

UNIVERSITÀ DEGLI
STUDI DI NAPOLI
FEDERICO II

Università degli Studi di Napoli "Federico II"
Dipartimento di Ingegneria Elettrica e Tecnologie dell'Informazione

Corso di Laurea in Informatica

Documentazione

Move'n'go

Insegnamento "Ingegneria del Software"

Anno Accademico 2019/2020

DOCENTE

Prof. Sergio Di Martino

AUTORI

Ciro Gallo N86002489

Gennaro Galante N86002527

ID Gruppo: INGSW1920_23

Indice

1 Presentazione del Progetto	6
1.1 Descrizione sintetica	6
1.2 Sorgenti	6
2 Documento dei Requisiti Software	7
2.1 Use Cases	7
2.1.1 Desktop	7
2.1.2 Mobile	8
2.2 Mockups	9
2.2.1 Desktop	9
2.2.2 Mobile	13
2.3 Tabelle di Cockburn	31
2.3.1 Desktop	31
2.3.2 Mobile	39
2.4 Modelli di Dominio	54
2.4.1 Class Diagrams	54
2.4.1.1 Amministratore effettua login	54
2.4.1.2 SuperAmministratore registra amministratore	55
2.4.1.3 AmministratoreLoggato aggiunge struttura	56
2.4.1.4 AmministratoreLoggato elimina strutture	57
2.4.1.5 AmministratoreLoggato visualizza struttura	58
2.4.1.6 Utente effettua login	59
2.4.1.7 Utente effettua registrazione	60
2.4.1.8 Visitatore visualizza struttura	61
2.4.1.9 Visitatore visualizza struttura su mappa	62
2.4.1.10 Visitatore visualizza recensioni	62
2.4.1.11 Visitatore aggiunge recensione	63
2.4.2 Sequence Diagrams	64
2.4.2.1 Amministratore effettua login	64
2.4.2.2 SuperAmministratore registra amministratore	65
2.4.2.3 AmministratoreLoggato aggiunge struttura	66
2.4.2.4 AmministratoreLoggato elimina strutture	66

2.4.2.5 AmministratoreLoggato modifica struttura	67
2.4.2.6 Utente effettua login	68
2.4.2.7 Utente effettua registrazione	68
2.4.2.8 Visitatore visualizza struttura	69
2.4.2.9 Visitatore visualizza struttura su mappa	69
2.4.2.10 Visitatore visualizza recensioni	70
2.4.2.11 Visitatore aggiunge recensione	70
2.4.3 Statechart Diagram	71
2.4.3.1 Visitatore visualizza recensioni	71
2.4.4 Activity Diagram	72
2.4.4.1 AmministratoreLoggato aggiunge struttura	72
2.4.5 Diagramma di Gantt	73
3 Documento di Design	74
3.1 Analisi dell'architettura	74
3.1.1 Architettura Server	75
3.1.1.1 Servizi esterni Server	76
3.1.2 Architettura Desktop	77
3.1.2.1 Servizi esterni Desktop	78
3.1.3 Architettura Android	79
3.1.3.1 Servizi esterni Android	80
3.1.4 Panoramica dei servizi esterni	80
3.1.5 Sicurezza	81
3.1.5.1 Gestione Filtri	82
3.1.5.2 Validazione del JWT	83
3.2 Class Diagram di Design	85
3.2.1 Desktop	85
3.2.2 Android	85
3.3 Sequence Diagram	86
3.3.1 Sequence Diagrams	86
3.3.1.1 Amministratore effettua login	86
3.3.1.2 SuperAmministratore registra amministratore	87
3.3.1.3 AmministratoreLoggato aggiunge struttura	88
3.3.1.4 AmministratoreLoggato elimina strutture	88
3.3.1.5 AmministratoreLoggato visualizza struttura	89
3.3.1.6 AmministratoreLoggato modifica struttura	89

3.3.1.7 Utente effettua login	91
3.3.1.8 Utente effettua registrazione	91
3.3.1.9 Visitatore visualizza struttura	92
3.3.1.10 Visitatore visualizza struttura su mappa	92
3.3.1.11 Visitatore visualizza recensioni	93
3.3.1.12 Visitatore aggiunge recensione	93
3.4 Statechart	94
3.4.1 Visitatore Visualizza Recensioni	94
3.5 CRC Cards	95
3.5.1 Desktop	95
3.5.2 Mobile	102
4 Documento di Testing	110
4.1 Test Plan	110
4.1.1 Desktop	110
4.1.2 Mobile	116
4.2 Testing del metodo prezzoDaStringaAIntero	125
4.2.1 Testing blackbox	126
4.2.2 Testing whitebox	128
4.3 Testing del metodo aggiungiValoreAMedia	130
4.3.1 Testing blackbox	131
4.3.2 Testing whitebox	134

1 Presentazione del Progetto

1.1 Descrizione sintetica

Move'n'Go è un Sistema Informativo complesso e distribuito finalizzato a offrire una piattaforma di raccomandazione di viaggi, attraverso la gestione di recensioni dei clienti.

Il sistema distribuito presenta una parte di Back-Office per la gestione delle strutture ricettive e delle recensioni da parte degli amministratori e un client su dispositivo mobile utilizzato per la ricerca delle strutture, la visualizzazione/pubblicazione delle recensioni e anche per funzionalità di localizzazione in mobilità (es: visualizza tutti i ristoranti in un raggio di 500m dalla mia posizione attuale).

I principali servizi offerti sono quattro:

1. Visualizzazione di strutture recettive, individuate attraverso filtri avanzati per la ricerca.
2. Visualizzazione di recensioni per la struttura individuata, con vari filtri (numero stelle, etc).
3. Aggiunta di una nuova Recensione.
4. Gestione delle Strutture.

Le funzionalità 1-3 sono disponibili in forma di app Android, mentre la funzionalità 4 in forma di applicativo Desktop. I due applicativi condividono lo stesso back-end.

1.2 Sorgenti

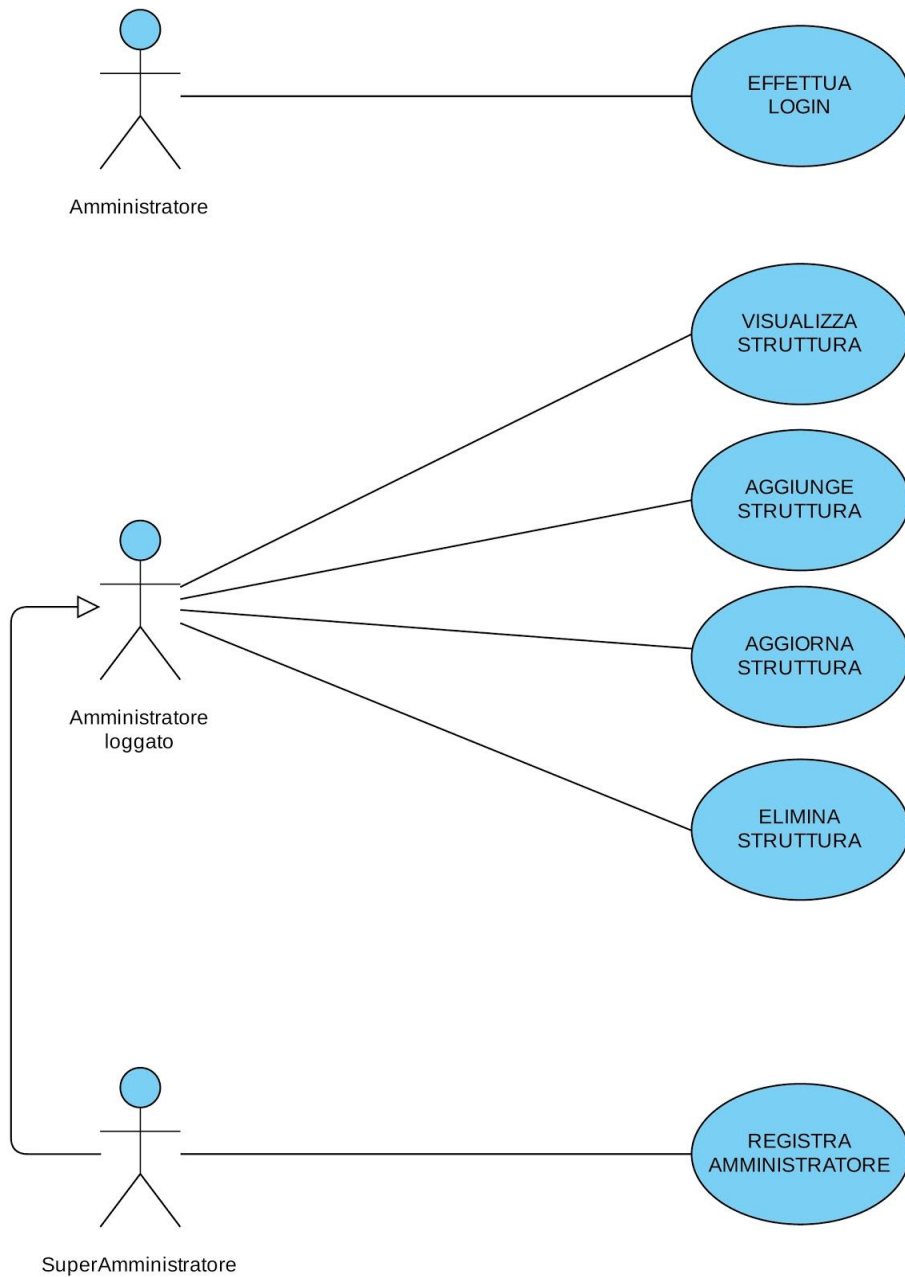
Per la visione dei sorgenti si rimanda ai seguenti repository GitHub:

- [Repository Server](#)
- [Repository Desktop](#)
- [Repository Android](#)

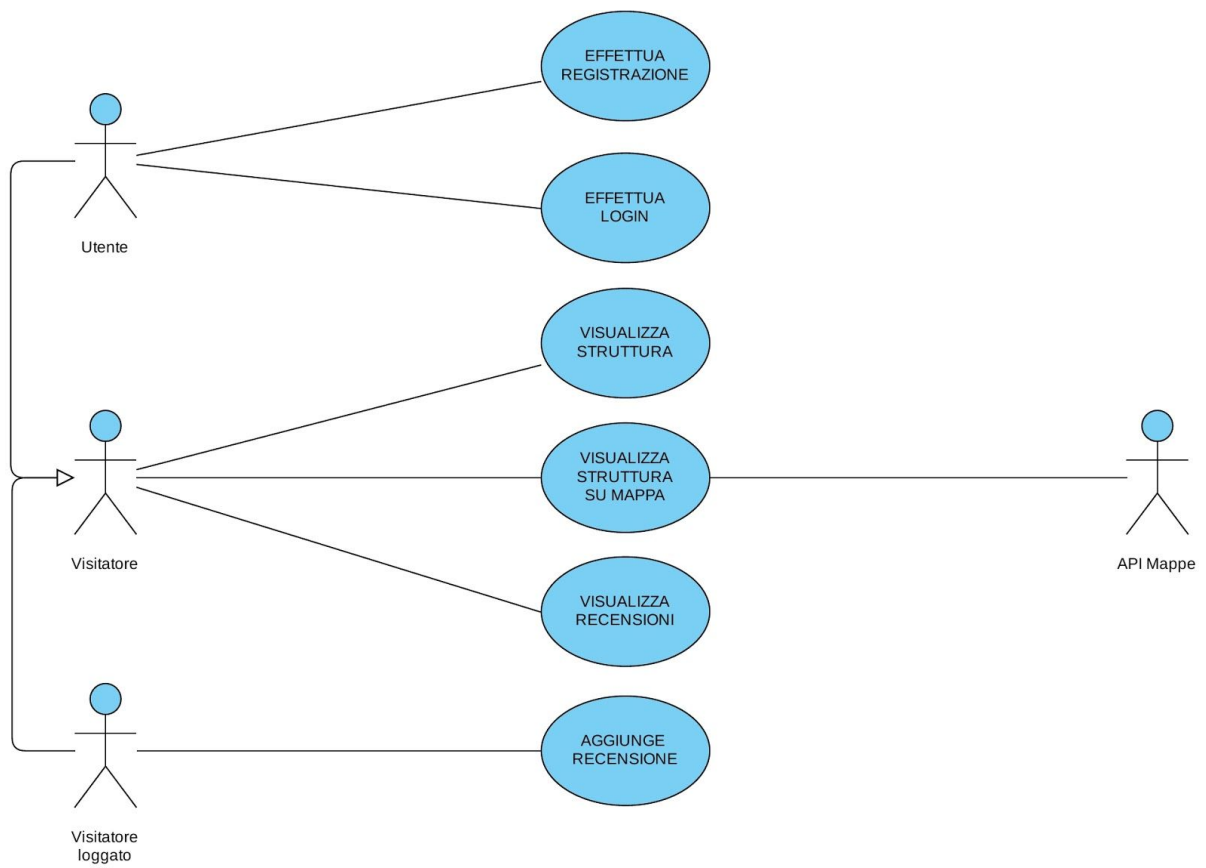
2 Documento dei Requisiti Software

2.1 Use Cases

2.1.1 Desktop



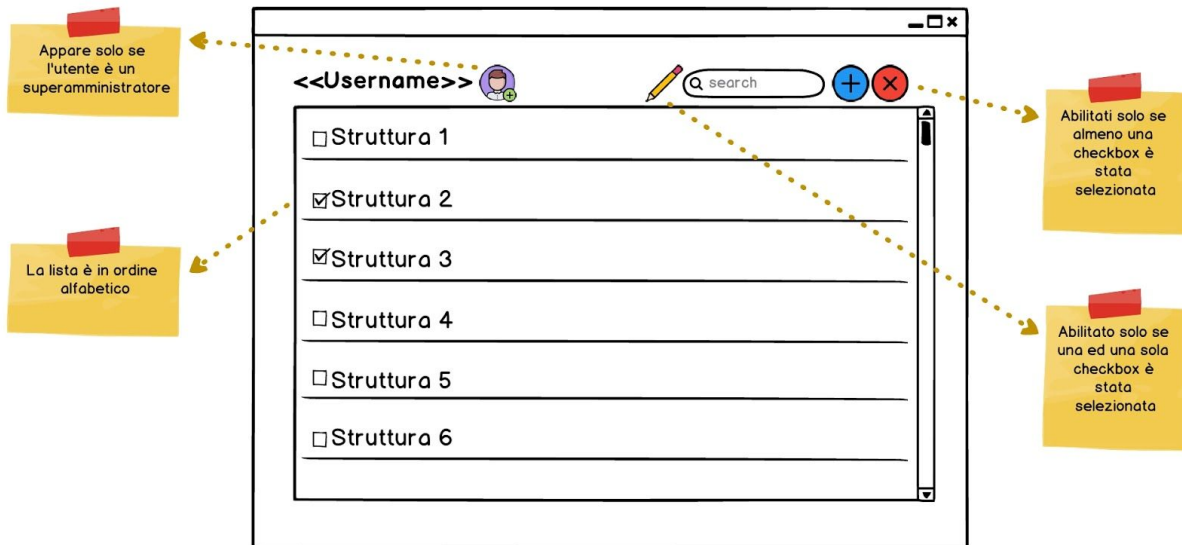
2.1.2 Mobile



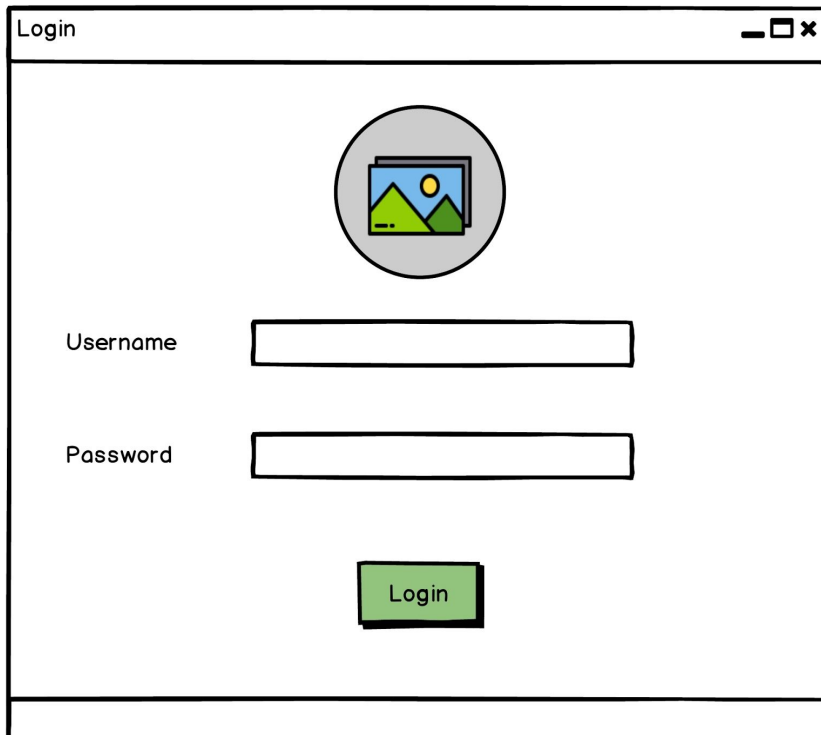
2.2 Mockups

2.2.1 Desktop

D0 - Main



D1 - Login



D2 - Registra Amministratore

The screenshot shows a web form titled "Registra Amministratore" with the following fields: Email, Nome, Cognome, Username, Password, and Conferma Password. At the bottom are "Annulla" and "Conferma" buttons. Two yellow sticky notes provide validation rules: one for the Password field stating it must be at least 8 characters with a lowercase letter and a special character, and another for the Username field stating the operation fails if the username is already in the database.

Registra Amministratore

Email

Nome

Cognome

Username

Password

Conferma Password

Annulla Conferma

La password deve contenere almeno 8 caratteri, una lettera maiuscola e un carattere speciale

L'operazione non ha successo se lo Username è già presente nel database

D3 - Aggiunge Struttura

The screenshot shows a web form titled "Aggiungi una nuova struttura" with fields for Nome, Indirizzo, Categoria (dropdown), Stelle (dropdown), Sito Web, Telefono, and Prezzo (dropdown). It also features a large image upload area with a description text area below it. "Annulla" and "Aggiungi" buttons are at the bottom. Three yellow sticky notes provide validation rules: one for the Categoria dropdown stating it is disabled for "Attrazione" and has a range of 0-3 for "Ristorante"; one for the Stelle dropdown stating it must have a valid format; and one for the Telefono field stating it must have a valid format.

Aggiungi una nuova struttura

Nome

Indirizzo

Categoria

Stelle

Sito Web

Telefono

Prezzo

Descrizione

Annulla Aggiungi

Disabilitato se Categoria = "Attrazione"

Range 0-3 se Categoria = "Ristorante"

Devono avere un formato valido

D4 - Visualizza Struttura

Grand Hotel Vesuvio

4.5

Albergo

5 Stelle

N.ro Recensioni 20000

Descrizione

Indirizzo Via Roma 22

Telefono +39 9999999

Sito Web www.miosito.it

Prezzo €€€€

Indietro

Mostra un'immagine di default se non è stata caricata alcuna immagine

Vengono mostrati solo i campi aggiunti dall'amministratore

D5 - Modifica Struttura

I campi sono automaticamente riempiti con le informazioni correnti

- Disabilitato se Categoria = "Attrazione"
- Range 0-3 se Categoria = "Ristorante"

Devono avere un formato valido

Modifica la struttura

Nome

Indirizzo

Categoria

Stelle

Sito Web

Telefono

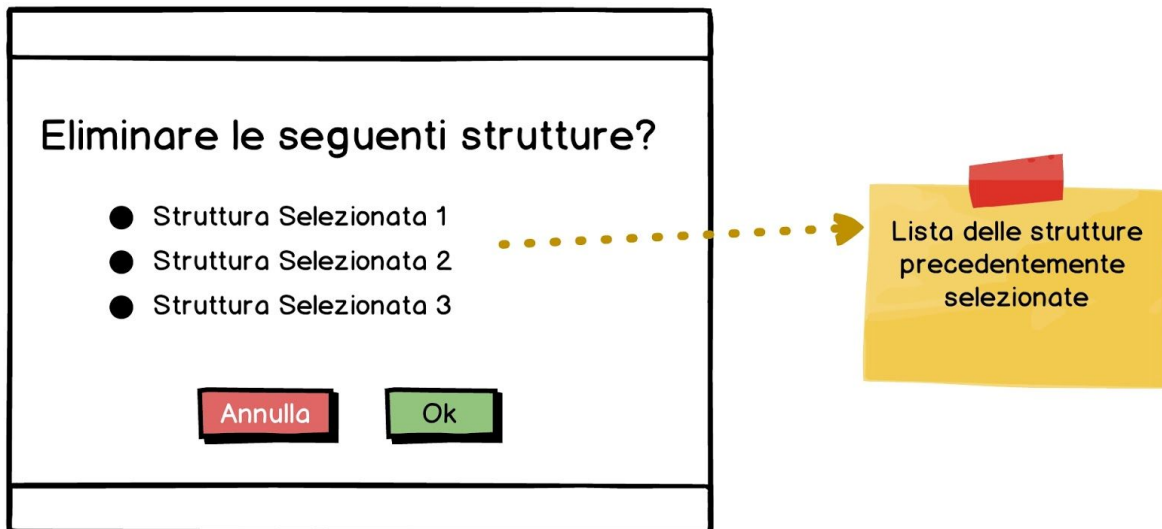
Prezzo

Descrizione

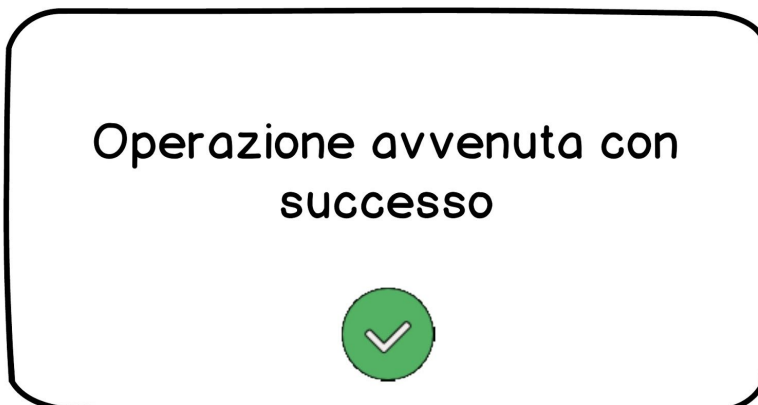
Annulla

Conferma

D6 - Elimina Struttura



D200 - Successo

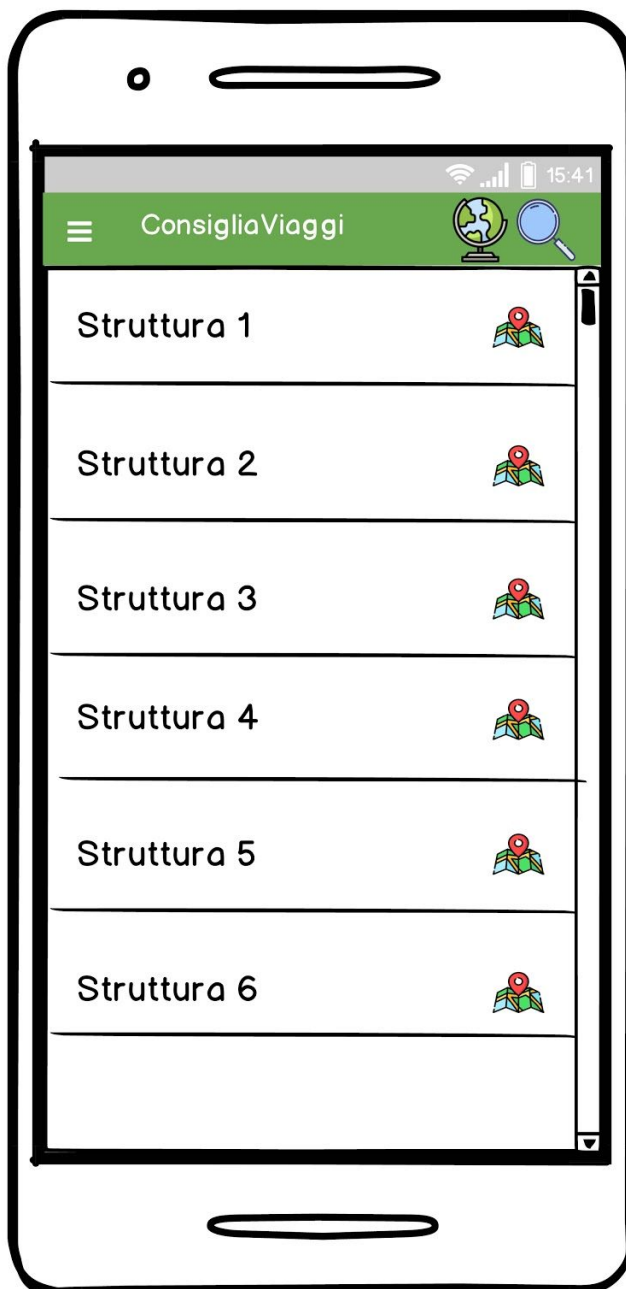


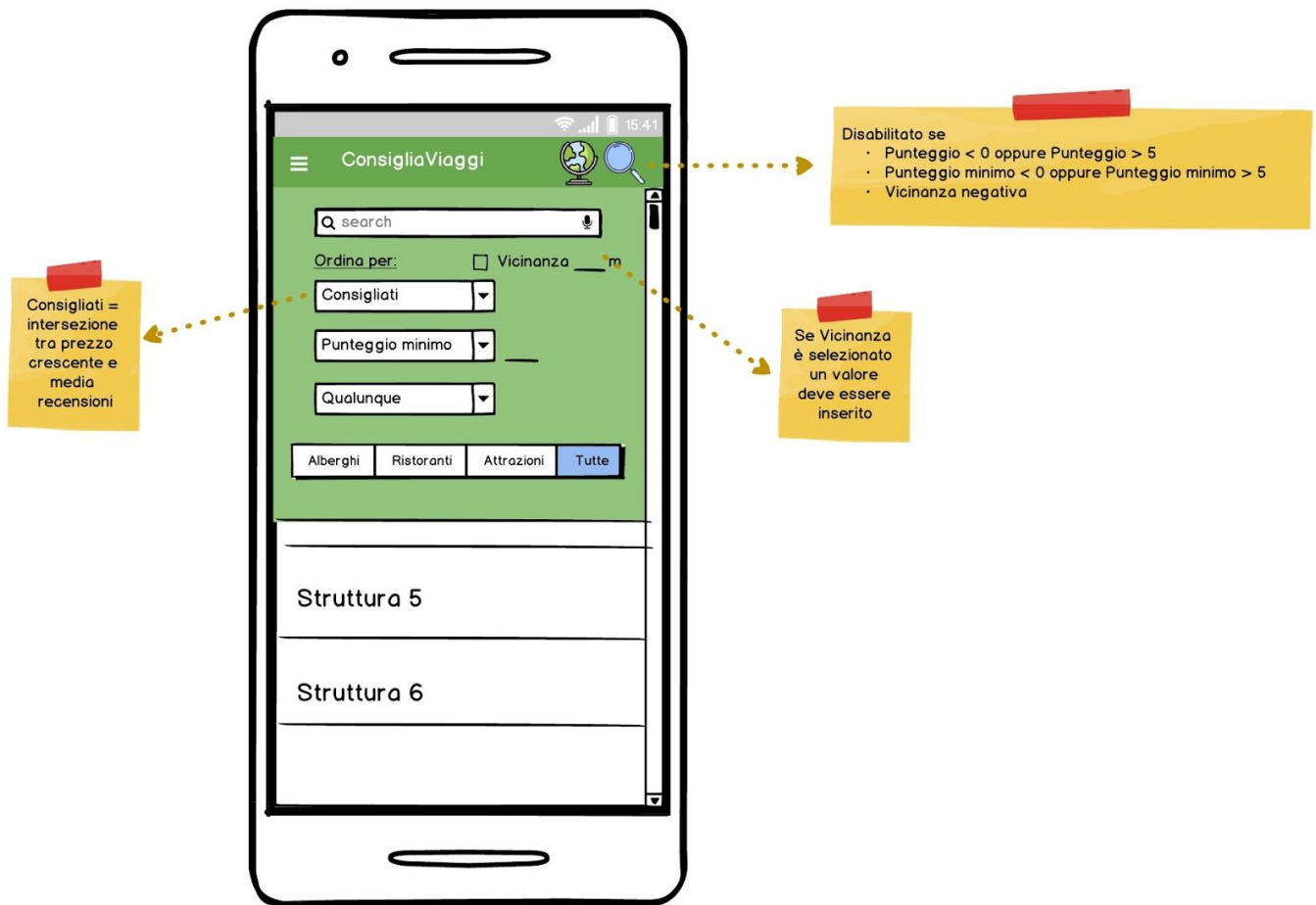
D404 - Errore



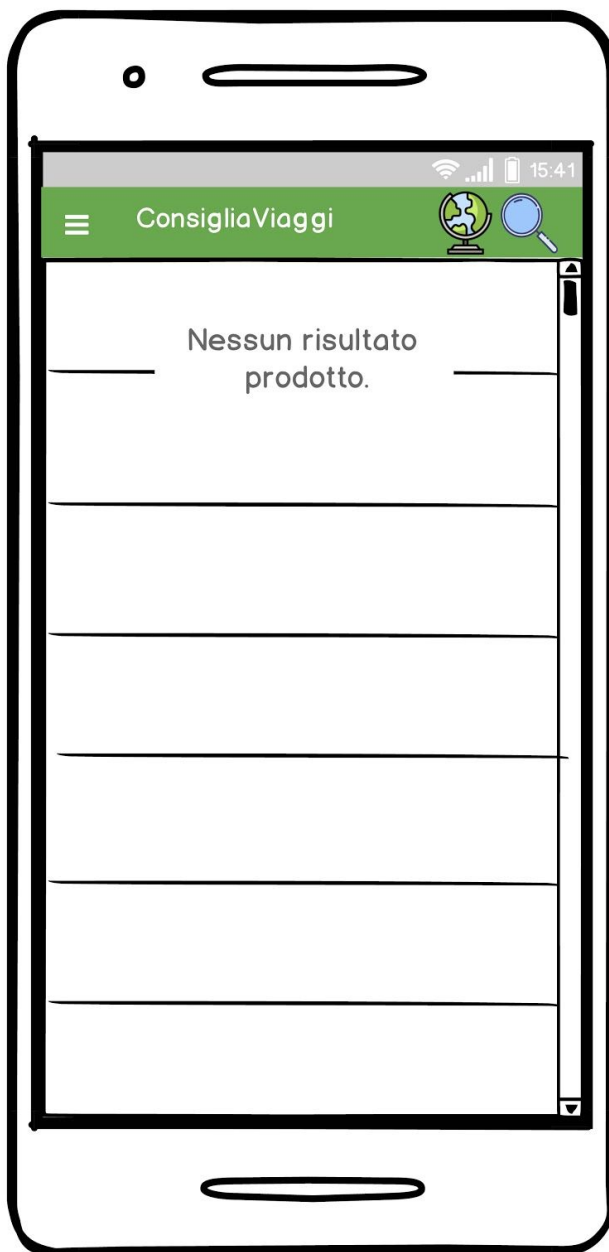
2.2.2 Mobile

M0 - Main

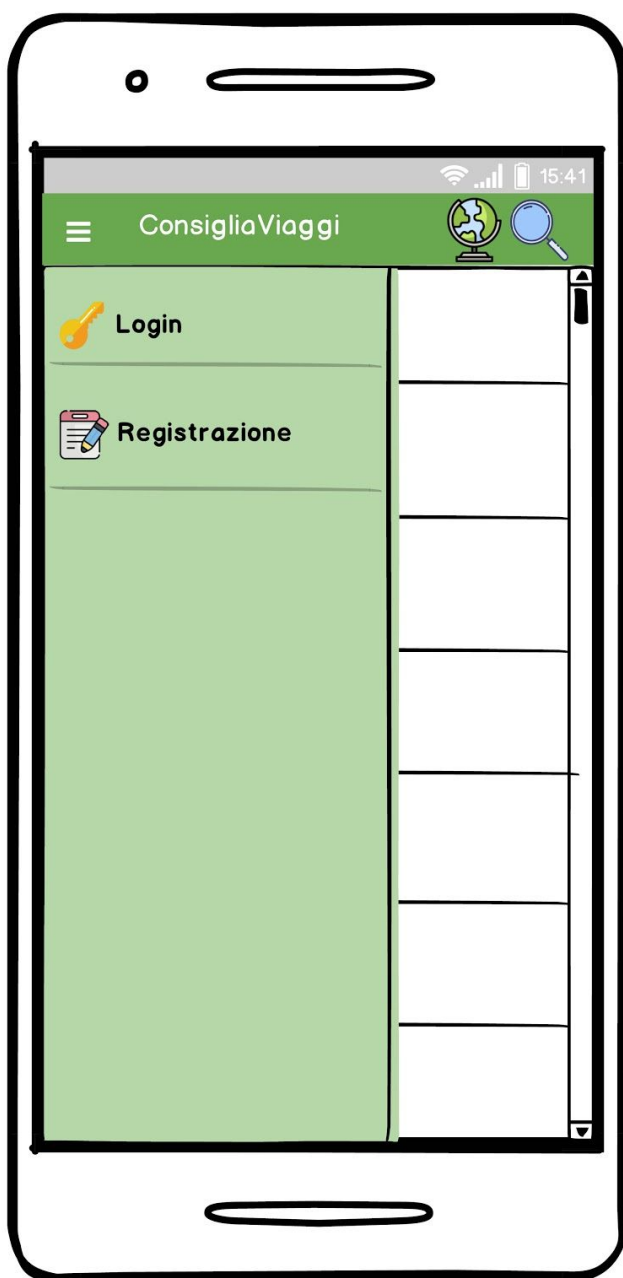




M404 - Main Errore



M2 - Menu Non Loggato



M3 - Login

A hand-drawn sketch of a mobile application interface for a login screen. The app is titled "ConsigliaViaggi" in a green header bar. The status bar at the top shows a Wi-Fi signal, cellular signal, battery level, and the time 15:41. The main content area features a circular profile picture placeholder with a landscape icon. Below this are two text input fields labeled "Username" and "Password", each with a horizontal underline. A green "Login" button is positioned below the password field. The entire interface is framed by a thick black outline representing the phone's bezel and home indicator.

ConsigliaViaggi

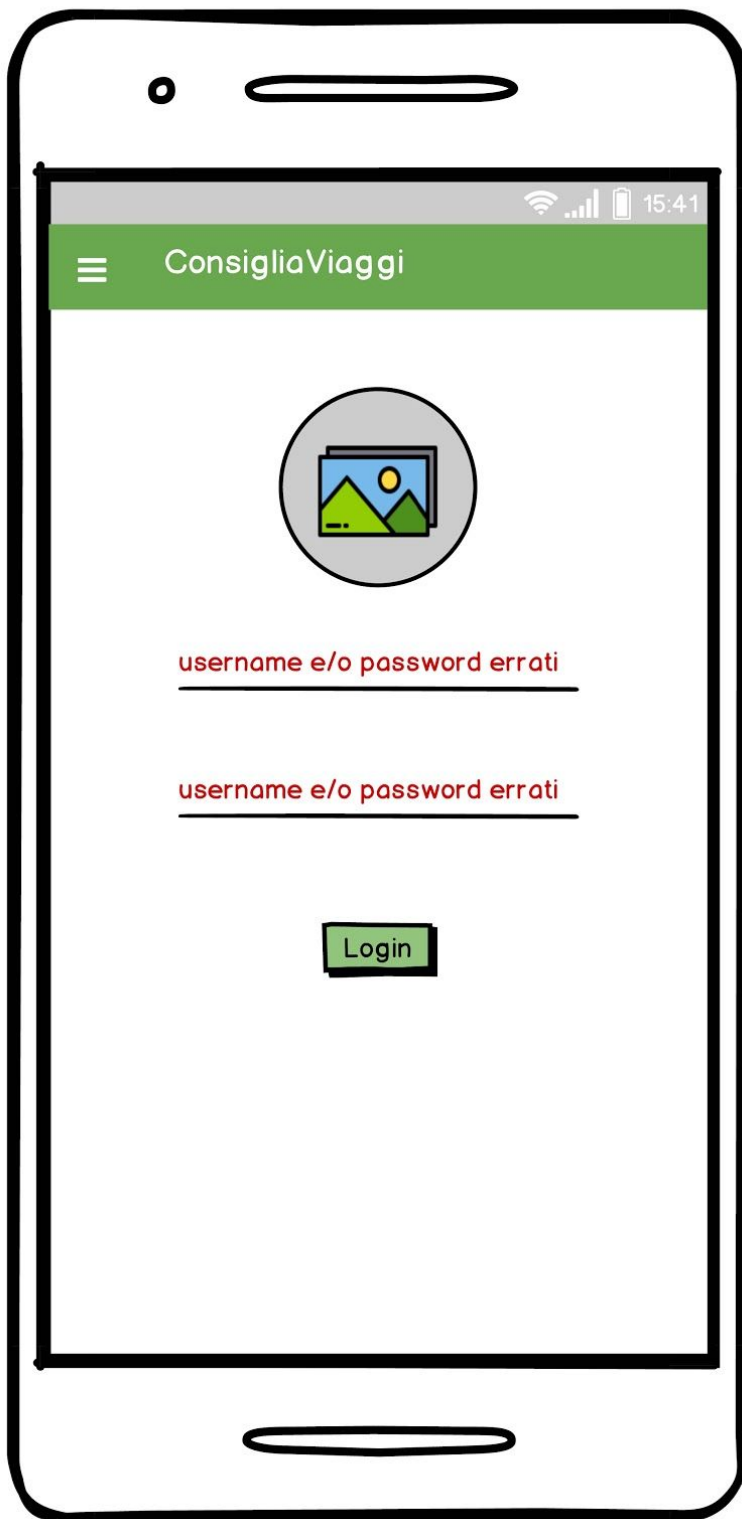
15:41

Username

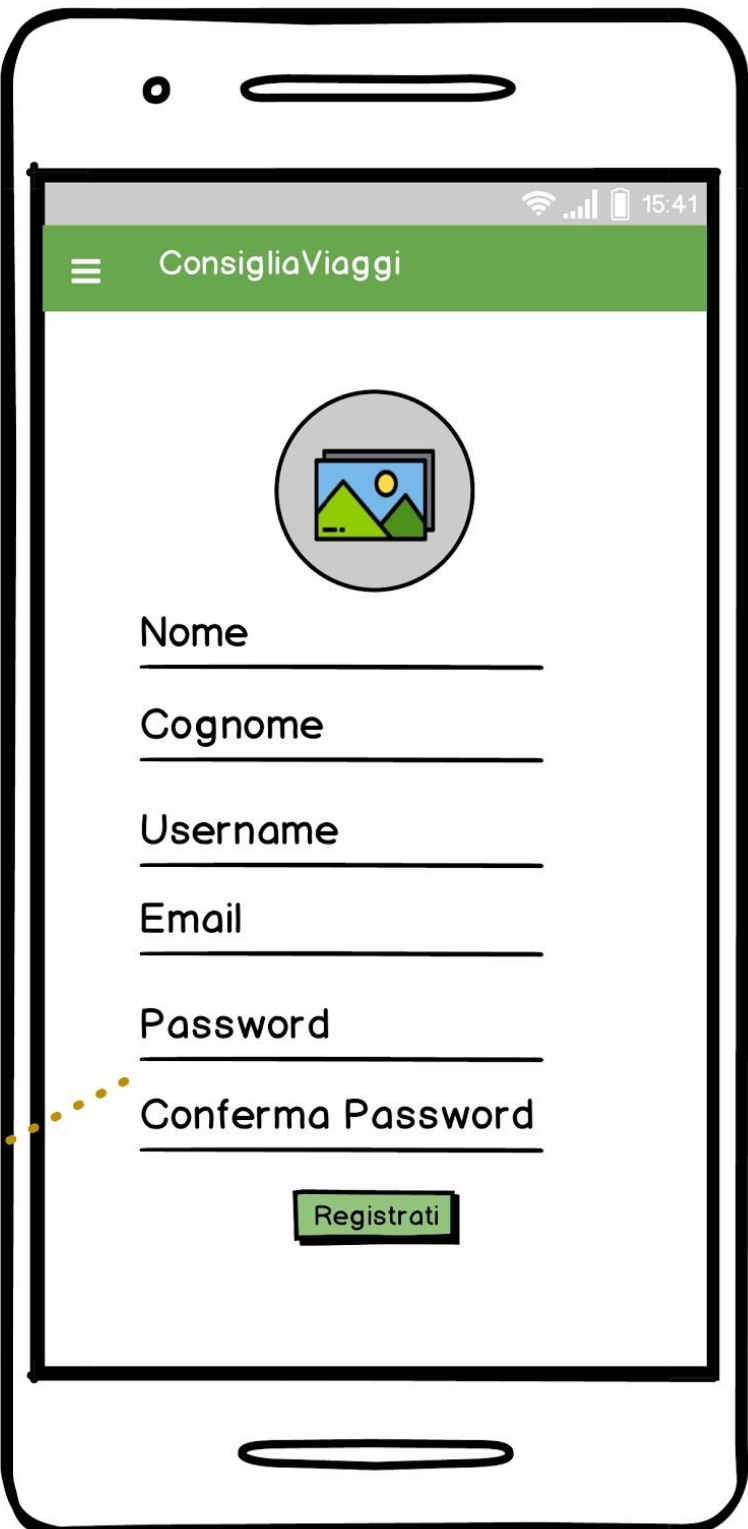
Password

Login

M4 - Login Errore



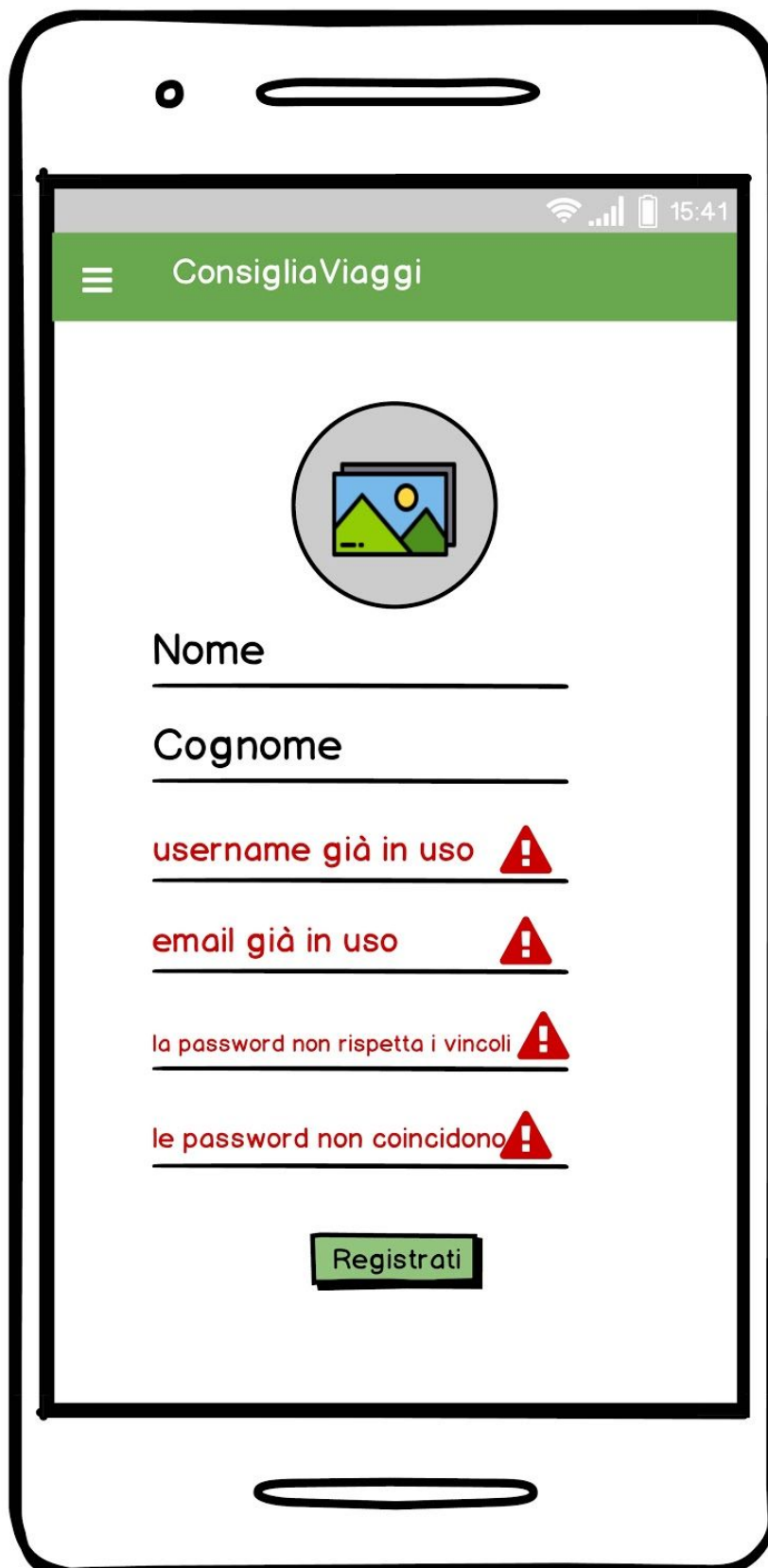
M5 - Registrazione



The image shows a smartphone screen with a registration form. The status bar at the top right shows signal strength, battery level, and the time 15:41. The app's header is green with a hamburger menu icon and the text 'ConsigliaViaggi'. Below the header is a circular profile picture placeholder with a landscape icon. The form consists of several text input fields labeled 'Nome', 'Cognome', 'Username', 'Email', 'Password', and 'Conferma Password'. A green 'Registrati' button is at the bottom of the form. A yellow sticky note with a red tab is attached to the left of the phone, with a dotted line pointing to the 'Password' field. The note contains the following text:


La password deve contenere almeno 8 caratteri, una lettera maiuscola e un carattere speciale

M6 - Registrazione Errore



The image shows a hand-drawn illustration of a smartphone screen. At the top, there is a status bar with a Wi-Fi icon, signal strength bars, a battery icon, and the time 15:41. Below the status bar is a green header bar with a hamburger menu icon on the left and the text "ConsigliaViaggi" in the center. The main content area is white and contains a circular profile picture placeholder with a landscape icon. Below the profile picture are four text input fields, each with a label and a horizontal line for text entry. The labels are "Nome", "Cognome", "username già in uso", and "email già in uso". Below the "username già in uso" and "email già in uso" fields are two red warning icons (exclamation marks inside triangles). Below the "email già in uso" field is a red warning icon. Below the first warning icon is a red warning icon. Below the second warning icon is a red warning icon. Below the third warning icon is a red warning icon. Below the fourth warning icon is a red warning icon. Below the warning icons is a green button with the text "Registrati".

ConsigliaViaggi



Nome

Cognome

username già in uso

email già in uso

la password non rispetta i vincoli

le password non coincidono

Registrati

M7 - Registrazione Effettuata

The image is a hand-drawn sketch of a smartphone screen. At the top, there is a status bar with a Wi-Fi icon, a cellular signal icon, a battery icon, and the time 15:41. Below the status bar is a green header bar with a white hamburger menu icon on the left and the text 'ConsigliaViaggi' in white. The main content area is white. In the center, there is a circular profile picture placeholder containing a landscape icon. Below the profile picture, the name 'Mario' is written in black. Underneath the name is a rounded rectangular box with the title 'Attenzione' and the text 'Come vuoi che gli altri utenti ti visualizzino?'. Below this box are two input fields: 'Username' and 'Nome e Cognome'. Below these fields are two rows of asterisks, each followed by an input line. At the bottom of the form is a green button with the text 'Registrati' in black. The entire screen is framed by a black outline representing the phone's bezel and home button area.

ConsigliaViaggi

Mario

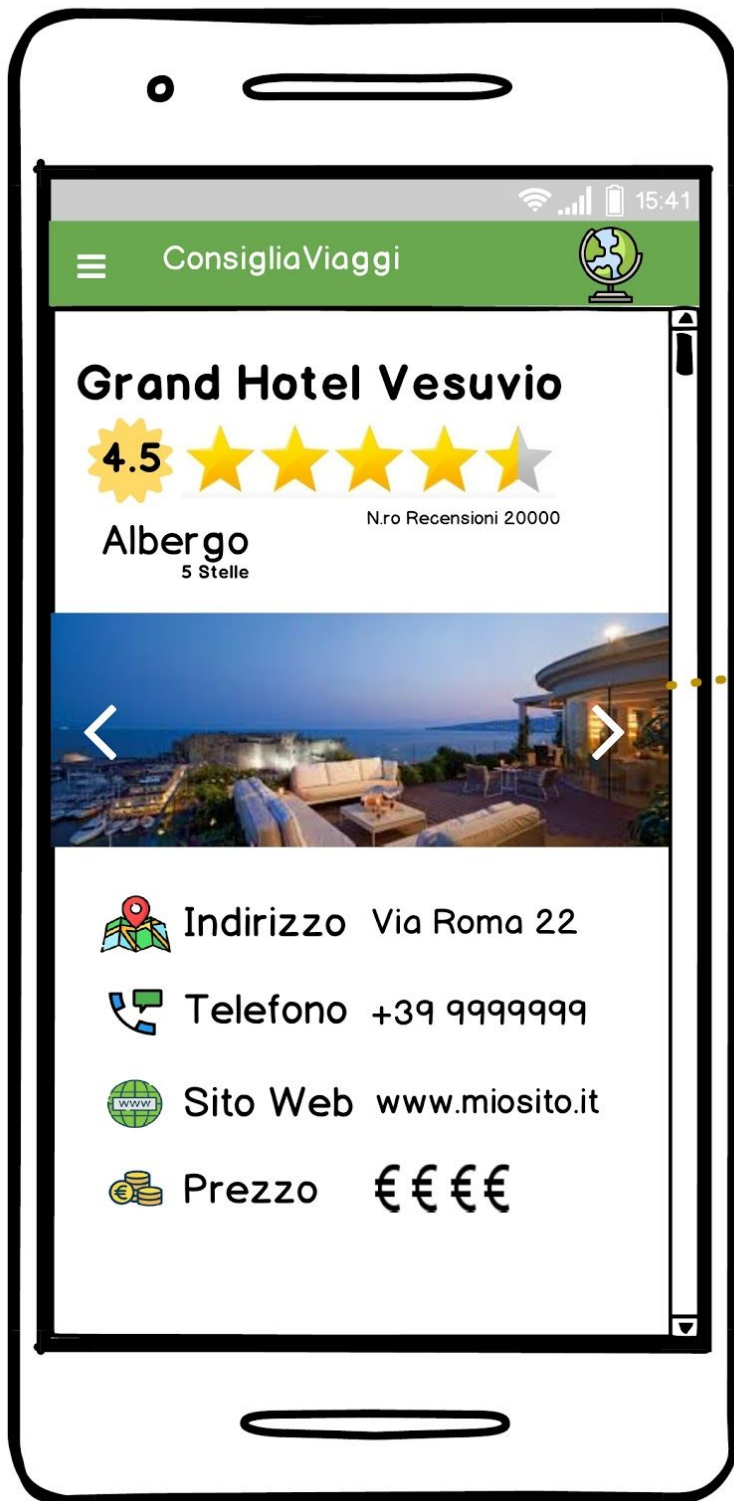
Attenzione

Come vuoi che gli altri utenti ti visualizzino?

Username

Nome e Cognome

Registrati



Mostra un'immagine di default se non è stata caricata alcuna immagine







ConsigliaViaggi

Grand Hotel Vesuvio

Scrivi la tua recensione

<<Identità Visitatore>>

★★★★★

Titolo Recensione [30]

[300]

Annulla Pubblica

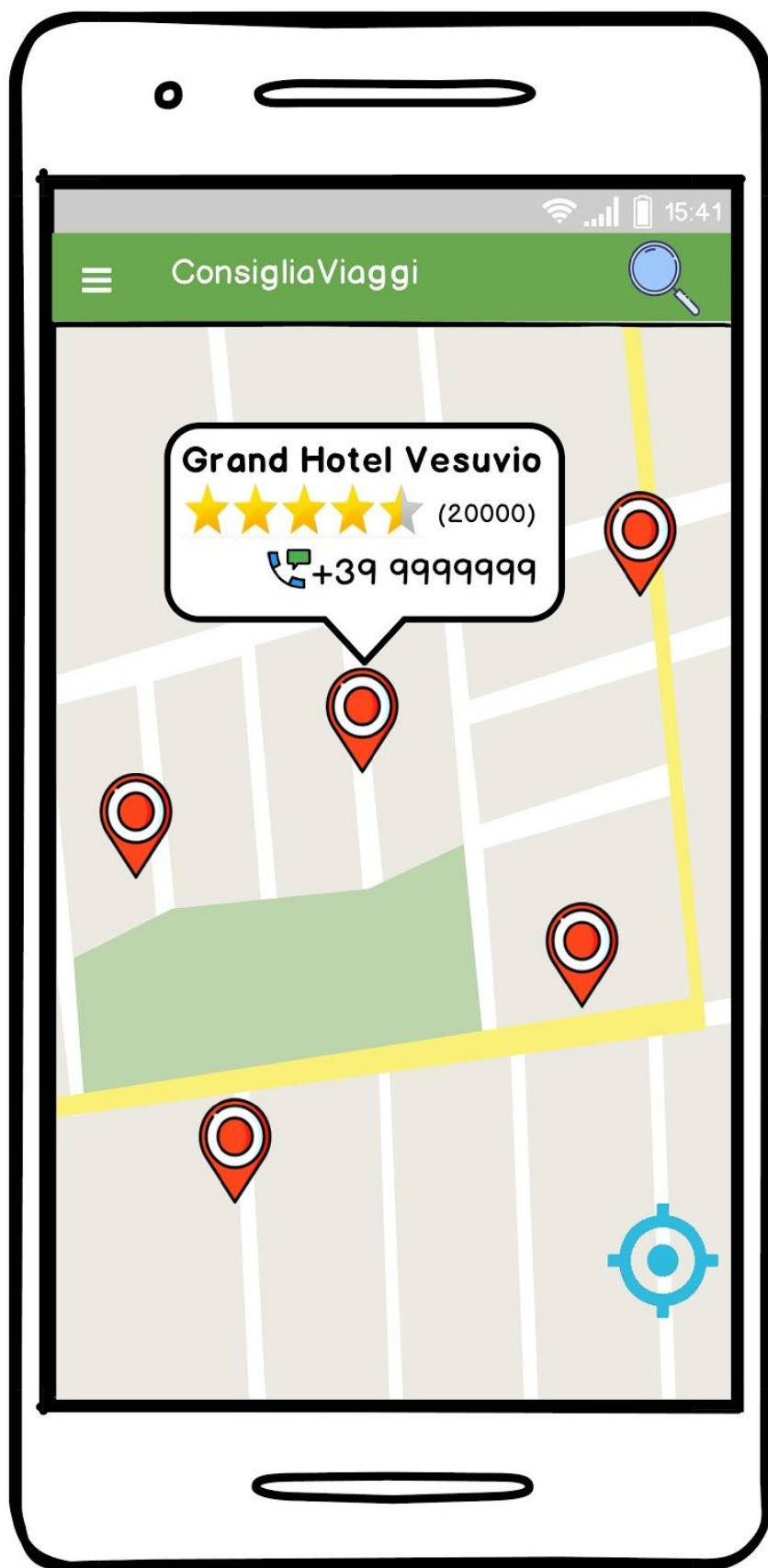
L'input non viene più letto se si supera la dimensione massima

Abilitato solo se sono stati inseriti tutti i campi

M13 - Aggiunge Recensione Successo



M14 - Visualizza su Mappa



M15 - Visualizza su Mappa Ricerca



Disabilitato se

- Punteggio < 0 oppure Punteggio > 5
- Punteggio minimo < 0 oppure Punteggio minimo > 5

M16 - Visualizza su Mappa Ricerca Errore



Disabilitato se

- Punteggio < 0 oppure Punteggio > 5
- Punteggio minimo < 0 oppure Punteggio minimo > 5

2.3 Tabelle di Cockburn

2.3.1 Desktop

Use Case	Effettua Login	
Preconditions	Nessuna	
Success End Condition	L'amministratore effettua con successo il login	
Failed End Condition	Credenziali errate; l'amministratore non inserisce (almeno) un campo	
Primary Actor	Amministratore	
Trigger	Nessuno	
Scenario di successo		
Step #	Amministratore	Sistema
1		Mostra D1
2	Inserisce username e password e clicca "Login"	
3		Mostra D0 con la lista in ordine alfabetico, senza il pulsante "aggiungi amministratore" e con "Username" uguale allo username dell'amministratore
Extension #1: Credenziali errate		
3.1		Mostra D404 con messaggio "Username e/o password errati"
4.1	Preme "Ok"	
5.1		Mostra D1
Extension #2: Username e/o password non inseriti		

2.2	Clicca il bottone "Login"	
3.2		Mostra D404 con messaggio "Username e/o password non inseriti"

Use Case	Registra Amministratore	
Preconditions	Nessuna	
Success End Condition	Il SuperAmministratore registra con successo un nuovo amministratore	
Failed End Condition	Username non disponibile; il SuperAmministratore annulla la registrazione; il SuperAmministratore non inserisce (almeno) un campo; il SuperAmministratore inserisce una password che non rispetta i vincoli	
Primary Actor	SuperAmministratore	
Trigger	Il SuperAmministratore clicca il bottone “Aggiungi Amministratore” dalla schermata D0	
Scenario di successo		
Step #	SuperAmministratore	Sistema
1	Clicca il bottone “Aggiungi Amministratore” dalla schermata D0	
2		Mostra D2
3	Inserisce nome, cognome, username e password e clicca su “Conferma”	
4		Mostra D200
Extension #1: Username non disponibile		
4.1		Mostra D404 con messaggio “Username non disponibile”
5.1	Preme “Ok”	
6.1		Mostra D2
Extension #2: Registrazione annullata		

3.2	Clicca su “Annulla”	
4.2		Mostra D0 con la lista in ordine alfabetico, con il pulsante “aggiungi amministratore” e con “Username” uguale allo username del SuperAmministratore
Extension #3: Almeno un campo non è stato inserito		
3.3	Clicca su “Conferma”	
4.3		Mostra D404 con messaggio “Tutti i campi sono obbligatori”
5.3	Preme “Ok”	
6.3		Mostra D2
Extension #4: la Password non rispetta i vincoli		
4.4	Clicca il bottone “Conferma”	
5.4		Mostra D404 con messaggio “La password deve contenere almeno 8 caratteri, una lettera maiuscola e un carattere speciale”
6.4	Preme “Ok”	
7.4		Mostra D2
Extension #5: la Password non rispetta i vincoli		
4.4	Clicca il bottone “Conferma”	
5.4		Mostra D404 con messaggio “Le password non coincidono”
6.4	Preme “Ok”	
7.4		Mostra D2

Use Case	Visualizza Struttura	
Preconditions	Nessuna	
Success End Condition	L'amministratore loggato visualizza la struttura di interesse	
Failed End Condition	Nessuna struttura rispetta i termini di ricerca	
Primary Actor	Amministratore Loggato	
Trigger	Nessuno	
Scenario di successo		
Step #	Amministratore Loggato	Sistema
1		Mostra D0 con la lista in ordine alfabetico e con "Username" uguale allo username dell'amministratore
2	Clicca sul nome di una struttura	
3		Mostra D4
Subvariation #1: Ricerca per parola chiave		
2.1	Scrive una o più parola/e chiave nella barra di ricerca	
3.1		Mostra D0 con la lista delle strutture il cui nome contiene la/e parola/e chiave e in ordine alfabetico
4.1	Clicca sul nome di una struttura	
5.1		Mostra D4
Extension #1: Nessuna struttura rispetta i termini di ricerca		
2.1	Scrive una o più parola/e chiave nella barra di ricerca	
3.1		Mostra D404 con messaggio "Nessuna struttura rispetta i

		termini di ricerca”
4.1	Preme “Ok”	
5.1		Mostra D0 con la lista in ordine alfabetico

Use Case	Aggiunge Struttura	
Preconditions	Nessuna	
Success End Condition	L'amministratore loggato aggiunge la nuova struttura nel sistema	
Failed End Condition	L'amministratore loggato annulla l'operazione; l'amministratore loggato non inserisce il nome o l'indirizzo; email non valida; sito web non valido; struttura già presente nel sistema	
Primary Actor	Amministratore Loggato	
Trigger	L'amministratore loggato clicca sul bottone "Aggiungi Struttura" nella schermata D0	
Scenario di successo		
Step #	Amministratore Loggato	Sistema
1	Clicca sul bottone "Aggiungi Struttura" nella schermata D0	
2		Mostra D3
3	Inserisce nome, indirizzo, categoria, stelle, sito web, telefono, range di prezzo, descrizione e foto e clicca su "Aggiungi"	
4		Mostra D200
Extension #1: "Aggiunge Struttura" annullato		
3.1	Clicca su "Annulla"	
4.1		Mostra D0 con la lista in ordine alfabetico e "Username" uguale allo username

		dell'amministratore loggato
Extension #2: Indirizzo e/o Nome mancanti		
3.2	Clicca su "Aggiungi"	
4.2		Mostra D404 con messaggio "Nome e Indirizzo sono campi obbligatori"
5.2	Clicca su "Ok"	
6.2		Mostra D3
Extension #3: Email non valida		
4.3		Mostra D404 con messaggio "Email non valida"
5.3	Clicca su "Ok"	
6.3		Mostra D3
Extension #4: Sito Web non valido		
4.4		Mostra D404 con messaggio "Sito Web non valido"
5.4	Clicca su "Ok"	
6.4		Mostra D3
Extension #5: Struttura già presente nel sistema		
4.5		Mostra D404 con messaggio "La struttura è già presente nel sistema"
5.5	Clicca su "Ok"	
6.5		Mostra D3

Use Case	Modifica Struttura	
Preconditions	Nessuna	
Success End Condition	L'amministratore loggato modifica la struttura	
Failed End Condition	L'amministratore loggato annulla l'operazione; nome o indirizzo mancanti; email non valida; sito web non valido;	
Primary Actor	Amministratore Loggato	
Trigger	L'amministratore loggato seleziona una struttura dalla lista e clicca sul bottone "Modifica" nella schermata D0	
Scenario di successo		
Step #	Amministratore Loggato	Sistema
1	Clicca sul bottone "Modifica" nella schermata D0	
2		Mostra D5 con le informazioni della struttura precompilate
3	Modifica nome, indirizzo, categoria, stelle, sito web, telefono, range di prezzo, descrizione e clicca su "Conferma"	
4		Mostra D200
Subvariation #1: Categoria "Attrazione"		
3.1	Modifica nome, indirizzo e seleziona "Attrazione"	
4.1		Disabilita "Stelle"
5.1	Clicca su "Conferma"	
6.1		Mostra D200
Extension #1: "Modifica Struttura" annullato		
3.1	Clicca su "Annulla"	
4.1		Mostra D0 con la lista in ordine alfabetico e

		"Username" uguale allo username dell'amministratore loggato
Extension #2: Nome e/o indirizzo mancanti		
3.2	Modifica i campi e lascia "Nome" e/o "Indirizzo" vuoti	
4.2		Mostra D404 con messaggio "Nome e Indirizzo sono campi obbligatori"
5.2	Clicca su "Ok"	
6.2		Mostra D5
Extension #3: Email non valida		
4.3		Mostra D404 con messaggio "Email non valida"
5.3	Clicca su "Ok"	
6.3		Mostra D5
Extension #4: Sito Web non valido		
4.4		Mostra D404 con messaggio "Sito Web non valido"
5.4	Clicca su "Ok"	
6.4		Mostra D5

Use Case	Elimina Struttura
Preconditions	Nessuna

Success End Condition	L'amministratore loggato rimuove la struttura dal sistema	
Failed End Condition	L'amministratore loggato annulla l'operazione	
Primary Actor	Amministratore Loggato	
Trigger	L'amministratore loggato seleziona una o più strutture dalla lista e clicca sul bottone "Elimina" nella schermata D0	
Scenario di successo		
Step #	Amministratore Loggato	Sistema
1	Clicca sul bottone "Elimina Struttura" nella schermata D0	
2		Mostra D6 con la lista delle strutture selezionate nella schermata D0
3	Clicca sul bottone "Ok"	
4		Mostra D0 con la lista in ordine alfabetico e "Username" uguale allo username dell'amministratore loggato
Extension #1: "Elimina Struttura" annullato		
3.1	Clicca su "Annulla"	
4.1		Mostra D1

2.3.2 Mobile

Use Case	Effettua Login
-----------------	----------------

Preconditions	Nessuna	
Success End Condition	L'utente effettua con successo il login	
Failed End Condition	Credenziali errate; username e/o password mancanti	
Primary Actor	Utente	
Trigger	L'utente clicca sull'hamburger menu di M0 e successivamente sul bottone "Login"	
Scenario di successo		
Step #	Utente	Sistema
1	L'utente clicca sull'Hamburger Menu di M0 e successivamente sul bottone "Login"	
2		Mostra M3
3	Inserisce username e password e clicca "Login"	
4		Mostra M0 con la lista ordinata per "Consigliati" e "Punteggio minimo" uguale a 0
Extension #1: Credenziali errate		
4.1		Mostra M4
Extension #2: Username e/o password non inseriti		
3.2	Clicca il tasto "Login"	
4.2		Mostra M4

Use Case	Effettua Registrazione
Preconditions	Nessuna

Success End Condition	L'utente effettua con successo la registrazione	
Failed End Condition	Username non disponibile; e-mail non disponibile; password non coincidenti; la password non rispetta i vincoli; almeno un campo mancante	
Primary Actor	Utente	
Trigger	L'utente clicca sull'hamburger menu di M0 e successivamente sul bottone "Registrazione"	
Scenario di successo		
Step #	Utente	Sistema
1	Clicca sull'Hamburger Menu di M0 e successivamente sul bottone "Registrazione"	
2		Mostra M5
3	Inserisce nome, cognome, username, e-mail, password e conferma password e clicca su "Registrati"	
4		Mostra M7
5	Clicca su "Username"	
6		Mostra M0
Subvariation #1: Utente sceglie "Nome e Cognome"		
5.1	Clicca su "Nome e Cognome"	
6.1		Mostra M0
Extension #1: Username non disponibile		
4.1		Mostra M6 con "Username già in uso"
Extension #2: Email non disponibile		
4.2		Mostra M6 con "E-mail già in uso"
Extension #3: Le password non coincidono		
4.3		Mostra M6 con "le

		password non coincidono"
Extension #4: La password non rispetta i vincoli		
4.4		Mostra M6 con "la password non rispetta i vincoli"
Extension #5: Almeno un campo mancante		
3.5	Clicca su "Registrati"	
4.5		Mostra M6 con i campi non inseriti in rosso

Use Case	Visualizza Struttura
Preconditions	Nessuna

Success End Condition	Il visitatore visualizza in dettaglio la struttura di interesse; il visitatore visualizza in dettaglio la struttura di interesse, filtrata e ordinata secondo le preferenze	
Failed End Condition	I filtri selezionati non corrispondono a nessuna struttura esistente; parola/e chiave non trovata/e	
Primary Actor	Visitatore	
Trigger	Nessuno	
Scenario di successo		
Step #	Visitatore	Sistema
1		Mostra M0 con la lista ordinata per “Consigliati” e “Punteggio minimo” uguale a 0
2	Clicca su una struttura presente nella lista	
3		Mostra M8
Subvariation #1: Visualizza per “Punteggio”		
2.1	Clicca su “Filtri”	
3.1		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.1	Seleziona “Punteggio” nella Combobox2, specifica un valore e clicca sul tasto “Filtri”	
5.1		Mostra M0 con la lista ordinata secondo le preferenze indicate al passo 4.1
6.1	Clicca su una struttura presente nella lista	
7.1		Mostra M8
Subvariation #2: Visualizza per “Range di Prezzo”		
2.2	Clicca su “Filtri”	

3.2		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.2	Seleziona “Range di Prezzo” e clicca sul tasto “Filtri”	
5.2		Mostra M0 con la lista filtrata secondo le preferenze indicate al passo 6.2
6.2	Clicca su una struttura presente nella lista	
7.2		Mostra M8
Subvariation #3: Visualizza per “Vicinanza”		
2.3	Clicca su “Filtri”	
3.3		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.3	Mette la spunta a “Vicinanza”, indica un valore in metri e clicca sul tasto “Filtri”	
5.3		Mostra M0 con la lista filtrata secondo le preferenze indicate al passo 4.3
6.3	Clicca su una struttura presente nella lista	
7.3		Mostra M8
Subvariation #4: Visualizza per parola chiave		
2.4	Clicca su “Filtri”	
3.4		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.4	Scrivi una o più parole chiave nella barra di ricerca e preme "invio"	
5.4		Mostra M0 con la lista delle strutture che contengono la/le

		parola/e chiave ricercata/e
6.4	Clicca su una struttura presente nella lista	
7.4		Mostra M8
Subvariation #5: Visualizza per parola chiave e filtri		
2.5	Clicca su "Filtri"	
3.5		Mostra M1 con "Consigliati" e "Punteggio minimo" uguale a 0
4.5	Scrivo una o più parole chiave nella barra di ricerca e preme "invio"	
5.5		Mostra M0 con la lista delle strutture che contengono la/le parola/e chiave ricercata/e
6.5	Clicca su "Filtri"	
7.5		Mostra M1 con la/e parola/e chiave inserita/e al passo 4.5
8.5	Inserisce le preferenze per i filtri	
9.5		Mostra M0 con la lista delle strutture che contengono la/le parola/e chiave ricercata/e e rispettano le preferenze dei filtri scelte nel passo 8.5
10.5	Clicca su una struttura presente nella lista	
11.5		Mostra M8
Extension #1: Nessun risultato per i filtri		
2.1	Clicca su "Filtri"	

3.1		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.1	Seleziona i filtri desiderati e clicca sul tasto “Filtri”	
5.1		Mostra M404 con messaggio “Nessun risultato per i filtri indicati”
Extension #2: Nessuna struttura rispetta i termini di ricerca		
2.2	Clicca su “Filtri”	
3.2		Mostra M1 con “Consigliati” e “Punteggio minimo” uguale a 0
4.2	Scrivo una o più parole chiave nella barra di ricerca e preme “Invio”	
5.2		Mostra M404 con messaggio “Nessun risultato per <<parola chiave>>” e “parola chiave” uguale alla parola scritta al passo 4.2

Use Case	Visualizza Recensioni
Preconditions	Nessuna

Success End Condition	Il visitatore visualizza le recensioni della struttura di interesse	
Failed End Condition	Parola/e chiave non trovata/e	
Primary Actor	Visitatore	
Trigger	Il visitatore clicca sul nome della struttura presente nella schermata M0 e scorre fino alla sezione delle recensioni	
Scenario di successo		
Step #	Visitatore	Sistema
1	Clicca sul nome della struttura presente nella schermata M0 e scorre fino alla sezione delle recensioni	
2		Mostra M9 con lista ordinata secondo la preferenza di default (“Data decrescente”)
Subvariation #1: Scelta di un ordinamento		
2.1	Clicca sul bottone “Filtri”	
3.1		Mostra M10 con la preferenza di default (“Data decrescente”)
4.1	Seleziona una preferenza in “Ordina per” e clicca sul bottone “Filtri”	
5.1		Mostra M9 con la lista di recensioni ordinata secondo la preferenza indicata al passo 4.1
Subvariation #2: Ricerca per parola chiave		
2.2	Clicca sul bottone “Filtri”	
3.2		Mostra M10 con la preferenza di default (“Data decrescente”)
4.2	Scriva una o più parole chiave nella barra di ricerca e preme "invio"	
5.2		Mostra M9 con la lista

		delle recensioni che contengono la/le parola/e chiave ricercata/e, ordinata secondo la preferenza di default ("Data decrescente")
Subvariation #3: Ricerca per parola chiave e filtro		
2.3	Clicca sul bottone "Filtri"	
3.3		Mostra M10 con la preferenza di default ("Data decrescente")
4.3	Scrivi una o più parole chiave nella barra di ricerca e preme "invio"	
5.3		Mostra M9 con la lista delle recensioni che contengono la/le parola/e chiave ricercata/e, ordinata secondo la preferenza di default ("Data decrescente")
6.3	Clicca sul bottone "Filtri"	
7.3		Mostra M10 con la/e parola/e chiave inserita/e nel passo 4.3
8.3	Seleziona una preferenza in "Ordina per"	
9.3		Mostra M9 con la lista di recensioni ordinata secondo la preferenza indicata e contenenti la/le parola/e chiave inserita/e al passo 4.3
Extension #1: Nessuna recensione rispetta i termini di ricerca		
2.1	Clicca sul bottone "Filtri"	
3.1		Mostra M10 con la preferenza di default ("Data decrescente")
4.1	Scrivi una o più parole chiave nella barra di ricerca e preme "invio"	
5.1		Mostra M11 con "parola

		chiave" uguale alla/e parola/e chiave inserita/e al passo 4.2
--	--	---

Use Case	Aggiunge Recensione	
Preconditions	Nessuna	
Success End Condition	Il visitatore loggato inserisce una recensione per la struttura di interesse	
Failed End Condition	Il visitatore loggato annulla l’operazione	
Primary Actor	Visitatore Loggato	
Trigger	Il visitatore clicca sul nome della struttura presente nella schermata M0, scorre fino alla sezione delle recensioni e clicca su “Scrivi una recensione”	
Scenario di successo		
Step #	Visitatore Loggato	Sistema
1	Clicca sul nome della struttura presente nella schermata M0, scorre fino alla sezione delle recensioni e clicca su “Scrivi una recensione”	
2		Mostra M12 con il tasto “Pubblica” disattivato e “Identità Visitatore” uguale a username o nome e cognome del visitatore (secondo le preferenze indicate durante la registrazione)
3	Inserisce “Stelle”, “Titolo” e “Descrizione”	
4		Abilita il tasto “Pubblica”
5	Preme il tasto “Pubblica”	
6		Mostra M13
Extension #1: “Aggiunge Recensione” annullato		
3.1	Clicca sul bottone “Annulla”	
4.1		Mostra M9

Use Case	Visualizza Struttura Mappa		
Preconditions	Nessuna		
Success End Condition	Il visitatore visualizza la struttura di interesse sulla mappa		
Failed End Condition	La struttura di interesse non esiste		
Primary Actor	Visitatore		
Secondary Actor	API Mappe		
Trigger	Nessuno		
Scenario di successo			
Step #	Visitatore	Sistema	API Mappe
1		Mostra M0 con la lista ordinata per “Consigliati” e “Punteggio minimo” uguale a 0	
2	Clicca sul tasto “Visualizza su Mappa” di una struttura presente nella lista		
3			Fornisce la mappa relativa alla posizione della struttura selezionata al passo 2
4		Mostra M14 relativa alla struttura scelta al passo 2 con relativo marker e popup informativo	
Subvariation #1:Struttura nelle vicinanze			
2.1	Clicca sul bottone “Mappa”		

3.1			Fornisce la mappa relativa alla posizione del visitatore
4.1		Mostra M14 con i marker delle eventuali strutture presenti nelle vicinanze	
5.1	Clicca sul marker di una struttura		
6.1		Mostra M14 relativa alla struttura scelta al passo 5.1 con popup informativo	
Subvariation #2: Struttura nelle vicinanze per range di prezzo			
2.2	Clicca sul bottone "Mappa"		
3.2			Fornisce la mappa relativa alla posizione del visitatore
4.2		Mostra M14 con i marker delle eventuali strutture presenti nelle vicinanze	
5.2	Clicca sul tasto "Filtri"		
6.2		Mostra M15 con "Punteggio Minimo" uguale a zero e "Range di Prezzo" uguale a "Qualunque"	
7.2	Seleziona "Categoria" e "Range di Prezzo" e clicca su "Filtri"		
8.2		Mostra M14 con i marker delle strutture che rispettano i filtri	
9.2	Clicca sul marker di una struttura		
10.2		Mostra M14 relativa alla struttura scelta al passo 7.2 con popup informativo	

Subvariation #3: Struttura nelle vicinanze per parola chiave			
2.3	Clicca sul bottone "Mappa"		
3.3			Fornisce la mappa relativa alla posizione del visitatore
4.3		Mostra M14 con i marker delle eventuali strutture presenti nelle vicinanze	
5.3	Clicca sul tasto "Filtri"		
6.3		Mostra M15 con "Punteggio Minimo" uguale a zero e "Range di Prezzo" uguale a "Qualunque"	
7.3	Scrivo una o più parola/e chiave nella barra di ricerca e preme "invio"		
8.3		Mostra M14 con il/i marker della/e struttura/e che rispetta/no la ricerca	
9.3	Clicca sul marker di una struttura		
10.3		Mostra M14 relativa alla struttura scelta al passo 7.3 con popup informativo	
Extension #1: Nessuna struttura rispetta i termini di ricerca			
2.1	Clicca sul bottone "Mappa"		
3.1			Fornisce la mappa relativa alla posizione del visitatore
4.1		Mostra M14 con i marker	

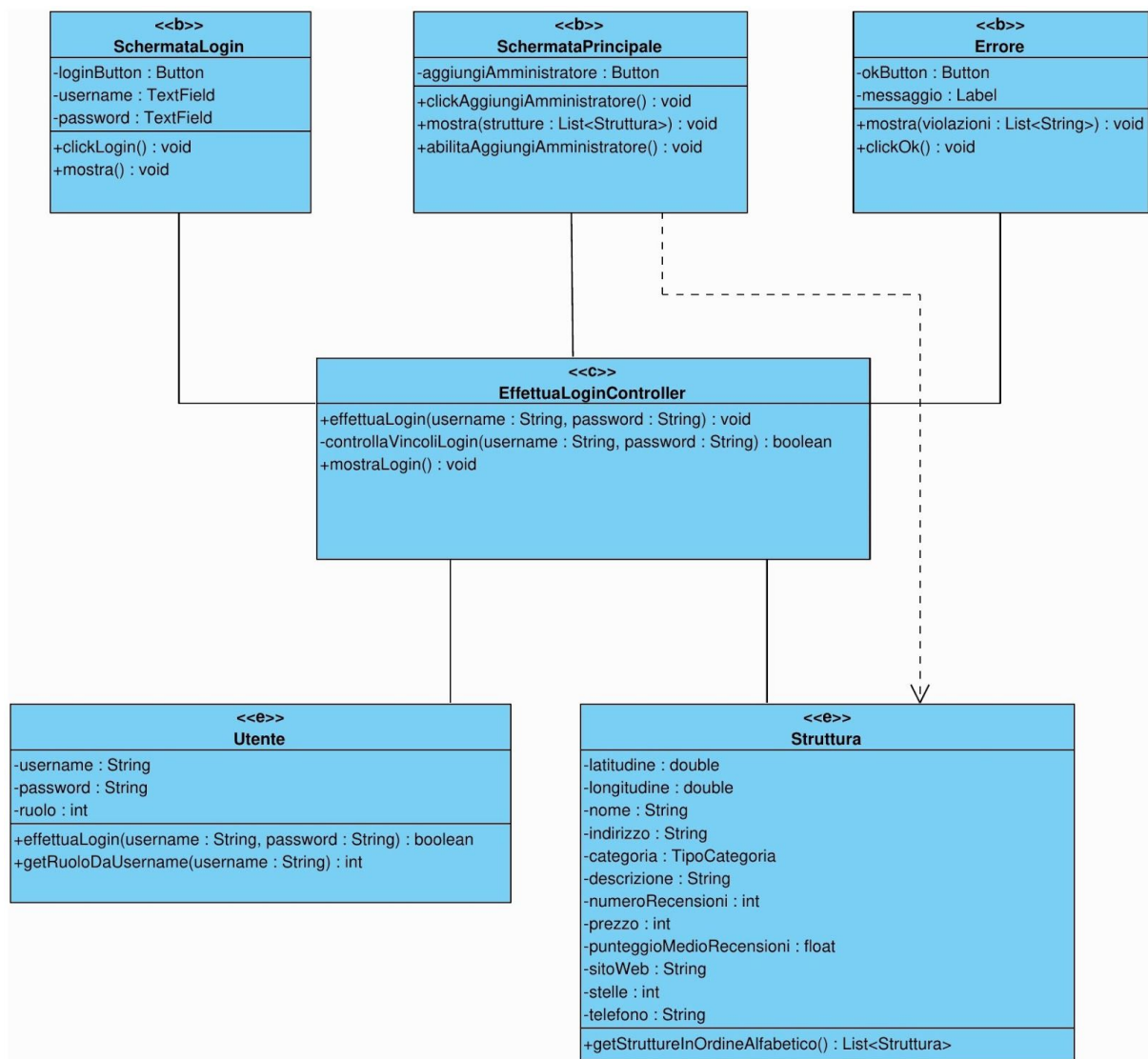
		delle eventuali strutture presenti nelle vicinanze	
5.1	Clicca sul tasto "Filtri"		
6.1		Mostra M15 con "Punteggio Minimo" uguale a zero e "Range di Prezzo" uguale a "Qualunque"	
7.1	Scrivi una o più parola/e chiave nella barra di ricerca e preme "invio"		
8.1		Mostra M16 con "input" uguale alla parola/e chiave ricercata/e	
Extension #2: Nessuna struttura rispetta i filtri			
2.1	Clicca sul bottone "Mappa"		
3.1		Contatta il provider di mappe	
4.1			Fornisce la mappa relativa alla posizione del visitatore
5.1		Mostra M14 con i marker delle eventuali strutture presenti nelle vicinanze	
6.1	Clicca sul tasto "Filtri"		
7.1		Mostra M15 con "Punteggio Minimo" uguale a zero e "Range di Prezzo" uguale a "Qualunque"	
8.1	Seleziona le preferenze per i filtri e clicca su "Filtri"		

9.1		Mostra M16 con "input" uguale a "i filtri indicati"	
-----	--	---	--

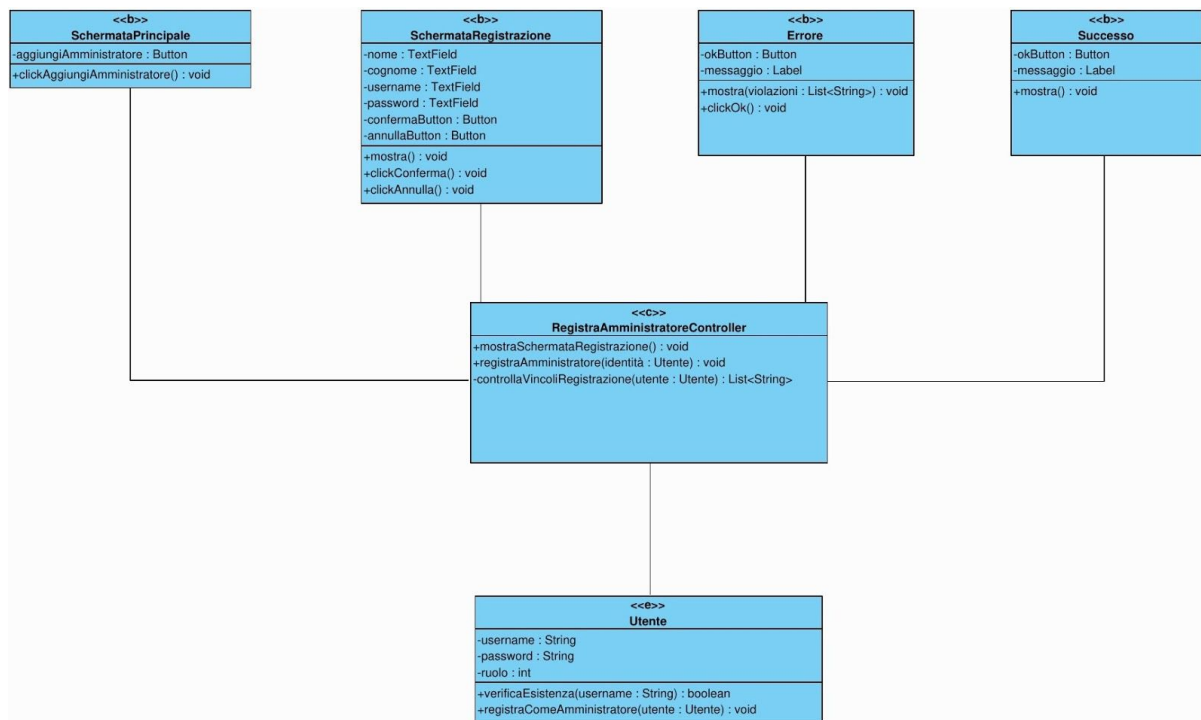
2.4 Modelli di Dominio

2.4.1 Class Diagrams

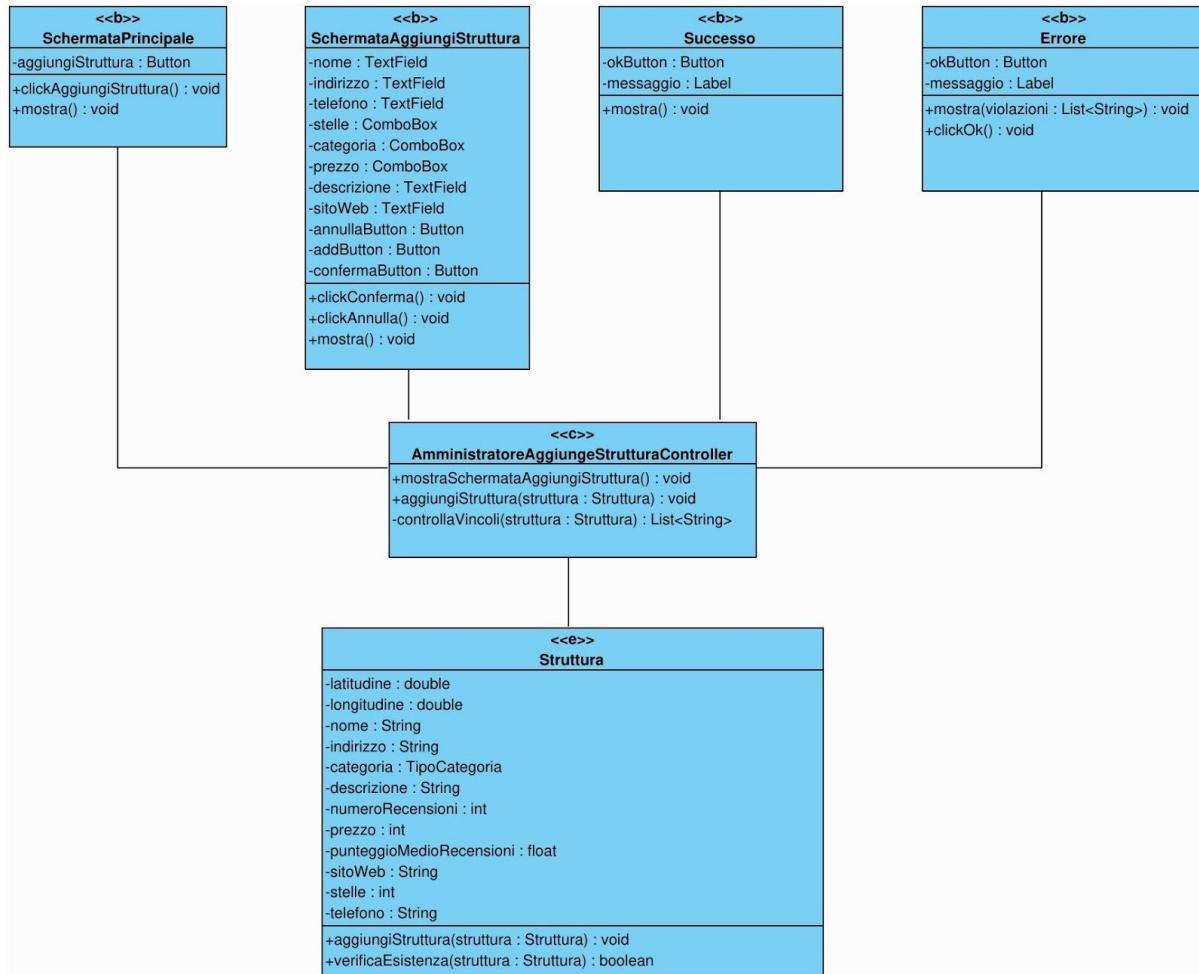
2.4.1.1 Amministratore effettua login



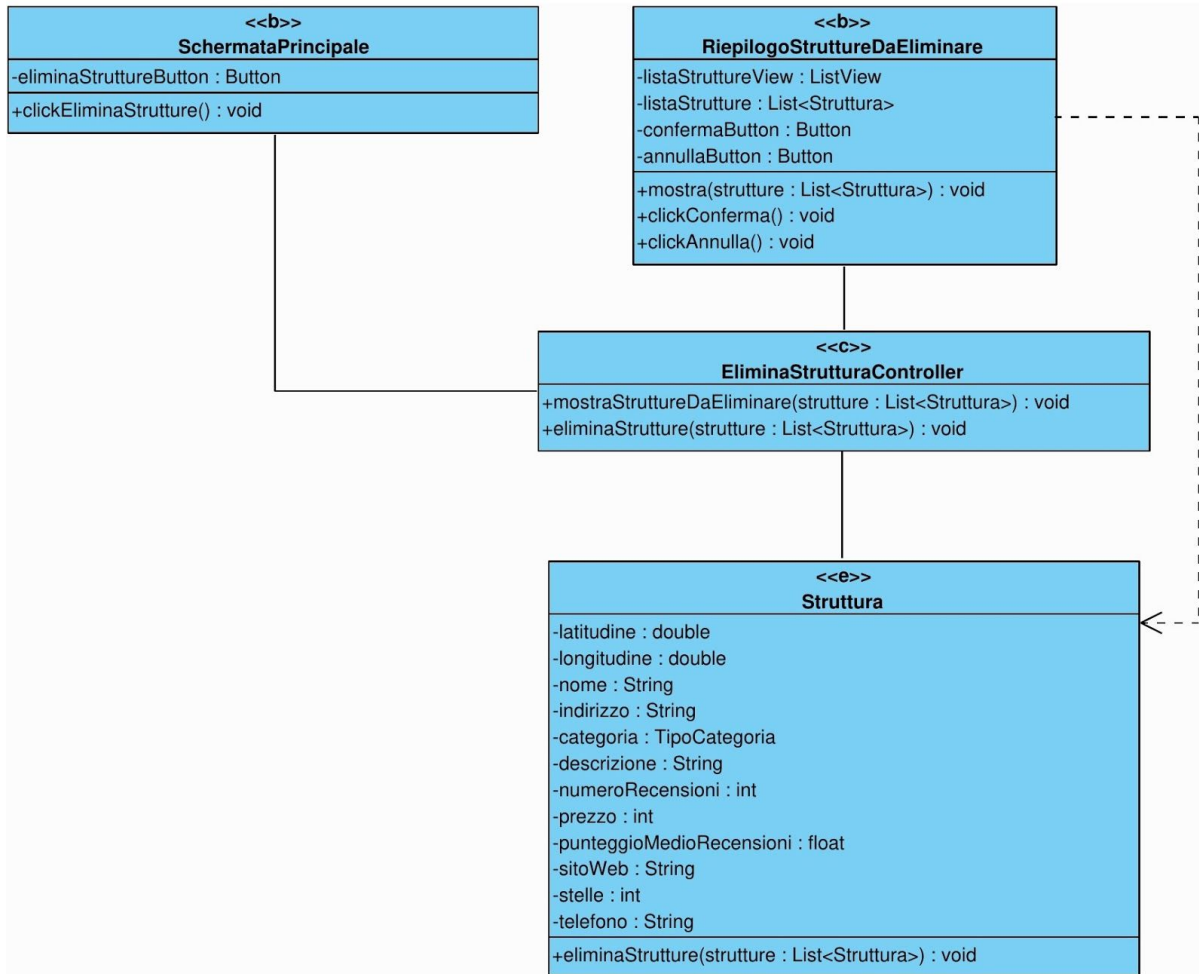
2.4.1.2 SuperAmministratore registra amministratore



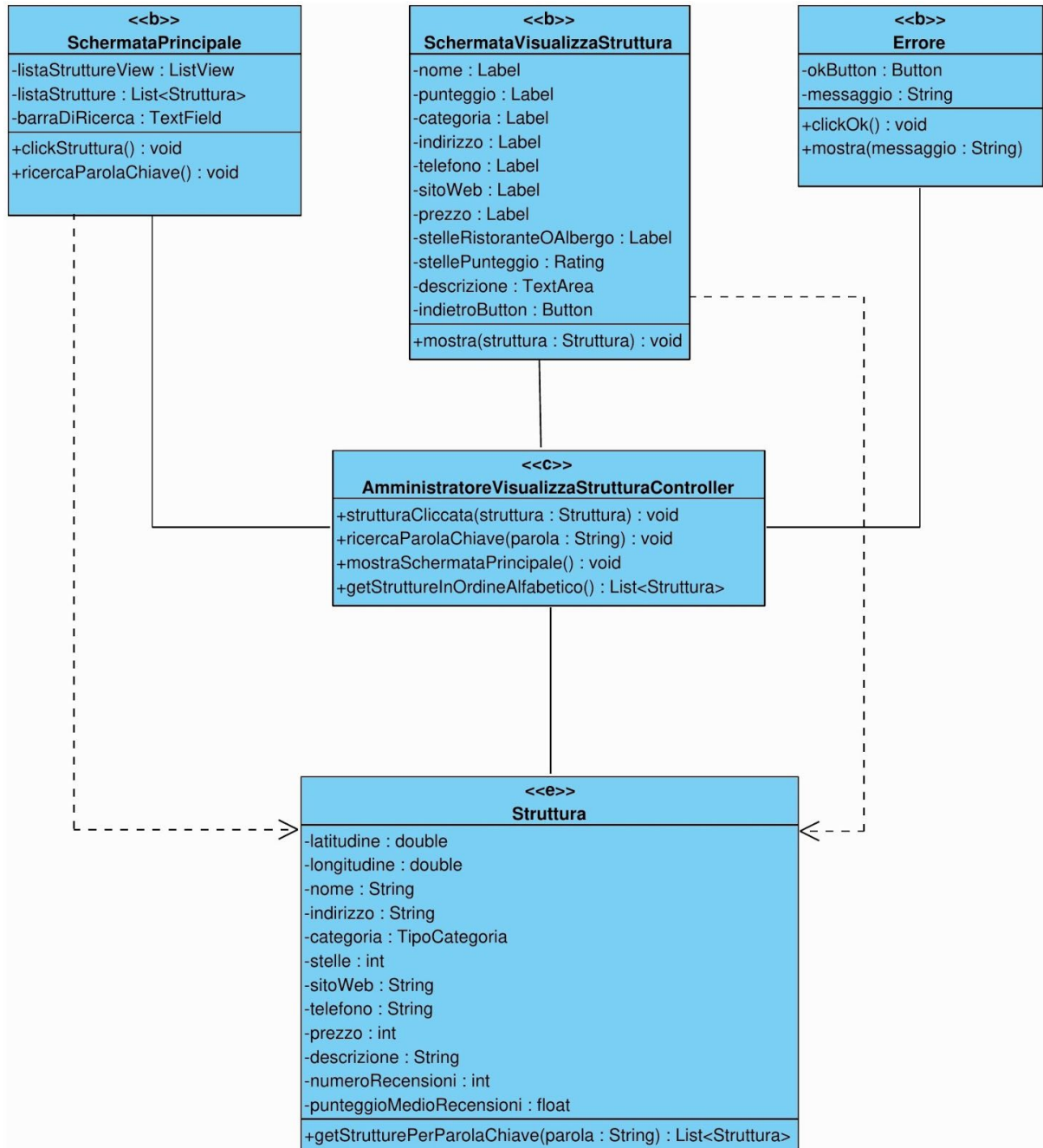
2.4.1.3 AmministratoreLoggato aggiunge struttura



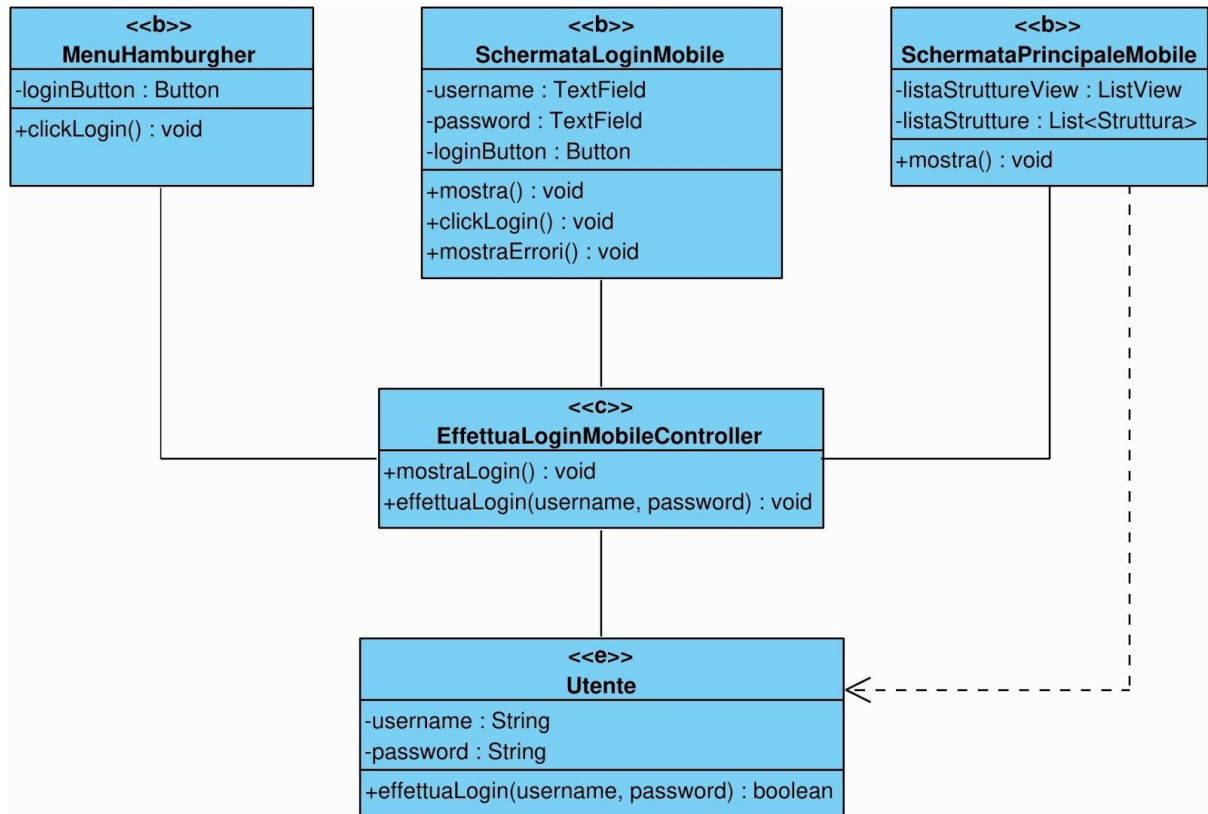
2.4.1.4 AmministratoreLoggato elimina strutture



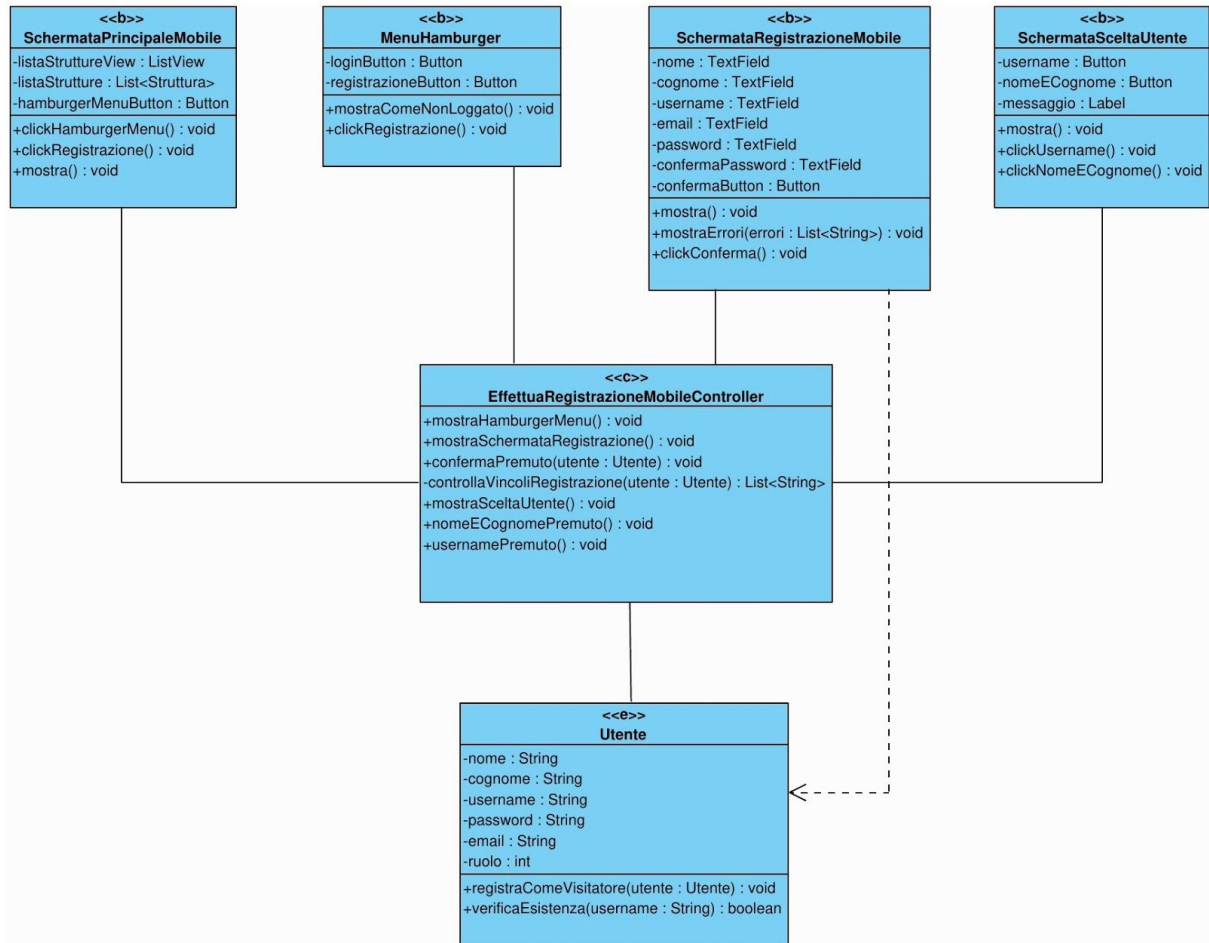
2.4.1.5 AmministratoreLoggato visualizza struttura



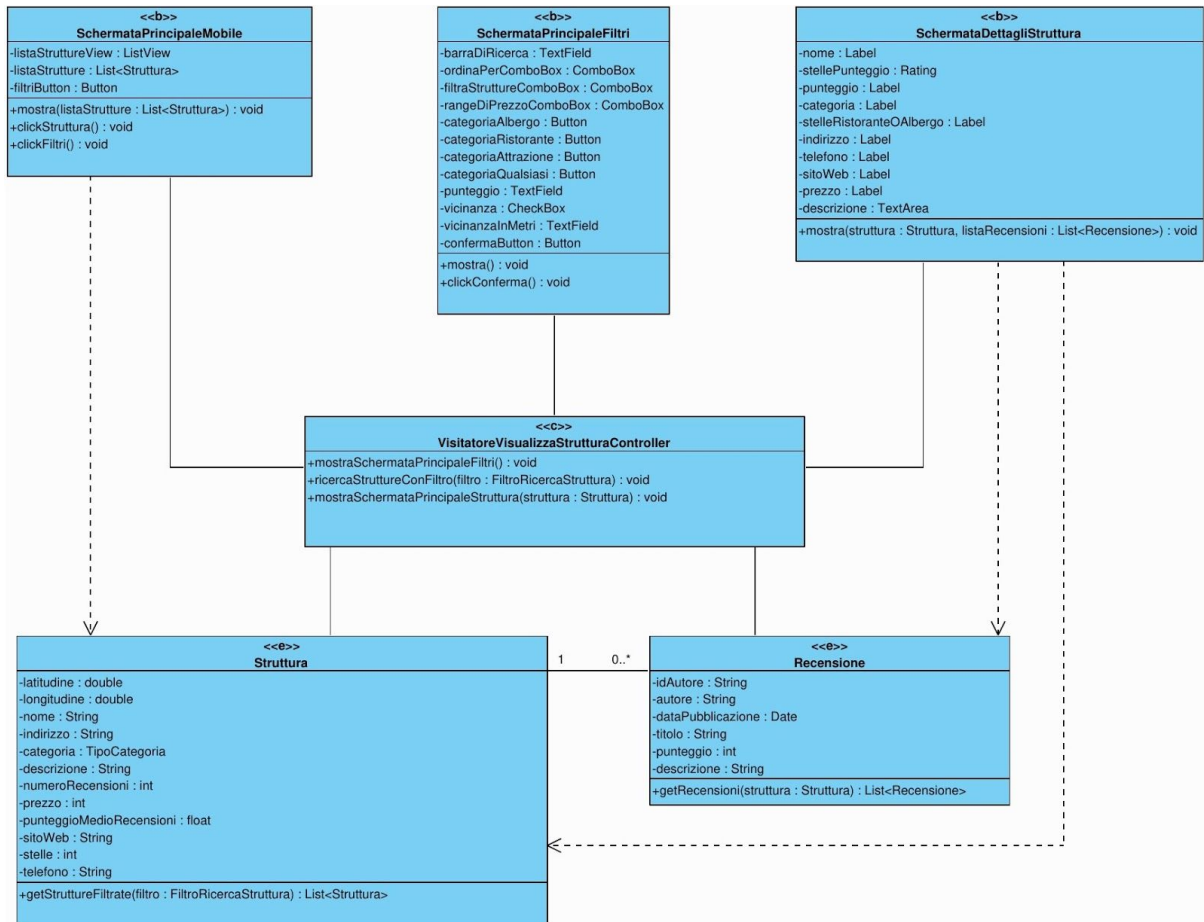
2.4.1.6 Utente effettua login



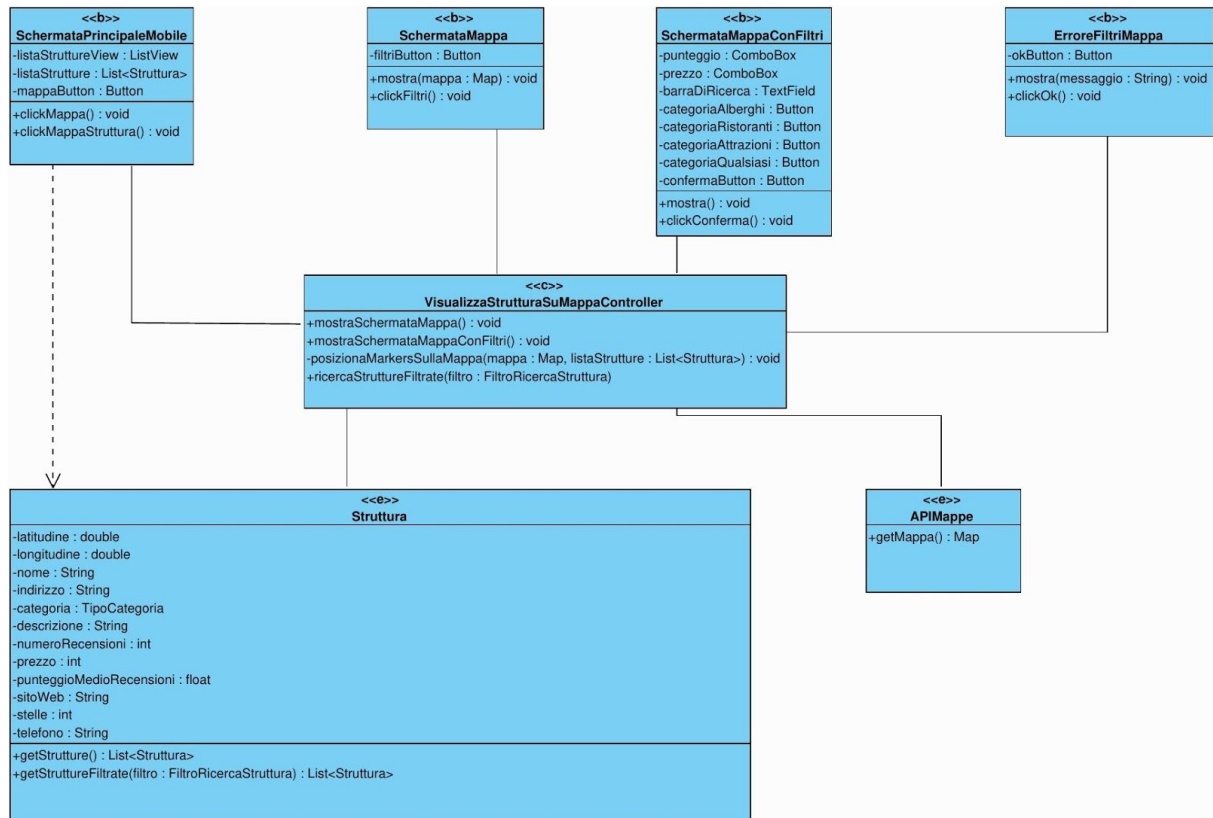
2.4.1.7 Utente effettua registrazione



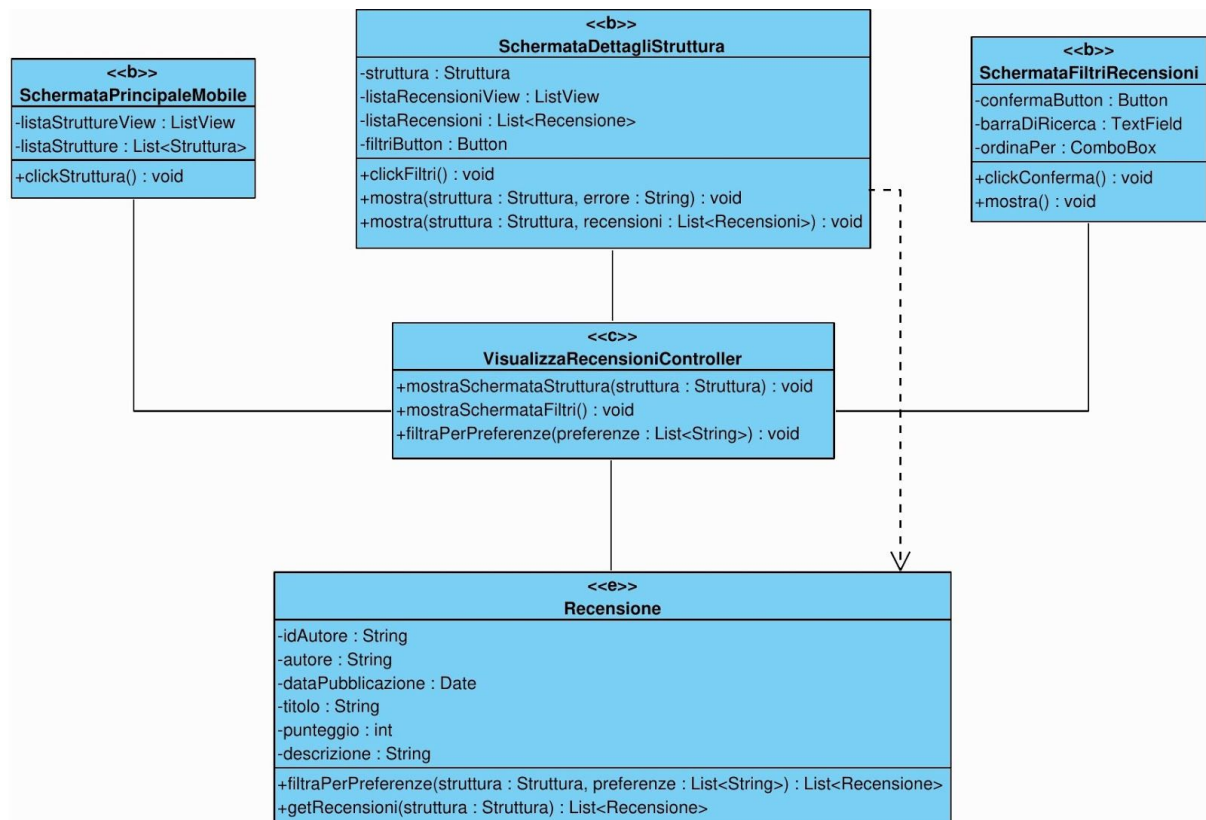
2.4.1.8 Visitatore visualizza struttura



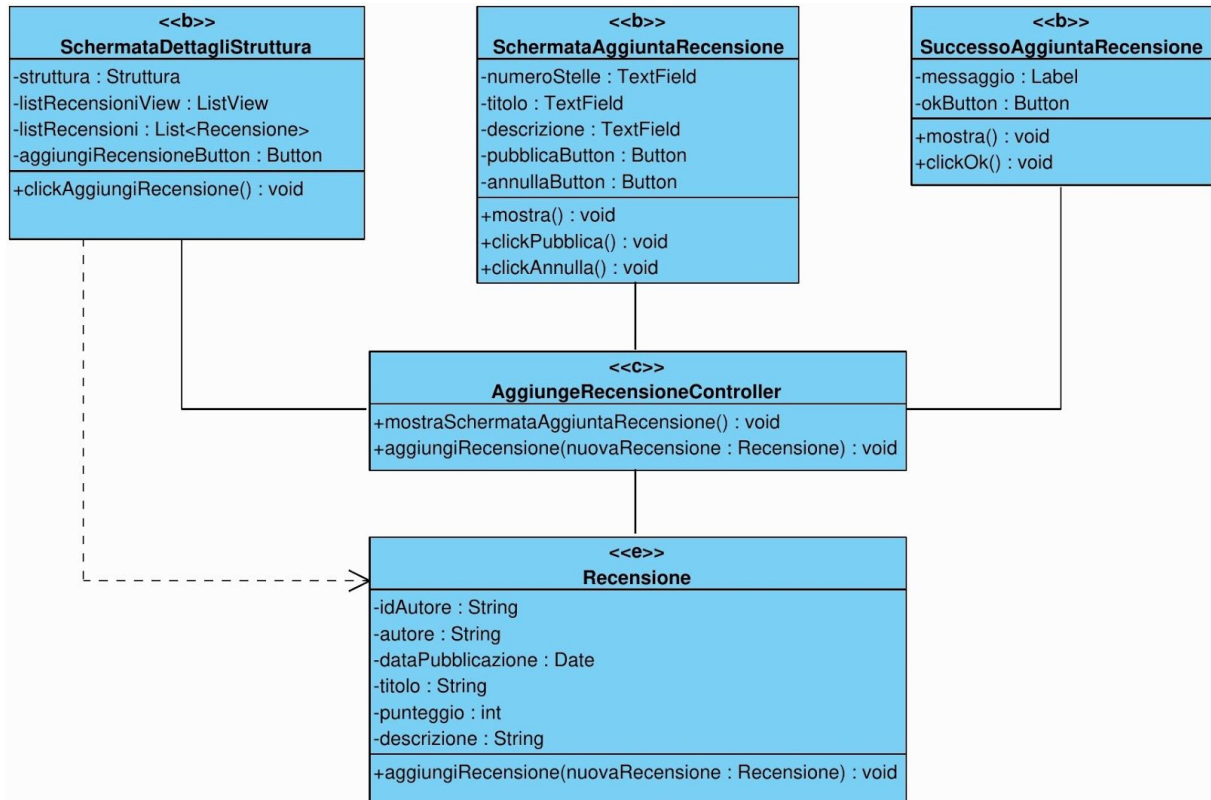
2.4.1.9 Visitatore visualizza struttura su mappa



2.4.1.10 Visitatore visualizza recensioni

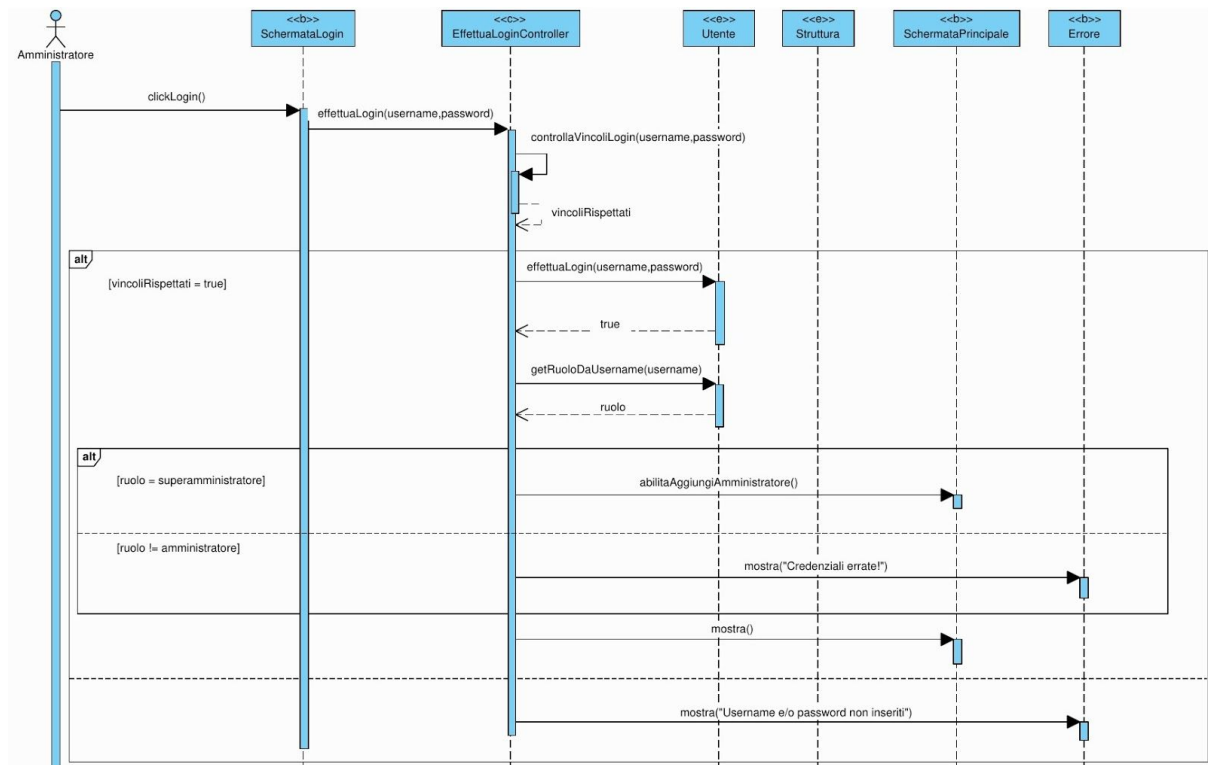


2.4.1.11 Visitatore aggiunge recensione

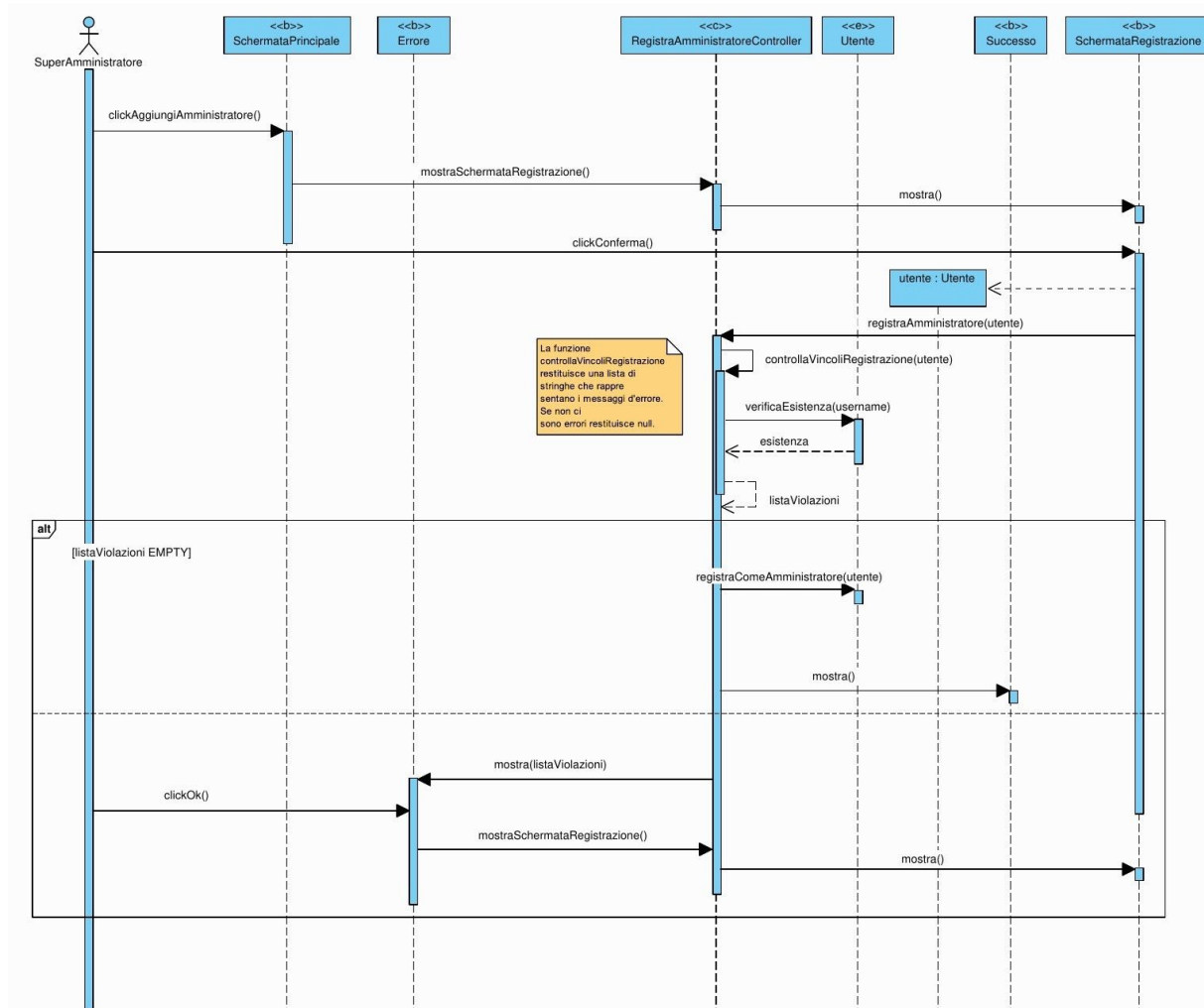


2.4.2 Sequence Diagrams

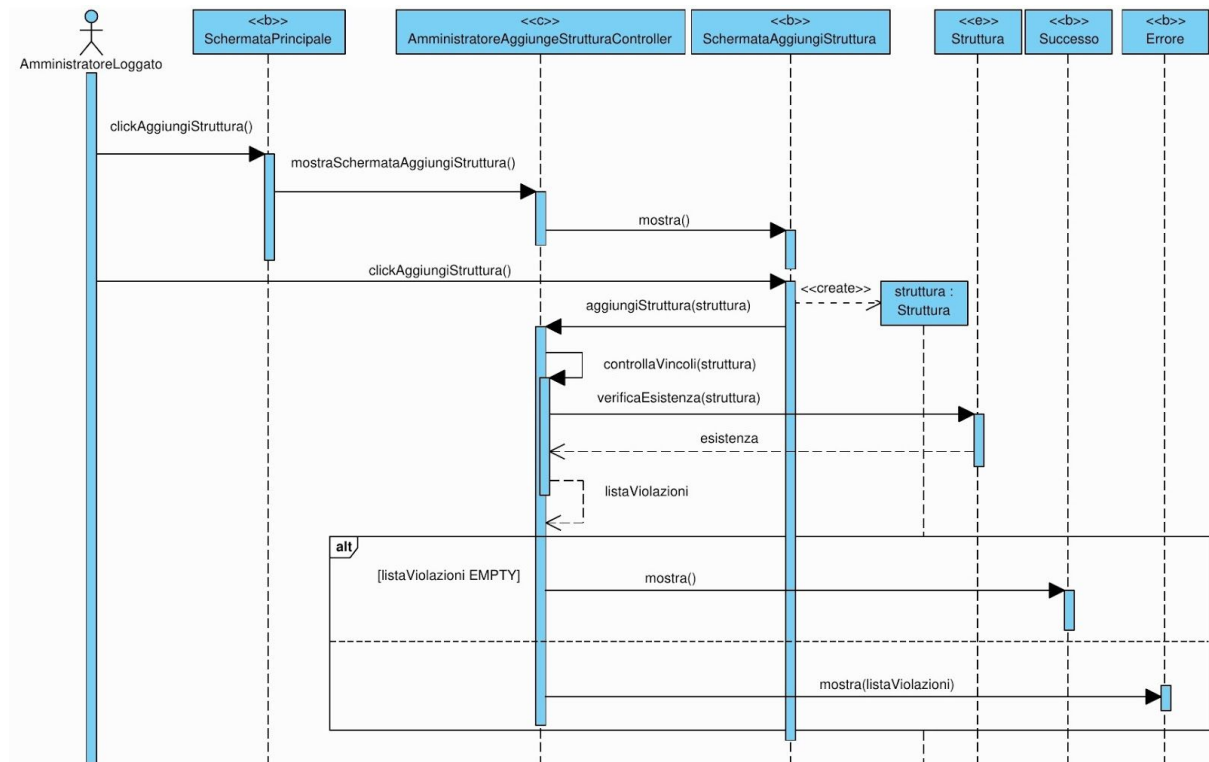
2.4.2.1 Amministratore effettua login



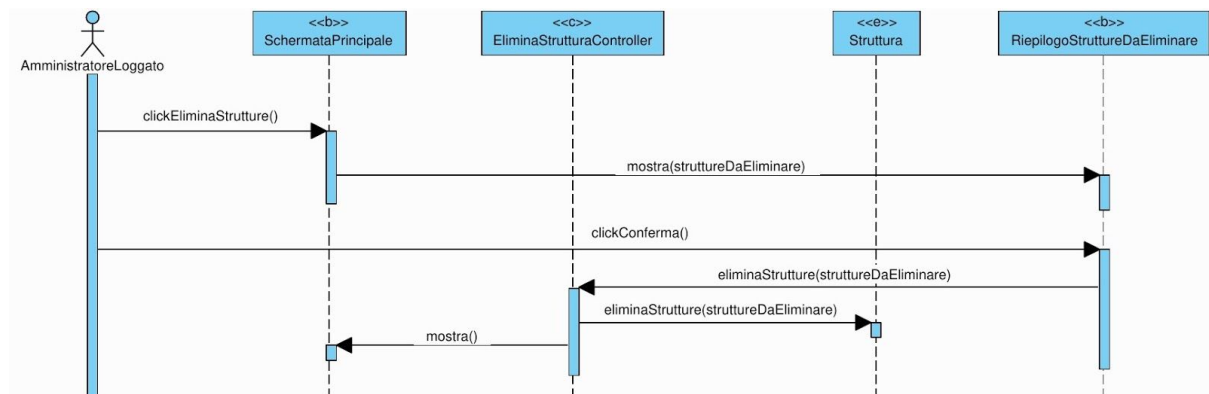
2.4.2.2 SuperAmministratore registra amministratore



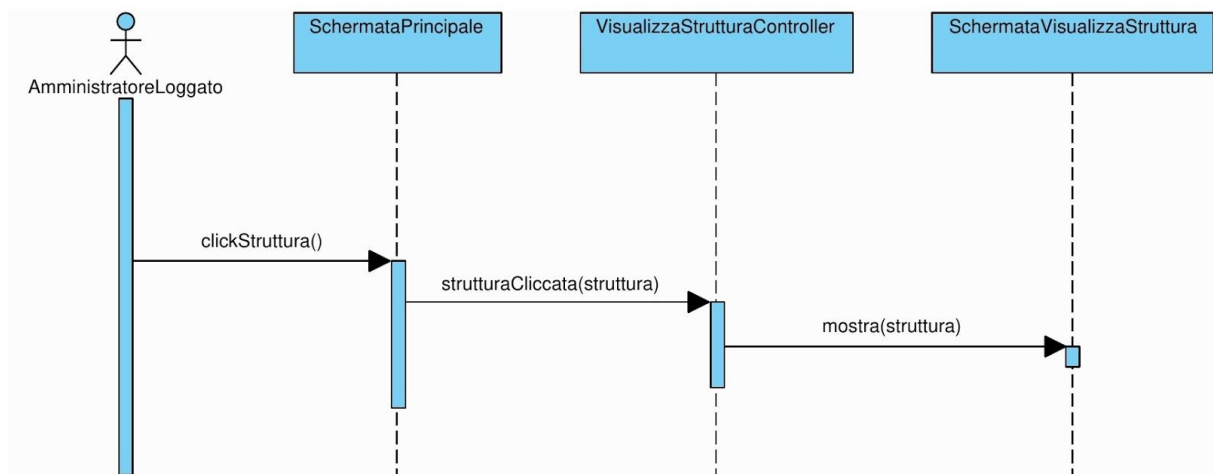
2.4.2.3 AmministratoreLoggato aggiunge struttura



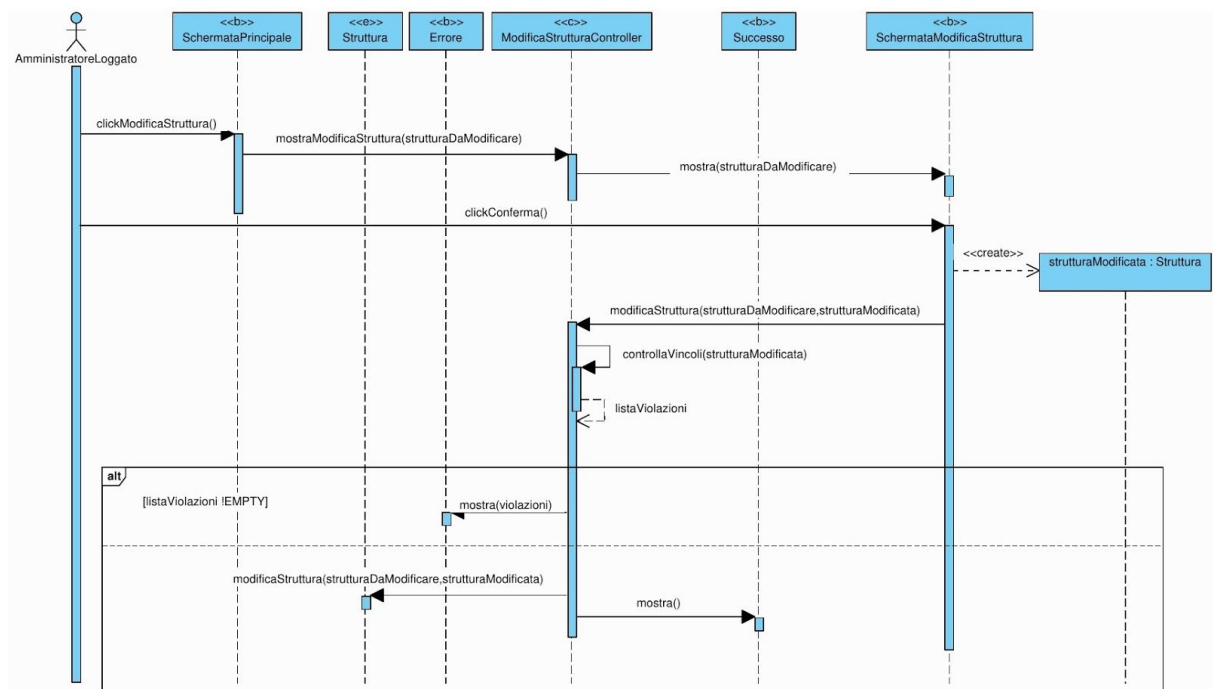
2.4.2.4 AmministratoreLoggato elimina strutture



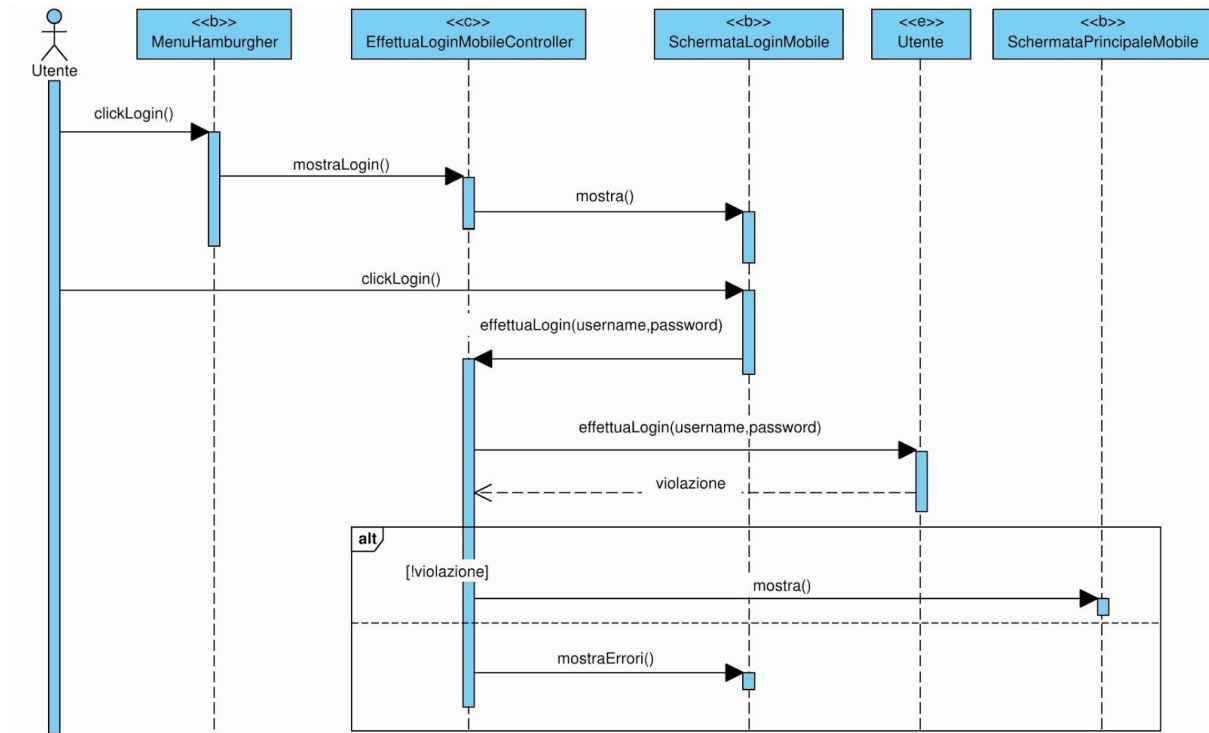
1.4.2.5 AmministratoreLoggato visualizza struttura



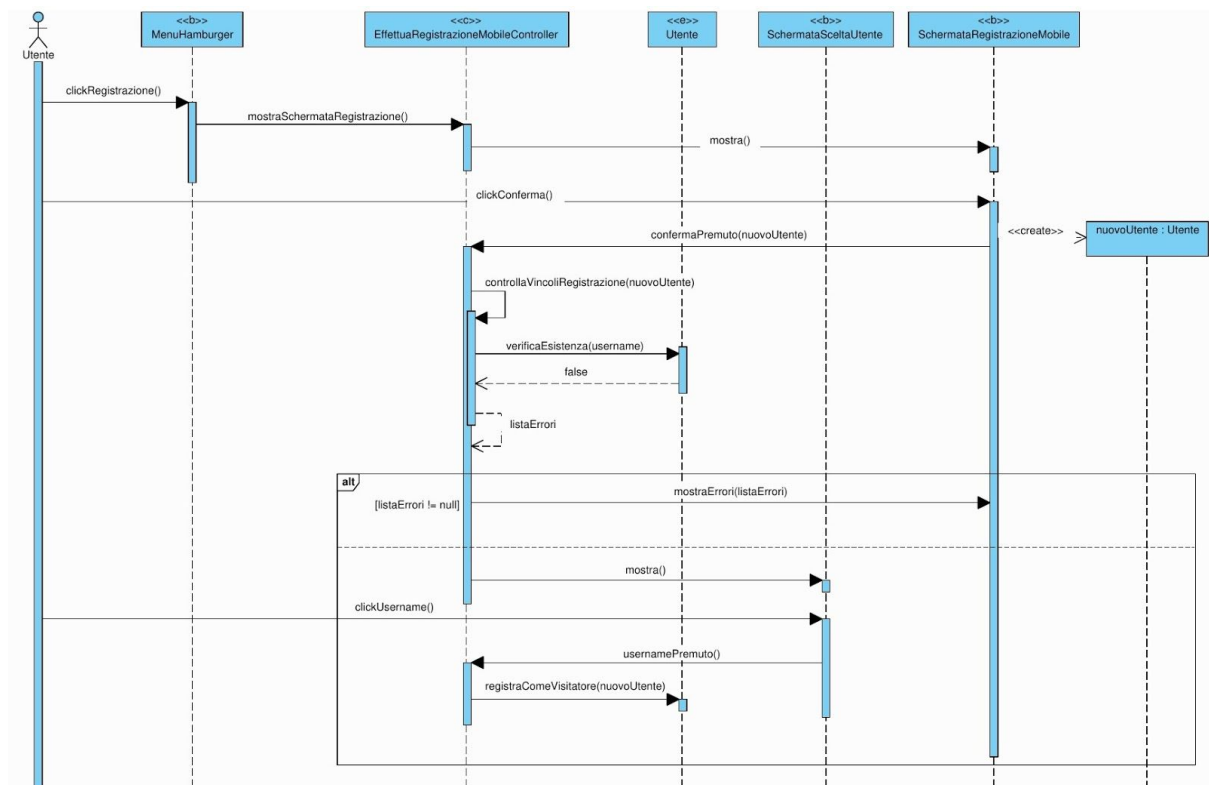
2.4.2.5 AmministratoreLoggato modifica struttura



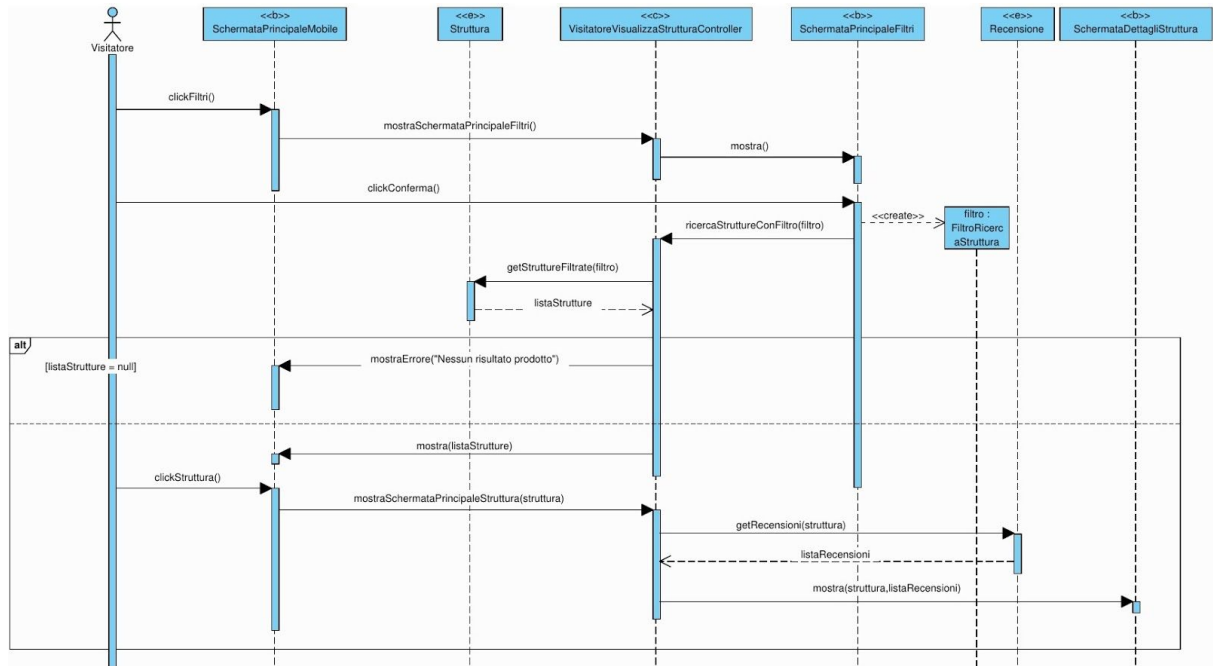
2.4.2.6 Utente effettua login



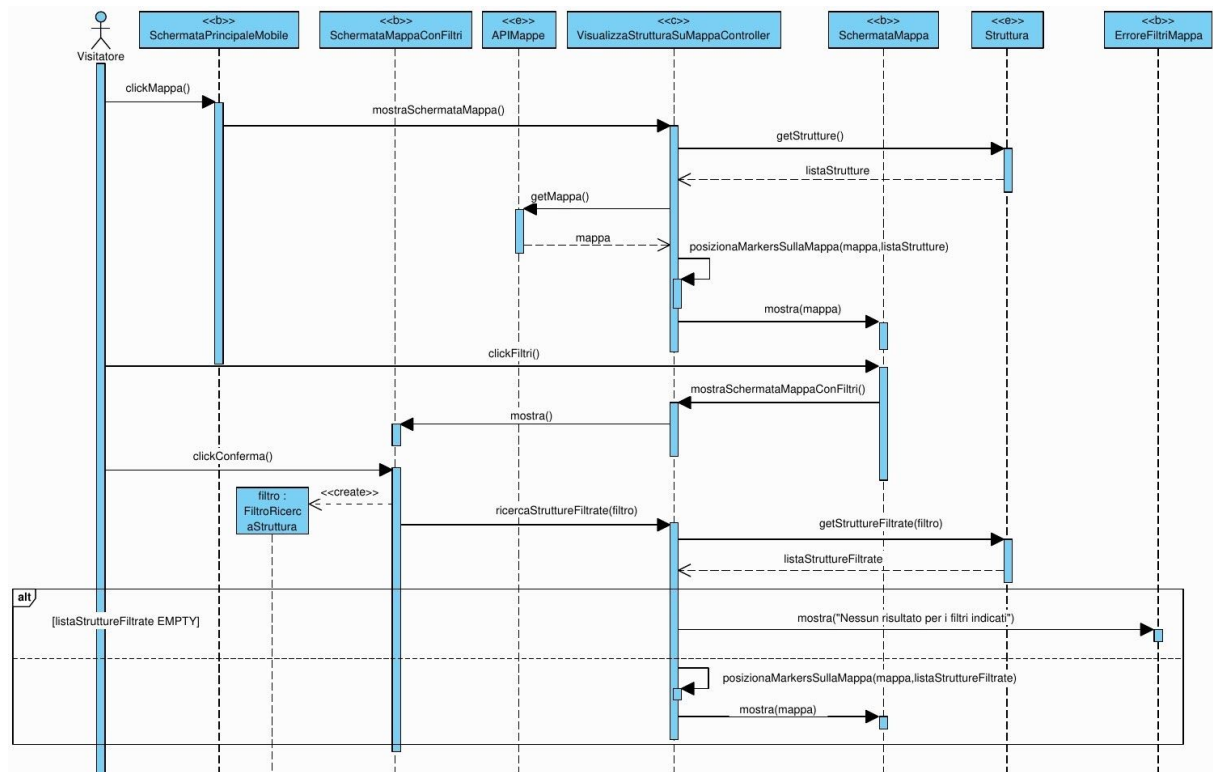
2.4.2.7 Utente effettua registrazione



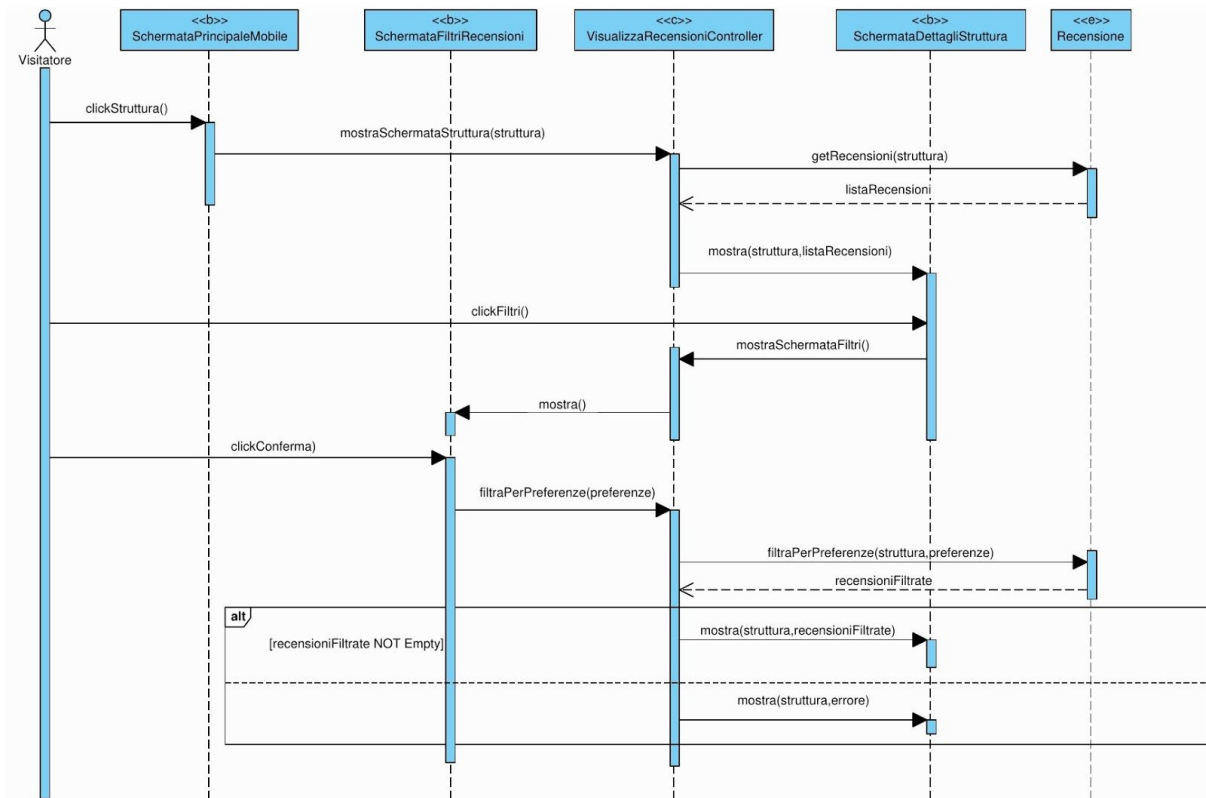
2.4.2.8 Visitatore visualizza struttura



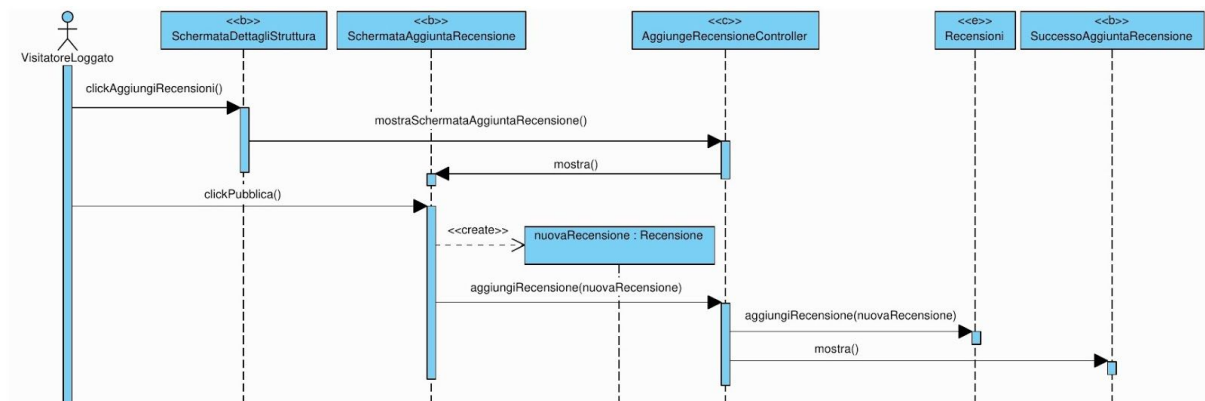
2.4.2.9 Visitatore visualizza struttura su mappa



2.4.2.10 Visitatore visualizza recensioni

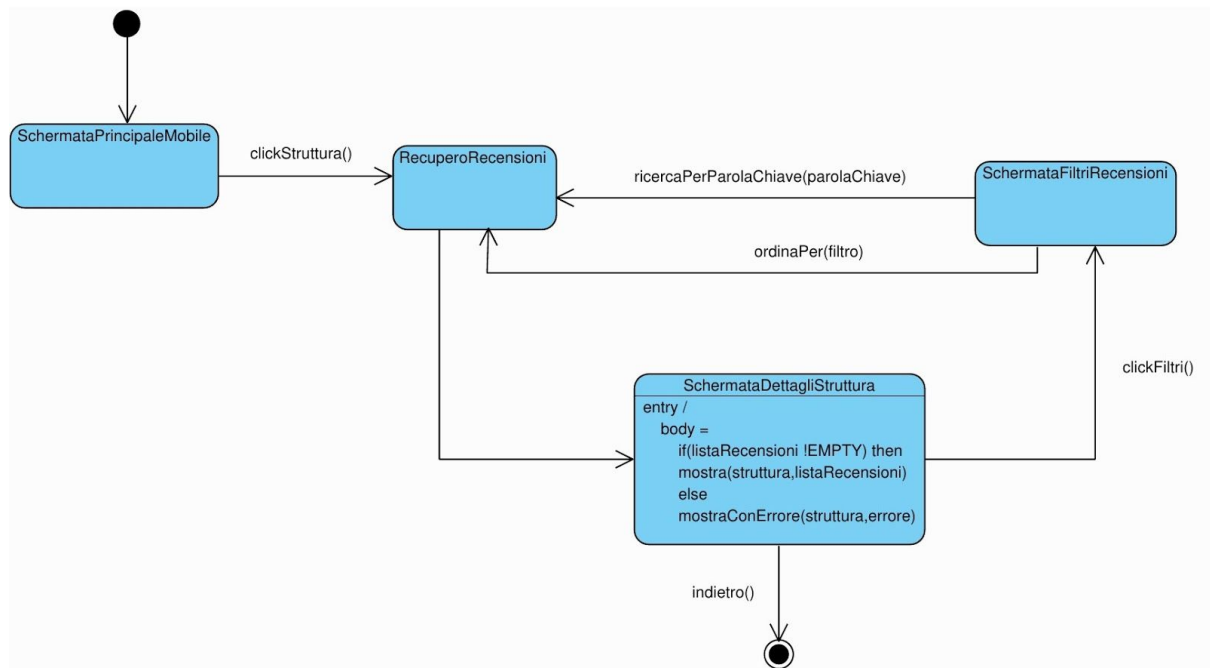


2.4.2.11 Visitatore aggiunge recensione



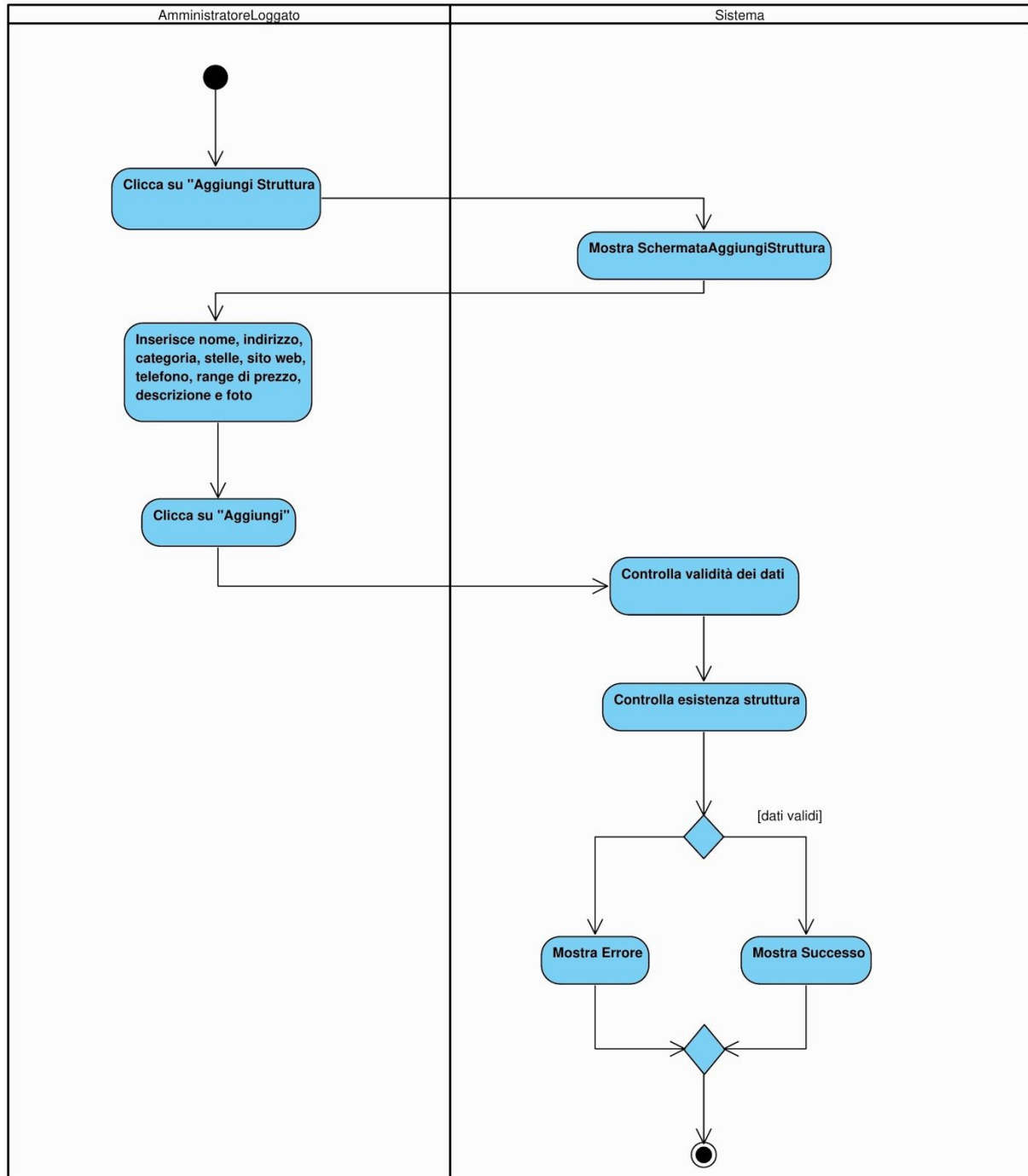
2.4.3 Statechart Diagram

2.4.3.1 Visitatore visualizza recensioni



2.4.4 Activity Diagram

2.4.4.1 AmministratoreLoggato aggiunge struttura



2.4.5 Diagramma di Gantt



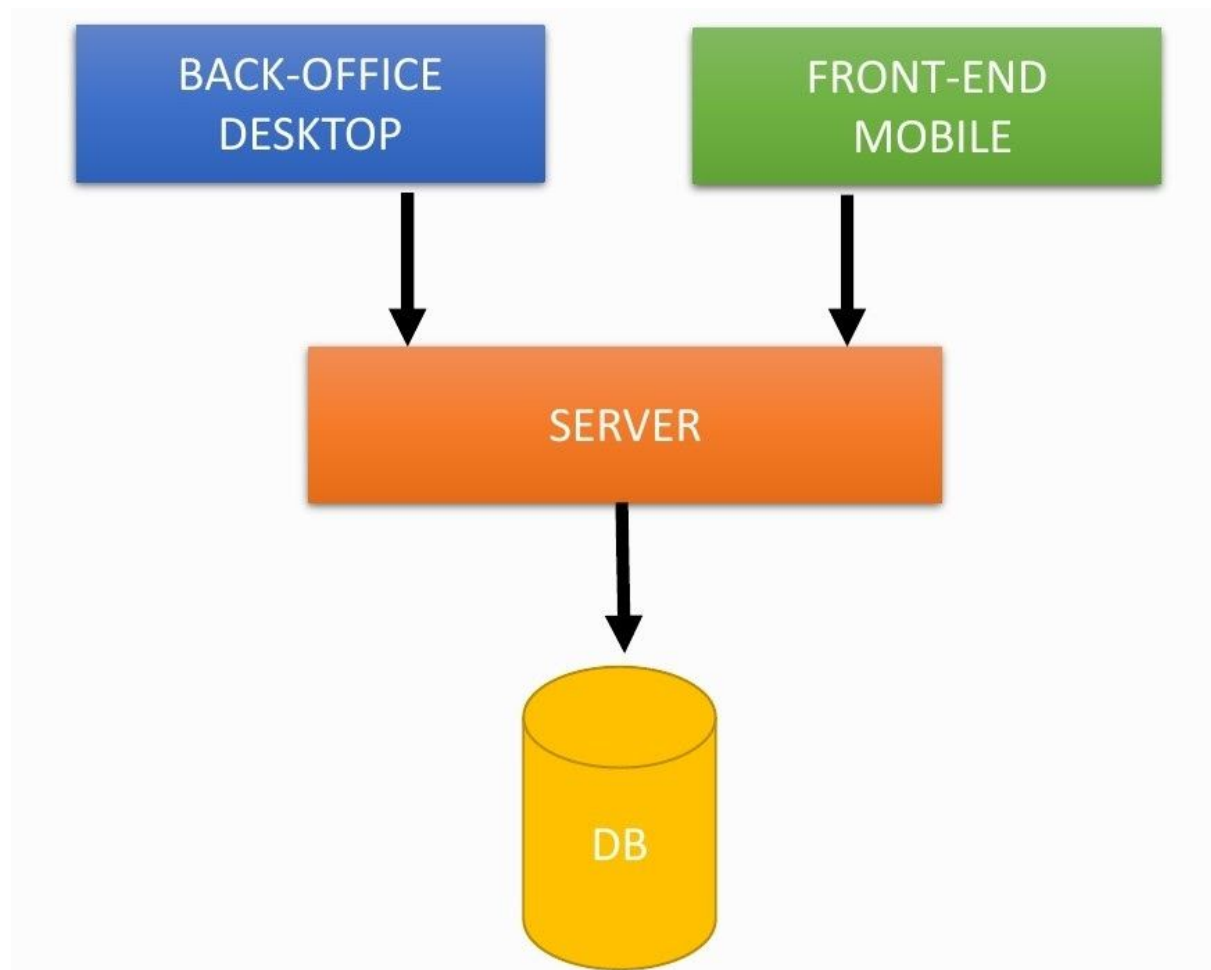
Focus fase di implementazione



3 Documento di Design

3.1 Analisi dell'architettura

Il CV'19 (Consiglia Viaggi) è un sistema informativo composto da 3 componenti fondamentali: un server, un back-office desktop e un front-end mobile (Android).



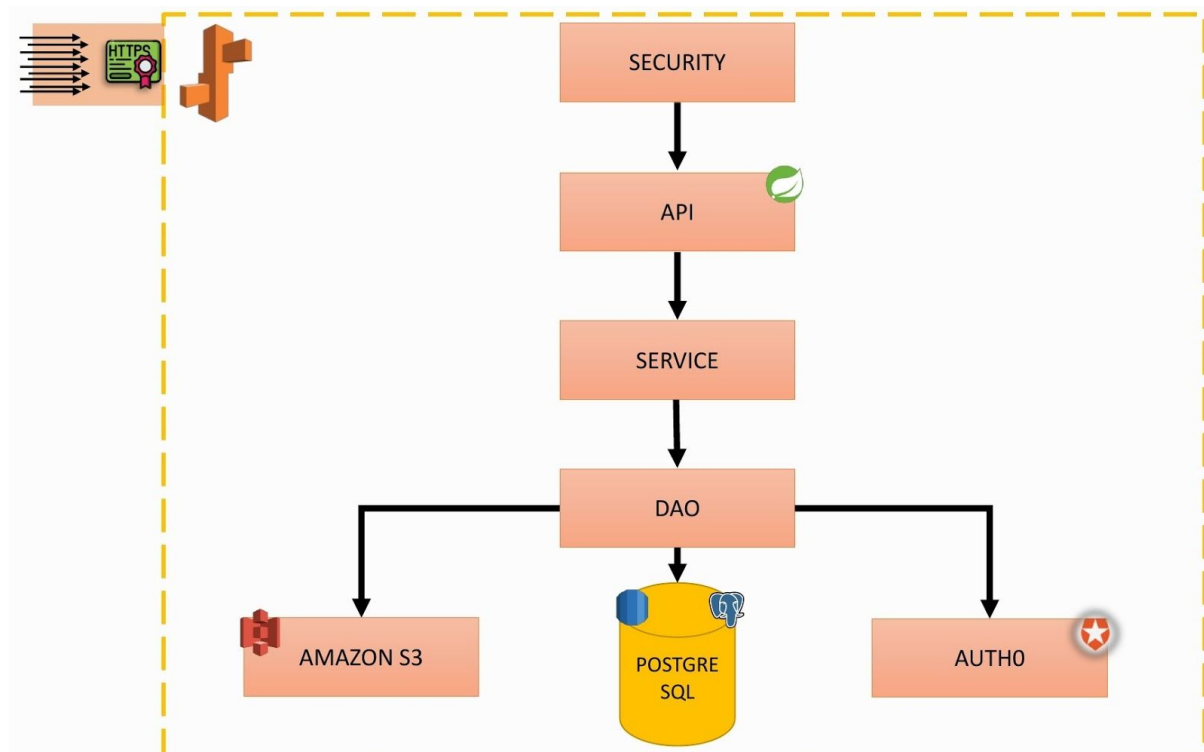
Per rispettare i principi di alta coesione e basso accoppiamento si è scelto di adottare un'architettura chiusa, in cui ogni livello offre servizi a quello soprastante ignorandone la natura. In questo modo, ad esempio, è garantita la possibilità di aggiungere in futuro ulteriori client senza dover modificare i livelli sottostanti oppure di cambiare il metodo di persistenza senza avere ripercussioni sui clients.

Tutti e tre gli applicativi sono stati scritti in linguaggio Java. Seguono la descrizione delle tecnologie adottate per ciascun componente, un paragrafo dedicato ai servizi di terze parti e un paragrafo dedicato alla sicurezza.

3.1.1 Architettura Server

Il server è stato realizzato con il supporto del framework Spring Boot (v.2.2.5). Questa scelta ha consentito di implementare un sistema di API STATEful in maniera molto agile sotto vari punti di vista:

- Routing delle richieste.
- Dependency injection a supporto del pattern MVC.
- Filtri Web, che sono stati utilizzati per la sicurezza e la validazione dei token.



Nota

- **Security** = filtra tutte le richieste http(s) in entrata e verifica se il client richiedente ha i permessi per accedere alle funzionalità richieste.
- **API** = espone le API, ma delega l'elaborazione delle richieste al service.
- **Service** = implementa la logica di business.
- **DAO** = si occupa della persistenza dei dati.

3.1.1.1 Servizi esterni Server

Per questa architettura sono stati utilizzati i seguenti servizi esterni:

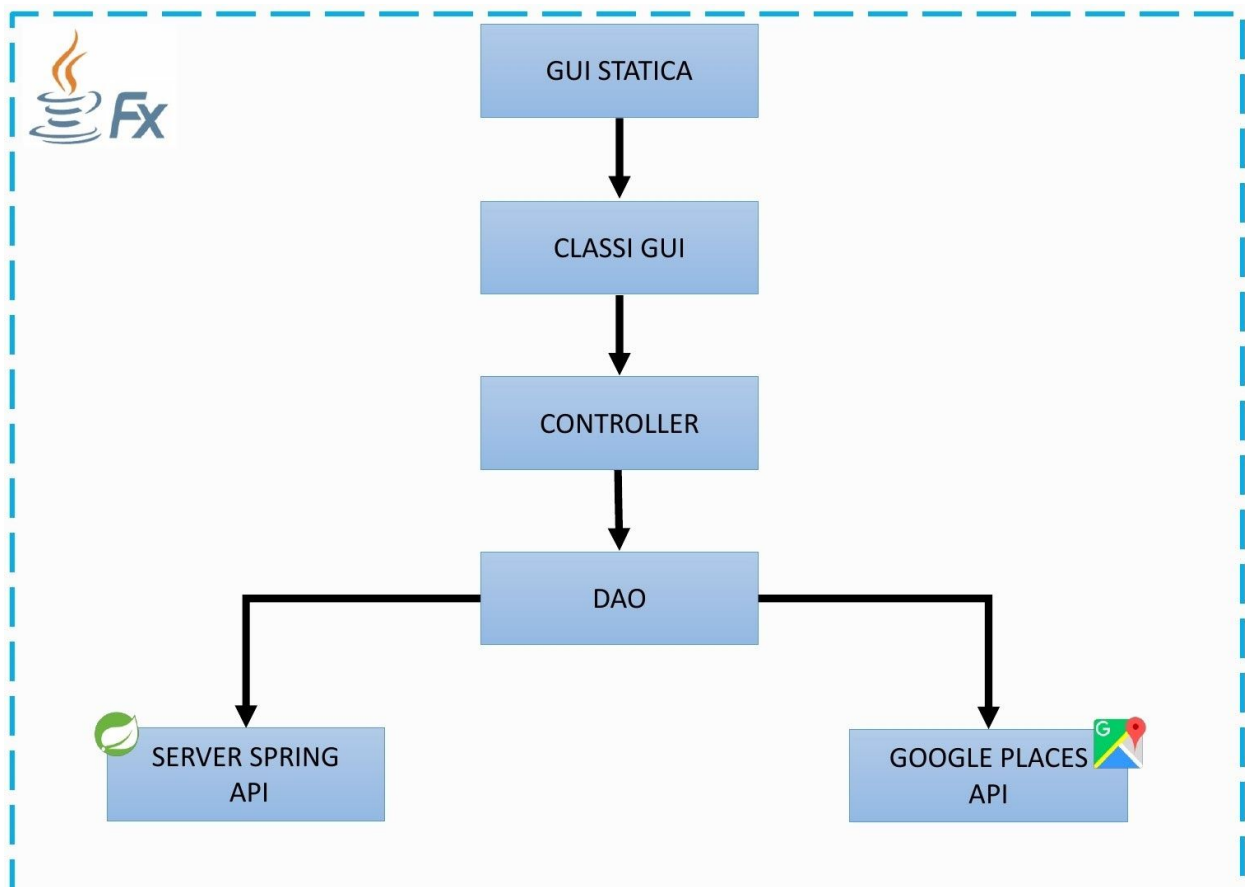
Auth0 per la gestione del pool di utenti, la generazione dei token e l'authentication server.

- **Amazon Elastic Beanstalk** per l'hosting.
- **Amazon RDS** con motore **PostgreSQL** per la persistenza dei dati.
- **Amazon S3** per la persistenza delle foto delle strutture.

3.1.2 Architettura Desktop

L'applicativo Desktop è stato realizzato con il supporto del framework JavaFX per la resa grafica. Anche in questo caso, si ha un beneficio in termini di responsabilità, dal momento che la maggior parte della grafica (statica) è stata realizzata tramite documenti FXML, senza dover necessariamente ricorrere al codice. Per quanto riguarda la comunicazione con il server web è stato utilizzato il client HTTP(s) Retrofit per i seguenti motivi:

- È type-safe.
- Gestisce la parametrizzazione delle richieste tramite annotazione.
- Permette di vedere le API utilizzate come un'interfaccia Java, il che è vantaggioso anche per il lavoro in team.
- È portabile.
-



Nota

- **GUI statica** = è l'insieme dei layout FXML.
- **Classi GUI** = carica i layout e cattura gli eventi scatenati dall'utente.
- **Service** = implementa la logica di business.
- **DAO** = si occupa della persistenza dei dati (dialogando con il web server).

3.1.2.1 Servizi esterni Desktop

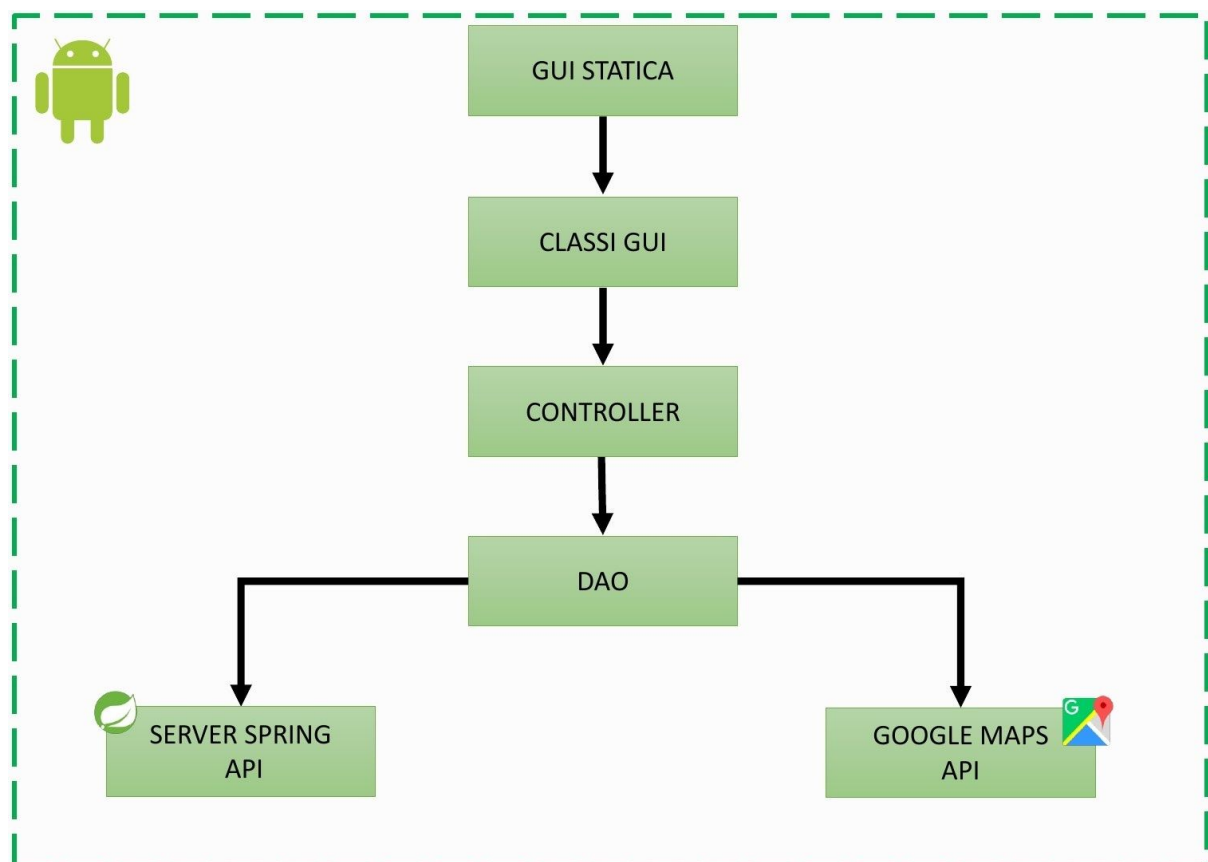
Per questa architettura è stata utilizzata la **API Google Places** per la geocodifica degli indirizzi inseriti dagli amministratori nel backoffice.

3.1.3 Architettura Android

L'applicativo mobile è stato realizzato per il sistema operativo Android con API level 21 (Android 5.0). Gli unici permessi richiesti, per la localizzazione del dispositivo, sono i seguenti:

```
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
<uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

Per la resa grafica, Android sfrutta nativamente i layout in formato XML, quindi anche in questo caso, come per l'applicativo Desktop, è garantita la separazione tra il layer di presentazione statica, il layer di presentazione dinamica e la logica di business. Per la comunicazione con il server web è stato anche qui utilizzato il client HTTP(s) Retrofit.



Nota

- **GUI statica** = è l'insieme dei layout XML.
- **Classi GUI** = carica i layout e cattura gli eventi scatenati dall'utente.
- **Service** = implementa la logica di business.
- **DAO** = si occupa della persistenza dei dati (dialogando con il web server).

3.1.3.1 Servizi esterni Android

Per questa architettura è stata utilizzata la **API Google Maps** per la generazione della mappa e la visualizzazione delle strutture su mappa (con markers e infowindow).

3.1.4 Panoramica dei servizi esterni

Segue una breve sintesi dei servizi utilizzati a supporto dell'intera architettura:

- **Auth0** per la gestione del pool di utenti, la generazione dei token e l'authentication server.
- **Amazon Elastic Beanstalk**, quindi **EC2**, per l'hosting del server.
- **Amazon RDS** con motore PostgreSQL per la persistenza dei dati.
- **Amazon S3** per la persistenza delle foto delle strutture.
- **Google Maps** per la generazione della mappa e la visualizzazione delle strutture su mappa (con markers e infowindow).
- **Google Places** per la geocodifica degli indirizzi inseriti dagli amministratori nel backoffice.

3.1.5 Sicurezza

La sicurezza di Move'n'Go è garantita da una combinazione di Web Filters Spring, e Utenti e Ruoli Auth0. Come già anticipato infatti, per l'autenticazione e la gestione di utenti e ruoli si è scelto di sfruttare Auth0, un'azienda che offre servizi di autenticazione e autorizzazione (compliant con la GDPR), nonché la possibilità di creare e gestire un proprio database di utenti (soluzione adottata nel nostro caso) hostato sui loro server. Tuttavia, l'intero sistema software (quindi sia server che clients) è progettato per limitare al massimo gli sforzi necessari nel caso si volesse o affidarsi ad un altro servizio o gestire il tutto in maniera proprietaria. Il processo di autenticazione è gestito tramite "access tokens" JWT con codifica RS256; supponiamo che un utente già registrato nel database voglia effettuare il login tramite il client (desktop o mobile):

- Il client effettua una richiesta di login al server, inviando in maniera sicura (protocollo HTTPS) le credenziali di accesso, username/email e password.
- Il server Spring fornisce le credenziali al server Auth0 che, in caso di successo, genera e risponde con un JWT avente la seguente struttura:

```
eyJhbGciOiJIUzI1NiIsInR5cCI6IkpXVCJ9.eyJzdWIiOiIxMjM0NTY3ODkwIiwibmFtZSI6IkpvaG4gRG9lIiwiaWF0IjoxNTE2MjM5MDIyfQ.Sf1KxwRJSMekKF2QT4fwpMeJf36P0k6yJV_adQssw5c
```



In particolare, grazie al RBAC (role based access control) nel token saranno inseriti solo i permessi corrispondenti al ruolo dell'utente (claim "permissions"). Dopodiché il server Spring raccoglie il token e lo restituisce al client.

- Dopo aver effettuato il login quindi il client inserirà il token ricevuto in ogni richiesta http verso il server. In particolare, si adotta lo schema http Bearer Authentication, per cui il token si presenterà nell'header della richiesta.

La scelta di adottare un accesso basato su tokens consente di avere maggiore efficienza e un migliore impiego delle risorse, pur restando in totale sicurezza, poiché ogni volta che il server Spring riceve una richiesta contenente un token, può sia validarlo, quindi verificarne l'autenticità, sia controllare che i permessi (contenuti al suo interno) siano sufficienti per effettuare la richiesta senza dover ricontattare l'authentication server. L'integrità e la riservatezza dei dati che viaggiano da e verso il server è garantita dall'uso del protocollo HTTPS (con certificato self-signed).

- La configurazione del protocollo sicuro è stata possibile impostando:

- Inbound rules sul security group dell'istanza EC2.
- Inbound rules sul security group del load balancer.
- Listener sulla porta 443 del load balancer.
- Certificato x509 su AWS IAM.

3.1.5.1 Gestione Filtri

Quando il server riceve una richiesta da parte di un client, a seconda dell'url viene attivato un determinato Web Filter che si occupa della [validazione del token](#) e della verifica dei permessi, che variano a seconda del ruolo dell'utente. Seguono i ruoli creati su Auth0, accompagnati dai relativi permessi inseriti nel token:

- VisitatoreLoggato: is:logguest
- Amministratore: is:logguest is:admin
- SuperAmministratore: is:logguest is:admin is:superadmin

La gerarchia di utenti disegnata in questo modo permette

- agli amministratori
 - di avere anche i permessi dei visitoriloggati nell'applicativo mobile
- ai superamministratori
 - di avere anche i permessi degli amministratori nell'applicativo desktop
 - di avere anche i permessi dei visitoriloggati nell'applicativo mobile.

Nota

Segue una panoramica dei web filters impostati sul server a seconda dei ruoli.

Visitatore: nessun filtro.

Endpoints:

- filtraPerPreferenze
- getRecensioni
- getStrutture
- getStruttureInOrdineAlfabetico

- `getStruttureFiltrate`

VisitatoreLoggato: necessario un token con almeno il permesso «is:logguest».

Endpoints:

- `aggiungiRecensione`

Amministratore: necessario un token con almeno il permesso «is:admin».

Endpoints:

- `aggiungiStruttura`
- `eliminaStruttura`
- `modificaStruttura`
- `verificaEsistenza`

SuperAdmin: necessario un token con almeno il permesso «is:superadmin»

Endpoints:

- `registraComeAmministratore`

3.1.5.2 Validazione del JWT

Quando il server riceve un access token, innanzitutto deve verificare che esso sia valido. Per farlo deve svolgere i seguenti step:

- Verificare che il token è ben formato.
- Verificare l'autenticità (crittografica) della firma.
- Verificare i claim standard.

Dalla [documentazione](#) Auth0, riguardo i claim standard, si legge:

<<Token expiration: the expiration date/time must be after the current date/time and should match what you set for your token lifetime.

Token issuer: the issuing authority inside the token must match the issuing authority (issuer) identified in your Auth0 tenant's discovery document.

Token audience: depending on the initial token request, the aud field could contain both an audience corresponding to your custom API and an audience corresponding to the userinfo endpoint. At least one of the audience values for the token must match the unique identifier of the target API as defined in your API's Settings in the Identifier field.>>

Dopodiché, superata la fase di validazione, sempre sfruttando i claims, il server deve **verificare che l'utente da cui ha ricevuto il token abbia i permessi per richiamare il metodo richiesto**. A tale scopo, si può utilizzare il claim "permissions", presente nel payload del jwt, il cui valore è un array di stringhe corrispondenti agli scope associati al token. Se RBAC è "enabled" allora in tale claim vengono anche inseriti i permessi associati al ruolo dell'utente che ha effettuato la richiesta. Dopo aver decodificato il token, basta verificare che tra le stringhe del claim permissions ci siano i permessi necessari per richiamare il metodo richiesto.

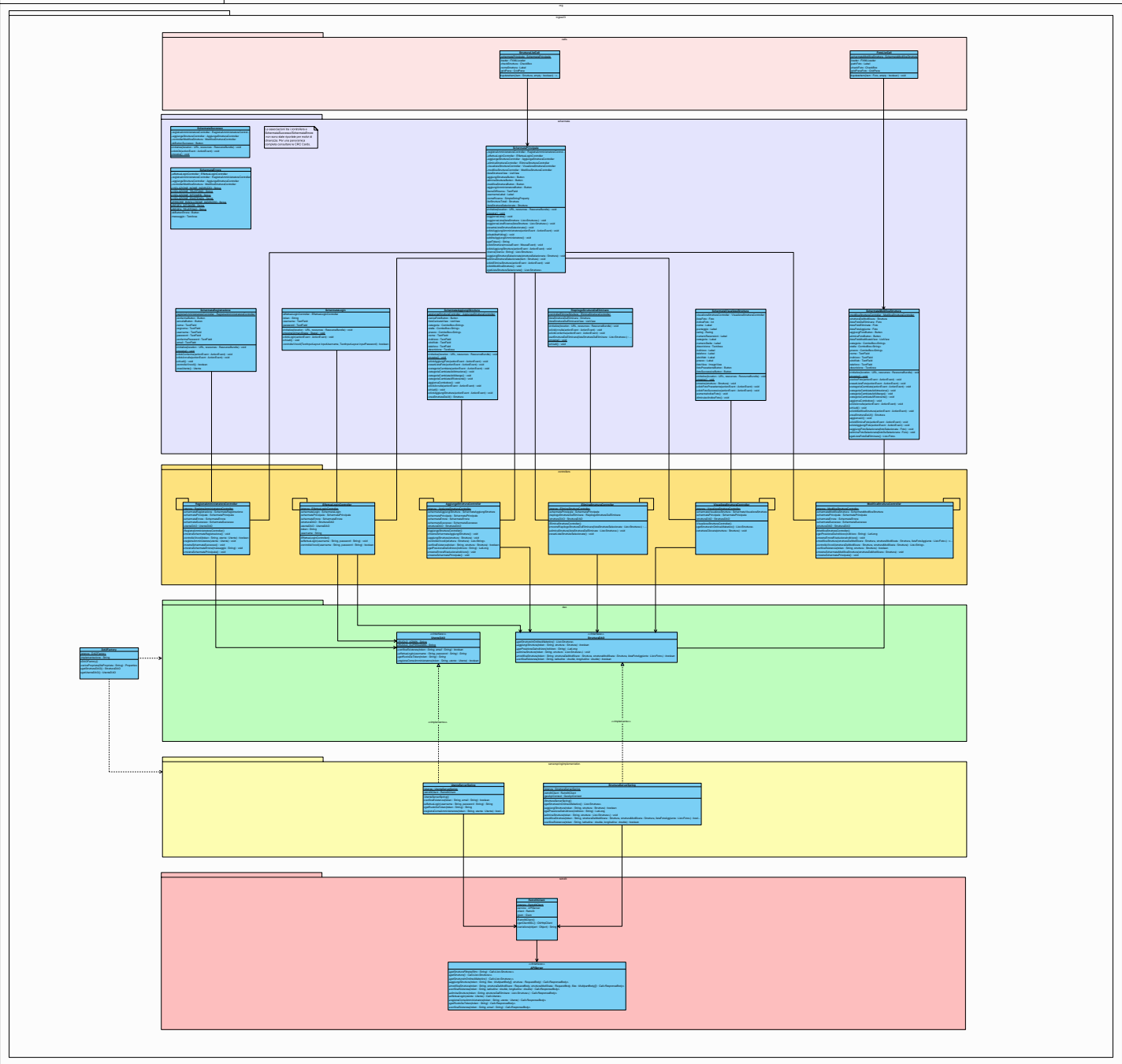
Dal punto di vista dell'implementazione, tutte queste operazioni vengono effettuate dalla classe **JWTUtility**, appositamente progettata per sfruttare il supporto delle librerie **auth0**, **java-jwt** e **jwt-rsa** fornite da Auth0.

3.2 Class Diagram di Design

Nelle due pagine seguenti si illustrano nel dettaglio i diagrammi delle classi delle componenti Desktop e Mobile.

Per motivi di chiarezza, i layer dell'architettura, separati in packages, sono stati evidenziati con colori diversi. Inoltre, si è scelto di adottare il consueto orientamento top-down per indicare che i livelli superiori sfruttano i servizi di quelli inferiori.

3.2.1 Desktop



The diagram is a UML Use Case diagram for a 'System for Managing the Distribution of Medicines'. It is organized into six swimlanes, each representing a different stakeholder or system component. The swimlanes are color-coded and labeled as follows:

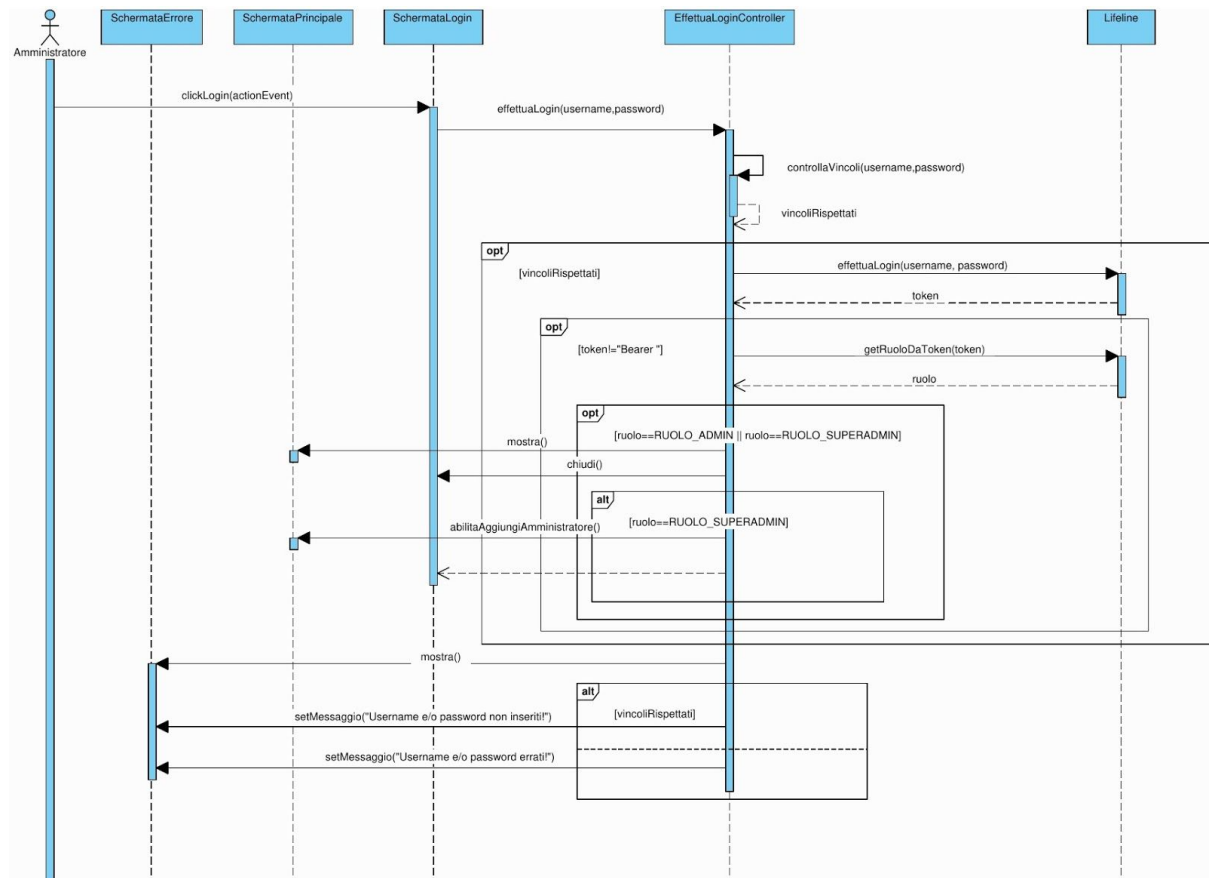
- Administrative Management (Pink):** Contains use cases for managing administrative data, including 'Manage Administrative Data', 'Manage Administrative Data', and 'Manage Administrative Data'.
- Medical Management (Purple):** Contains use cases for managing medical data, including 'Manage Medical Data', 'Manage Medical Data', 'Manage Medical Data', and 'Manage Medical Data'.
- Pharmacy Management (Orange):** Contains use cases for managing pharmacy data, including 'Manage Pharmacy Data', 'Manage Pharmacy Data', 'Manage Pharmacy Data', and 'Manage Pharmacy Data'.
- Patient Management (Green):** Contains use cases for managing patient data, including 'Manage Patient Data', 'Manage Patient Data', 'Manage Patient Data', and 'Manage Patient Data'.
- Medicine Management (Yellow):** Contains use cases for managing medicine data, including 'Manage Medicine Data', 'Manage Medicine Data', and 'Manage Medicine Data'.
- System Management (Brown):** Contains use cases for managing system data, including 'Manage System Data', 'Manage System Data', and 'Manage System Data'.

The diagram shows a complex network of relationships between these use cases, represented by solid and dashed lines. Solid lines indicate direct relationships, while dashed lines indicate dependencies or indirect relationships. The diagram is a high-level overview of the system's functionality and the interactions between different components.

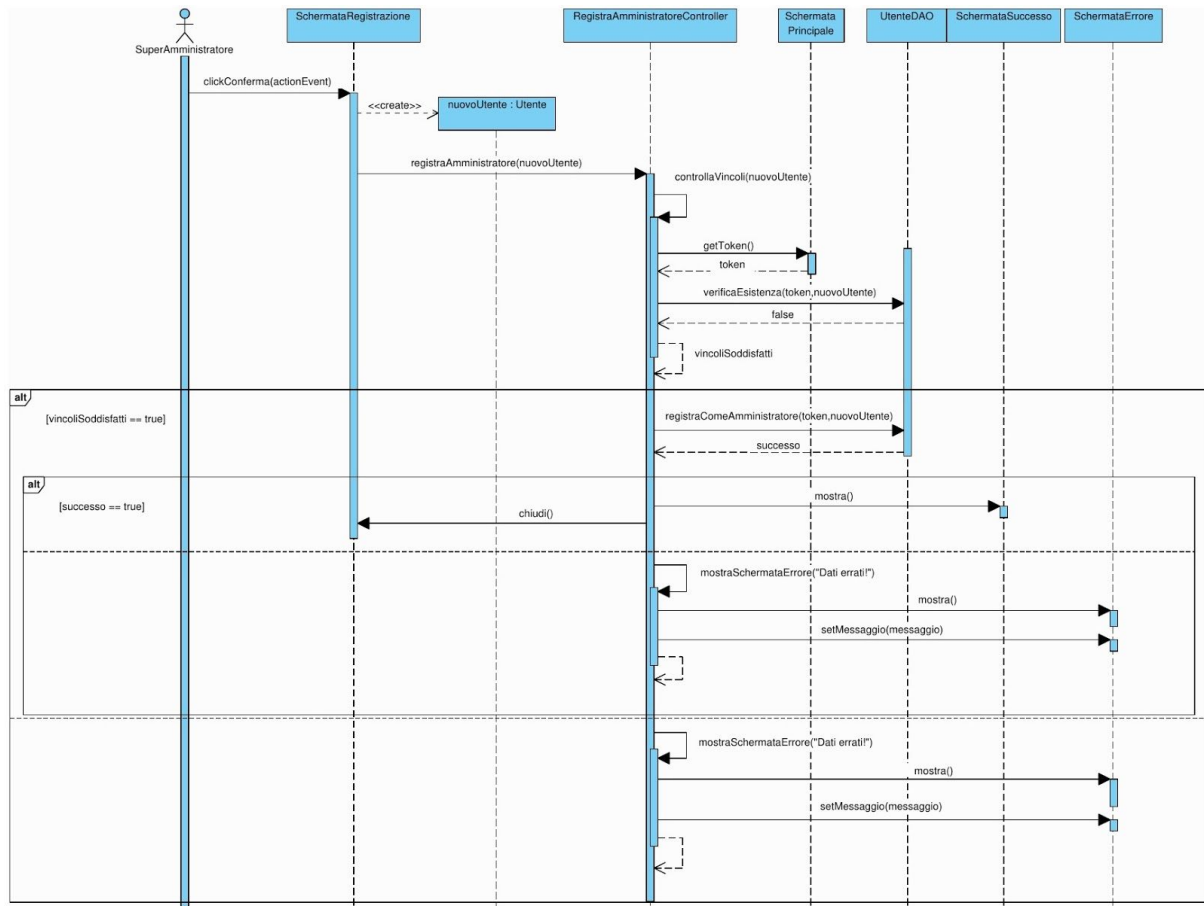
3.3 Sequence Diagram

3.3.1 Sequence Diagrams

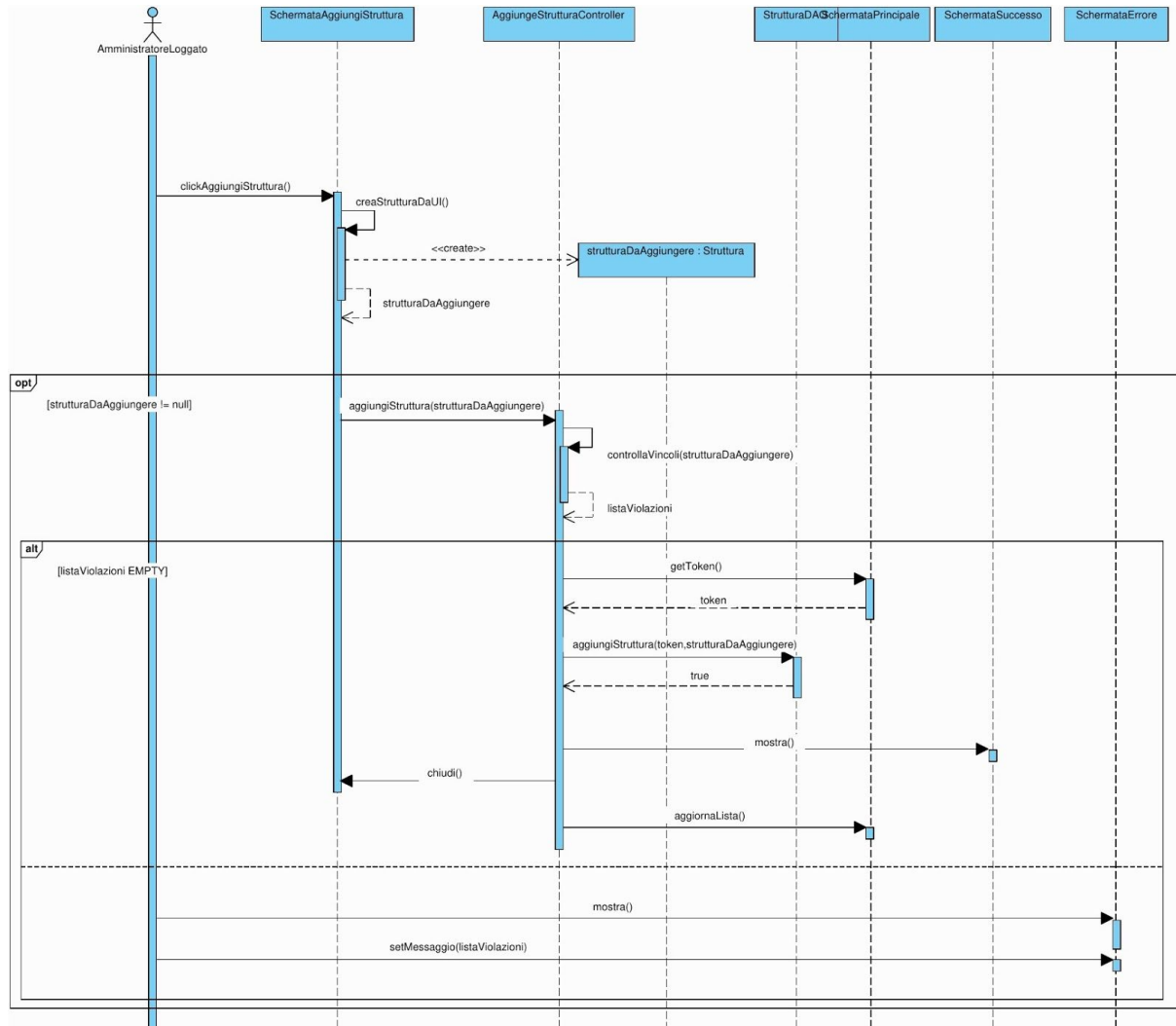
3.3.1.1 Amministratore effettua login



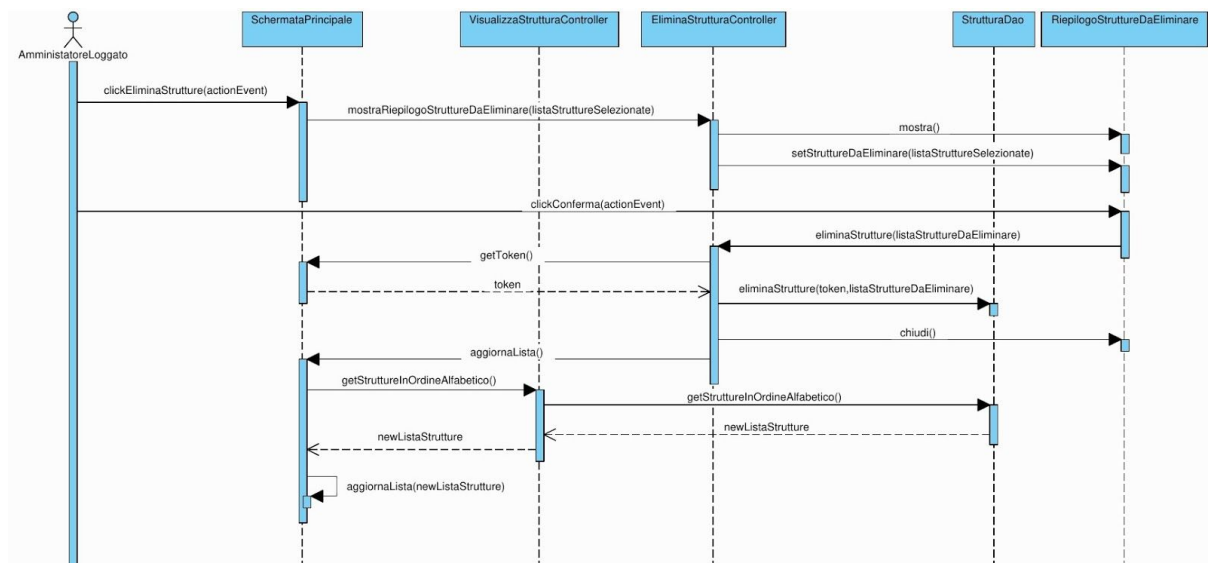
3.3.1.2 SuperAmministratore registra amministratore



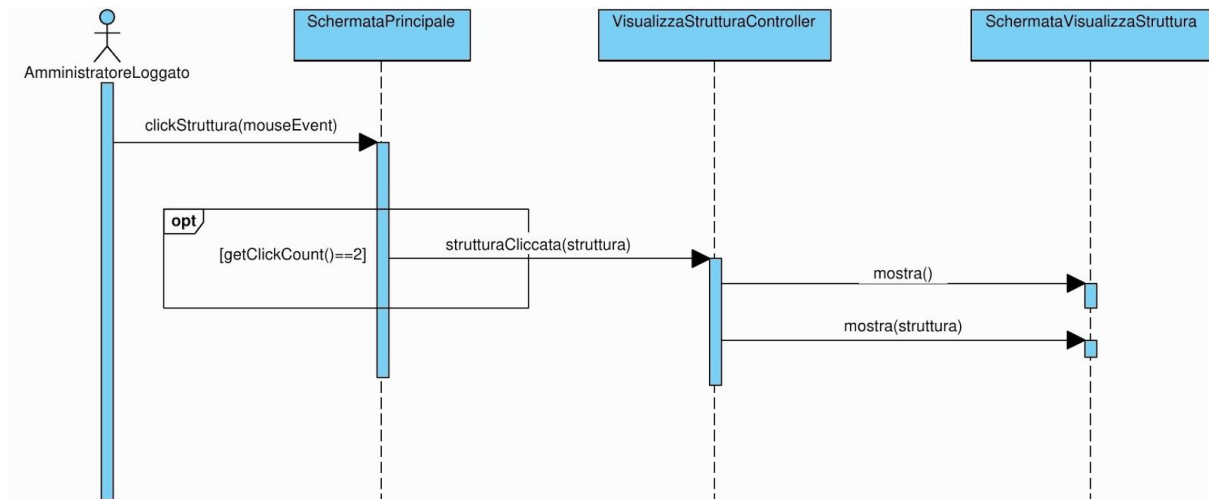
3.3.1.3 AmministratoreLoggato aggiunge struttura



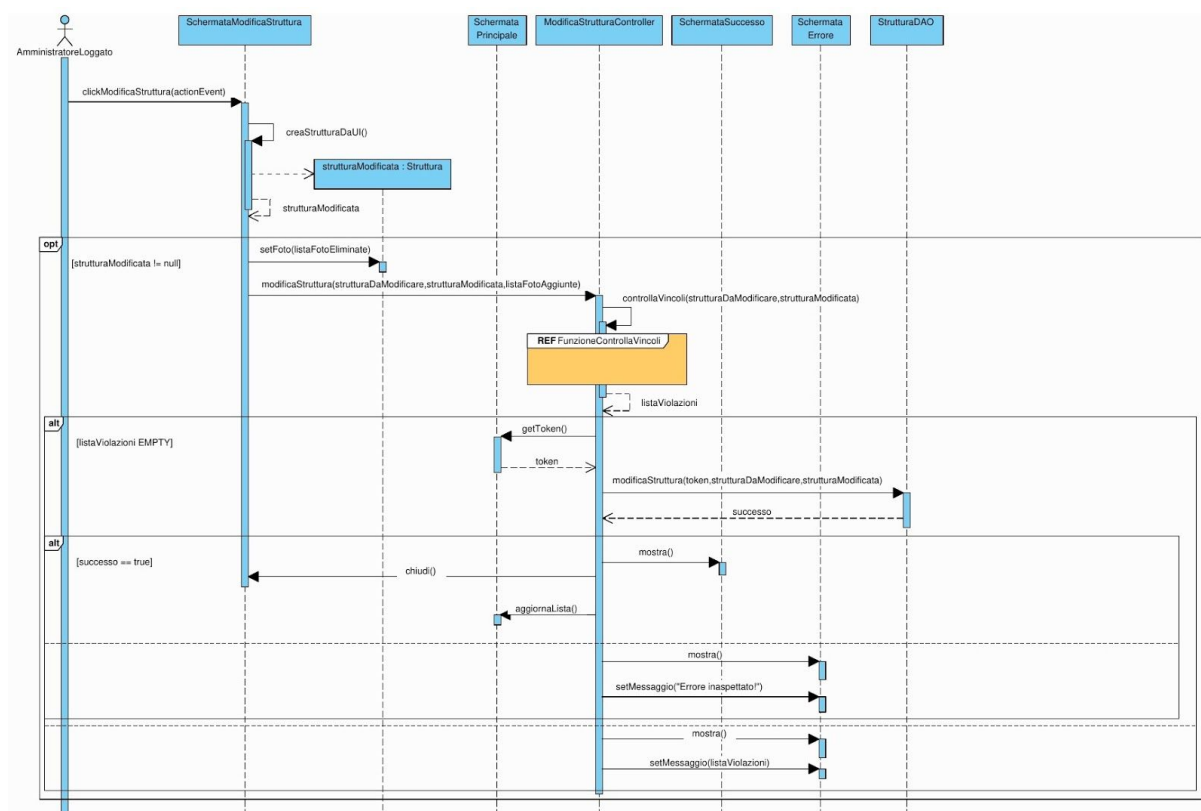
3.3.1.4 AmministratoreLoggato elimina strutture



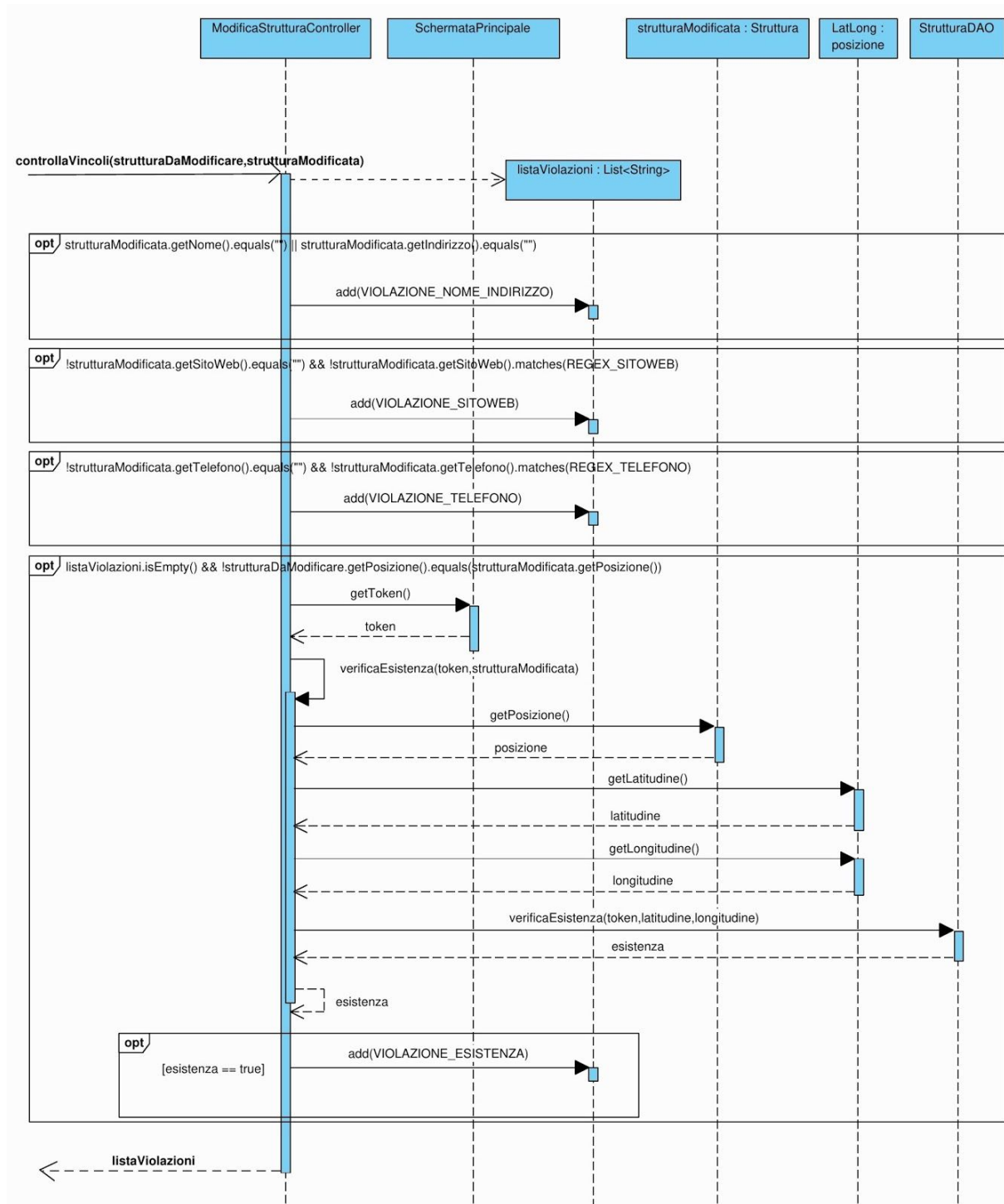
3.3.1.5 AmministratoreLoggato visualizza struttura



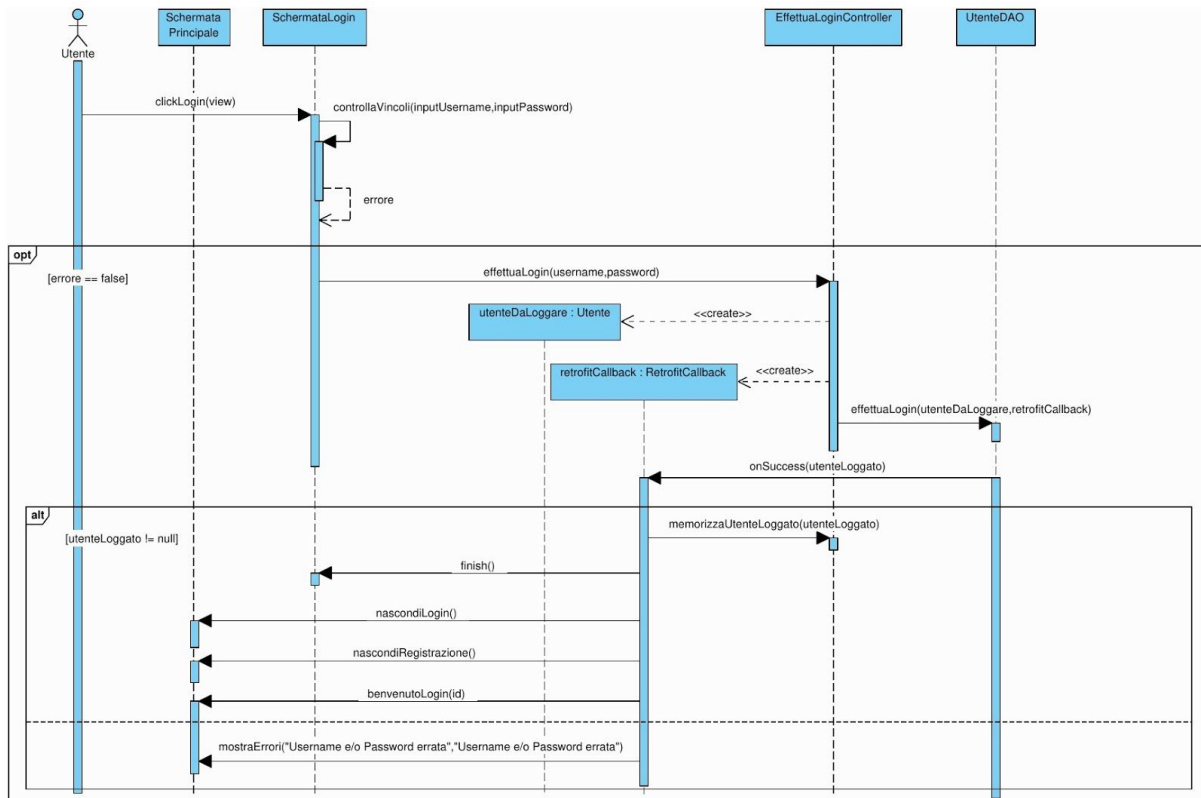
3.3.1.6 AmministratoreLoggato modifica struttura



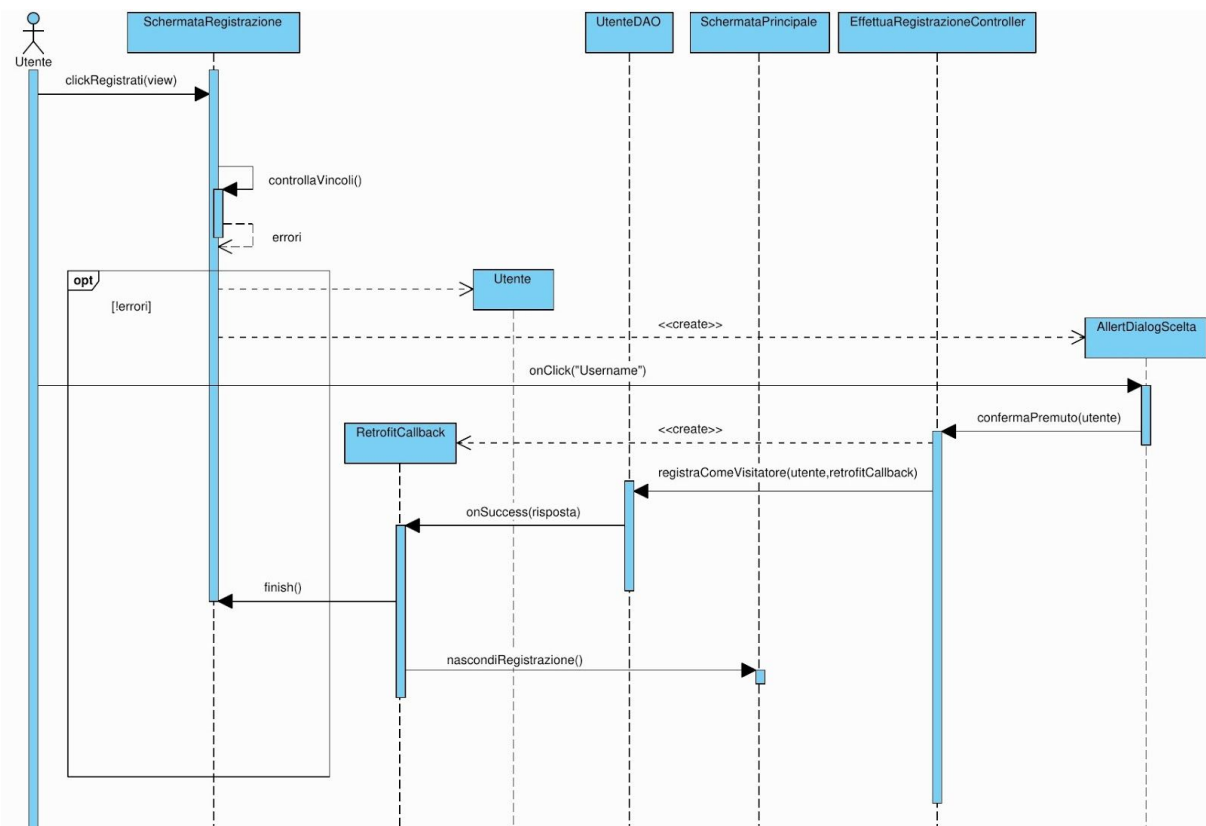
#FunzioneControllaVincoli



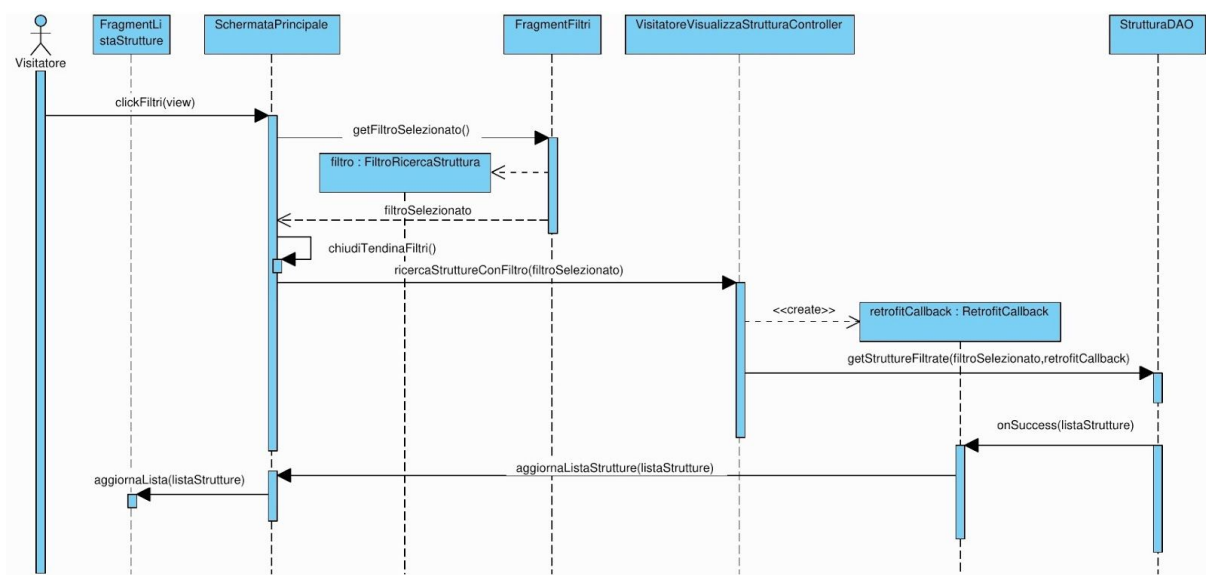
3.3.1.7 Utente effettua login



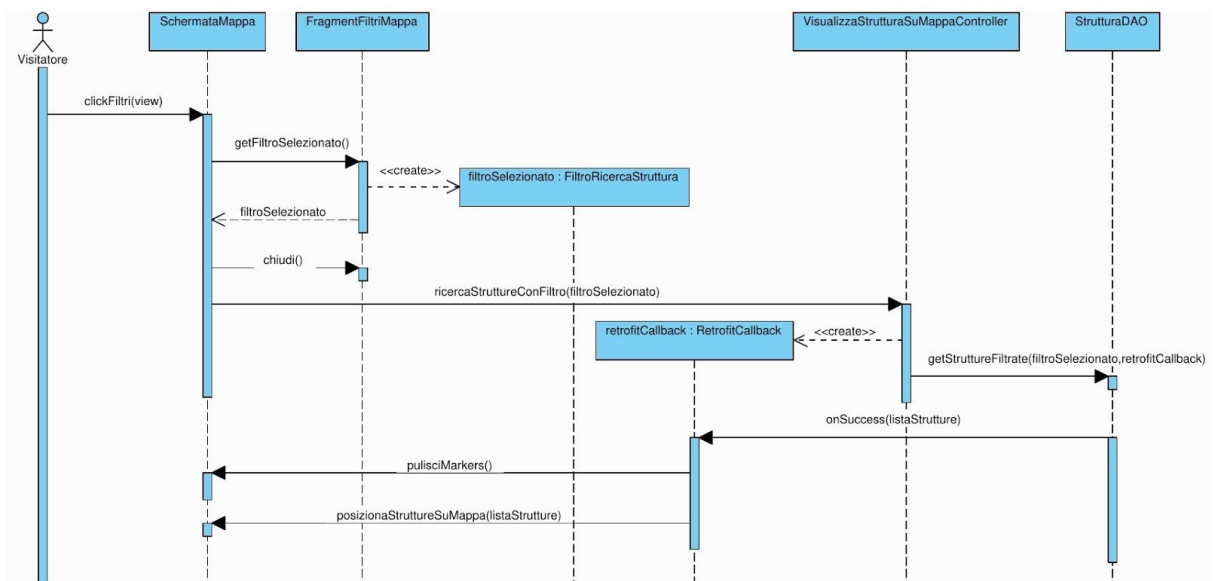
3.3.1.8 Utente effettua registrazione



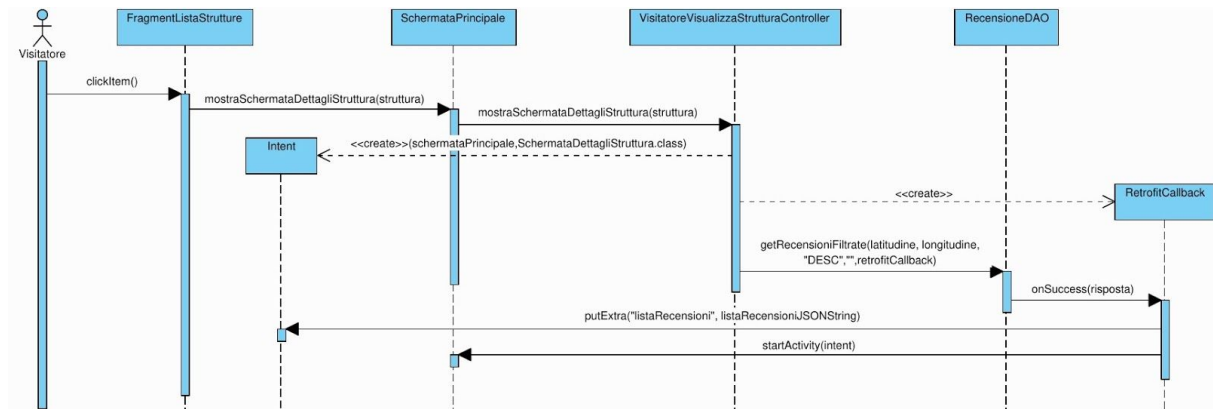
3.3.1.9 Visitatore visualizza struttura



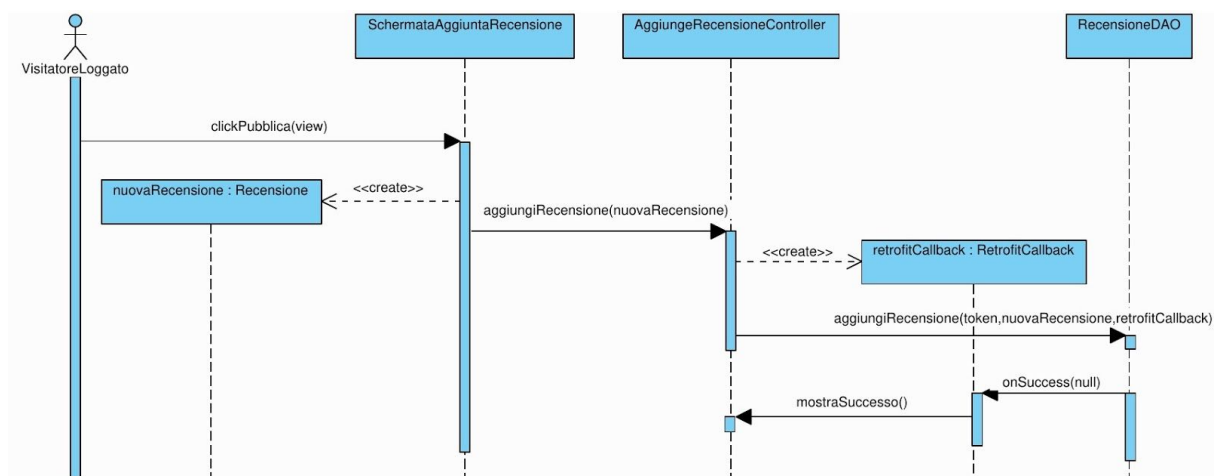
3.3.1.10 Visitatore visualizza struttura su mappa



3.3.1.11 Visitatore visualizza recensioni

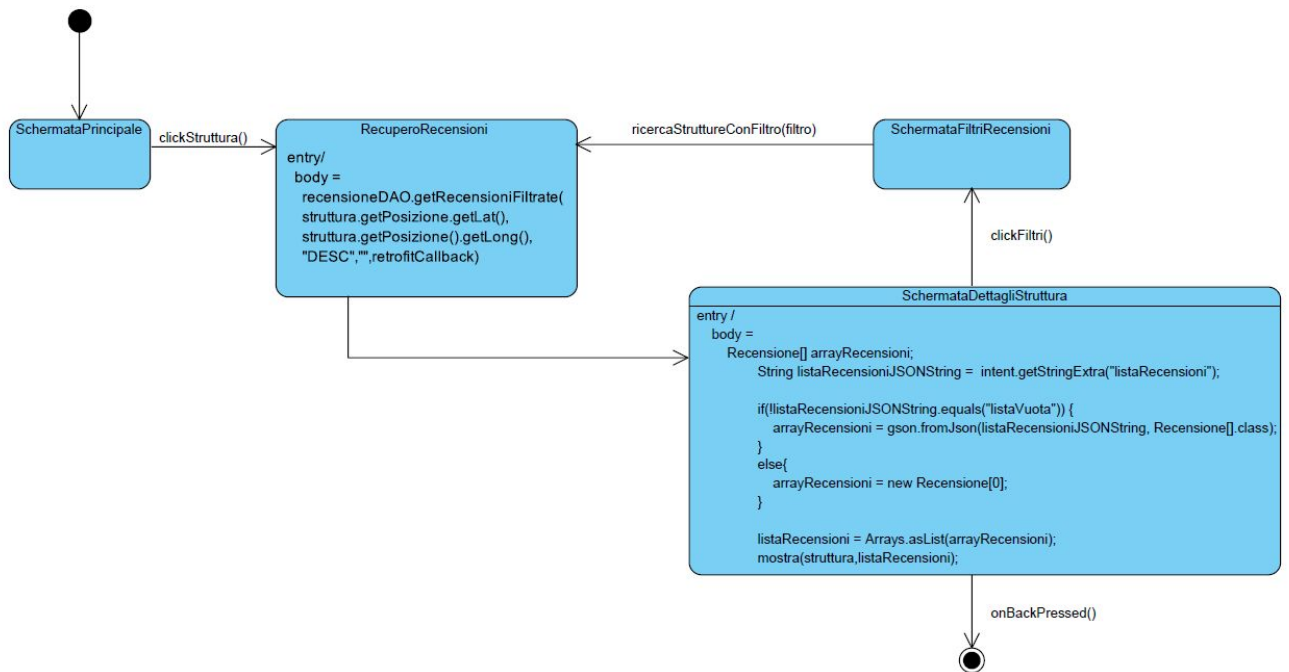


3.3.1.12 Visitatore aggiunge recensione



3.4 Statechart

3.4.1 Visitatore Visualizza Recensioni



3.5 CRC Cards

3.5.1 Desktop

Nome classe	StrutturaListCell	
Superclasse	ListCell<Struttura>	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta una singola cella della lista di strutture presente nella SchermataPrincipale.	SchermataPrincipale FXMLLoader	

Nome classe	FotoListCell	
Superclasse	ListCell<Foto>	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta una singola cella della lista di strutture presente nella SchermataModificaStruttura.	SchermataModificaStruttura FXMLLoader	

Nome classe	SchermataSuccesso	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di successo.	RegistraAmministratoreController AggiungeStrutturaController ModificaStrutturaController	

Nome classe	SchermataErrore	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di errore, personalizzabile con un messaggio.	EffettuaLoginController RegistraAmministratoreController AggiungeStrutturaController ModificaStrutturaController	

Nome classe	SchermataPrincipale	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata principale del back office.	RegistraAmministratoreController EffettuaLoginController AggiungeStrutturaController EliminaStrutturaController VisualizzaStrutturaController ModificaStrutturaController Struttura	

Nome classe	SchermataLogin	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di login del backoffice.	EffettuaLoginController Utente	

Nome classe	SchermataRegistrazione	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di registrazione di un nuovo amministratore.	RegistraAmministratoreController Utente	

Nome classe	SchermataAggiungiStruttura	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di aggiunta di una nuova struttura nel backoffice.	AggiungeStrutturaController Struttura	

Nome classe	RiepilogoStruttureDaEliminare	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata contenente la lista di strutture da eliminare nel backoffice.	EliminaStrutturaController Struttura	

Nome classe	SchermataVisualizzaStruttura	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di visualizzazione di una struttura dal backoffice.	VisualizzaStrutturaController Struttura	

Nome classe	SchermataModificaStruttura	
Superclasse	<<implements>> Initializable	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la schermata di modifica di una struttura dal backoffice.	ModificaStrutturaController Struttura	

Nome classe	RegistraAmministratoreController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "SuperAmministratore registra amministratore".	SchermataRegistrazione SchermataPrincipale SchermataErrore SchermataSuccesso UtenteDAO Utente	

Nome classe	EffettuaLoginController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "Amministratore effettua login".	SchermataLogin SchermataPrincipale SchermataErrore StrutturaDAO UtenteDAO	

Nome classe	AggiungeStrutturaController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "Amministratore loggato aggiunge struttura".	SchermataAggiungiStruttura SchermataPrincipale SchermataErrore SchermataSuccesso StrutturaDAO Struttura	

Nome classe	EliminaStrutturaController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "Amministratore loggato elimina struttura".	RiepilogoStruttureDaEliminare SchermataPrincipale StrutturaDAO Struttura	

Nome classe	VisualizzaStrutturaController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "Amministratore loggato visualizza struttura".	SchermataVisualizzaStruttura SchermataPrincipale StrutturaDAO Struttura	

Nome classe	ModificaStrutturaController	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la business logic dello use case "Amministratore loggato modifica struttura".	SchermataModificaStruttura SchermataPrincipale SchermataErrore SchermataSuccesso StrutturaDAO Struttura Foto LatLong	

Nome classe	<<interface>> UtenteDAO	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Dichiara i metodi necessari per la persistenza di un utente.	Utente	

Nome classe	<<interface>> StrutturaDAO	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Dichiara i metodi necessari per la persistenza di una struttura.	Struttura Foto LatLong	

Nome classe	DAOFactory	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Istanza opportunamente le implementazioni dei DAO a seconda di un file di configurazione.	StrutturaDAO StrutturaServerSpring UtenteDAO UtenteServerSpring	

Nome classe	UtenteServerSpring	
Superclasse	<<implements>> UtenteDAO	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la persistenza degli utenti sul server Spring.	RetrofitClient Utente	

Nome classe	StrutturaServerSpring	
Superclasse	<<implements>> StrutturaDAO	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la persistenza delle strutture sul server Spring.	RetrofitClient Struttura LatLong Foto GeoApiContext	

Nome classe	RetrofitClient	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Effettua le richieste al web server Spring.	Retrofit Gson	

	APIServer OkHttpClient
--	---------------------------

Nome classe	<<interface>> APIServer	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Dichiara i metodi che bisogna implementare per effettuare chiamate al web server tramite Retrofit.	Nessuno	

3.5.2 Mobile

Nome classe	AdapterStrutture	
Superclasse	ArrayAdapter<Struttura>	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Realizzare una view per ogni struttura contenuta nel dataset.	ViewHolder	

Nome classe	ViewHolder	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Mantiene i riferimenti alle View create dall'AdapterStruttura.	Nessuno	

Nome classe	AdapterRecensioni	
Superclasse	ArrayAdapter<Recensione>	
Sottoclassi	Nessuna	

Responsabilità	Collaboratori
Realizzare una view per ogni recensione contenuta nel dataset.	ViewHolder

Nome classe	ViewHolder	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Mantiene i riferimenti alle View create dall'AdapterRecensioni.	Nessuno	

Nome classe	FragmentListaStrutture	
Superclasse	Fragment	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la GUI Logic della lista delle strutture.	SchermataPrincipale AdapterStrutture	

Nome classe	FragmentFiltri	
Superclasse	Fragment	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Rappresenta la GUI Logic dei filtri della SchermataPrincipale.	SchermataPrincipale	

Nome classe	FragmentFiltriMappa	
Superclasse	Fragment	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	

Rappresenta la GUI Logic dei filtri in SchermataMappa.	SchermataMappa
--	----------------

Nome classe	CustomInfoMarker
Superclasse	<<implements>> GoogleMap.InfoWindowAdapter
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestire la GUI Logic degli InfoWindows presenti sulla mappa.	Nessuno

Nome classe	FragmentFiltriRecensioni
Superclasse	Fragment
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Rappresenta la GUI Logic dei filtri in SchermataDettagliStruttura.	Nessuno

Nome classe	SchermataLogin
Superclasse	AppCompatActivity
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la GUI Logic relativa al login.	EffettuaLoginController

Nome classe	SchermataRegistrazione
Superclasse	AppCompatActivity
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la GUI Logic relativa alla registrazione.	EffettuaRegistrazioneController

Nome classe	SchermataPrincipale
-------------	---------------------

Superclasse	AppCompatActivity	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la GUI Logic relativa alla SchermataPrincipale.	FragmentListaStrutture FragmentFiltri VisitatoreVisualizzaStrutturaController AggiungeRecensioneController EffettuaLoginController EffettuaRegistrazioneController VisualizzaStrutturaSuMappaController	

Nome classe	SchermataMappa	
Superclasse	AppCompatActivity, <<implements>> OnMapReadyCallback	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la GUI Logic relativa alla Mappa.	FragmentFiltriMappe CustomInfoMarker VisualizzaStrutturaSuMappaController	

Nome classe	SchermataDettagliStruttura	
Superclasse	AppCompatActivity	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la GUI Logic relativa alla visualizzazione di una struttura con le sue recensioni.	AdapterRecensioni FragmentFiltriRecensioni AggiungeRecensioneController VisualizzaRecensioniController VisitatoreVisualizzaStrutturaController	

Nome classe	SchermataAggiuntaRecensione	
Superclasse	AppCompatActivity	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la GUI Logic relativa	AggiungeRecensioneController	

all'aggiunta di una recensione per una struttura.	
---	--

Nome classe	EffettuaLoginController
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Effettua Login".	SchermataLogin SchermataPrincipale UtenteDAO

Nome classe	EffettuaRegistrazioneController
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Effettua Registrazione".	SchermataRegistrazione SchermataPrincipale UtenteDAO

Nome classe	VisualizzaStrutturaSuMappaController
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Visualizza Struttura Mappa".	SchermataPrincipale SchermataMappa StrutturaDAO

Nome classe	VisitatoreVisualizzaStrutturaController
Superclasse	Nessuna

Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Visualizza Struttura".	SchermataPrincipale SchermataDettagliStruttura StrutturaDAO RecensioneDAO

Nome classe	VisualizzaRecensioniController
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Visualizza Recensioni".	SchermataDettagliStruttura RecensioneDAO

Nome classe	AggiungeRecensioneController
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Gestisce la business logic dello use case "Aggiunge Recensione".	SchermataDettagliStruttura SchermataAggiuntaRecensione SchermataPrincipale RecensioniDAO

Nome classe	<<interface>> UtenteDAO
Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Dichiara i metodi necessari per la persistenza di un utente.	Utente

Nome classe	<<interface>> StrutturaDAO
-------------	----------------------------

Superclasse	Nessuna
Sottoclassi	Nessuna
Responsabilità	Collaboratori
Dichiara i metodi necessari per la persistenza di una struttura.	Struttura

Nome classe	<<interface>> RecensioneDAO	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Dichiara i metodi necessari per la persistenza di una recensione.	Recensioni	

Nome classe	DAOFactory	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Istanza opportunamente le implementazioni dei DAO a seconda di un file di configurazione.	UtenteDAO StrutturaDAO RecensioneDAO UtenteServerSpring StrutturaServerSpring RecensioneServerSpring.	

Nome classe	UtenteServerSpring	
Superclasse	<<implements>> UtenteDAO	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la persistenza degli utenti sul server Spring.	RetrofitClient Utente	

Nome classe	StrutturaServerSpring	
Superclasse	<<implements>> StrutturaDAO	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la persistenza delle strutture sul server Spring.	RetrofitClient FiltroRicercaStruttura	

Nome classe	RecensioneServerSpring	
Superclasse	<<implements>> RecensioneDAO	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Gestisce la persistenza delle recensioni sul server Spring.	RetrofitClient Recensione	

Nome classe	RetrofitClient	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Effettua le richieste al web server Spring.	Retrofit Gson APIServer OkHttpClient	

Nome classe	<<interface>> APIServer	
Superclasse	Nessuna	
Sottoclassi	Nessuna	
Responsabilità	Collaboratori	
Dichiara i metodi che bisogna implementare per effettuare chiamate al web server tramite Retrofit.	Nessuno	

4 Documento di Testing

4.1 Test Plan

4.1.1 Desktop

Test ID	T_D1	
Test name	Amministratore effettua login	
Test description	L'obiettivo del test è di verificare lo use case "Amministratore effettua login"	
Input	Expected	Result
Inserisce dati validi e clicca su "Login".	Login effettuato, visualizza la schermata principale con la lista delle strutture.	Superato
Inserisce username valido ma password errata e clicca su "Login".	Login fallito, visualizza il messaggio d'errore "Username e/o password errati".	Superato
Inserisce il campo password e uno username non esistente e clicca su "Login".	Login fallito, visualizza il messaggio d'errore "Username e/o password errati".	Superato
Inserisce username valido e lascia vuoto il campo password e clicca su "Login".	Login fallito, visualizza il messaggio d'errore "Username e/o password non inseriti!"	Superato
Inserisce una password e lascia vuoto il campo username.	Login fallito, visualizza il messaggio d'errore "Username e/o password non inseriti!"	Superato
Lascia tutti i campi vuoti e clicca su "Login".	Login fallito, visualizza il messaggio d'errore "Username e/o password non inseriti!"	Superato

Test ID	T_D2	
Test name	SuperAmministratore registra amministratore	
Test description	L'obiettivo del test è di verificare lo use case "SuperAmministratore registra amministratore"	
Input	Expected	Result
Inserisce dati validi e clicca su "Conferma".	Registrazione effettuata, visualizza il messaggio "Operazione avvenuta con successo".	Superato
Lascia tutti i campi vuoti e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Inserisce tutti campi validi, ma uno username di lunghezza superiore a 15 caratteri e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Inserisce tutti campi validi, ma uno username di lunghezza inferiore a 8 caratteri e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Lascia almeno un campo vuoto e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Inserisce tutti i campi, ma "Password" diverso da "Conferma Password" e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Le password non corrispondono!".	Superato
Inserisce tutti i campi ma una password debole ¹ .	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Clicca su "Annulla".	Ritorna alla schermata principale.	Superato

¹ Una password è valida (non debole) se contiene almeno 8 caratteri, di cui almeno una lettera minuscola, una lettera maiuscola e un numero.

Test ID	T_D3	
Test name	Amministratore Loggato visualizza struttura	
Test description	L'obiettivo del test è di verificare lo use case "Amministratore Loggato visualizza struttura"	
Input	Expected	Result
Fa doppio click sulla struttura "Grand Hotel Vesuvio".	Visualizza tutti i dettagli sulla struttura "Grand Hotel Vesuvio", foto comprese.	Superato
Digita "Grand Hotel Vesuvio" nella barra di ricerca.	Visualizza la lista delle strutture con solo "Grand Hotel Vesuvio".	Superato
Digita "Grand" nella barra di ricerca.	Visualizza la lista delle strutture con solo "Grand Hotel Vesuvio".	Superato
Digita "grand" nella barra di ricerca.	Visualizza la lista delle strutture con la struttura "Grand Hotel Vesuvio".	Superato
Digita "zzzz" nella barra di ricerca.	La lista delle strutture si svuota.	Superato
Inserisce tutti i campi, ma "Password" diverso da "Conferma Password" e clicca su "Conferma".	Registrazione fallita, visualizza il messaggio d'errore "Le password non corrispondono!".	Superato
Inserisce tutti i campi ma una password debole ² .	Registrazione fallita, visualizza il messaggio d'errore "Dati errati".	Superato
Clicca su "Annulla".	Ritorna alla schermata principale.	Superato

² Una password è valida (non debole) se contiene almeno 8 caratteri, di cui almeno una lettera minuscola, una lettera maiuscola e un numero.

Test ID	T_D4	
Test name	Amministratore Loggato aggiunge struttura	
Test description	L'obiettivo del test è di verificare lo use case "Amministratore Loggato aggiunge struttura"	
Input	Expected	Result
Inserisce tutti i campi (validi) di una struttura non presente nel sistema.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Inserisce tutti i campi (validi) di una struttura non presente nel sistema e aggiunge una foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Inserisce tutti i campi (validi) di una struttura non presente nel sistema e aggiunge due foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Lascia tutti i campi vuoti e clicca su "Conferma".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi eccetto "Nome".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi eccetto "Indirizzo".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi e telefono "abcd" (formato non valido)	Visualizza messaggio d'errore "Telefono non valido".	Superato
Inserisce tutti i campi e sito web "abcd" (formato non valido)	Visualizza messaggio d'errore "Sito Web non valido".	Superato
Inserisce tutti i campi e indirizzo "abcd" (inesistente)	Visualizza messaggio d'errore "Errore nella risoluzione dell'indirizzo".	Superato
Inserisce tutti i campi e un indirizzo che è già stato assegnato ad un'altra struttura.	Visualizza messaggio d'errore "Errore nella risoluzione dell'indirizzo".	Superato
Clicca su "Annulla".	Ritorna alla schermata principale.	Superato

Test ID	T_D5	
Test name	Amministratore Loggato modifica struttura	
Test description	L'obiettivo del test è di verificare lo use case "Amministratore Loggato modifica struttura"	
Input	Expected	Result
Modifica tutti i campi con dati validi.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi e aggiunge una foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi e aggiunge due foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi ed elimina una foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi ed elimina due foto.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi ed elimina tutte le foto della struttura.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Modifica tutti i campi con dati validi, elimina una foto e ne aggiunge una.	Visualizza il messaggio "Operazione avvenuta con successo.	Superato
Cancella tutti i campi e clicca su "Conferma".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi eccetto "Nome".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi eccetto "Indirizzo".	Visualizza messaggio d'errore "Nome e indirizzo sono campi obbligatori!".	Superato
Inserisce tutti i campi e telefono "abcd" (formato non valido)	Visualizza messaggio d'errore "Telefono non valido".	Superato

Inserisce tutti i campi e sito web "abcd" (formato non valido)	Visualizza messaggio d'errore "Sito Web non valido".	Superato
Inserisce tutti i campi e indirizzo "abcd" (inesistente)	Visualizza messaggio d'errore "Errore nella risoluzione dell'indirizzo".	Superato
Inserisce tutti i campi e un indirizzo che è già stato assegnato ad un'altra struttura.	Visualizza messaggio d'errore "Errore nella risoluzione dell'indirizzo".	Superato
Clicca su "Annulla".	Ritorna alla schermata principale.	Superato

Test ID	T_D6	
Test name	Amministratore Loggato elimina struttura	
Test description	L'obiettivo del test è di verificare lo use case "Amministratore Loggato elimina struttura"	
Input	Expected	Result
Seleziona una struttura e clicca sul tasto "Elimina"	Visualizza un riepilogo contenente la sola struttura selezionata.	Superato
Seleziona due strutture e clicca sul tasto "Elimina"	Visualizza un riepilogo contenente le due strutture selezionate.	Superato
Clicca "Conferma" sulla schermata che mostra il riepilogo delle strutture selezionate.	Visualizza la lista delle strutture mancante delle strutture presenti nel riepilogo.	Superato
Clicca "Annulla" sulla schermata che mostra il riepilogo delle strutture selezionate.	Visualizza la lista delle strutture senza modifiche.	Superato

4.1.2 Mobile

Test ID	T_M1	
Test name	Utente effettua login	
Test description	L'obiettivo del test è di verificare lo use case "Utente effettua login"	
Input	Expected	Result
Inserisce dati validi e clicca su "Login".	Login effettuato, visualizza la schermata principale con la lista delle strutture.	Superato
Inserisce username valido ma password errata e clicca su "Login".	Login fallito, evidenzia i campi username e password di rosso e visualizza il messaggio d'errore "Username e/o password errati".	Superato
Inserisce il campo password e uno username non esistente e clicca su "Login".	Login fallito, evidenzia i campi username e password di rosso e visualizza il messaggio d'errore "Username e/o password errati".	Superato
Inserisce username valido e lascia vuoto il campo password e clicca su "Login".	Login fallito, evidenzia il campo password di rosso e visualizza il messaggio "Numero caratteri insufficiente".	Superato
Inserisce una password e lascia vuoto il campo username.	Login fallito, evidenzia il campo username di rosso e visualizza il messaggio "Numero caratteri insufficiente"	Superato
Lascia tutti i campi vuoti e clicca su "Login".	Login fallito, evidenzia i campi username e password di rosso e visualizza il messaggio "Numero caratteri insufficiente"	Superato
Inserisce username di lunghezza < 8 e password	Login fallito, evidenzia il campo username di rosso e	Superato

valida.	visualizza il messaggio "Numero caratteri insufficiente"	
Inserisce password di lunghezza < 8 e username valido.	Login fallito, evidenzia il campo password di rosso e visualizza il messaggio "Numero caratteri insufficiente"	Superato

Test ID	T_M2	
Test name	Utente effettua registrazione	
Test description	L'obiettivo del test è di verificare lo use case "Utente effettua registrazione"	
Input	Expected	Result
Inserisce dati validi e clicca su "Conferma".	Registrazione effettuata, visualizza il popup "Come vuoi essere visualizzato?".	Superato
Lascia tutti i campi vuoti e clicca su "Conferma".	Registrazione fallita, evidenzia tutti i campi di rosso.	Superato
Inserisce tutti campi validi, ma uno username di lunghezza superiore a 15 caratteri e clicca su "Conferma".	Registrazione fallita, evidenzia il campo username di rosso e visualizza il messaggio "Username troppo lungo".	Superato
Inserisce tutti campi validi, ma uno username di lunghezza inferiore a 8 caratteri e clicca su "Conferma".	Registrazione fallita, evidenzia il campo username di rosso e visualizza il messaggio "Username troppo corto".	Superato
Lascia almeno un campo vuoto e clicca su "Conferma".	Registrazione fallita, evidenzia di rosso i campi non inseriti.	Superato
Inserisce tutti i campi, ma "Password" diverso da "Conferma Password" e clicca su "Conferma".	Registrazione fallita, evidenzia i campi password e conferma password di rosso e visualizza il messaggio "Le password non coincidono".	Superato

Inserisce tutti i campi ma una password debole ³ .	Registrazione fallita, evidenza di rosso i campi password e conferma password e visualizza il messaggio "Password debole"	Superato
Inserisce tutti i campi, ma uno username o una email già esistente.	Registrazione fallita, evidenza di rosso i campi email e username e visualizza il messaggio "Username e/o email già esistenti".	Superato

Test ID	T_M3	
Test name	Visitatore visualizza struttura	
Test description	L'obiettivo del test è di verificare lo use case "Visitatore visualizza struttura"	
Input	Expected	Result
Clicca sulla struttura "Grand Hotel Vesuvio"	Visualizza la schermata contenente i dettagli sulla struttura "Grand Hotel Vesuvio".	Superato
Digita "Grand Hotel Vesuvio" nella barra di ricerca.	Visualizza la lista delle strutture con solo "Grand Hotel Vesuvio".	Superato
Digita "Grand" nella barra di ricerca.	Visualizza la lista delle strutture con solo "Grand Hotel Vesuvio".	Superato
Digita "grand" nella barra di ricerca.	Visualizza la lista delle strutture con la struttura "Grand Hotel Vesuvio".	Superato
Digita "zzzz" nella barra di ricerca.	La lista delle strutture si svuota.	Superato

³ Una password è valida (non debole) se contiene almeno 8 caratteri, di cui almeno una lettera minuscola, una lettera maiuscola e un numero.

Test ID	T_M4	
Test name	Visitatore visualizza recensioni	
Test description	L'obiettivo del test è di verificare lo use case "Visitatore visualizza recensioni"	
Input	Expected	Result
Clicca sulla struttura "Grand Hotel Vesuvio".	Visualizza la schermata contenente i dettagli sulla struttura "Grand Hotel Vesuvio" e il box con la lista delle recensioni ordinata per data decrescente.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e seleziona "Data decrescente" nella tendina dei filtri.	Visualizza la schermata contenente i dettagli sulla struttura "Grand Hotel Vesuvio" e il box con la lista delle recensioni ordinata per data decrescente.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e seleziona "Data crescente" nella tendina dei filtri.	Visualizza la schermata contenente i dettagli sulla struttura "Grand Hotel Vesuvio" e il box con la lista delle recensioni ordinata per data crescente.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e digita "Esplosivo" nella tendina dei filtri.	Mostra le tre recensioni di Pippo Bello, Pippo Brutto e Pippo Franco contenenti la parola "esplosivo" sia nel titolo che nella descrizione.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e digita "esplosivo" nella tendina dei filtri.	Mostra le tre recensioni di Pippo Bello, Pippo Brutto e Pippo Franco contenenti la parola "esplosivo" sia nel titolo che nella descrizione.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e digita "espl" nella tendina dei filtri.	Mostra le tre recensioni di Pippo Bello, Pippo Brutto e Pippo Franco contenenti la parola "esplosivo" sia nel titolo che nella descrizione.	Superato
Clicca sulla struttura "Grand Hotel Vesuvio" e digita "zzzzz" nella tendina dei filtri.	Mostra il messaggio "Nessuna recensione disponibile!"	Superato

Test ID	T_M5	
Test name	Visitatore Loggato aggiunge recensione	
Test description	L'obiettivo del test è di verificare lo use case "Visitatore Loggato aggiunge recensione"	
Input	Expected	Result
Inserisce titolo, punteggio e descrizione e clicca su "Pubblica".	Visualizza il messaggio "Operazione avvenuta con successo".	Superato
Clicca su ok nella schermata di successo.	Visualizza la schermata dei dettagli della struttura con la recensione e il punteggio aggiornati. Il pulsante per aggiungere una nuova recensione è disattivato.	Superato
Clicca su "Annulla".	Ritorna alla schermata dei dettagli della struttura. Il pulsante per aggiungere una nuova recensione è attivato.	Superato
Inserisce titolo e punteggio, ma lascia vuoto il campo descrizione.	Il pulsante "Pubblica" è disabilitato.	Superato
Inserisce descrizione e punteggio, ma lascia vuoto il campo titolo.	Il pulsante "Pubblica" è disabilitato.	Superato
Inserisce titolo e descrizione, ma selezione punteggio zero..	Il pulsante "Pubblica" è disabilitato.	Superato

Test ID	T_M6	
Test name	Visitatore visualizza struttura	
Test description	L'obiettivo del test è di verificare lo use case "Visitatore visualizza struttura"	
Precondizioni	Il database deve contenere le seguenti strutture: Albergo "Andromeda". Rating 5 prezzo 3 Ristorante "Burger Queen". Rating 4 prezzo 1 Albergo "Grand Hotel Vesuvio". Rating 3 prezzo 4 Attrazione "Parco Fontana". Rating 2 prezzo // Ristorante "La Taverna". Rating 1 prezzo 4	
Input	Expected	Result
Fa una ricerca con i filtri di default, cioè per "Consigliati"	Visualizza le strutture nel seguente ordine: 1. "Burger Queen". 2. "Parco Fontana". 3. "Andromeda". 4. "Grand Hotel Vesuvio". 5. "La Taverna".	Superato
Fa una ricerca con "Punteggio" uguale a 5.	Visualizza nella lista la struttura "Andromeda".	Superato
Fa una ricerca con "Punteggio" uguale a 3.	Visualizza nella lista la struttura "Grand Hotel Vesuvio".	
Fa una ricerca con "Categoria" uguale ad "Attrazioni".	Visualizza nella lista la struttura "Parco Fontana".	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Punteggio minimo" uguale a 2. • "Prezzo decrescente" 	Visualizza le strutture nel seguente ordine: 1. "Grand Hotel Vesuvio". 2. "Andromeda". 3. "Burger Queen". 4. "Parco Fontana".	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Categoria" uguale ad "Alberghi". • "Media Recensioni" 	Visualizza le strutture nel seguente ordine: 1. "Andromeda". 2. "Grand Hotel Vesuvio".	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Prezzo" uguale a 4. • "Punteggio minimo" uguale a 0. • "Media recensioni" 	Visualizza le strutture nel seguente ordine: 1. "Grand Hotel Vesuvio". 2. "La Taverna".	Superato

<p>Fa una ricerca con</p> <ul style="list-style-type: none"> • “Punteggio minimo” uguale a 2. • “Prezzo decrescente”. • “Parola Chiave” uguale a “g”. 	<p>Visualizza le strutture nel seguente ordine:</p> <ol style="list-style-type: none"> 1. “Grand Hotel Vesuvio” 2. “Burger Queen”. 	Superato
<p>Fa una ricerca con</p> <ul style="list-style-type: none"> • “Categoria” uguale ad “Alberghi”. • “Media Recensioni” • “Parola Chiave” uguale a “m” 	<p>Visualizza nella lista la struttura “Andromeda”.</p>	Superato
<p>Fa una ricerca con</p> <ul style="list-style-type: none"> • “Prezzo” uguale a 4. • “Punteggio minimo” uguale a 0. • “Media recensioni” • “Parola Chiave” uguale a “ve”. 	<p>Visualizza le strutture nel seguente ordine:</p> <ol style="list-style-type: none"> 1. “Grand Hotel Vesuvio”. 2. “La Taverna”. 	Superato
<p>Fa una ricerca con “Parola Chiave” uguale a “zzz”.</p>	<p>Visualizza una lista vuota.</p>	

Test ID	T_M7	
Test name	Visitatore visualizza struttura su mappa	
Test description	L'obiettivo del test è di verificare lo use case "Visitatore visualizza struttura su mappa"	
Precondizioni	Il database deve contenere le seguenti strutture: Albergo "Andromeda". Rating 5 prezzo 3 Ristorante "Burger Queen". Rating 4 prezzo 1 Albergo "Grand Hotel Vesuvio". Rating 3 prezzo 4 Attrazione "Parco Fontana". Rating 2 prezzo // Ristorante "La Taverna". Rating 1 prezzo 4	
Input	Expected	Result
Fa una ricerca con i filtri di default, cioè <ul style="list-style-type: none"> • "Prezzo" uguale a "Qualunque". • "Punteggio minimo" uguale a 0. 	Visualizza sulla mappa tutte le strutture.	Superato
Fa una ricerca con "Punteggio" uguale a 5.	Visualizza sulla mappa la struttura "Andromeda".	Superato
Fa una ricerca con "Punteggio" uguale a 3.	Visualizza sulla mappa la struttura "Grand Hotel Vesuvio".	
Fa una ricerca con "Categoria" uguale ad "Attrazioni".	Visualizza sulla mappa la struttura "Parco Fontana".	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Punteggio minimo" uguale a 2. 	Visualizza sulla mappa le seguenti strutture: <ul style="list-style-type: none"> • "Grand Hotel Vesuvio". • "Andromeda". • "Burger Queen". • "Parco Fontana". 	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Prezzo" uguale a 4. • "Punteggio minimo" uguale a 0. 	Visualizza sulla mappa le seguenti strutture: <ul style="list-style-type: none"> • "Grand Hotel Vesuvio". • "La Taverna". 	Superato
Fa una ricerca con <ul style="list-style-type: none"> • "Punteggio minimo" uguale a 2. • "Parola Chiave" uguale a "g". 	Visualizza sulla mappa le seguenti strutture: <ul style="list-style-type: none"> • "Grand Hotel Vesuvio" • "Burger Queen". 	Superato

<p>Fa una ricerca con</p> <ul style="list-style-type: none"> • “Categoria” uguale ad “Alberghi”. • “Parola Chiave” uguale a “m” 	<p>Visualizza sulla mappa la struttura “Andromeda”.</p>	<p>Superato</p>
<p>Fa una ricerca con</p> <ul style="list-style-type: none"> • “Prezzo” uguale a 4. • “Punteggio minimo” uguale a 0. • “Parola Chiave” uguale a “ve”. 	<p>Visualizza sulla mappa le seguenti strutture:</p> <ul style="list-style-type: none"> • “Grand Hotel Vesuvio”. • “La Taverna”. 	<p>Superato</p>
<p>Fa una ricerca con “Parola Chiave” uguale a “zzz”.</p>	<p>Visualizza la mappa priva di markers.</p>	<p>Superato</p>

4.2 Testing del metodo prezzoDaStringaAIntero

Il metodo appartiene alla classe model FiltroRicercaStruttura utilizzata da tutti e tre i componenti software del sistema: server, applicativo Android, applicativo desktop. Il suo scopo è convertire una stringa nell'insieme {€,€,€,€,€€€€} in un intero corrispondente al numero di simboli «€». In caso di errore lancia RuntimeException.

La signature è la seguente:

```
public static int prezzoDaStringaAIntero(String prezzo)
```

Parametro di input:

- String Prezzo: stringa appartenente all'insieme {€,€,€,€,€€€€}.

Output:

- Numero intero corrispondente al numero di simboli “€” presenti nella stringa di input.

Esempio d'uso

USO	OUTPUT
prezzoDaStringaAIntero(“€”)	1
prezzoDaStringaAIntero(“€€”)	2
prezzoDaStringaAIntero(“€€€”)	3
prezzoDaStringaAIntero(“€€€€”)	4
prezzoDaStringaAIntero(“nonValido”)	RuntimeException

4.2.1 Testing blackbox

Poiché il parametro di input appartiene ad un insieme discreto, allora è stata definita una classe di equivalenza valida per ogni valore dell'insieme e una non valida per tutti i valori non appartenenti ad esso. In particolare:

- CE1: {"€"} valida
- CE2: {"€€"} valida
- CE3: {"€€€"} valida
- CE4: {"€€€€"} valida
- CE5: String – {CE1 U CE2 U CE3 U CE4} non valida

Trattandosi di un metodo che prende in input un solo parametro, le strategie WECT e SECT coincidono. Pertanto è stato scritto un test case per ogni classe:

- prezzoValido1()
- prezzoValido2()
- prezzoValido3()
- prezzoValido4()
- prezzoNonValido()

```
. public class PrezzoDaStringaAInteroUnitTest {  
.     /*  
.     Il metodo prende in input una stringa nell'insieme {€,€,€,€€€€} e  
.     restituisce un intero corrispondente al numero di simboli "€". In caso di  
.     errore lancia RuntimeException.  
.     */  
.     @Test  
.     public void prezzoValido1(){  
.         int uno = FiltroRicercaStruttura.prezzoDaStringaAIntero("€");  
.         assertEquals(1,uno);  
.     }  
. }
```



```

.      @Test
.      public void prezzoValido2(){
.          int due = FiltroRicercaStruttura.prezzoDaStringaAIntero("€€");
.          assertEquals(2,due);
.      }
.
.      @Test
.      public void prezzoValido3(){
.          int tre = FiltroRicercaStruttura.prezzoDaStringaAIntero("€€€");
.          assertEquals(3,tre);
.      }
.
.      @Test
.      public void prezzoValido4(){
.          int quattro = FiltroRicercaStruttura.prezzoDaStringaAIntero("€€€€");
.          assertEquals(4,quattro);
.      }
.
.      @Test(expected = RuntimeException.class)
.      public void prezzoNonValido() {
.          FiltroRicercaStruttura.prezzoDaStringaAIntero("nonValido");
.      }
.  }

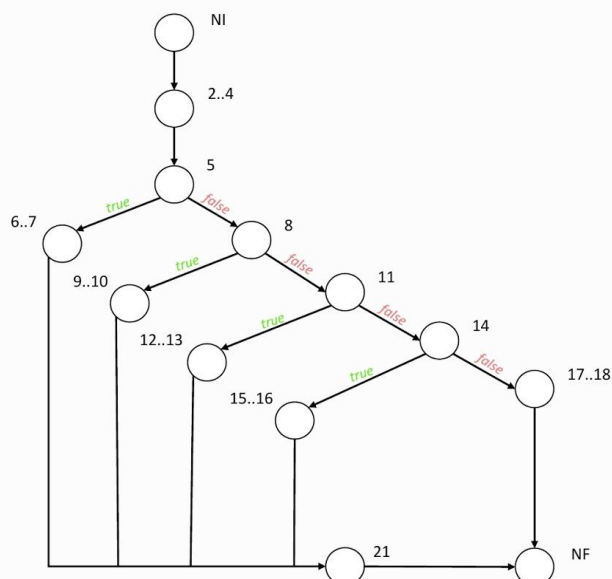
```

4.2.2 Testing whitebox

Il metodo ha il seguente corpo:

```
1 public static int prezzoDaStringaAIntero(String prezzo) {
2     int risultato;
3
4     switch (prezzo) {
5         case "€":
6             risultato = 1;
7             break;
8         case "€€":
9             risultato = 2;
10            break;
11         case "€€€":
12             risultato = 3;
13             break;
14         case "€€€€":
15             risultato = 4;
16             break;
17         default:
18             throw new RuntimeException("Errore di conversione: \""+prezzo+"\" non
==          rappresenta un prezzo valido!");
19     }
20
21     return risultato;
22 }
```

Il GFC ottenuto dal codice è il seguente:



In questo specifico caso, si noti che i test cases utilizzati per il test black box di fatto compiono già una Node e Branch Coverage del grafo. Nello specifico:

- `prezzoValido1()` copre il path $2..4 \rightarrow 5 \rightarrow 6..7 \rightarrow 21 \rightarrow \text{NF}$
- `prezzoValido2()` copre il path $2..4 \rightarrow 5 \rightarrow 8 \rightarrow 9..10 \rightarrow 21 \rightarrow \text{NF}$
- `prezzoValido3()` copre il path $2..4 \rightarrow 5 \rightarrow 8 \rightarrow 11 \rightarrow 12..13 \rightarrow 21 \rightarrow \text{NF}$
- `prezzoValido4()` copre il path $2..4 \rightarrow 5 \rightarrow 8 \rightarrow 11 \rightarrow 14 \rightarrow 15..16 \rightarrow \text{NF}$
- `prezzoNonValido()` $2..4 \rightarrow 5 \rightarrow 8 \rightarrow 11 \rightarrow 14 \rightarrow 17..18 \rightarrow \text{NF}$

4.3 Testing del metodo aggiungiValoreAMedia

Il metodo appartiene all'activity *SchermataDettagliStruttura* e il suo scopo è quello di aggiungere un nuovo valore ad una media senza doverla ricalcolare, quindi senza dover conoscere tutti i valori che la compongono. In caso di errore lancia *RuntimeException*.

La signature è la seguente:

```
public float aggiungiValoreAMedia(float vecchiaMedia, int numeroValoriVecchiaMedia, int nuovoValore)
```

Parametri di input:

- float vecchiaMedia: la media a cui si vuole aggiungere il nuovo valore.
- int numeroValoriVecchiaMedia: numero valori aggreganti la vecchia media.
- int nuovoValore: valore da aggiungere alla media.

Output:

- Media comprensiva del nuovo valore.

Esempio d'uso

USO	OUTPUT
aggiungiValoreAMedia(3,2,3)	3
aggiungiValoreAMedia(0,0,2)	2
aggiungiValoreAMedia(2.3,30,7000)	RuntimeException: "Il nuovo valore deve essere compreso tra 1 e 5"
aggiungiValoreAMedia(5,20,-77)	RuntimeException: "Il nuovo valore deve essere compreso tra 1 e 5"
aggiungiValoreAMedia(2,-500,3)	RuntimeException: "Il numero dei valori della vecchia media non può essere negativo"
aggiungiValoreAMedia(70.23,22,3)	RuntimeException: "La vecchia media deve essere compresa tra 0 e 5"
aggiungiValoreAMedia(-10,1,4)	RuntimeException: "La vecchia media deve essere compresa tra 0 e 5"

4.3.1 Testing blackbox

Segue la definizione delle classi di equivalenza per ciascun parametro.

Parametro **vecchiaMedia**

- CE1: {Float.MIN_VALUE ... -1} non valida
- CE2: {0 ... 5} valida
- CE3: {6 ... Float.MAX_VALUE} non valida

Parametro **numeroValoriVecchiaMedia**

- CE4: {Integer.MIN_VALUE ... -1} non valida
- CE5: {0 ... Integer.MAX_VALUE} valida

Parametro **nuovoValore**:

- CE6: {Integer.MIN_VALUE ... 0} non valida
- CE7: {1 ... 5} valida
- CE8: {6 ... Integer.MAX_VALUE} non valida

Si è scelto di adottare una strategia WECT, pertanto è stato effettuato un test case per ogni classe non valida e uno che massimizzi i casi di validità. Pertanto, l'elenco dei test cases è il seguente:

- aggiungiValoreAMediaMinoreDiZero()
- aggiungiValoreAMediaConNumeroValoriMinoreDiZero()
- aggiungiValoreMinoreDiZeroAMedia()
- aggiungiValoreMaggioreDiCinqueAMedia()
- aggiungiValoreAMediaConParametriValidi()

```
. public class AggiungiValoreAMediaUnitTest {  
.     /*  
.         Il metodo aggiunge un nuovo valore ad una media senza doverla ricalcolare, .  
.         quindi senza dover conoscere  
.         tutti i valori che la compongono. In caso di errore lancia RuntimeException.  
.     */  
.     SchermataDettagliStruttura schermataDettagliStruttura;
```

```

.
.
.   @Before
.   public void init(){
.       schermataDettagliStruttura = new SchermataDettagliStruttura();
.   }
.
.
.   @Test(expected = RuntimeException.class)
.   public void aggiungiValoreAMediaMinoreDiZero() {
.       schermataDettagliStruttura.aggiungiValoreAMedia(-10,1,4);
.       fail();
.   }
.
.
.   @Test(expected = RuntimeException.class)
.   public void aggiungiValoreAMediaMaggioreDiCinque() {
.       schermataDettagliStruttura.aggiungiValoreAMedia(70.23f,22,3);
.       fail();
.   }
.
.
.   @Test(expected = RuntimeException.class)
.   public void aggiungiValoreAMediaConNumeroValoriMinoreDiZero() {
.       schermataDettagliStruttura.aggiungiValoreAMedia(2,-500,3);
.       fail();
.   }
.
.
.   @Test(expected = RuntimeException.class)
.   public void aggiungiValoreMinoreDiZeroAMedia() {
.       schermataDettagliStruttura.aggiungiValoreAMedia(5,20,-77);
.       fail();
.   }
.
.
.   @Test(expected = RuntimeException.class)
.   public void aggiungiValoreMaggioreDiCinqueAMedia() {
.       schermataDettagliStruttura.aggiungiValoreAMedia(2.3f,30,7000);
.       fail();
.   }
.

```

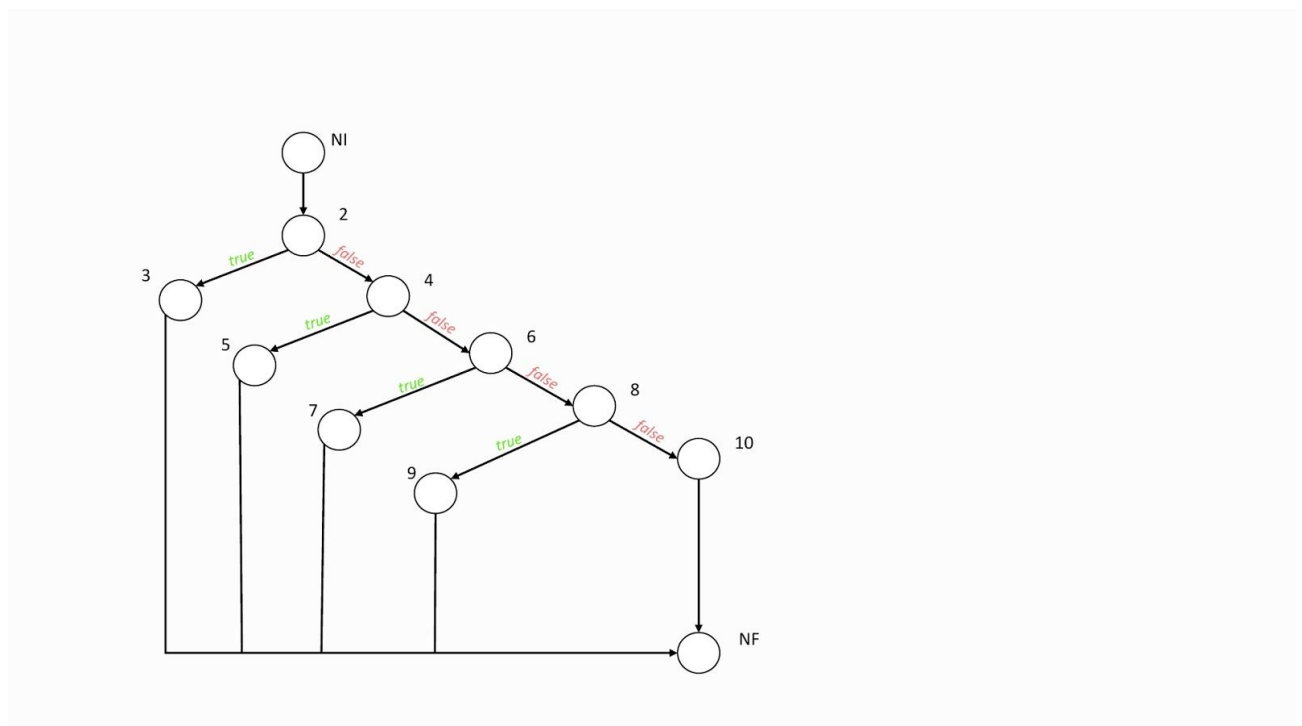
```
.    @Test
.    public void aggiungiValoreAMediaConParametriValidi() {
.        float nuovaMedia = schermataDettagliStruttura.aggiungiValoreAMedia(3,2,3);
.        assertEquals(3,nuovaMedia,0.1);
.    }
. }
```

4.3.2 Testing whitebox

Il metodo ha il seguente corpo:

```
1  public float aggiungiValoreAMedia(float vecchiaMedia, int numeroValoriVecchiaMedia, int nuovoValore){
2      if(numeroValoriVecchiaMedia < 0)
3          throw new RuntimeException("Il numero dei valori della vecchia media non può essere negativo");
4      if(vecchiaMedia < 0 || vecchiaMedia > 5)
5          throw new RuntimeException("La vecchia media deve essere compresa tra 0 e 5");
6      if(nuovoValore < 1 || nuovoValore > 5)
7          throw new RuntimeException("Il nuovo valore deve essere compreso tra 1 e 5");
8      if(numeroValoriVecchiaMedia == 0)
9          return nuovoValore;
10     return (vecchiaMedia + ((nuovoValore-vecchiaMedia)/(numeroValoriVecchiaMedia+1)));
11 }
```

Il GFC ottenuto dal codice è il seguente:



Si noti che i test cases della strategia blackbox coprono:

- 8 archi su 14, quindi una Branch Coverage all'86%.
- 10 nodi su 11, quindi una Node Coverage al 91%.

Aggiungendo un test case **aggiungiValoreAMediaConNumeroValoriZero()**, in cui tutti i parametri sono validi e in particolare il parametro `numeroValoriVecchiaMedia` è uguale a 0, si ottiene un 100% di Node Coverage, Branch Coverage e Condition Coverage.

```
.    @Test
.    public void aggiungiValoreAMediaConNumeroValoriZero() {
.        float nuovaMedia = schermataDettagliStruttura.aggiungiValoreAMedia(0,0,2);
.        assertEquals(2,nuovaMedia,0.1);
.    }
```