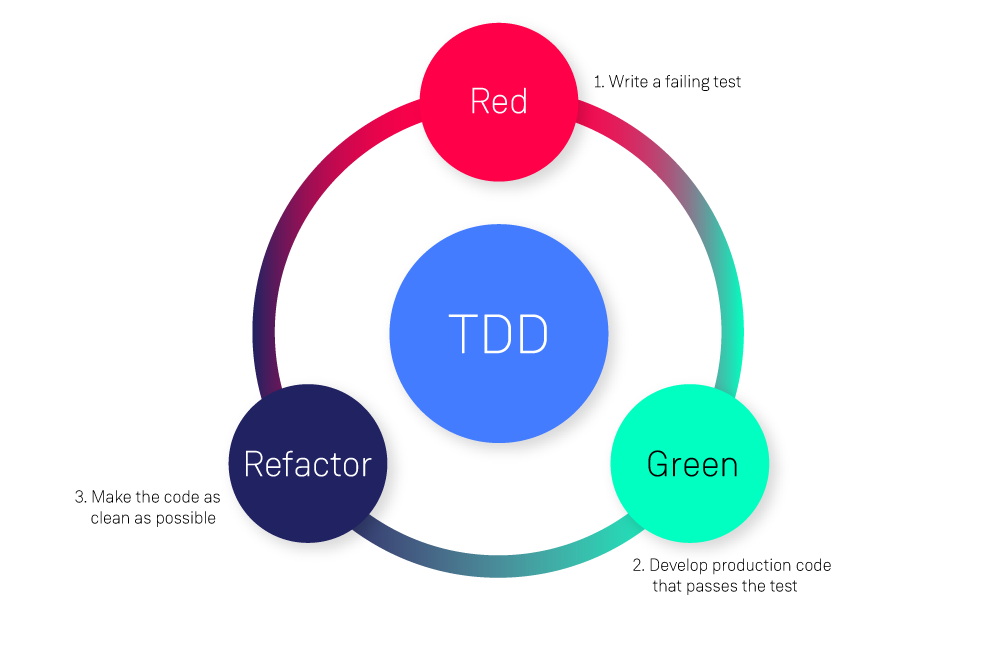
Seguono diagrammi di sequenza che mostrano le dinamiche di alcune interazioni tra classi particolarmente complesse od interessanti. Sono stati rappresentati soltanto gli scambi salienti.



Cciicccc

**Attività di test e validazione**

Nella costruzione dell’intero progetto lato software abbiamo concordato di procedere adottando il metodo TDD (Test-Driven Development). Questo metodo ci permette di sviluppare il codice e al contempo testarlo. Solo quando tutti i test previsti vengono soddisfatti dal codice sviluppato possiamo passare alla seguente funzionalità da sviluppare, precedendo al cosiddetto sviluppo incrementale.



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

1. Ricognizione del documento delle specifiche e confronto con i diagrammi prodotti

2. Verifica della consistenza tra diagrammi e codice prodotto

3. Ispezione del codice e ricerca di pratiche errate

4. Test degli sviluppatori sul software

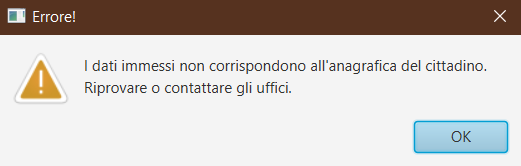
5. Test utente generico sul software

Nella prima fase si è semplicemente rivisto il documento delle specifiche e lo si è confrontato con gli use cases, activity diagrams e class diagrams precedentemente sviluppati. 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 infrazioni, malepratiche e cattivi usi generali di metodi e funzioni.

## Test degli sviluppatori

In questa fase lo sviluppatore ha immesso nel sistema degli input (sia corretti sia errati) per vedere se la reazione del software fosse quella attesa. Di seguito alcuni casi svolti:

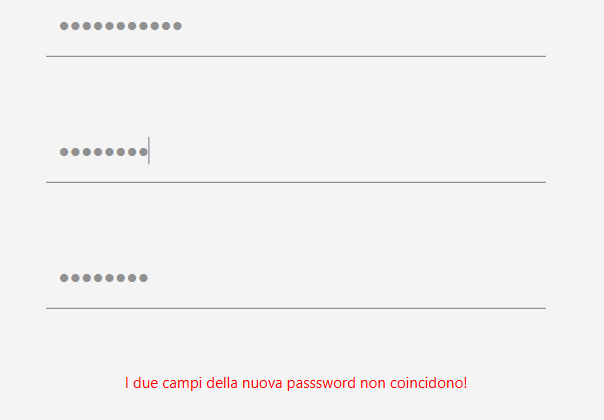
Registrazione nuovo utente: in questa sezione del programma sono state effettuate diverse prove di inserimento dei dati. Nella fase di progettazione si è deciso che fosse buona regola far inserire al nuovo utente tutti i dati personali utili. Solo qualora tutti i dati inseriti fossero uguali, dapprima il codice fiscale, ai dati in possesso nella base di dati generale “anagrafiche”, si poteva procedere alla registrazione dell’utente cittadino nel sistema. Prima della vera e propria creazione dell’utente, però, è richiesto il doppio inserimento della password personale, per evitare errori nella prima digitazione della password stessa, dal momento che, per motivi di sicurezza e privacy, tutti i campi “password” del sistema sono stati resi non leggibili. La registrazione viene, quindi, resa possibile solo quando tutti i dati corrispondono e le password rispettano i vincoli. Nel caso di un codice fiscale errato viene mostrato al cliente un errore specifico che invita, nel caso di un corretto inserimento da parte dell’utente, a contattare i tecnici del sistema per verificare la presenza del cittadino nella base di dati “anagrafiche”. Nel caso un utente già registrato tenti di registrarsi nuovamente verrà visualizzato un errore da parte dell’utente. Di seguito un esempio di risposta a dati inseriti non correttamente o di una mancanza del cittadino all’interno della tabella “anagrafiche” in fase di registrazione.



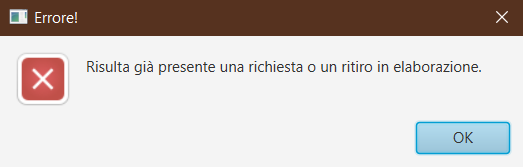
Autenticazione: nel nostro progetto sono richieste due sezioni per l’autenticazione, una per il personale amministrativo, una per l’utente cittadino. Ovviamente queste due sezioni sono ben distinte sia dal punto di vista software sia dal punto di vista grafico. All’utente cittadino viene richiesto l’inserimento del proprio codice fiscale e della password scelta in fase di registrazione (o cambiata successivamente dalla propria home). Il campo “password” viene attivato solo quando viene inserito un nome utente pari a 16 caratteri, come il codice fiscale. L’accesso viene permesso solo quando codice fiscale, che sia minuscolo o maiuscolo o misto, e password sono corretti e corrispondono ai dati in possesso nella tabella “cittadini” della base di dati. Lo stesso vale per l’autenticazione del personale amministrativo, con l’unica differenza che il nome utente è composto da 5 cifre. Di seguito le due sezioni di accesso, personale amministrativo e cittadino.



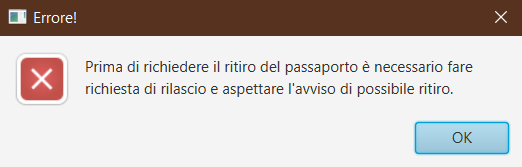
Richiesta nuova password: questa sezione presente nella home del cittadino permette il cambio password solo qualora venga inserita la password attuale corretta e la nuova password (inserita due volte) rispetti i vincoli di sicurezza. Di seguito un esempio di cambiamento di password, la nuova password, inserita due volte non corrisponde alla sua omologa del campo precedente.



Sezione richiesta passaporto: questa sezione è accessibile solo quando non ci sia una pratica già avviata dal cittadino, che sia di richiesta o di ritiro passaporto. Di seguito un esempio di accesso alla sezione con una pratica già avviata.



Sezione ritiro passaporto: questa sezione è accessibile solo quando è possibile ritirare il passaporto, una volta chiusa la precedente richiesta erogazione del passaporto. Di seguito un esempio di richiesta di ritiro del passaporto senza, però, aver avviato prima una pratica di richiesta del passaporto.



Entrambe le procedure qui sopra descritte sono estremamente guidate e non permettono alcun errore di inserimento.

## Test utente generico

Come ultimo scaglione il software è stato sottoposto ad un test da parte di alcuni individui con limitata dimestichezza informatica ed 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 il soggetto navigasse liberamente il sistema. Non è stata data nessuna spiegazione sull’utilizzo del software, se non una generale indicazione dei suoi fini; ci si è limitati a rispondere alle domande, quando sollevate. L’unico scopo del test era quello di rilevare errori invisibili allo sviluppatore; in realtà, gli utenti a cui è stato mostrato il software hanno anche aiutato ad individuare nuove funzionalità per migliorare l’usabilità generale del sistema.

N. B. tutti i test effettuati in fase di sviluppo del software che hanno dato risultato positivo sia dal software stesso sia dalla validazione degli sviluppatori sono salvati su un database MySql esistente che può essere consultato su richiesta al team di sviluppo.

**Tecnologie utilizzate**

Per lo sviluppo di un progetto come questo è stato necessario ricorrere a più tecnologie, scelte accuratamente per ottimizzare lo sviluppo e i vari flussi di lavoro.

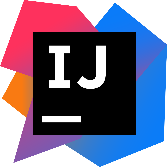
## Sincronizzazione dei flussi di sviluppo

Si è scelto di utilizzare GitHub e lo strumento di controllo versione distribuito Git per ottimizzare lo sviluppo software di diversi utenti su diversi PC allo stesso tempo.



## Sviluppo software

Come programma per lo sviluppo software si è deciso di optare per IntelliJ Idea, grazie alla sua eccellente ottimizzazione con il protocollo di GitHub e agli svariati Plugins disponibili, come il collegamento diretto alla base di dati direttamente dalla console. Come linguaggio di sviluppo e tecnologia abbiamo adottato Java e per lo sviluppo della parte grafiche ci siamo affidati a Java FX.



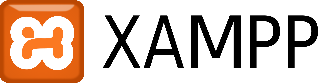
## Base di dati

Come base di dati si è deciso di sfruttare il sistema open source di gestione di database relazionali MySql. È stato scelto per gli svariati sistemi e [linguaggi di programmazione](https://it.wikipedia.org/wiki/Linguaggi_di_programmazione) che lo supportano, come [ODBC](https://it.wikipedia.org/wiki/ODBC), [Java](https://it.wikipedia.org/wiki/Java_(linguaggio_di_programmazione)), [Mono](https://it.wikipedia.org/wiki/Mono_(progetto)), [.NET](https://it.wikipedia.org/wiki/Microsoft_.NET), [PHP](https://it.wikipedia.org/wiki/PHP), [Python](https://it.wikipedia.org/wiki/Python) ecc.



## Server web

Come emulazione del server web si è deciso di lavorare con XAMPP, per creare un server web multipiattaforma tramite tecnologie open source. Inoltre XAMPP è composto da Apache HTTP Server e MySQL ed è anche compatibile con i sistemi operativi Windows, Linux, Mac e Solaris. L’ idea per il prodotto finale è di servirsi di un servizio server in cloud con opzione PAY-AS-YOU-GO e load balancing per gestire autonomamente il carico di utenti tra diversi server.

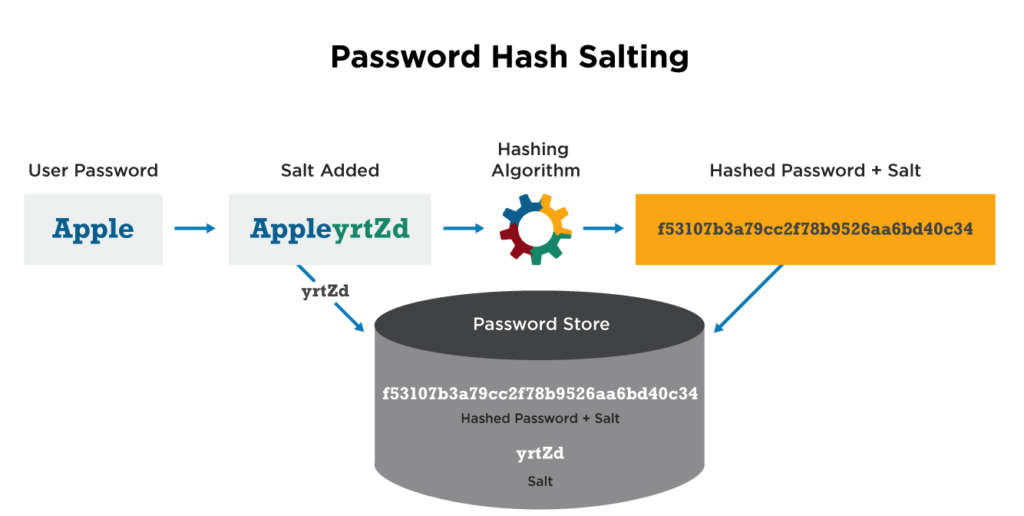


**Scelte implementative**

Implementazione degli stati per gestione pratiche: lo stato può essere: “aperta”, “in elaborazione”, “pronta”, “chiusa” e si riferisce alla singola pratica, che sia di richiesta o di ritiro passaporto. Pensando che nell'anno ci sono momenti dove la "lista d'attesa" è molto lunga e momenti dove c'è poca gente, abbiamo gestito il vincolo della successione temporale tra richiesta passaporto e ritiro passaporto tramite gli stati e non affidandoci al lasso di tempo passato. Il personale della questura gestirà lo stato di ciascuna richiesta. Quando una richiesta viene creata è posta nello stato di “aperta” fino a quando non verrà prenotata una data da parte del cittadino; in quest’ultimo caso lo stato passa a “in elaborazione”. Quando il passaporto fisico verrà creato dal Ministero lo stato passerà a “pronta” e verrà inviato il passaporto fisico alla questura generale della provincia di residenza. Una volta arrivato il passaporto fisico alla questura della provincia di residenza, verrà posta su “chiusa” la pratica e verrà creata automaticamente una pratica di ritiro passaporto posta su “aperta”. Da quel momento il cittadino potrà prenotare l’appuntamento per il ritiro del passaporto. Una volta ritirato, verrà chiusa la pratica.

Campi d’inserimento “password” e privacy: per motivi di sicurezza e privacy, tutti i campi dove si inseriscono le password sono resi illeggibili, così un mal intenzionato non potrà “sbirciare” la password inserita da un utente distratto.

Campo “salt” per criptare le password: sempre per motivi di sicurezza si è deciso di implementare la funzione di encrypt con salt per rendere criptate le password salvate nella base di dati.



Il salting della password comporta l'aggiunta di una stringa di 32 o più caratteri a una password. Il password salting è uno dei modi più sicuri per proteggere le password memorizzate per l'autenticazione futura senza esporle in caso di violazione del tuo sito Web in futuro.

Gestione sessioni utente: per l’implementazione del sistema lato utente cittadino si è deciso di sviluppare una gestione della sessione di accesso all’applicazione. Questo per tenere traccia di tutte le azioni svolte da un utente cittadino e per rendere più “user friendly” l’interfaccia grafica, con un messaggio di benvenuto personalizzato. Questo sistema si presta ottimamente per il possibile sviluppo di una tabella “log” nella base di dati, estremamente utile per gli sviluppatori, sistemisti o addetti amministrativi in caso di problemi del sistema.

Creazioni slot disponibilità delle prenotazioni: il personale amministrativo è incaricato di assegnare gli slot giornalieri di una data sede in base al personale presente distinguendo la tipologia dello slot in riferimento alla pratica: richiesta di passaporto o ritiro dello stesso. Il personale indicherà dapprima quale sede gestire, quale giorno modificare, quale fascia oraria, quanti slot e per quale tipologia di richiesta.

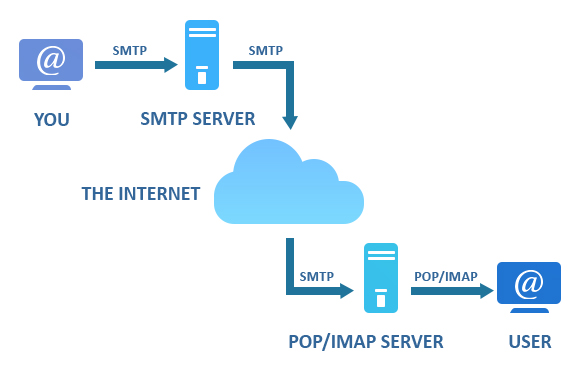
Registrazione nuovo utente: quando un utente si registra, dopo aver inserito tutti i dati corretti, dovrà scegliere una password. L’inserimento della password è doppio, per evitare errori di distrazione da parte dell’utente.

Registrazione personale amministrativo: l’utente amministrativo e la relativa password vengono create esternamente per evitare errori e possibili azioni non permesse.

**Future implementazioni**

Sistema di “log”: con l’introduzione in fase di sviluppo delle sessioni utente è possibile lo sviluppo di una tabella “log” nella base di dati, che tenga in memoria per 30, 60 o 90 giorni le azioni svolte dall’utente e l’ora di tali azioni. Questo per motivi di sicurezza e per facilitare la correzione di eventuali bug e problematiche varie. Questa tabella, insieme alla tabella “personale”, verrebbe gestita esternamente al sistema.

Sistema di avviso stato pratica tramite mail: inserendo il campo “mail” durante la registrazione del cittadino, implementando una classe “mail” ad hoc e sfruttando un server di invio mail SMTP esterno (come può essere il servizio offerto gratuitamente da Google) è possibile inviare mail al cittadino riguardo il cambiamento della propria pratica avviata. Ad ogni aggiornamento di stato si invierebbe una mail con l’informazione voluta o, per esempio, per gestire il reminder degli appuntamenti. Un giorno prima dell’appuntamento, infatti, l’utente verrebbe informato adeguatamente.



# Opere citate

“Password salting” - [Craig Taylor](https://cyberhoot.com/author/craig/) - cyberhoot.com

Github - github.com

IntelliJ Idea - jetbrains.com

Java & Java FX - java.com

MySql - mysql.com

XAMPP - apachefriends.org