

Sistema di Gestione Passaporti

Andrea Doci, Michele Todeschini.

VR445930, VR443680

Programmazione 2 e Ingegneria del Software.

12/07/2023

Requisiti ed interazioni utente-sistema

Specifiche casi d'uso

Note generali.

Il sistema si propone come un portale per la gestione di creazione, da parte dei dipendenti, e selezione, da parte dei cittadini, di appuntamenti per il ritiro o il rilascio dei passaporti.

E' stato scelto di usare dei file XML come appoggio per il salvataggio dei dati e strutture create.

Nella Home si potrà scegliere se entrare come dipendente o se registrarsi o accedere come cittadino, in entrambi i casi verranno richiesti dati ed eventuali mail e passw; in caso di errore nell'inserimento di dati verrà aperto un popup che indicherà il problema. Al contrario nel caso in cui vada tutto a buon fine ognuno verrà reindirizzato alla rispettiva pagina.

Casi d'uso relativi ai dipendenti.

Nel caso dei dipendenti, esso dovrà poter inserire prenotazioni nel sistema e poter avere visione di esse.

Inserimento prenotazioni:

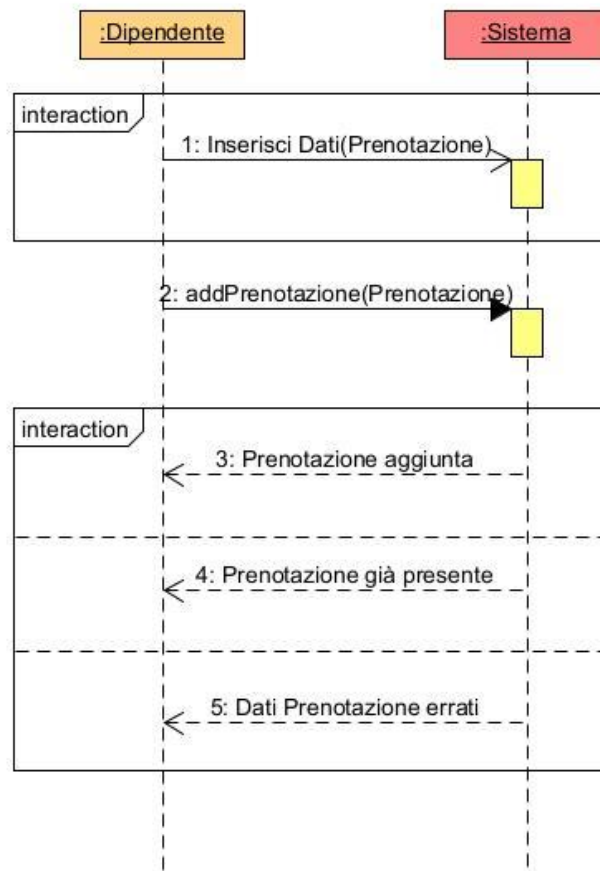
Attore: Dipendente

Precondizioni: Il dipendente deve avere le credenziali offerte dalla questura

Passi:

1. Il dipendente avvia l'applicazione.
2. Il dipendente è introdotto all'interfaccia principale.
3. Il dipendente clicca il pulsante "Area Dipendente".
4. Compare a schermo un popup dove il dipendente dovrà inserire le sue credenziali.
5. Il popup si chiude e la schermata principale cambia interfaccia, mostrando l'interfaccia del dipendente.
6. Nella tab che viene mostrata, il dipendente può inserire una prenotazione nel sistema, compilando i vari campi.
7. Una volta premuto il pulsante, il sistema inserirà la prenotazione nei file xml del sistema.

Sequence diagram:



Visione delle prenotazioni già esistenti:

Attore: Dipendente

Precondizioni: Il dipendente deve avere le credenziali offerte dalla questura

Passi:

1. Il dipendente avvia l'applicazione.
2. Il dipendente è introdotto all'interfaccia principale.
3. Il dipendente clicca il pulsante "Area Dipendente".
4. Compare a schermo un popup dove il dipendente dovrà inserire le sue credenziali.
5. Il popup si chiude e la schermata principale cambia interfaccia, mostrando l'interfaccia del dipendente.
6. Nella seconda tab, sarà presente la lista di tutte le prenotazioni presenti nel sistema, suddivise tra quelle con cittadini già iscritti e quelle senza.

Casi d'uso relativi ai cittadini.

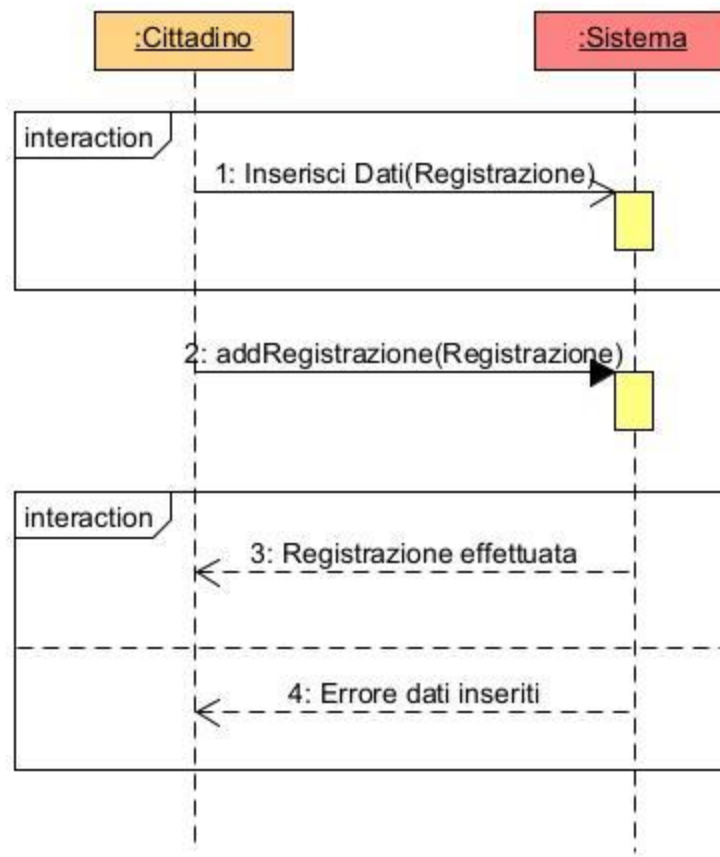
I cittadini dovranno esser in grado di potersi registrare per la prima volta oppure accedere e selezionare il giorno relativo al tipo di prenotazione che si desidera effettuare; e avrà anche la possibilità di prendere visione delle prenotazioni già esistenti.

Effettuare la registrazione:

Attore: Cittadino
Precondizioni: nessuna
Passi:

1. Il cittadino avvia l'applicazione
2. Il cittadino è introdotto all' interfaccia principale
3. Il cittadino, non iscritto, preme il pulsante di registrazione, e viene introdotto alla scheda di registrazione
4. Il cittadino compila i campi richiesti
5. Dopo aver premuto il pulsante di conferma e controllato la presenza dei dati nell'anagrafe di Stato, si aprirà un popup che richiede l'inserimento di una mail e password, le quali saranno usate nella fase di login.
Nel caso di dati sbagliati o omessi, si aprirà un popup di errore.
6. Il cittadino viene reindirizzato all'interfaccia iniziale.

Sequence diagram:



Scelta e conferma della prenotazione nel calendario:

Attore: Cittadino

Precondizioni: Il cittadino deve essere iscritto al sistema (Vedi Case 1)

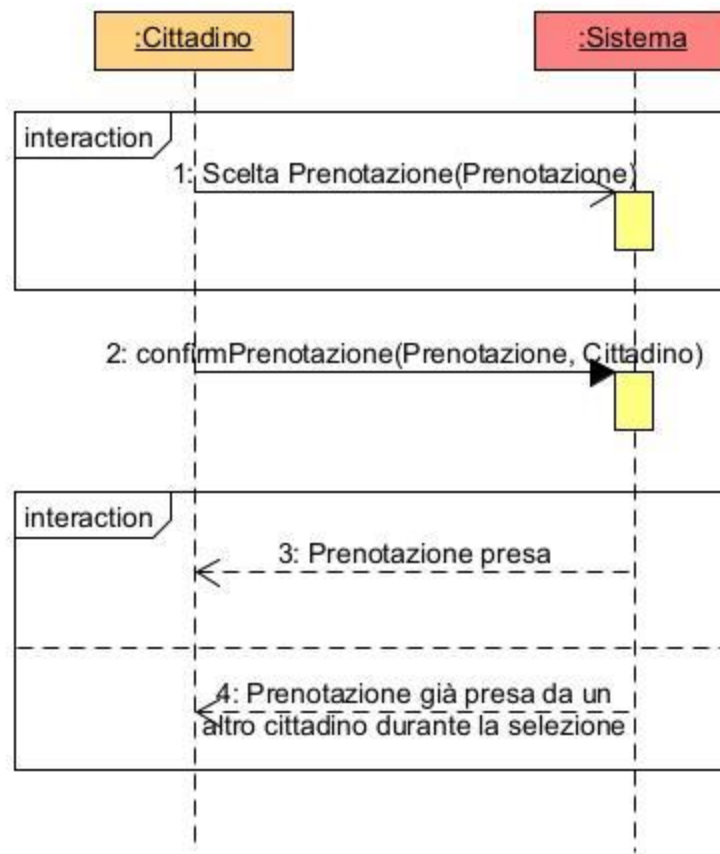
Passi:

1. Il cittadino avvia l'applicazione
2. Il cittadino è introdotto all' interfaccia principale
3. Il cittadino, iscritto, inserisce la mail e la password inserite durante la fase di registrazione e preme il tasto di Login
4. Il cittadino è introdotto all'interfaccia di scelta prenotazioni.
5. Qui il cittadino ha a disposizione la visione di un calendario, che mostra le disponibilità delle prenotazioni, ed anche una serie di filtri sulla sede e sul tipo di prenotazione, con i quali filtrare in base alle proprie esigenze.
6. Una volta selezionato il giorno, si aprirà un popup con una lista di tutte le possibili prenotazioni in base ai filtri inseriti
7. Una volta selezionata la prenotazione che si vuole prendere e premuto il pulsante di conferma, il sistema controlla se nel frattempo la prenotazione è già stata presa da qualcun'altro.

Nel caso in cui non è stata presa, il sistema conferma all'utente che la prenotazione è stata presa correttamente.

Nel caso negativo, il sistema mostra un popup a schermo con un errore.

Sequence diagram:



Visione delle prenotazioni già esistenti:

Attore: Cittadino

Precondizioni: Il cittadino deve essere iscritto al sistema (Vedi Case 1)

Passi:

1. Il cittadino avvia l'applicazione
2. Il cittadino è introdotto all' interfaccia principale
3. Il cittadino, iscritto, inserisce la mail e la password inserite durante la fase di registrazione e preme il tasto di Login
4. Il cittadino è introdotto all'interfaccia di scelta prenotazioni.
5. Cliccando sulla seconda tab, il cittadino potrà vedere le sue prenotazioni.

DIAGRAMMI DI ATTIVITA'

Nota: Di seguito vengono rappresentati i diagrammi delle attività relativi alle singole azioni eseguite dal punto di vista dell'utente.

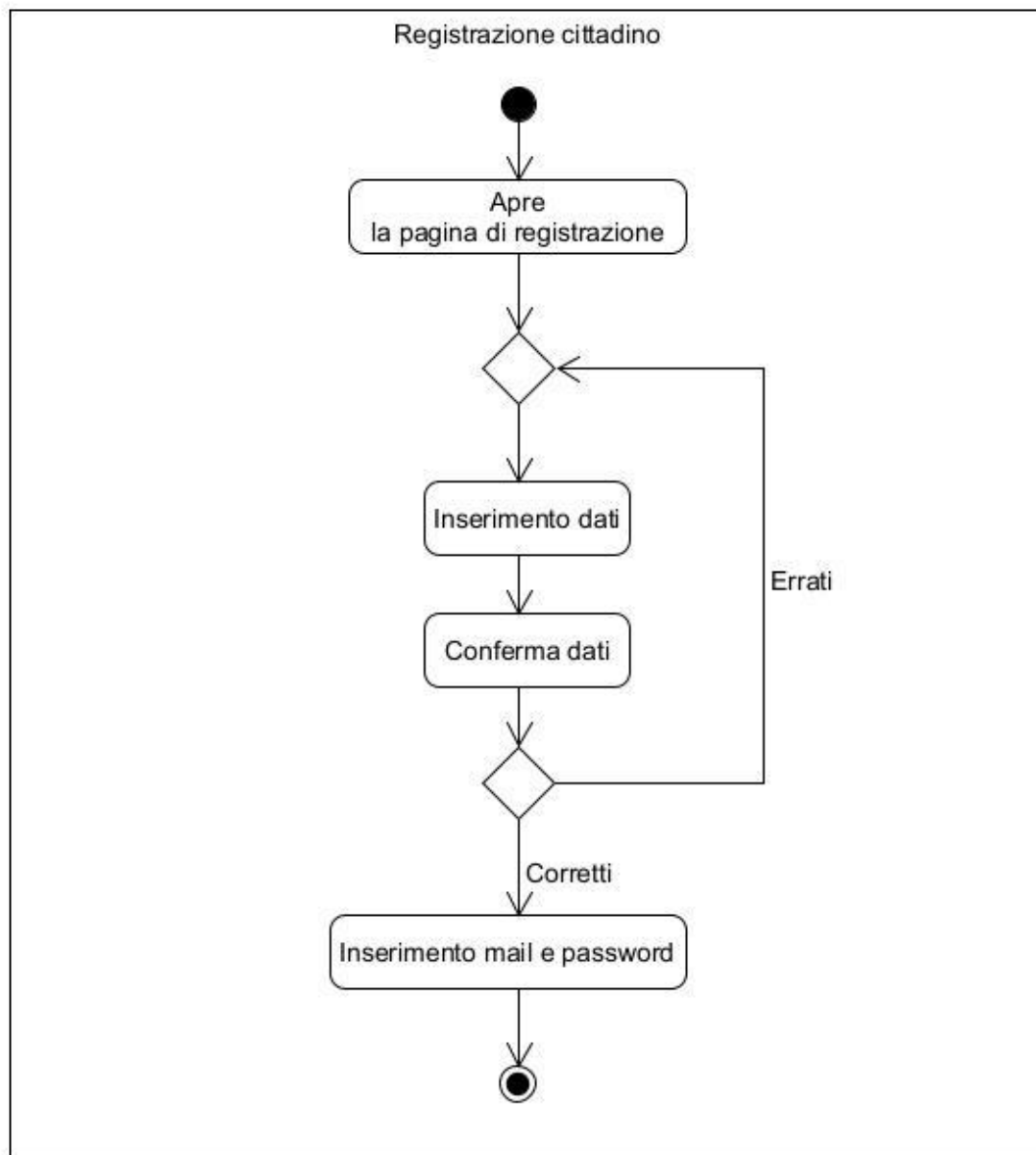


Figura 1: Diagramma Registrazione Cittadino

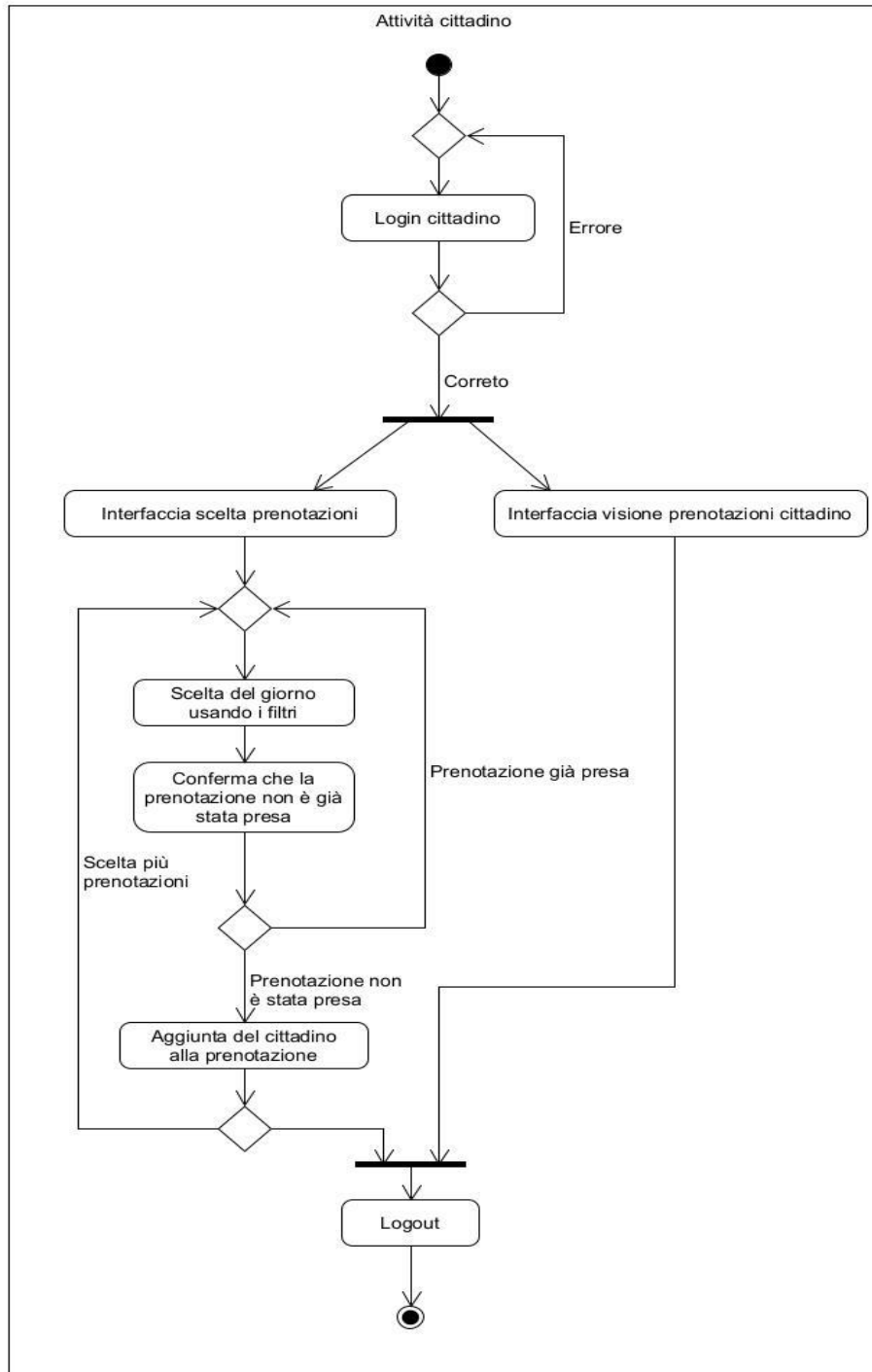


Figura 2: Diagramma attività Cittadino

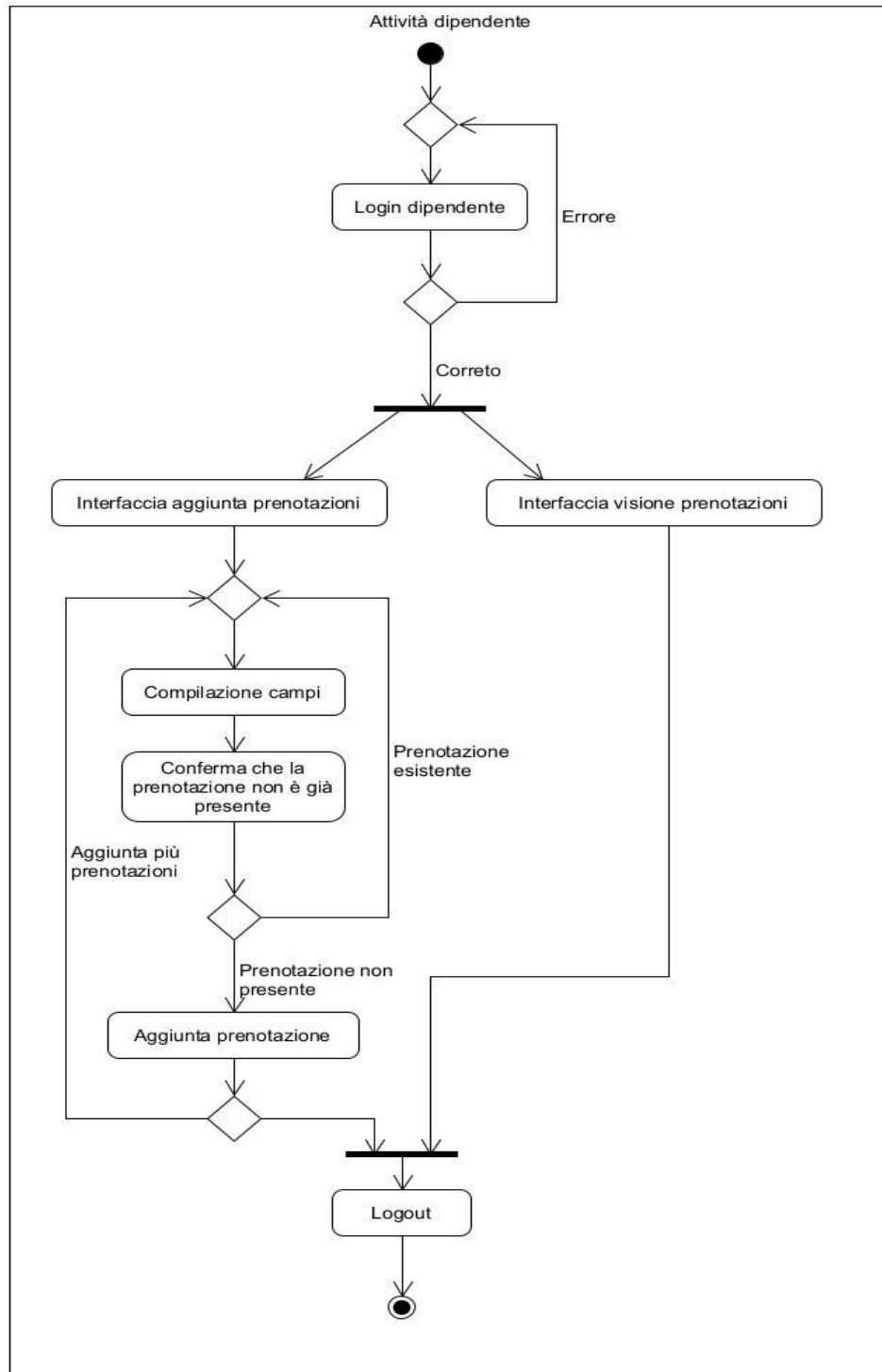
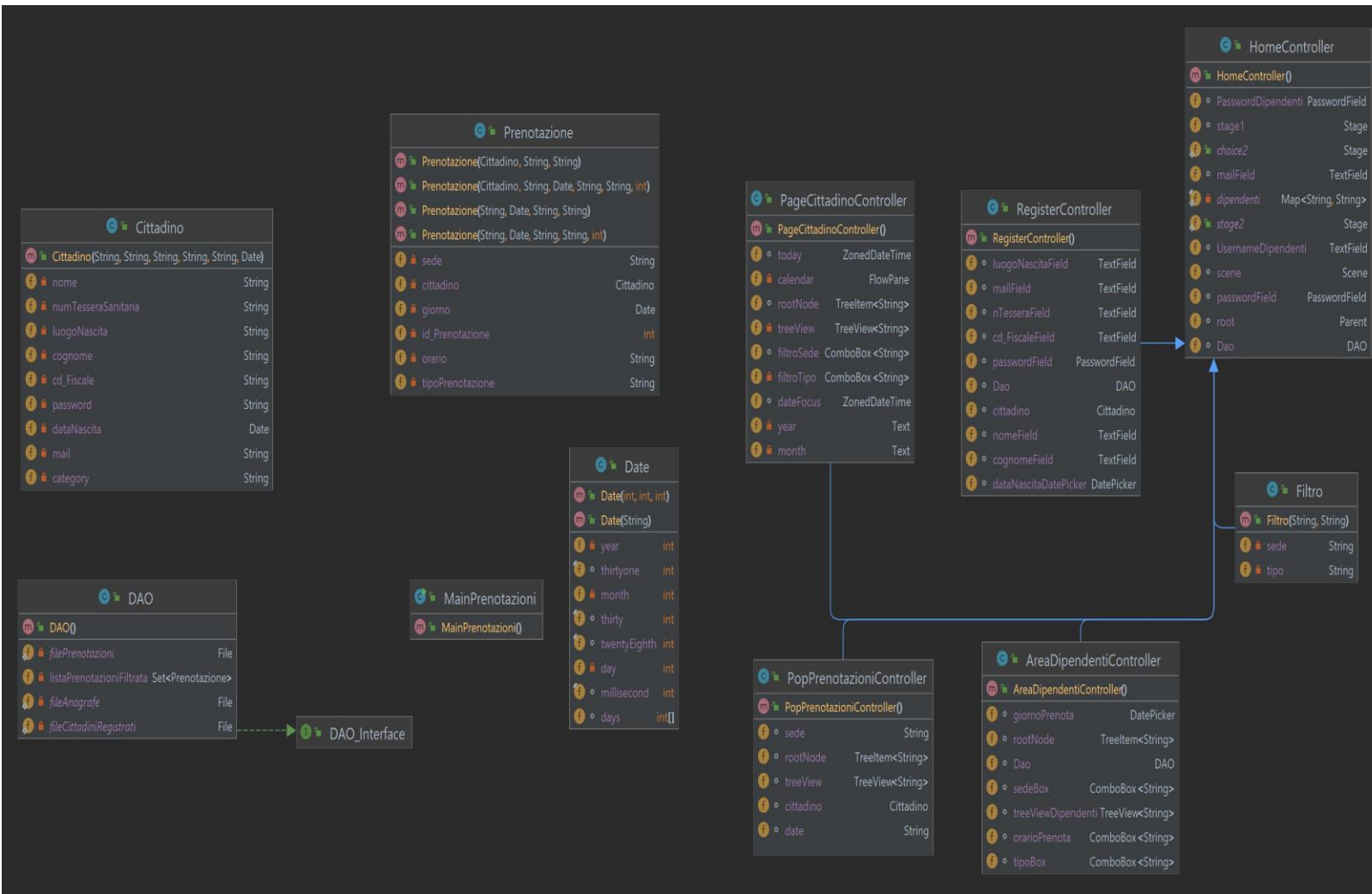


Figura 3: Diagramma attività Dipendente

PROGETTAZIONE

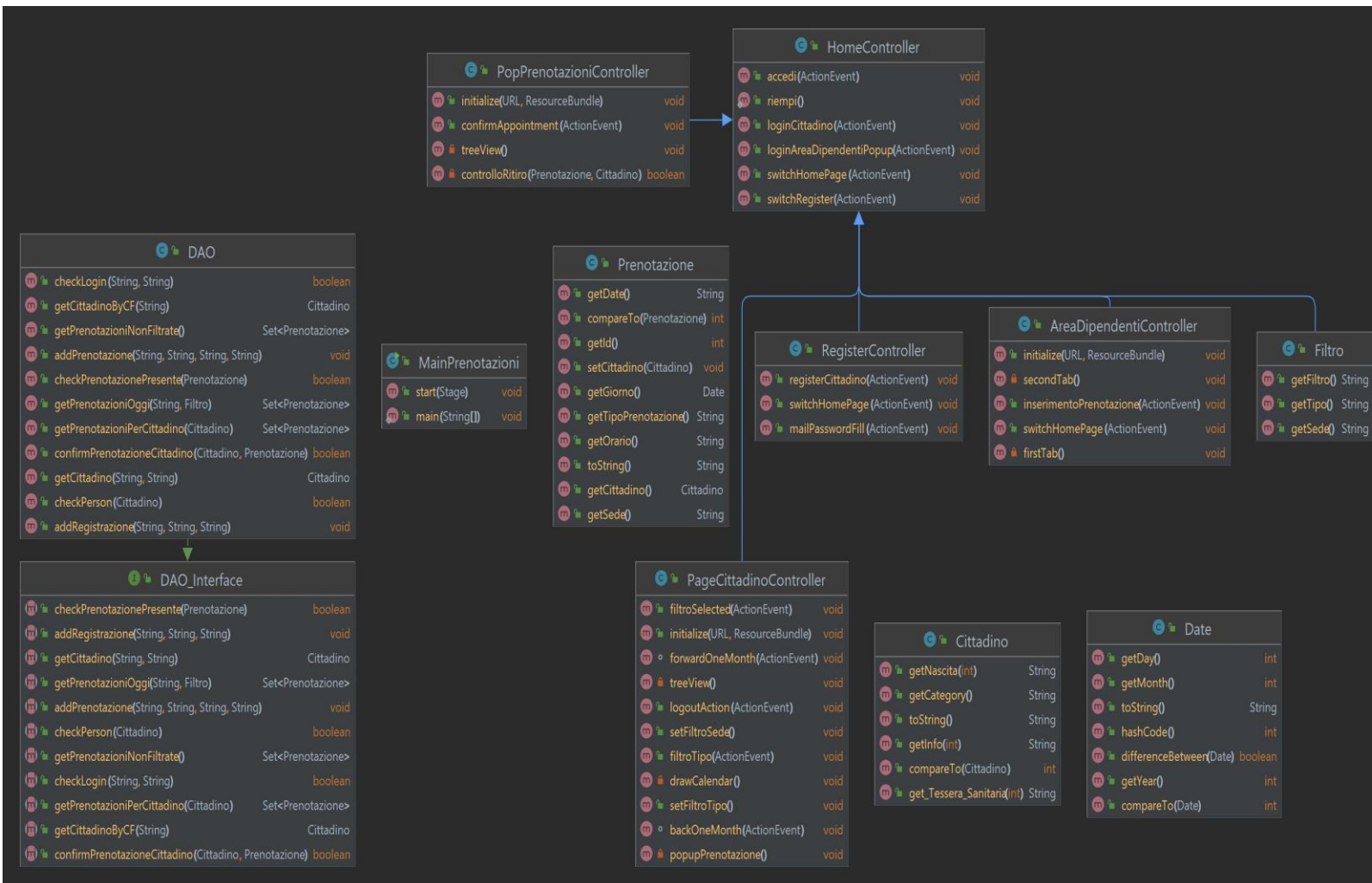
Di seguito illustreremo degli schemi riguardanti il numero di classi create e utilizzate, e la loro struttura mostrando di conseguenza poi variabili, metodi, costruttori e dipendenze ecc...

Diagramma variabili e costruttori:

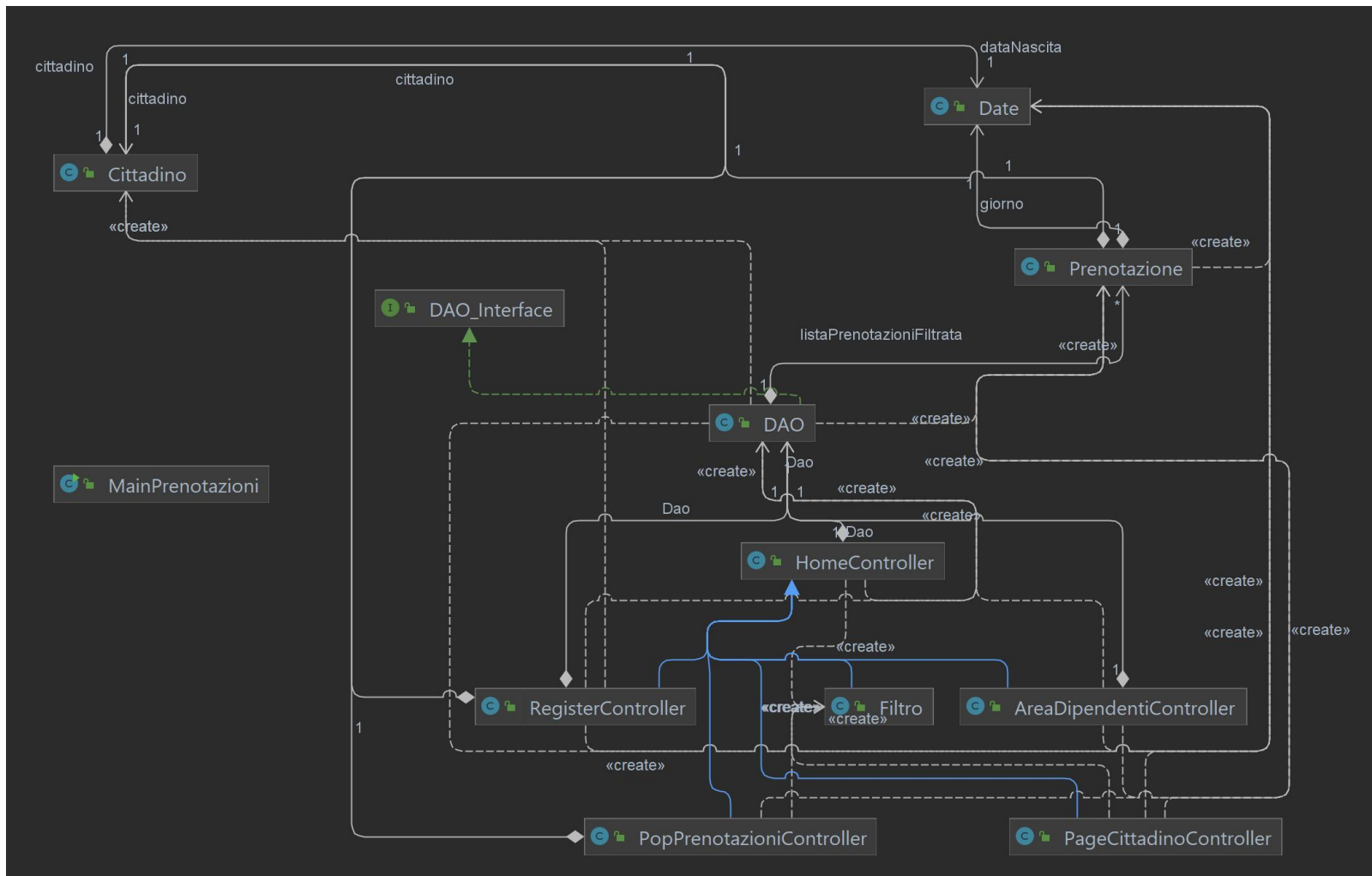


Qui si potrà prendere visione di tutte le classi e dei rispettivi costruttori e variabili utilizzati, assieme ad le eventuali dipendenze.

Diagramma relativo ai metodi dentro le classi:



Qui sopra vengono illustrati metodi creati e i loro parametri.



Qui sono illustrate le dipendenze:

- I percorsi che partono da una classe e proseguono dall'oggetto di forma romboidale e finiscono fino ad un altro oggetto indicano le molteplicità dell'oggetto a cui punta il romboide dentro la classe da cui parte il percorso;
- I percorsi tratteggiati bianchi indicano che la classe da cui parte tale percorso crea e contiene oggetto/i relativo alla classe da esso puntato;
- I percorsi verdi tratteggiati indicano l'implementazione di una interfaccia;
- I percorsi Blu indicano l'ereditarietà tra classi;

NOTA SULL'INTERFACCIA "DAO_INTERFACE"

Metodo utilizzato per ottenere le prenotazioni di un dato giorno.

```
public Set<Prenotazione> getPrenotazioniOggi(String giorno, Filtro filtro);
```

Metodo per ottenere la lista di tutte le prenotazioni.

```
public Set<Prenotazione> getPrenotazioniNonFiltrate();
```

Metodo per ottenere la lista di tutte le prenotazioni di un cittadino.

```
public Set<Prenotazione> getPrenotazioniPerCittadino(Cittadino cittadino);
```

Metodo per ottenere il cittadino attraverso la mail e la passw.

```
public Cittadino getCittadino(String mail, String psw);
```

Metodo per ottenere il cittadino attraverso solo il codice fiscale.

```
public Cittadino getCittadinoByCF(String cf);
```

Metodo per controllare se il cittadino è presente nell'anagrafe.

```
public boolean checkPerson(Cittadino cittadino);
```

Metodo per aggiungere una registrazione da parte del cittadino.

```
public void addRegistrazione(String mail, String password, String cd_fiscale);
```

Metodo per aggiungere una prenotazione, utilizzato prevalentemente dal dipendente.

```
public void addPrenotazione(String sede, String giorno, String orario, String tipo) throws Exception;
```

Metodo per controllare le credenziali del cittadino.

```
public boolean checkLogin(String mail, String password) throws Exception;
```

Metodo per la scelta e conferma della prenotazione effettuata dal cittadino.

```
public boolean confirmPrenotazioneCittadino(Cittadino cittadino, Prenotazione temp) throws Exception;
```

Metodo per controllare il caso in cui una prenotazione fosse già presente.

```
public boolean checkPrenotazionePresente(Prenotazione temp) throws Exception;
```

PATTERN UTILIZZATI

I pattern utilizzati sono il pattern "DAO" (ovvero il Data Access Object Pattern) ed il pattern "Factory".

1. Il pattern DAO è utilizzato per progettare la gestione tra applicazione e il file XML, che abbiamo come storage per dati.
2. Il pattern FACTORY è stato utilizzato per il "building" e l'uso di documenti logici basati su il file XML.

DAO:

La classe DAO si occupa della lettura e scrittura di dati direttamente nei file XML a disposizione e ci offre degli strumenti per la creazione/gestione di prenotazioni, conferme di prenotazioni, registrazioni di cittadini, controlli sui dati dei cittadini e eventuali errori.

Secondo come è stato strutturato esso richiede la presenza di un interfaccia che dovrà esser poi implementata.

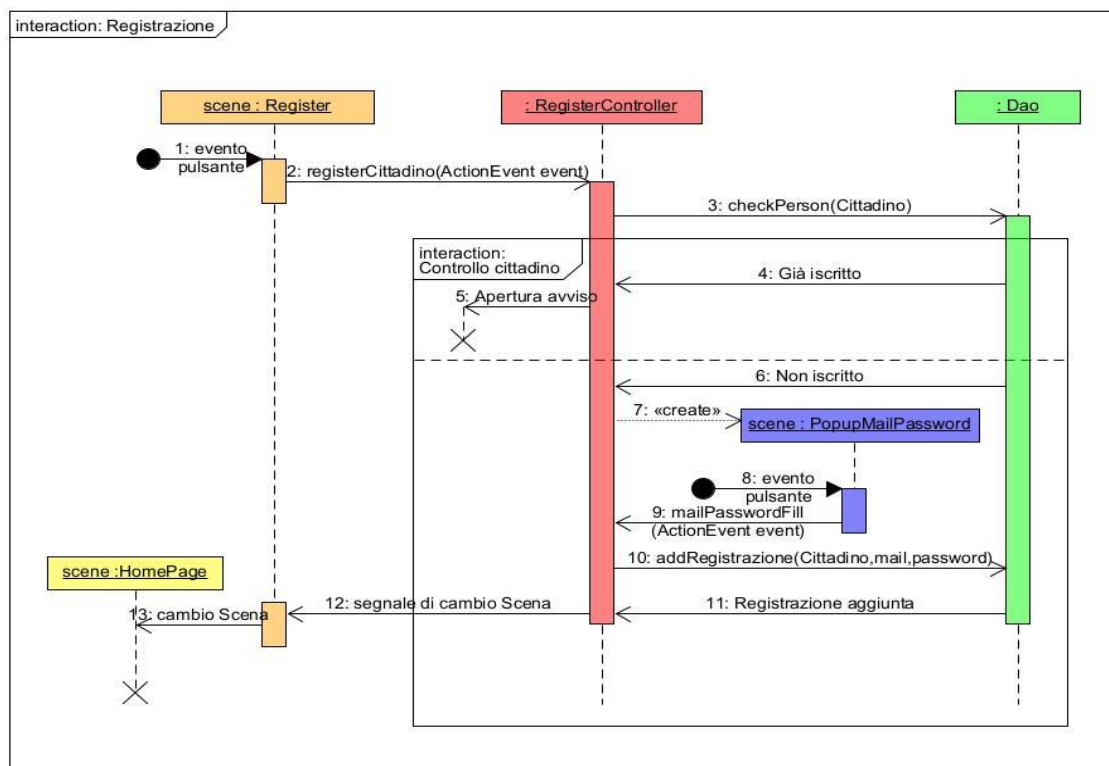
FACTORY:

Il pattern FACTORY è stato utilizzato per fornire strumenti per la creazione, e la manipolazione, di documenti logici di tipo XML.

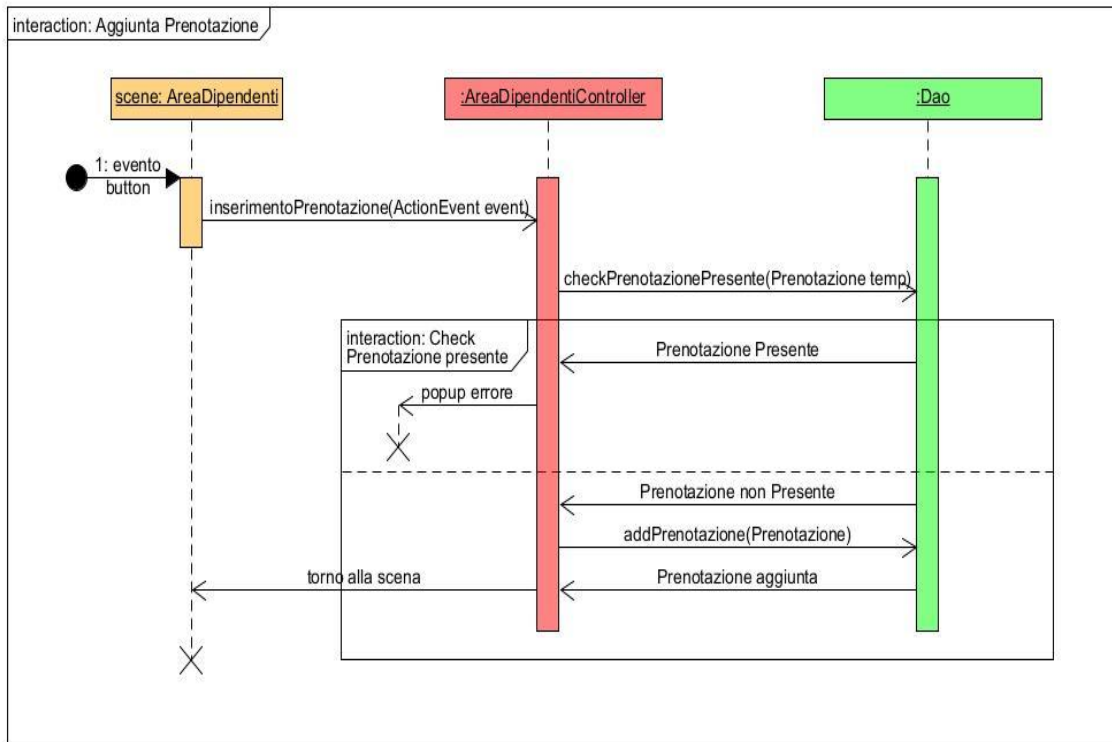
DIAGRAMMI DI SEQUENZA DEL SW IMPLEMENTATO

Seguono diagrammi di sequenza che mostrano le dinamiche di alcune interazioni tra classi particolarmente complesse o interessanti.

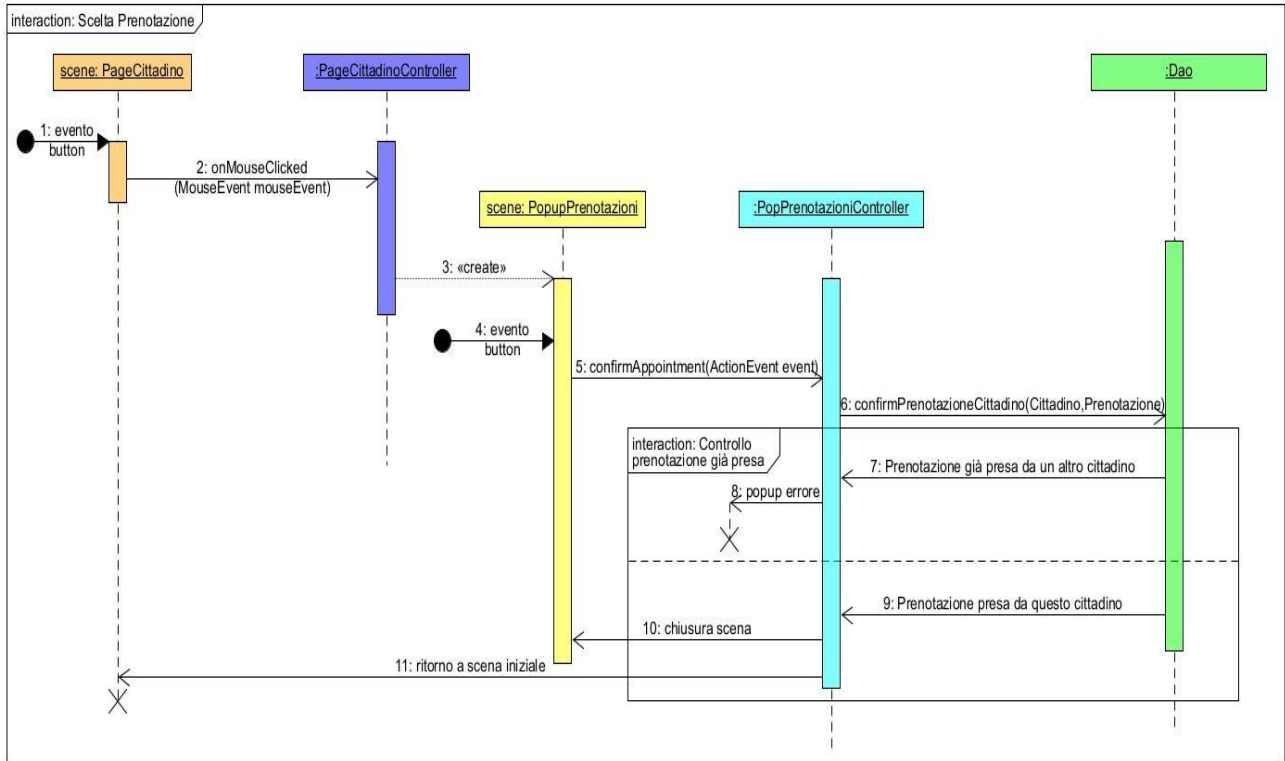
RegisterSequence



AddPrenotazioneSequence



ChoosePrenotazioneSequence



ATTIVITA' DI TEST E VALIDAZIONE

Per verificare la correttezza e le funzionalità del software si sono fatti I seguenti controlli e test:

- Verifica del documento delle specifiche rispetto a quanto implementato
- Controllo del codice, per controllare se I vari elementi usati sono stati usati in modo corretto
- Verifica della funzionalità delle varie classi tramite test da parte degli sviluppatori
- Verifica del software da parte di individui estranei al progetto

Come primo passaggio, abbiamo effettuato un controllo rispetto al documento con le specifiche iniziali, controllando di aver implementato tutti i punti richiesti (sincronizzazione tra varie istanze dello stesso software, funzionalità del calendario per mostrare il numero di appuntamenti disponibili per giorno, ecc.)

E' stato fatto un confronto anche con gli schemi classe e il documento per controllare se sono presenti tutte le variabili richieste.

Dopodichè si è controllato il codice scritto, se presenta errori o usi impropri di classi e oggetti. Si è anche controllato l'uso corretto dei due pattern usati (Data Access Object e il Factory).

Come terzo punto, si sono fatti vari test da parte degli sviluppatori, registrando cittadini (Fig. 1), aggiungendo prenotazioni dall'interfaccia del dipendente (Fig. 2), scegliendo prenotazioni dall'interfaccia del cittadino (Fig. 3), ecc.

Questi test hanno portato a delle modifiche sostanziali a vari settori del DAO e della parte grafica.

The image displays two side-by-side screenshots of a web application titled "Prenotazione Passaporti".


The left screenshot shows the "Registration Page". It features a green header with the Italian coat of arms and the text "Repubblica Italiana". Below the header, there are several input fields, some labeled "TEST", and a date field showing "01/01/2001". A green button labeled "REGISTER" is positioned at the bottom center.

The right screenshot shows the same "Registration Page" but with a modal window titled "Mail and password" open. The modal has a green header "Mail e Password" and a message: "Si prega di inserire un indirizzo mail e una password validi per poter accedere al sistema". It contains input fields for "Mail" and "Password", and a green button labeled "CONFIRM". The "REGISTER" button from the background page is still visible at the bottom.

Figura 1 (I dati del cittadino sono presenti nel file Anagrafe.xml)

Prenotazione Passaporti

INSERIMENTO PRENOTAZIONI LISTA PRENOTAZIONI

 **Appointments**


Venezia 25/07/2023

13:30 Rilascio passaporto per furto

Inserisci Prenotazione

Prenotazione Passaporti

INSERIMENTO PRENOTAZIONI LISTA PRENOTAZIONI

 **Appointments**

Prenotazione aggiunta

La prenotazione è stata aggiunta correttamente

OK

Venezia 25/07/2023

13:30 Rilascio passaporto per furto

Inserisci Prenotazione

Figura 2 (Inserimento di una prenotazione da parte di un dipendente)

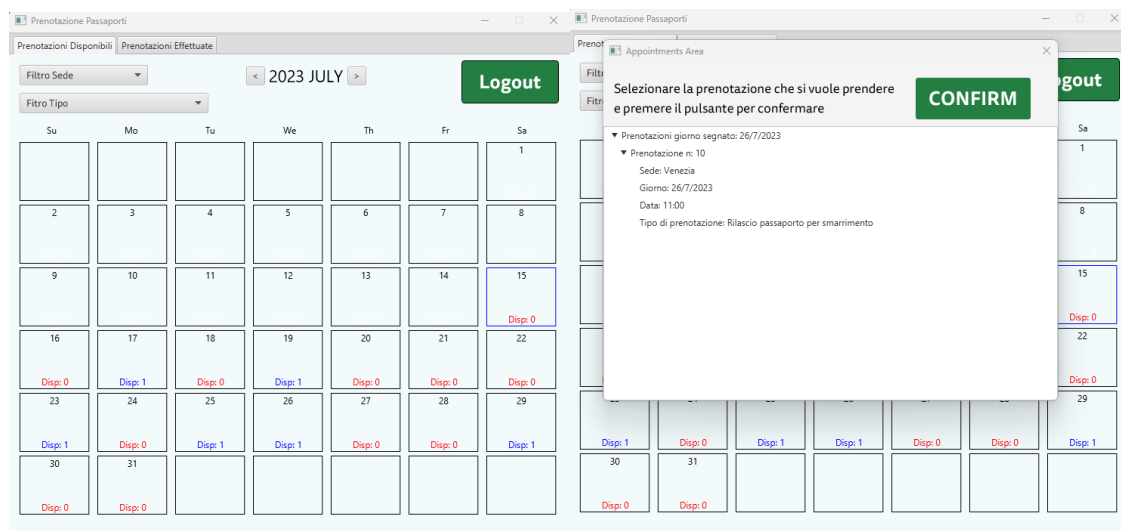


Figura 3 (Selezione di un appuntamento dal calendario)

Come ultimo test, sono stati presi degli individui al di fuori del progetto e lo si è fatto testare a loro. Questo all'inizio ha portato a molti errori, i quali hanno portato ad un cambiamento sostanziale della parte grafica per aiutare gli utenti a navigare l'applicazione.