

UNIVERSITÀ DI PISA

COMPUTER ENGINEERING

SOFTWARE SYSTEM ENGINEERING

**Sharing Everywhere
Documentazione**

Autori:

Martina TROSCIA
Gionatan GALLO
Daniele MODICA
Chiara CAIAZZA

Supervisori:

G. VAGLINI
M. CIMINO

27 giugno 2017



INDICE

| | |
|--|----|
| Introduzione..... | v |
| Capitolo 1: Modello del processo..... | 6 |
| 1.1 Modello a cascata | 6 |
| Capitolo 2: Stima delle ore | 8 |
| 1.1 Modello..... | 8 |
| 1.2 Stima del numero di righe di codice | 8 |
| 1.3 Scale Driver | 9 |
| 1.4 Cost Driver | 10 |
| 1.5 Risultato..... | 11 |
| 1.6 Rettifica alla stima delle ore | 12 |
| Capitolo 3: Analisi dei rischi | 13 |
| 3.1 Identificazione dei rischi | 13 |
| 3.2 Analisi dei rischi | 14 |
| 3.3 Piano per minimizzare gli effetti dei rischi..... | 15 |
| 3.4 Monitoraggio dei rischi..... | 16 |
| Capitolo 4: Software Requirements Specification..... | 17 |
| 4.1 Introduzione | 17 |
| 4.2 Descrizione generale | 18 |
| 4.3 Specifiche sulle interfacce esterne | 20 |
| 4.4 Caratteristiche del sistema | 21 |
| 4.4.1 Requisiti funzionali | 21 |
| 4.4.2 Requisiti non funzionali | 27 |
| Capitolo 5: Piano sulla qualità | 30 |
| 5.1 Introduzione | 30 |
| 5.2 Suddivisione del lavoro..... | 30 |
| 5.3 Milestone | 31 |
| 5.4 Modello ISO – IEC 9126 | 32 |
| 5.5 Gestione dei rischi | 33 |
| 5.5.1 Rischi sulla sicurezza | 34 |
| Capitolo 6: Analisi..... | 36 |
| 6.1 Organizzazione della fase | 36 |

| | | |
|-------|--|----|
| 6.2 | Casi d'uso | 38 |
| 6.2.1 | Offerta casa | 39 |
| 6.2.2 | Richiesta casa | 47 |
| 6.2.3 | Attrazioni | 59 |
| 6.2.4 | Servizi accessori..... | 67 |
| 6.2.5 | SharingEverywhere..... | 73 |
| 6.3 | Classi | 76 |
| 6.3.1 | Offerta casa | 76 |
| 6.3.2 | Richiesta casa | 77 |
| 6.3.3 | Attrazioni | 78 |
| 6.3.4 | Servizi accessori..... | 79 |
| | Capitolo 7: Design | 80 |
| 7.1 | Organizzazione della fase | 80 |
| 7.2 | Diagrammi SoaML delle interfacce dei servizi | 82 |
| 7.2.1 | Offerta casa | 82 |
| 7.2.2 | Richiesta casa | 82 |
| 7.2.3 | Attrazioni | 83 |
| 7.2.4 | Servizi accessori..... | 83 |
| 7.3 | Diagramma di sequenza SoaML per il protocollo | 84 |
| 7.4 | Diagrammi UML interface-like delle classi..... | 87 |
| 7.4.1 | Offerta casa | 87 |
| 7.4.2 | Richiesta casa | 87 |
| 7.4.3 | Attrazioni | 87 |
| 7.4.4 | Servizi accessori..... | 88 |
| 7.5 | Diagrammi strutturali delle classi e degli oggetti | 88 |
| 7.5.1 | Offerta casa | 88 |
| 7.5.2 | Richiesta casa | 89 |
| 7.5.3 | Attrazioni | 91 |
| 7.5.4 | Servizi accessori..... | 92 |
| | Capitolo 8: Implementazione | 94 |
| 8.1 | Organizzazione della fase | 94 |
| 8.1.1 | Tipi di attività | 96 |

| | | |
|-------------|--|--|
| 8.2 | Classi Java per le interfacce | 97 |
| 8.2.1 | Interfaccia Offerta casa | 97 |
| 8.2.2 | Interfaccia Richiesta casa | 97 |
| 8.2.3 | Interfaccia Attrazioni | 98 |
| 8.2.4 | Interfaccia Servizi accessori..... | 98 |
| 8.3 | File XSD e XML | 99 |
| 8.3.1 | Offerta casa | 99 |
| 8.3.2 | Richiesta casa | 103 |
| 8.3.3 | Attrazioni | 108 |
| 8.3.4 | Servizi accessori..... | 112 |
| 8.4 | Orchestrazione locale in codice Java | 114 |
| 8.5 | Diagramma di sequenza generato dal codice Java dell'orchestrazione | 121 |
| 8.5.1 | Confronto tra R e K..... | 124 |
| 8.6 | Axis WS | 125 |
| 8.6.1 | WSDL Offerta casa..... | 125 |
| 8.6.2 | WSDL Richiesta casa | 128 |
| 8.6.3 | WSDL Attrazioni..... | 137 |
| 8.6.4 | WSDL Servizi accessori | 143 |
| 8.7 | Messaggi XML Soap generati per ogni caso di test..... | 147 |
| 8.7.1 | Offerta casa | 147 |
| 8.7.2 | Richiesta casa | 148 |
| 8.7.3 | Attrazioni | 150 |
| 8.7.4 | Servizi accessori..... | 153 |
| Capitolo 9: | Piano di test | Errore. Il segnalibro non è definito. |
| 9.1 | Introduzione | 154 |
| 9.2 | Oggetto del test | 154 |
| 9.3 | Tipi e tecniche di testing | 154 |
| 9.3.1 | Test d'unità..... | 154 |
| 9.3.2 | Test d'integrazione..... | 155 |
| 9.4 | Criteri d'inizio e di fine test..... | 157 |
| 9.5 | Deliverable..... | 157 |
| 9.6 | Responsabilità..... | 157 |

| | |
|---|-----|
| 9.7 Rischi..... | 158 |
| Capitolo 10: Testing..... | 159 |
| 10.1 Organizzazione della fase..... | 159 |
| 10.2 Test d'unità | 161 |
| 10.2.1 Offerta casa | 161 |
| 10.2.2 Richiesta casa | 164 |
| 10.2.3 Attrazioni | 171 |
| 10.2.4 Servizi accessori..... | 175 |
| 10.2.5 Commenti ai test d'unità | 177 |
| 10.3 Logica di Coordinamento: Black-box testing | 178 |
| 10.3.1 Black-box testing | 178 |
| 10.3.2 White-box testing..... | 190 |
| Capitolo 11: Manutenibilità | 192 |
| 11.1 Manutenzione del progetto..... | 192 |
| 11.2 Indice di gestibilità..... | 192 |
| 11.2.1 Metrica di complessità di Halstead | 192 |
| 11.2.2 Complessità ciclomatica di Mc Cabe | 193 |
| 11.2.3 Numero di linee di codice..... | 193 |
| 11.3 Risultato..... | 194 |
| Capitolo 8: Diagramma di Gantt..... | 195 |
| 12.1 Commenti al diagramma di Gantt | 196 |

Introduzione

Questa documentazione descrive il lavoro di gruppo per il progetto di Software System Engineering del corso di Computer Engineering dell'Università di Pisa. Verranno discusse tutte le principali fasi di sviluppo del progetto: analisi, design, implementazione, integrazione e testing. Il gruppo di lavoro consiste di quattro studenti, ognuno dei quali ha svolto un ruolo specifico:

Martina Troscia Project Manager

Gionatan Gallo Tool Specialist

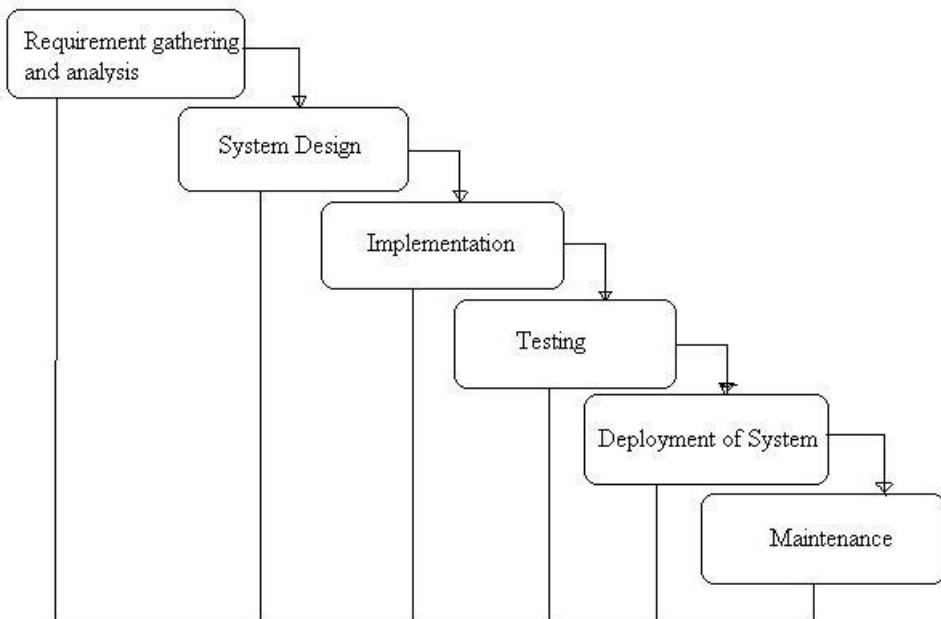
Daniele Modica Librarian

Chiara Caiazza Quality Engineer

Capitolo 1

Modello del processo

1.1 Modello a cascata



1.1.1 Descrizione

Il modello a cascata è un processo di design sequenziale, usato nei processi di sviluppo dei software. Secondo questo modello, il processo di realizzazione del software è strutturato in sequenza lineare di fasi o passi che comprendono l'analisi dei requisiti, design, sviluppo, collaudo e manutenzione. Il modello a cascata prevede le seguenti fasi:

1. **Studio di fattibilità:** ha lo scopo di determinare se intraprendere lo sviluppo del sistema
2. **Analisi dei requisiti:** ha lo scopo di determinare cosa farà il sistema
3. **Design:** ha lo scopo di determinare come il sistema farà quanto stabilito nella prima fase, e in particolare la sua suddivisione in moduli e le relazioni fra di essi
4. **Codifica:** creazione dei moduli con un linguaggio di programmazione
5. **Collaudo:** esecuzione di prove per verificare la correttezza dell'implementazione dei singoli moduli

6. **Integrazione:** esecuzione di prove per verificare la correttezza del funzionamento complessivo del sistema
7. **Manutenzione:** comprende tutte le attività volte a migliorare, estendere e correggere il sistema nel tempo

Il modello afferma che si debba passare alla fase successiva solo quando quella precedente è stata riesaminata e verificata.

1.1.2 Motivazioni della scelta

Per lo sviluppo del progetto è stato scelto il modello a cascata. Poiché questo prevede che non si possa procedere alla fase successiva senza che la precedente sia stata analizzata completamente e con attenzione, il team è portato a concentrarsi sull'attività che sta svolgendo, minimizzando la possibilità che si accorga di non aver considerato qualche aspetto importante e che siano necessari dei cambiamenti. In questo modo il processo di sviluppo è analizzato di fase in fase e i passi che devono essere seguiti vengono scoperti in modo incrementale.

Il problema di questo modello è che non si adatta bene a cambiamenti delle specifiche da parte dei committenti ma nel nostro caso le specifiche sono state definite all'inizio e aggiustate nella fase di analisi quindi una volta conclusa questa fase non dovrebbero più esserci problemi.

Capitolo 2

Stima delle ore

Per stimare i costi per ogni fase del progetto in termini di uomo/mese è stato utilizzato il software SystemStar. Tutte le scelte nella pianificazione del lavoro sono state guidate da questo strumento.

2.1 Modello

Il modello usato per la stima è COCOMO II.2000 con fasi tradizionali. La stima di uomo/mese M viene effettuata con la seguente formula:

$$M = C \cdot P^S \cdot EAF$$

dove C è un fattore di complessità (moltiplicatore), P è la dimensione del prodotto (di solito espressa in numero di righe di codice), S è l'esponente di complessità dovuto agli scale driver e EAF è un fattore che tiene conto dei cost driver.

2.2 Stima del numero di righe di codice

2.2.1 Function point adjustment worksheet

I valori sono stati scelti in modo tale che il fattore di adattamento sia basso poiché non è richiesto di dover scrivere molte righe di codice per il progetto.

- Comunicazioni dei dati: 2.0 – intermedie
- Funzioni distribuite: 0 – nessuna
- Performance: 1.0 – moderata
- Configurazioni usate pesantemente: 0 – nessuna
- Alto tasso di transazioni: 0 – nessuno
- Entrate on-line di dati: 0 – nessuna
- Efficienza dell'utente finale: 1.0 – intermedia
- Aggiornamento on-line: 1.0 – insignificante
- Elaborazione complessa: 0 – nessuna
- Riusabilità del codice: 1.0 – insignificante
- Facilità di installazione: 1.0 – intermedia
- Facilità di utilizzo: 1.0 – intermedia
- Siti multipli: 0 – nessuno
- Facilità nell'introdurre cambiamenti: 1.0 – media

Il fattore calcolato è 0.74.

2.2.2 Linee per function point

Le linee per function point sono 53 (Java). Nel progetto non è utilizzato solo il linguaggio Java, ma 53 è uno dei valori più bassi che poteva essere scelto.

2.2.3 Impostazioni per i function point

- Input esterni: 2 di media complessità (il sistema necessita di un'Offerta e di una Richiesta per poter generare un abbinamento)
- Output esterni: 1 complesso (il sistema restituisce una Richiesta di condivisione contenente l'offerta selezionata, una proposta di attività e una pianificazione dei servizi accessori scelti)

2.2.4 Risultato

Sono state calcolate 588 linee di codice.

2.3 Scale Driver

I cinque scale driver determinano l'esponente S nelle equazioni usate per la stima. L'equazione usata in COCOMO generalmente ha un esponente maggiore di 1.0: ciò corrisponde a uno svantaggio dal punto di vista economico, che significa che la produttività è più bassa in progetti più grandi.

- **PREC – Novità:** Il nuovo progetto è paragonabile a altri progetti che il tuo team ha fatto prima?
Molto basso – Assolutamente senza precedenti.
- **FLEX – Flessibilità di sviluppo:** Le tue specifiche sono flessibili o devi rispettarle tutte?
Nominale – Un po' di flessibilità è ammessa.
- **RESL – Decisione dell'architettura/rischi:** Fino a che punto hai già definito l'architettura?
Molto alto – Completamente (100%). (Si considera che gli applicativi sono già stati sviluppati e devono solo essere orchestrati.)
- **TEAM – Coesione del team:** Come descriveresti la relazione tra le parti interessate?
Nominale – Fondamentalmente cooperativa.
- **PMAT – Maturità del processo:** Come è valutato il punteggio della tua organizzazione in base a SEI Maturity Scale?
Nominale – SEI CMM Level 2. “Ripetibile”: i processi di base sono stati stabiliti. (Il processo è documentato in modo sufficiente a poter provare a ripetere gli stessi passi.)

2.4 Cost Driver

Insieme alla dimensione e agli scale driver, COCOMO II include diciassette altri parametri di input chiamati cost driver. I cost driver includono i fattori principali che influenzano gli sforzi richiesti per portare a termine il progetto e riguardano personale, progetto, piattaforma e attributi del prodotto.

I cost driver vengono impostati per descrivere come il nuovo progetto si differenzia da ogni altro progetto software che sia mai stato fatto in precedenza. Ogni cost driver ha un impatto moltiplicativo sul risultato finale.

- **ACAP – Capacità dell'analista:** Quanto sono capaci gli analisti di questo progetto?
Basso – 35° percentile.
- **APEX – Esperienza nelle applicazioni:** Quanta esperienza ha il tuo team con questo tipo di applicazioni?
Molto basso – Meno di 2 mesi.
- **PCAP – Abilità del programmatore:** Quanto sono capaci i programmatore di questo progetto?
Molto alto – 90° percentile (il miglior team). (Nel progetto non c'è bisogno di scrivere molte righe di codice.)
- **PLEX – Esperienza con la piattaforma:** Quanta esperienza ha il team con la piattaforma?
Alto – 3 anni. (La piattaforma non deve essere gestita.)
- **LTEX – Esperienza con il linguaggio e gli strumenti:** Quanta esperienza ha il team con il linguaggio e gli strumenti?
Basso – 6 mesi.
- **PCON – Continuità del personale:** Qual è il tasso annuale di ricambio per la tua organizzazione?
Molto alto – 3% di ricambio all'anno. (Non ci sono cambi nella composizione del team.)
- **TOOL – Uso di strumenti software:** Che strumenti utilizza il tuo team?
Nominale – Strumenti per il ciclo di vita di base, moderatamente integrati.
- **SITE – Sviluppo in più sedi:** Il team è diviso in molte sedi? Le sedi come comunicano tra loro?
Estremamente alto – Collocati in un'unica sede. Multimedia interattivo.
- **SCED – Piano di sviluppo:** Il piano di sviluppo è compresso rispetto al piano nominale (di default)?
Molto basso – 75% del piano di sviluppo nominale. (“Molto basso” viene utilizzato per indicare la più veloce soluzione che sia anche fattibile.)
- **TIME – Vincoli sul tempo di esecuzione:** Quanta CPU richiede il tuo software?
Nominale – Uso minore del 50% del tempo di esecuzione a disposizione.
- **STOR – Vincoli sullo spazio di archiviazione principale:** Quanta memoria richiede il tuo software?
Nominale – Uso minore del 50% dello spazio di archiviazione a disposizione.

- **PVOL – Volatilità della piattaforma:** Quanto spesso cambia la piattaforma (OS, DBMS, ecc.)?
Basso – I maggiori cambiamenti avvengono ogni 12 mesi; cambiamenti minori ogni mese.
- **RELY – Affidabilità richiesta:** Quali sono le conseguenze di un guasto software?
Basso – Poche perdite facilmente recuperabili.
- **DATA – Dimensione del database:** Quanti dati sono necessari per testare il software?
Nominale – I byte del database e/o SLOC sono tra 10 e 100.
- **CPLX – Complessità del prodotto:** Quanto è complesso il software?
Molto basso – Codice lineare, computazioni semplici, semplice I/O.
- **RUSE – Riusabilità richiesta:** Stai progettando le tue componenti software perché siano riutilizzabili?
Basso – Nessuna. (Sviluppare per consentire il riuso è più costoso.)
- **DOCU – Corrispondenza della documentazione ai bisogni del ciclo di vita:** Quanta documentazione stai producendo?
Nominale – Della dimensione adatta ai bisogni del ciclo di vita.

2.5 Risultato

Di seguito sono riportati i valori che sono stati restituiti dal software SystemStar.

| Fase | Uomo/Mese | Durata |
|---------------------------------|----------------|---|
| RQ – Specifiche | 0,1 M/M | $0,1 \cdot 152 = 15,2 \text{ h}$ |
| PD – Design del prodotto | 0,2 M/M | $0,2 \cdot 152 = 30,4 \text{ h}$ |
| DD – Design dettagliato | 0,3 M/M | $0,3 \cdot 152 = 45,6 \text{ h}$ |
| CT – Codice e Test d'unità | 0,3 M/M | $0,3 \cdot 152 = 45,6 \text{ h}$ |
| IT – Integrazione e Test finale | 0,2 M/M | $0,2 \cdot 152 = 30,4 \text{ h}$ |
| Totale | 1,1 M/M | $1,1 \cdot 152 = 167,2 \text{ h}$ |

La durata è calcolata considerando le ore di lavoro al mese per una persona che lavora a tempo pieno = 152h (come suggerito dal sito web di SystemStar).

| Nostre fasi | Fasi su SystemStar | Durata |
|-----------------------|--------------------|-----------------------------|
| Analisi | RQ + PD | $6 + 12 - 1 = 17 \text{ h}$ |
| Design | DD | $16 - 1 = 15 \text{ h}$ |
| Implementazione | CT | $16 - 2 = 14 \text{ h}$ |
| Test | IT | $10 - 1 = 9 \text{ h}$ |
| Documentazione finale | | 5 h |
| Totale | | 60 h |

La durata restituita dal software è stata aggiustata basandosi sulle 60h imposte dal progetto. Il risultato ottenuto sono il numero di ore di lavoro individuale o collettivo richiesto a ogni membro del gruppo per ogni fase.

È stata rimossa un'ora o due da ogni fase per riservare del tempo alla stesura della documentazione finale.

2.6 Rettifica alla stima delle ore

Dopo la prima fase di analisi che ha richiesto anche più del tempo che le era stato riservato, siamo riusciti a portare a termine tutte le altre fasi in meno ore di quelle previste. Pertanto ho deciso di ripartire le ore in eccesso per fare una seconda breve iterazione di controllo e sistemare alcuni errori nelle fasi precedenti che sono emersi mentre procedevamo con il lavoro.

Inoltre, ci siamo resi conto che gran parte della documentazione andava completata al termine di una fase perché necessaria alle successive. Quindi ho pensato che 4 ore sono sufficienti per portare a termine quello che resta della documentazione finale.

La ripartizione delle ore per la seconda iterazione è la seguente:

| Fase | Ore |
|-----------------------|-------|
| Analisi | 4 h |
| Design | 2,5 h |
| Implementazione | 2,5 h |
| Test | 1 h |
| Documentazione finale | 4 h |

Questa ripartizione è da utilizzare solo come un'indicazione poiché al termine della prima iterazione c'erano delle lievi variazioni sulle ore rimaste ai membri del team (dalle 17h alle 13h).

Capitolo 3

Analisi dei rischi

Prima di procedere con la fase di analisi, è necessario analizzare i rischi. È infatti fondamentale, per tutte le fasi da realizzare, conoscere e prevedere quali siano gli eventi che potrebbero incidere negativamente sulla realizzazione del progetto. I rischi possono essere:

- Superamento del costo inizialmente stimato
- Prodotto finale non soddisfacente
- Tempo di programmazione non rispettato
- Test per individuare la presenza di fallimenti approssimati e non effettuati su larga scala

La probabilità che un rischio si verifichi è influenzata da diversi fattori come per esempio requisiti incompleti o mancanza di risorse. Queste condizioni devono essere tenute in considerazione durante lo sviluppo del progetto per evitare che si possano verificare. La gestione dei rischi richiede la loro identificazione e la stesura di un piano al fine di minimizzare gli effetti negativi nel caso in cui si verifichino. L'analisi dei rischi è pertanto riportata di seguito.

3.1 Identificazione dei rischi

3.1.1 Rischi tecnologici

- 1 Il servizio da sviluppare deve interagire con altri quattro moduli. Dal momento che non si può chiedere che vengano realizzate delle interfacce apposite, il servizio deve adattarsi a quello che viene messo a disposizione.
- 2 Il servizio da sviluppare deve essere in grado di lavorare in un contesto ristretto. Se in futuro il numero di utilizzatori crescerà, è possibile che il servizio debba essere reso più scalabile.

3.1.2 Rischi sulla persona

- 3 Ogni membro del gruppo ricopre un ruolo specifico e, in caso di una sua assenza o ritardo, ci sono delle attività che non possono essere portate a termine da altri. Ciò provoca un rallentamento del lavoro nel progetto e può causare ritardi anche per gli altri membri.
- 4 Le attività collettive devono essere svolte in presenza di tutti i membri e l'assenza di uno di essi impedisce di poter procedere, rallentando l'intero lavoro.

3.1.3 Rischi sul gruppo

- 5 Durante un'attività di gruppo, possono sorgere delle discussioni che deviano dal problema principale e questo porta a ritardi in quella determinata fase.

6 Possono sorgere incomprensioni tra i membri del gruppo che impediscono un clima di lavoro sereno e il raggiungimento degli obiettivi del progetto.

3.1.4 Rischi sugli strumenti utilizzati

7 La scarsa conoscenza dei software utilizzati può portare a un uso inefficiente degli strumenti e a errori o ritardi che potrebbero essere evitati.
8 Una scarsa integrazione del software può portare problemi nel risultato finale.

3.1.5 Rischi sulle specifiche

9 Requisiti poco chiari possono portare a dover modificare molte volte il lavoro svolto prima di arrivare alla giusta versione. Ogni revisione ha un effetto importante sulle varie fasi del progetto.
10 I committenti non riescono a capire che anche dei minimi cambiamenti nelle specifiche potrebbero portare a dover modificare diverse parti del lavoro.

3.1.6 Rischi sulle stime

11 Sottovalutare la dimensione del progetto porta a fare stime errate sui costi da affrontare e sul tempo che servirà a portare a termine il lavoro.

3.2 Analisi dei rischi

L'obiettivo di questa analisi è di evidenziare quale sia la probabilità che si verifichino i rischi identificati in precedenza e la loro gravità.

La probabilità che un rischio si verifichi può essere:

- Molto bassa
- Bassa
- Moderata
- Alta
- Molto alta

Le conseguenze possono essere:

- Catastrofiche
- Serie
- Tollerabili
- Insignificanti

I rischi sono stati ordinati in modo da mettere in evidenza quali siano quelli con le conseguenze più importanti.

| Rischio | Probabilità che si verifichi | Conseguenze |
|---|------------------------------|---------------|
| Sui requisiti: scarsa chiarezza (9) | Alta | Catastrofiche |
| Sui requisiti: interazione con i clienti (10) | Alta | Catastrofiche |
| Sulle stime: stime errate (11) | Alta | Catastrofiche |
| Tecnologico: adattamento (1) | Moderata | Serie |
| Tecnologico: scalabilità (2) | Bassa | Serie |
| Sugli strumenti: scarsa conoscenza (7) | Moderata | Serie |
| Sugli strumenti: scarsa integrazione (8) | Alta | Serie |
| Sulla persona: ritardi dovuti a un membro (3) | Moderata | Tollerabili |
| Sulla persona: attività collettive (4) | Moderata | Tollerabili |
| Sul gruppo: discussioni off-topic (5) | Moderata | Tollerabili |
| Sul gruppo: scarso affiatamento (6) | Bassa | Tollerabili |

3.3 Piano per minimizzare gli effetti dei rischi

Di seguito vengono mostrate alcune delle misure che possono essere adottate per evitare che si verifichi un rischio o per minimizzarne le conseguenze nel caso in cui si sia già verificato.

- **Rischi tecnologici:** l'idea è di rendere gli input e output il più standard possibili per integrare meglio i moduli dell'applicativo.
- **Rischi sulla persona:** la strategia è di usare cartelle condivise in cui ogni componente del team può mettere il risultato del proprio lavoro così che in caso di necessità un membro possa controllare il lavoro di un altro. L'uso di programmi come Skype o TeamViewer permette di lavorare anche quando non c'è la possibilità di incontrarsi direttamente.
- **Rischi sul gruppo:** l'idea è di definire in precedenza in maniera chiara di cosa si intende discutere in una certa attività collettiva e darsi dei tempi massimi entro cui deve essere portata a termine. I membri del gruppo devono sforzarsi di mantenere dei rapporti amichevoli per favorire la coesione del team.
- **Rischi sugli strumenti utilizzati:** l'utilizzo di programmi come Visual Paradigm guida nella creazione del codice perché consente di generarlo automaticamente partendo da modelli UML.
- **Rischi sulle specifiche:** è necessario analizzare a fondo il problema in esame per evitare di lasciare dubbi o incertezze sull'interpretazione del lavoro da svolgere. Pertanto è importante generare un SRS chiaro e dettagliato che non consenta il sorgere di fraintendimenti con i committenti. È necessario far capire loro come un certo cambiamento alle specifiche si ripercuoterà sui tempi e costi del progetto.
- **Rischi sulle stime:** la strategia è di usare software come SystemStar per avere un'idea di quanto tempo è richiesto per portare a termine una fase. Queste stime possono essere usate per organizzare il proprio lavoro e dovrebbero impedire che sorgano troppi problemi, se il team si sforza di seguire la pianificazione.

3.4 Monitoraggio dei rischi

È necessario tenere sotto controllo i rischi per capire se stanno diventando più probabili e pertanto se le strategie che erano state pensate per gestirli sono ancora efficaci o vanno riviste.

- **Indicatore di rischi tecnologici**
 - difficoltà nel passaggio dei parametri tra i diversi applicativi che devono scambiare informazioni e comunicare
- **Indicatore di rischi sulla persona**
 - difficoltà nel partecipare agli incontri a causa di problemi sulla connettività
 - difficoltà nell'utilizzo della cartella condivisa con la generazione di file in conflitto
- **Indicatore di rischi sul gruppo**
 - difficoltà nel rispettare le scadenze per la consegna dei documenti richiesti da una certa fase del progetto
 - difficoltà nel pianificare il lavoro e nel capire come si debba procedere
 - difficoltà di comunicazione tra i membri e rapporti poco amichevoli all'interno del team
- **Indicatore di rischi sugli strumenti utilizzati**
 - riluttanza dei membri nell'utilizzare dei software specifici per aiutarsi in una certa fase o attività
 - difficoltà nell'uso del software per ottenere risultati consistenti
- **Indicatore di rischi sulle specifiche**
 - difficoltà da parte dei clienti nel comprendere chiaramente i punti elencati nel documento SRS
 - frequente richiesta di cambiamenti al progetto da parte dei clienti
- **Indicatore di rischi sulle stime**
 - il team non riesce a rispettare le scadenze prefissate

Capitolo 4

Software Requirements Specification

Prima che le fasi di analisi e design possano iniziare, si deve avere un'idea di ciò che si cerca di realizzare e questo è lo scopo dei requisiti: scoprire e raggiungere un accordo su ciò che il sistema dovrebbe fare, utilizzando un linguaggio comprensibile all'utente del sistema.

Un requisito può essere definito come "una specifica di ciò che deve essere implementato".

Ci sono fondamentalmente due tipi di requisiti:

- Requisiti funzionali: descrivono le funzionalità del sistema e cosa dovrebbe offrire
- Requisiti non funzionali: una proprietà specifica o un vincolo sul sistema

I requisiti sono alla base di tutti i sistemi e dovrebbero solo essere una dichiarazione di ciò che il sistema dovrebbe fare, non di come dovrebbe essere fatto.

Un Software Requirements Specification (SRS) è una descrizione del sistema software che deve essere sviluppato. L'SRS stabilisce i requisiti funzionali e non funzionali e può includere un insieme di casi d'uso che descrivono le interazioni dell'utente con il software. Definisce inoltre le basi per un accordo tra i clienti e i contraenti o fornitori su quello che il prodotto finale deve fare così come quello che non si prevede di fare. L'SRS consente una rigorosa valutazione dei requisiti prima che la fase di design possa iniziare. Dovrebbe inoltre fornire una base realistica per valutare i costi e i rischi.

4.1 Introduzione

4.1.1 Scopo del documento

Lo scopo di questo documento è quello di effettuare una descrizione dettagliata dei requisiti di SharingEverywhere, un servizio nato allo scopo di facilitare la condivisione d'immobili a livello nazionale ed internazionale. Questo documento descrive i casi d'uso per ogni applicazione utilizzata dal servizio software.

SharingEverywhere è un web service che automatizza le principali azioni compiute per effettuare con successo una condivisione di un immobile. Il software utilizza quattro servizi che si occupano della pubblicazione delle offerte d'immobili, della pubblicazione delle richieste di condivisione, della generazione di una lista di servizi accessori attivabili e della generazione di una lista di attività d'intrattenimento che un richiedente potrebbe voler effettuare nella città di destinazione.

Ogni applicazione utilizzata fornisce interfacce specifiche che guidano gli utenti nella pianificazione del loro scambio in ogni sua fase.

4.1.2 Definizioni, acronimi ed abbreviazioni

- 1 **Agente:** colui che interagisce con i quattro applicativi al fine di pianificare uno scambio tra proprietario e richiedente.
- 2 **Proprietario:** colui che mette a disposizione, per future condivisioni, il proprio immobile.
- 3 **Richiedente:** colui che richiede un immobile in condivisione.
- 4 **Classe d'attrazione:** rappresenta la categoria di un'attrazione (per esempio: siti d'interesse, musei, ristoranti, locali notturni, ...).
- 5 **Proposte d'attività:** una lista di attività eseguibili in una data fascia oraria di un dato giorno.
- 6 **Lista di proposte d'attività:** un insieme di proposte attività per ogni fascia oraria di ogni giorno compreso nel periodo dello scambio.
- 7 **Limitazioni:** condizioni poste dal proprietario che devono essere rispettati dal richiedente qualora lo scambio venga accettato.
- 8 **Servizi accessori:** servizi attivabili opzionalmente per migliorare la permanenza del richiedente nell'immobile di destinazione.

4.2 Descrizione generale

4.2.1 Funzionalità del prodotto

I quattro applicativi che forniscono le funzioni fondamentali per automatizzare il processo di scambio sono i seguenti:

- *Offerta casa* è utilizzato sia dal proprietario che dall'agente. Il modulo fornisce un'interfaccia per permettere al proprietario di inserire un'offerta per il suo immobile. L'agente utilizza l'applicativo per visionare la lista di offerte riguardanti la destinazione desiderata dal richiedente e per comunicare al proprietario l'esito dello scambio.
- *Richiesta casa* è utilizzato sia dal richiedente che dall'agente. Il richiedente si interfaccia con il modulo per inserire la sua richiesta specificando la destinazione, il periodo di tempo e le preferenze. L'agente interagisce con il servizio per ottenere una richiesta da gestire e proporre una proposta di condivisione al richiedente.
- *Attrazioni* è utilizzato sia dal richiedente che dall'agente. Quest'ultimo utilizza il servizio per creare una lista di proposte d'attività da sottoporre al richiedente, che la visualizza attraverso il modulo. Il richiedente può interagire per richiedere una modifica della lista di proposta attività.
- *Servizi accessori* è utilizzato sia dal richiedente che dall'agente. Quest'ultimo interagisce con il modulo per creare una pianificazione di servizi accessori da sottoporre al richiedente, il quale può decidere se accettarla o meno.

4.2.2 Tipologie d'attore e caratteristiche

Tre tipologie di attori interagiscono con il sistema:

- L'*agente* utilizza tutti i quattro applicativi allo scopo di creare un abbinamento tra le offerte di condivisione poste dai proprietari e le richieste effettuate dai richiedenti.
- Il *proprietario* utilizza *Offerta casa* per pubblicare un'offerta relativa all'immobile che vuole offrire in condivisione.
- Il *richiedente* utilizza *Richiesta casa* per postare la sua richiesta di condivisione e per controllare le proposte effettuate dall'agente. Accede ai servizi *Attrazioni* e *Servizi accessori* per valutare le pianificazioni proposte dall'agente.

4.2.3 Vincoli di design e d'implementazione

I quattro applicativi sono accessibili agli attori attraverso siti web, pertanto le applicazioni devono essere implementate in modo che rispettino gli standard esistenti in materia di progettazione web così che siano raggiungibili attraverso la maggior parte dei web-browser.

L'utilizzo di tutti gli applicativi deve avvenire necessariamente tramite una connessione alla rete Internet. L'agente interagisce in modo continuativo con tutti gli applicativi, pertanto necessita di una connessione affidabile. Al contrario, il proprietario ed il richiedente accedono agli applicativi in maniera sporadica per verificare lo stato delle loro operazioni, pertanto non sono richiesti vincoli stringenti sulla loro connettività.

Tutti gli applicativi gestiscono i loro database separatamente. Quindi non scambiano informazioni in maniera diretta in quanto non vi è alcuna interazione tra di essi, ma possono comunicare indirettamente tramite l'agente.

L'applicativo finale è in grado di gestire non solo più proprietari e richiedenti allo stesso tempo, ma è stata considerata anche la possibilità di avere più agenti che lavorano in contemporanea.

I servizi *Offerta casa*, *Richiesta casa* ed *Attrazioni* possiedono un ulteriore database contenente le credenziali d'autenticazione.

4.2.4 Assunzioni e dipendenze

Assumiamo che i tre attori utilizzino un web browser compatibile con HTML5 e CSS3, in esecuzione su un pc con performance sufficientemente alte.

È stato supposto che i quattro applicativi siano indipendenti e che interagiscano tra di loro solo ed esclusivamente attraverso l'agente.

È stato supposto che il proprietario abbia effettuato il log-in al servizio *Offerta casa*, il richiedente abbia effettuato l'accesso ai servizi *Richiesta casa* ed *Attrazioni* e l'agente abbia effettuato il log-in a tutti e tre i servizi. *Servizi accessori* non richiede alcun meccanismo d'autenticazione.

Per concludere, è stato supposto che tutti gli attori si comportino nel massimo rispetto reciproco, evitando tutti i comportamenti che possano minare il meccanismo di scambio, che per sua natura è basato sui principi della fiducia e della condivisione. Essi quindi si impegnano ad effettuare le richieste e a valutare le proposte ricevute nei modi e nei tempi previsti dal sistema, senza abbandonare le operazioni in corso in maniera arbitraria ed ingiustificata.

4.3 Specifiche sulle interfacce esterne

4.3.1 Interfacce utente

Il proprietario inserisce l'offerta del proprio immobile tramite il servizio *Offerta casa*. Il richiedente compila una richiesta di condivisione tramite il servizio *Richiesta casa*, esprimendo le sue preferenze. Le interazioni con i due servizi vengono svolte in maniera asincrona.

L'agente utilizza il modulo *Richiesta casa* per ottenere la lista di richieste non gestite e ne sceglie una da prendere in carico. Interagendo con il modulo *Offerta casa* ottiene una la lista ordinata d'immobili, filtrata in base ai parametri espressi nella richiesta. Per ogni offerta l'agente interagisce con il modulo *Attrazioni* per procurarsi la lista d'attrazioni presenti nelle vicinanze dell'indirizzo di destinazione ed assegna un punteggio tenendo conto di quante classi d'attrazioni presenti nella richiesta sono presenti anche nella lista ottenuta. L'agente prende l'immobile con il maggior punteggio e genera una lista di proposte d'attività e una pianificazione di servizi accessori per proporli al richiedente. A questo punto l'agente gli suggerisce una proposta di condivisione.

Il richiedente visualizza l'offerta selezionata dall'agente tramite *Richiesta casa*, ottenendo anche l'accesso alla lista di attività e alla lista dei servizi accessori.

Il richiedente controlla il risultato della ricerca attraverso il servizio *Richiesta casa* e se non è soddisfatto del primo immobile presentato, può chiedere che venga visualizzato un altro risultato, pertanto la richiesta sarà ripresa in carico da parte dell'agente. Una volta che il richiedente ha accettato l'immobile proposto l'agente deve comunicarlo al proprietario, fornendo tutte le generalità del richiedente per permettere alle due parti di entrare in contatto per stipulare un contratto.

Il richiedente interagisce con il servizio *Attrazioni* per esprimere il proprio gradimento, che può essere visualizzato dall'agente. Fino a che il richiedente non accetta la lista di attività proposta, l'agente interagisce con il modulo aggiungendo alternative laddove ci sono fasce orarie vuote e rimuovendo le attività che non appartengono alle classi d'attrazioni richieste. L'agente analizza la lista finale di proposte d'attività approvate dal richiedente e utilizza le fasce orarie per richiedere al modulo *Servizi accessori* di generare una pianificazione. Il richiedente può visualizzare e accettare la pianificazione mostrata dall'agente.

4.4 Caratteristiche del sistema

4.4.1 Requisiti funzionali

4.4.1.1 Offerta casa

Proprietario

Il servizio permette al proprietario di inserire un'offerta per un suo immobile e di verificare se l'agente ha effettuato una proposta di condivisione.

Agente

L'agente deve poter accedere alle offerte pubblicate dai proprietari e notificare l'esito di uno scambio.

ID: OC1

TITOLO: Inserimento offerta

DESCRIZIONE: Il proprietario inserisce il periodo dello scambio, l'indirizzo (via, numero civico, CAP, città e stato), la tipologia dell'immobile, la foto, una descrizione dell'immobile, una descrizione della zona, una descrizione dei mezzi pubblici accessibili, la possibilità di utilizzare autoveicoli e le limitazioni presenti (indica se la casa è adatta ad ospitare bambini, se è possibile fumare all'interno, se è possibile portare animali e se è richiesto particolare silenzio in orari notturni). Il sistema non accetta offerte con campi obbligatori non compilati.

SCOPO: il sistema pubblica l'offerta inserita dal proprietario

DIPENDENZE: NESSUNA

ID: OC2

TITOLO: Ricerca offerte salvate

DESCRIZIONE: il sistema ricerca tutte le offerte appartenenti al proprietario autenticato. Questa operazione permette al proprietario di accedere nuovamente alle offerte poste in precedenza. Il sistema controlla che esistano richieste appartenenti al proprietario.

SCOPO: mostrare la lista delle offerte appartenenti all'utente

DIPENDENZE: NESSUNA

ID: OC3

TITOLO: Visualizza offerte

DESCRIZIONE: il proprietario sceglie un'offerta dalla sua lista di offerte al fine di visualizzarla.

SCOPO: visualizzare un'offerta

DIPENDENZE: OC2

ID: OC4

TITOLO: Ricerca case

DESCRIZIONE: l'agente compila una form inserendo l'indirizzo e le sue preferenze, per ricercare le offerte pertinenti ai parametri inseriti. L'indirizzo può anche non essere completo ma composto solo dalla città. L'agente ha la possibilità di visualizzare le offerte trovate dal sistema.

SCOPO: trovare le offerte presenti in una zona circoscritta

DIPENDENZE: NESSUNA

ID: OCS

TITOLO: Comunica scambio

DESCRIZIONE: L'agente accede per sottoporre al proprietario una proposta, inserendo le generalità del richiedente, l'indirizzo e-mail, il numero di telefono e di cellulare. Il sistema non accetta proposte che non contengano almeno un contatto valido.

SCOPO: inserire un contatto che permetta al proprietario di contattare in futuro il richiedente

DIPENDENZE: OC4

4.4.1.2 Servizi Accessori

Richiedente

Il richiedente può visualizzare e selezionare i servizi accessori a cui è interessato. Egli può accettare o meno la pianificazione offerta dal sistema.

Agente

L'agente può visualizzare i servizi disponibili in una determinata zona e per ogni servizio inserisce la fascia oraria e chiede al sistema di generare una pianificazione.

ID: SA1

TITOLO: Seleziona servizi accessori

DESCRIZIONE: il richiedente visualizza e seleziona i servizi a cui è interessato.

SCOPO: avere una lista di servizi scelti dal richiedente

DIPENDENZE: NESSUNA

ID: SA2

TITOLO: Rispondi pianificazione

DESCRIZIONE: il richiedente visualizza la pianificazione proposta dal sistema e sceglie se accettarla o meno.

SCOPO: permette al richiedente di accettare o rifiutare la pianificazione.

DIPENDENZE: NESSUNA

ID: SA3

TITOLO: Richiedi lista servizi accessori

DESCRIZIONE: L'agente inserisce un indirizzo (Via, Numero Civico, Città, CAP, Stato) per ricercare una lista di servizi accessori.

SCOPO: permette all'agente di visualizzare una lista di servizi in una determinata zona.

DIPENDENZE: NESSUNA

ID: SA4

TITOLO: Crea pianificazione

DESCRIZIONE: L'agente visualizza la lista dei servizi scelti dal richiedente e per ogni servizio inserisce le fasce orarie in cui saranno disponibili.

SCOPO: permette di creare una pianificazione di servizi.

DIPENDENZE: NESSUNA

4.4.1.3 Attrazioni

Richiedente

Il servizio permette al richiedente di consultare una lista di proposte d'attività, visualizzare nel dettaglio le attrazioni presenti e rimuoverle se non sono di suo gradimento. Può accettare la lista di proposte attività offerta dall'agente.

Agente

L'agente può ottenere una lista di attrazioni presenti in una determinata zona, suggerire al richiedente una lista di attività che potrebbe svolgere e aggiungere o rimuovere attrazioni.

ID: A1

TITOLO: Ottieni lista attrazioni

DESCRIZIONE: l'agente inserisce un indirizzo e il sistema restituisce una lista di attrazioni specificando la categoria, l'indirizzo dell'attrazione e la distanza dall'indirizzo passato. L'indirizzo dovrebbe essere completo in ogni sua parte così che il sistema possa restituire risultati più accurati.

SCOPO: mostrare all'agente una lista di attrazioni in una determinata zona con le relative informazioni

DIPENDENZE: NESSUNA

ID: A2

TITOLO: Creazione lista proposta attività

DESCRIZIONE: l'agente crea una proposta inserendo il periodo e la zona della vacanza e le classi di attrazioni da includere. Il sistema deve controllare che tutti i campi siano compilati.

SCOPO: creare una proposta di attività

DIPENDENZE: NESSUNA

ID: A3

TITOLO: Visualizzazione lista proposta attività

DESCRIZIONE: l'agente inserisce l'identificatore della proposta d'attività. Il sistema controlla se la proposta esiste e in caso affermativo mostra la sua lista di proposte di attività, altrimenti visualizza un messaggio d'errore.

SCOPO: Visualizzare la lista di proposte di attività relativa a un dato utente

DIPENDENZE: NESSUNA

ID: A4

TITOLO: Inserimento alternative

DESCRIZIONE: L'agente visualizza la lista di proposte di attività di un utente e seleziona le fasce orarie in cui vorrebbe che il sistema aggiungesse nuove attrazioni. Il sistema cerca delle alternative e aggiorna lo stato e l'aspetto della proposta.

SCOPO: Inserire nuove attrazioni in una determinata fascia oraria

DIPENDENZE: A3

ID: A5

TITOLO: Rimuovi alternative

DESCRIZIONE: L'agente visualizza la lista di proposte di attività di un utente e seleziona le attività che vorrebbe rimuovere. Il sistema rimuove le attività selezionate e aggiorna lo stato della proposta.

SCOPO: Rimuove delle attività da una lista di proposte di attività

DIPENDENZE: A3

ID: A6

TITOLO: Visualizza dettagli di un'attrazione

DESCRIZIONE: Il sistema mostra al richiedente una lista di proposte di attività. Per ogni attrazione presente nella lista, il richiedente può chiedere di visualizzarne le informazioni dettagliate. Il sistema mostrerà quindi l'indirizzo dell'attrazione selezionata, la durata consigliata, un sito web e un recapito telefonico.

SCOPO: Visualizzare in dettaglio le informazioni relative a una determinata attrazione

DIPENDENZE: NESSUNA

ID: A7

TITOLO: Rimuove attrazione sgradita

DESCRIZIONE: il richiedente visualizza la lista di proposte di attività e, per ogni attrazione che non gradisce, può chiedere di rimuoverla. Il sistema rimuove l'attrazione scelta dal richiedente e aggiorna la lista e lo stato della proposta.

SCOPO: Rimuove le attrazioni che non sono gradite

DIPENDENZE: NESSUNA

ID: A8

TITOLO: Accetta lista proposta attività

DESCRIZIONE: il richiedente visualizza la lista di proposte di attività e decide se accettare o meno la lista. In caso affermativo, il sistema aggiorna lo stato della proposta.

SCOPO: permette al richiedente di accettare o rifiutare una lista di proposte di attività

DIPENDENZE: NESSUNA

4.4.1.4 Richiesta Casa

Richiedente

Il richiedente può compilare una richiesta di condivisione. Può visualizzare, modificare e accettare una proposta di condivisione suggerita dall'agente.

Agente

Il servizio permette all'agente di visualizzare e gestire le richieste di condivisione del richiedente. Può suggerire, visualizzare e modificare una proposta di condivisione.

ID: RC1

TITOLO: Compila richiesta

DESCRIZIONE: il richiedente compila una richiesta inserendo la destinazione (nazione, città), il periodo, il numero di posti letto e di bagni, la tipologia dell'immobile, eventuali restrizioni (se può fumare, portare bambini piccoli e animali, suonare in orari notturni), eventuali preferenze (presenza di garage, piscina e terrazza), una serie di classi di attrazioni che vorrebbe che l'immobile avesse nelle vicinanze, e una preferenza sulla presenza di mezzi pubblici e sulla possibilità di condivisione dell'auto. Il richiedente può confermare o annullare la richiesta compilata e chiedere di cancellare i campi.

SCOPO: permette al richiedente di compilare una richiesta di condivisione

DIPENDENZE: NESSUNA

ID: RC2

TITOLO: Ottieni proposta

DESCRIZIONE: Il richiedente può controllare una proposta di condivisione, visualizzando la destinazione e il periodo, il trasporto e le restrizioni, una descrizione, le foto e il contesto dell'immobile.

SCOPO: Visualizzare una proposta di condivisione

DIPENDENZE: NESSUNA

ID: RC3

TITOLO: Risolvi proposta

DESCRIZIONE: il richiedente visualizza la proposta di condivisione e può decidere se accettarla o rifiutarla. In caso affermativo, il sistema informa che il proprietario

dell'immobile verrà contattato. In caso negativo, attenderà che l'agente presenti una nuova proposta. In entrambi i casi, il sistema ne cambia lo stato.

SCOPO: permette al richiedente di accettare o rifiutare una proposta di condivisione

DIPENDENZE: RC2

ID: RC4

TITOLO: Visualizza richiesta

DESCRIZIONE: l'agente può visualizzare e selezionare le richieste gestite e quelle che ancora nessuno sta gestendo. Il sistema provvederà a mostrare le informazioni dettagliate della richiesta selezionata.

SCOPO: permette all'agente di visualizzare le richieste gestite e quelle non gestite

DIPENDENZE: NESSUNA

ID: RC5

TITOLO: Gestisci richiesta

DESCRIZIONE: l'agente può selezionare una delle richieste non ancora gestite. Il sistema chiede conferma e, in caso affermativo, mostra all'agente alcune informazioni (ID, nome, destinazione, periodo) della richiesta appena gestita.

SCOPO: permette all'agente di selezionare le richieste di condivisione che vuole gestire

DIPENDENZE: NESSUNA

ID: RC6

TITOLO: Crea proposta

DESCRIZIONE: l'agente crea una proposta di condivisione inserendo la posizione dell'immobile (stato, città, via, numero civico, CAP), il periodo di condivisione, una descrizione dell'immobile, le informazioni sul contesto, le preferenze e le restrizioni (mezzi pubblici, condivisione auto, fumatore, bambini piccoli, animali, suonare in locali notturni) e può caricare le foto dell'immobile.

SCOPO: permette all'agente di creare una proposta di condivisione

DIPENDENZE: NESSUNA

ID: RC7

TITOLO: Ottieni proposta agente

DESCRIZIONE: l'agente può visualizzare le informazioni relative alle proposte di condivisione da lui gestite, compreso lo stato della proposta.

SCOPO: permette all'agente di visualizzare le proposte di condivisione gestite

DIPENDENZE: NESSUNA

ID: RC8

TITOLO: Modifica proposta

DESCRIZIONE: l'agente, dopo aver visualizzato le informazioni relative a una proposta di condivisione, può modificare i vari campi e aggiungere o rimuovere le foto dell'immobile. Una volta modificata la proposta, il sistema ne cambia lo stato.

SCOPO: permette all'agente di modificare una proposta di condivisione

DIPENDENZE: NESSUNA

4.4.2 Requisiti non funzionali

L'applicazione non presenta dei rigorosi requisiti non funzionali, in quanto essi sono a carico degli applicativi utilizzati.

4.4.2.1 Requisiti del prodotto

Usabilità

Tutte gli applicativi utilizzati devono possedere un'interfaccia user-friendly per facilitare l'utente nella fruizione dei servizi, permettendo l'accesso anche a coloro che non possiedono approfondite capacità informatiche. L'interfaccia dovrebbe essere tale che tutti gli utenti siano in grado di interagire con il sistema sin dal primo utilizzo, senza bisogno di essere istruiti.

Il requisito d'usabilità viene considerato raggiunto se un utente non esperto è in grado d'apprendere come utilizzare il servizio entro 48 ore e se, successivamente, è in grado d'utilizzarlo senza effettuare più di 2 errori ogni 24 ore.

Efficienza

Le operazioni offerte dai servizi dovrebbero essere veloci in modo da offrire all'utente un'esperienza d'uso ottimale. Il DBMS utilizzato deve garantire la consistenza e la correttezza delle informazioni memorizzate.

Prestazioni

Le performance del sistema implementato sono fortemente correlate a quelle dei singoli applicativi utilizzati. Guasti, problemi e ritardi su di essi comporteranno una perdita di prestazioni sul servizio finale.

L'agente, ogni qualvolta un proprietario od un richiedente interagisce con un servizio, deve garantire una risposta entro e non oltre le 3 ore, eventualmente fermandosi per una nuova interazione con gli utenti. Se non riceve alcun tipo di risposta per oltre 1 mese, egli può annullare la transazione.

Requisiti di spazio

I servizi utilizzati dovrebbero possedere una capacità di memorizzazione sufficiente a conservare le informazioni inserite dagli utenti e dovrebbero essere in grado di garantire la scalabilità del

sistema al variare del numero di utenti registrati. Per quanto riguarda i singoli applicativi, i requisiti di spazio possono essere considerati raggiunti se non si verificano problemi di memorizzazione di dati nell'arco di 3 mesi.

SharingEverywhere conserva localmente una quantità ridotta di dati e li mantiene per un tempo limitato pertanto, considerata la capacità di memorizzazione media di un server moderno, i requisiti di spazio risultano sempre raggiunti.

Dependability

Il sistema può subire fallimenti, sia hardware che software, dovuti a guasti verificatisi sui singoli applicativi. Sfortunatamente questi guasti non dipendono dal software sviluppato quindi non solo non sono controllabili ma sono anche difficili da testare e quindi da gestire. Il requisito risulta raggiunto se gli applicativi esterni non causano più di 25 fallimenti nell'arco di un mese e se il ripristino da situazioni d'errore non supera i 70 minuti per guasto.

Il sistema può subire fallimenti hardware dovuti a guasti dei server sui quali si appoggia. Il protocollo di comunicazione implementato, come tutti i software, può presentare anche fallimenti a livello software dovuti a errori contenuti nelle specifiche, design o implementazione. Il processo di sviluppo dovrebbe essere eseguito in modo da minimizzare la possibilità che quest'eventualità si verifichi. Il requisito risulta raggiunto se il sistema non subisce più di 10 fallimenti nell'arco di un mese e se il rispristino da situazioni d'errore non supera i 50 minuti per guasto.

I fallimenti operazionali sono dovuti a errori causati dagli operatori umani. Sebbene il coordinatore non risenta direttamente di questi problemi, i singoli applicativi potrebbero soffrirne: il verificarsi di tali problemi su un applicativo potrebbe influire negativamente sull'intero sistema, per questo motivo i singoli servizi dovrebbero essere il più robusti possibile rispetto agli errori umani. Il requisito risulta raggiunto se errori operazionali sugli applicativi esterni non causano più di 5 fallimenti nell'arco di un mese e se il ripristino da situazioni d'errore non supera i 30 minuti per guasto.

Sebbene questi valori siano piuttosto alti, il risultato complessivo rimane accettabile: il sistema implementato non è critico, pertanto il notevole incremento del costo del progetto non è compensato dai benefici portati da un maggior grado di dependability.

Sicurezza

Gli utenti che utilizzano il servizio devono autenticarsi con i tre applicativi *Offerta casa*, *Richiesta Casa* ed *Attrazioni* (*Servizi Accessori* non richiede alcuna autenticazione). La gestione dei database contenenti le credenziali d'autenticazione è lasciata al personale incaricato dei singoli applicativi, che deve assicurarsi che siano messe in atto tutte le strategie e i metodi per garantire la massima confidenzialità agli utenti, sia nella gestione delle autenticazioni che nell'accesso dell'utente alle sue aree personali. Il personale incaricato della manutenzione dei singoli applicativi dovrebbe eseguire controlli periodici del software per verificare la presenza di vulnerabilità ed eventualmente correggerle.

4.4.2.2 Requisiti esterni

Requisiti etici

Gli utenti dovrebbero interagire con i servizi in modo etico, nel massimo rispetto reciproco evitando tutti i comportamenti che possano minare il meccanismo di scambio, che per sua natura è basato sui principi della fiducia e della condivisione. Essi dovrebbero evitare di inserire spam, informazioni false o ingannevoli.

Requisiti legali

Il servizio non è soggetto a vincoli legali relazionati alla stipula del contratto, il quale avviene in separata sede dopo l'utilizzazione di questo sistema.

I servizi hanno responsabilità legale sui dati forniti dall'utente. L'utilizzo che il servizio può farne è stabilito nei termini d'uso che l'utente accetta in fase di registrazione.

Capitolo 5

Piano sulla qualità

5.1 Introduzione

Questo documento riporta tutte le considerazioni effettuate per garantire una certa qualità del prodotto finale. Verranno definiti i ruoli e le responsabilità di tutti i membri coinvolti nello sviluppo di SharingEverywhere, cercando di massimizzare la flessibilità laddove possibile.

5.2 Suddivisione del lavoro

Ogni componente è responsabile della gestione di uno dei quattro applicativi usati. La tabella sottostante mostra l'assegnazione degli applicativi:

| Componente | Applicativo | Descrizione |
|-----------------|-------------------|--|
| Chiara Caiazza | Offerta Casa | Permette l'inserimento d'offerte relative ad immobili e permette di effettuare ricerche al fine di visualizzare le informazioni salvate. |
| Gionatan Gallo | Richiesta Casa | Permette di richiedere immobili e di visualizzare le richieste pendenti. |
| Daniele Modica | Servizi Accessori | Permette di realizzare una pianificazione di servizi accessori attivabili presso un dato immobile. |
| Martina Troscia | Attrazioni | Crea una lista di attività fruibili a partire da un dato immobile. |

5.2.1 Gestione degli spazi di lavoro

I componenti cercano di massimizzare il tempo che passeranno insieme, eventualmente richiedendo la prenotazione di aule quando possibile: questa metodologia è stata scelta per riunire tutti i membri in un'unica stanza anche se stanno eseguendo un lavoro individuale.

La presenza di tutti i componenti in uno spazio comune permette di:

- Risolvere il prima possibile eventuali problemi/dubbi non rilevati in precedenza;
- Richiedere un parere aggiuntivo da parte degli altri membri del gruppo che, sebbene coinvolti nell'intero processo, hanno una visione del lavoro indipendente dalle scelte effettuate dal singolo durante la progettazione;
- Permettere di eseguire riunioni anche se non precedentemente pianificate.

5.2.2 Piano di ricerca

I membri del gruppo cercano di ispirarsi quanto più possibile ad applicativi già esistenti durante il loro lavoro individuale, evitando di reinventare componenti già esistenti.

5.2.3 Comunicazione interna

I membri utilizzano una cartella condivisa su Google Drive, al fine di garantire la massima trasparenza sul lavoro svolto e per permettere a ognuno di poter visualizzare il lavoro di ogni altro componente ogni qualvolta ne abbia necessità. Tutti i componenti concordano sull'utilizzo della piattaforma Google in quanto hanno esperienze di utilizzo pregresse, sia nell'accesso alle funzionalità di upload/download che alla gestione delle versioni.

Per la comunicazione i membri utilizzano canali di comunicazioni informali (chat/gruppo Telegram, videoconferenze Skype/TeamViewer).

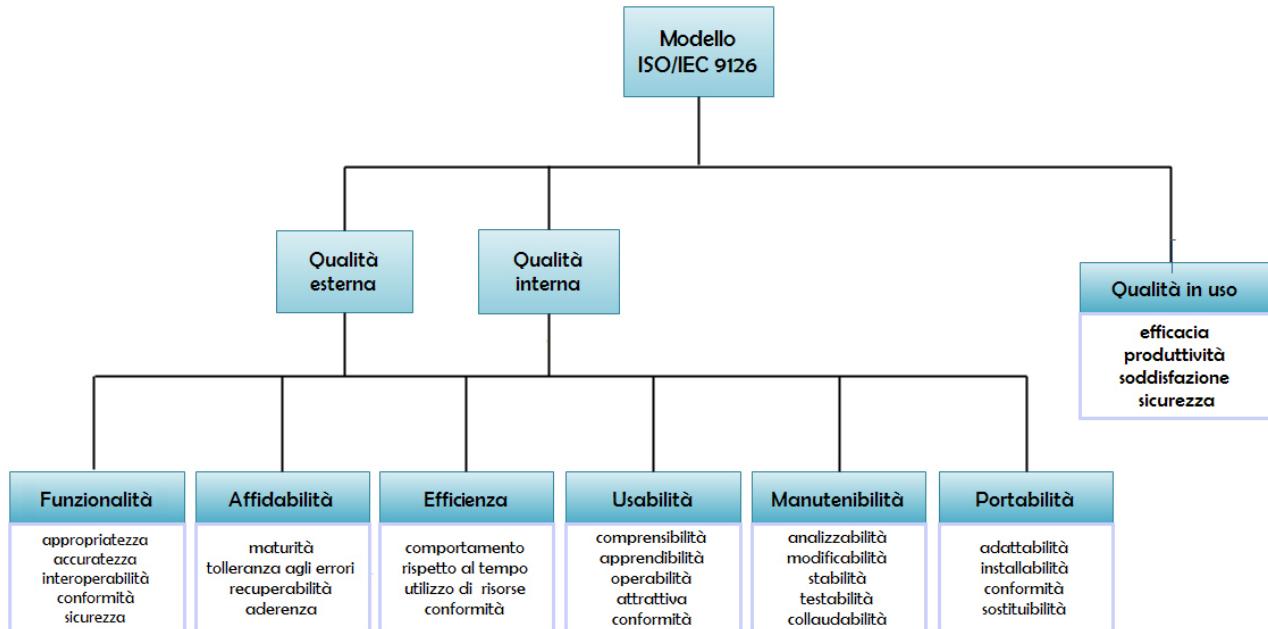
5.3 Milestone

La progettazione del sistema è suddivisa in sei fasi:

| Fase | Deliverable | Deadline |
|-----------------|--|-----------------|
| Requirements | Descrizione testuale del sistema. | |
| Analisi | Descrizione delle funzionalità del sistema ad alto livello, atta ad essere presentata a coloro che hanno commissionato il lavoro. | 17 h |
| Design | Descrizione dettagliata delle funzionalità del sistema. Questa fase ha lo scopo di presentare documentazione che tenga conto di dettagli implementativi per facilitare la fase successiva. | 16 h |
| Implementazione | Implementazione delle funzionalità del sistema precedentemente descritte. | 14 h |
| Testing | Verifica e validazione del codice prodotto al passo precedente. | 9 h |
| Final reporting | Stesura della documentazione. | 5 h |

5.4 Modello ISO – IEC 9126

Lo standard ISO – IEC 9126 specifica quali siano le caratteristiche che il codice dovrebbe avere per essere considerato di qualità dagli utenti:



- **Funzionalità:** capacità del sistema di compiere il lavoro per il quale è stato progettato, coerentemente con i requisiti stilati.
- **Affidabilità:** capacità del sistema di operare a regime, senza sperimentare fallimenti e mantenendo le performance per un periodo stabilito.
- **Efficienza:** capacità del sistema di mantenere performance e quantità di risorse accettabili.
- **Usabilità:** capacità del sistema di essere utilizzabile da utenti con poca esperienza con il minor sforzo possibile, minimizzando l'addestramento cui l'utente deve essere sottoposto.
- **Manutenibilità:** facilità d'apportare modifiche al sistema utilizzato.
- **Portabilità:** capacità del sistema di essere eseguito su sistemi software diversi.

5.4.1 Garanzie di qualità

La tabella di seguito mostra in che modo verranno garantiti gli attributi di qualità appena presentati:

| Attributo di qualità | Provvedimento |
|----------------------|--|
| Funzionalità | L'esecuzione del codice è coerente con lo scenario proposto dal cliente, in quanto il codice Java, generato a partire dai diagrammi di sequenza, soddisfa i requisiti del cliente. |

| | |
|----------------|---|
| Affidabilità | Il sistema garantisce l'affidabilità se, una volta eseguito, rimarrà in esecuzione per tutta la durata dell'attività della compagnia. Tuttavia il software utilizza servizi di terze parti che possono essere soggetti a fallimenti. Dalla fase di test deve risultare che questa percentuale è accettabile e permette di continuare il lavoro. |
| Efficienza | Il sistema è efficiente se garantisce performance ed utilizzo di risorse accettabili nel tempo. Una valutazione dell'efficienza deve essere legata alla stime dei tempi d'esecuzione del servizio e delle risorse necessarie al suo funzionamento. |
| Usabilità | Il sistema garantisce usabilità tramite le interfacce degli applicativi, le quali devono essere intuitive e di facile utilizzo, permettendo anche agli utenti meno esperti una completa fruizione del servizio. |
| Manutenibilità | La manutenibilità è garantita utilizzando un'elevata modularità all'interno degli applicativi, in questo modo il sistema è aperto ad ogni tipo di modifica. Fintanto che le interfacce dei singoli moduli rimangono invariate, sono possibili cambiamenti all'implementazione di una parte del sistema: cambiamenti ad un modulo non influiscono sugli altri poiché le modalità di comunicazione rimangono invariate. |
| Portabilità | Il sistema è implementato utilizzando il linguaggio Java che garantisce portabilità, infatti la compilazione non produce codice binario da eseguire direttamente su una specifica macchina ma byte-code indipendente dalla piattaforma utilizzata. |

5.5 Gestione dei rischi

Il manager durante la sua fase d'analisi aveva individuato i seguenti rischi, calcolando la probabilità che essi si verifichino e il loro possibile impatto sul progetto.

| Rischio | | Probabilità che si verifichi | Conseguenze |
|------------------|-------------------------------|------------------------------|---------------|
| Sulle specifiche | scarsa chiarezza | Alta | Catastrofiche |
| | interazione con i committenti | Alta | Catastrofiche |
| Sulle stime | stime errate | Alta | Catastrofiche |
| Tecnologico | adattamento | Moderata | Serie |
| | Scalabilità | Bassa | Serie |
| Sugli strumenti | scarsa conoscenza | Moderata | Serie |
| | scarsa integrazione | Alta | Serie |
| Sulla persona | ritardi dovuti a un membro | Moderata | Tollerabili |
| | attività collettive | Moderata | Tollerabili |

| | | | |
|------------|--|-------------------|----------------------------|
| Sul gruppo | discussioni off-topic scarso affiatamento | Moderata Bassa | Tollerabili Tollerabili |
|------------|--|-------------------|----------------------------|

Aveva individuato i seguenti piani per minimizzare gli effetti dei rischi:

- **Rischi tecnologici:** l'idea è di rendere gli input e output il più standard possibili per integrare meglio i moduli dell'applicativo.
- **Rischi sulla persona:** la strategia è di usare cartelle condivise in cui ogni componente del team può mettere il risultato del proprio lavoro così che in caso di necessità un membro possa controllare il lavoro di un altro. L'uso di programmi come Skype o TeamViewer permette di lavorare anche quando non c'è la possibilità di incontrarsi direttamente.
- **Rischi sul gruppo:** l'idea è di definire in precedenza in maniera chiara di cosa si intende discutere in una certa attività collettiva e darsi dei tempi massimi entro cui deve essere portata a termine. I membri del gruppo devono sforzarsi di mantenere dei rapporti amichevoli per favorire la coesione del team.
- **Rischi sugli strumenti utilizzati:** l'utilizzo di programmi come Visual Paradigm guida nella creazione del codice perché consente di generarlo automaticamente partendo da modelli UML.
- **Rischi sulle specifiche:** è necessario analizzare a fondo il problema in esame per evitare di lasciare dubbi o incertezze sull'interpretazione del lavoro da svolgere. Pertanto è importante generare un SRS chiaro e dettagliato che non consenta il sorgere di fraintendimenti con i committenti. È necessario far capire loro come un certo cambiamento alle specifiche si ripercuoterà sui tempi e costi del progetto.
- **Rischi sulle stime:** la strategia è di usare software come SystemStar per avere un'idea di quanto tempo è richiesto per portare a termine una fase. Queste stime possono essere usate per pianificare il proprio lavoro e dovrebbero impedire che sorgano troppi problemi, se il team si sforza di seguire la pianificazione.

5.5.1 Rischi sulla sicurezza

Ai rischi individuati dal Project Manager è stato aggiunto un ulteriore rischio legato alla sicurezza:

| Rischio | Probabilità che si verifichi | Conseguenze |
|---|------------------------------|-------------|
| Sulla sicurezza Attacchi informatici | Bassa | Serie |

Nel caso in cui gli applicativi fossero esposti ad attacchi esterni da parte di malintenzionati, che potrebbero sottrarre o manomettere i dati personali inseriti dagli utenti, il servizio subirà seri danni d'immagine dovuti alla compromissione del rapporto di fiducia instaurato tra SharingEverywhere e gli utilizzatori del servizio.

Dal momento che la sicurezza deve essere implementata dai quattro applicativi, il servizio non può approvare piani per la minimizzazione del rischio, né può applicare piani di contingenza per

limitare gli eventi avversi nel caso in cui quest'eventualità si verifichi. Tutto ciò che il servizio può fare è effettuare, in fase di progettazione, una scelta oculata dei servizi da utilizzare.

Gli applicativi dovrebbero mettere in pratica tutte le norme di buona programmazione al fine di costruire un codice robusto, richiedere l'utilizzo di protocolli di comunicazione sicuri (come https) per garantire confidenzialità ed autenticazione e mantenere un buon firewall per bloccare attacchi di tipo DoS, DDoS, network mapping e port scanning.

Capitolo 6

Analisi

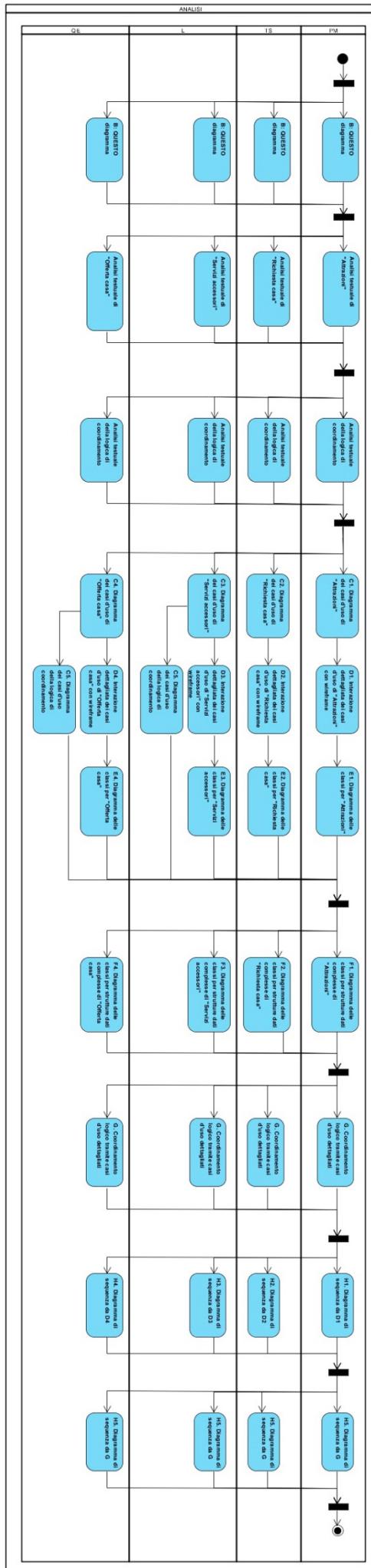
6.1 Organizzazione della fase

Le attività riguardanti la fase di analisi sono state organizzate e pianificate secondo lo schema riportato in Figura 6.1. Questo diagramma indica inoltre i documenti che devono essere elaborati:

- C** Diagramma UML del caso d'uso per ogni modulo esistente e per la logica di coordinamento.
- D** Interazione dettagliata dei casi d'uso con wireframe, per ogni applicativo esistente.
- E** Diagramma delle classi (Analisi) per ogni applicativo esistente.
- F** Diagramma delle strutture dati complesse per ogni applicativo esistente.
- G** Coordinamento logico tramite casi d'uso dettagliati.
- H** Diagrammi di sequenza (Analisi) prodotti da **D** e **G**.

6.1.1 Tipi di attività

Le attività possono essere individuali, in modo che possano essere eseguite in parallelo da ciascun membro del team, o cooperativa, in cui tutti i membri del team devono svolgere il lavoro. Il diagramma B ha lo scopo di pianificare nel migliore dei modi la fase di analisi in modo da ottenere le migliori prestazioni in termini di tempo speso in ogni fase e qualità del prodotto.



Attività individuali

Attività C richiede che ogni membro produca uno schema sul proprio applicativo e non c'è bisogno di alcuna informazione di come gli altri moduli siano sviluppati. È stata assegnata l'elaborazione del caso d'uso UML del coordinatore logico al Librarian e il Quality Engineer, i quali avevano una quantità di lavoro inferiore rispetto agli altri membri del gruppo.

Attività D richiede che l'attività **C** sia completata, prima di poter essere eseguita. Quindi **D** segue logicamente **C** e non richiede nessuna sincronizzazione tra i membri.

Attività E richiede una sincronizzazione dopo il suo completamento, in modo da condividere le informazioni necessarie per le attività successive.

Attività F richiede una sincronizzazione dopo il suo completamento.

Attività cooperative

Attività G produce la logica che utilizza le funzioni di tutti i moduli. Ogni membro deve quindi fornire le operazioni del suo applicativo al processo di coordinamento.

Attività H deriva da D e G e richiede la validazione da parte di tutti i componenti del gruppo.

6.2 Casi d'uso

La modellazione di casi d'uso è una forma di ingegneria dei requisiti e richiede di:

- Identificare il confine del sistema
- Identificare gli attori
- Identificare i casi d'uso

Il risultato di queste attività rappresenta il modello del caso d'uso. Esistono quattro componenti che definiscono questo modello:

Attori: ruoli svolti dalle persone o cose che utilizzano il sistema.

Casi d'uso: cose che gli attori possono fare con il sistema.

Relazioni: relazioni significative tra gli attori e i casi d'uso.

Confine del sistema: quadrato disegnato attorno ai casi d'uso per definire il confine e il bordo del sistema che è stato modellato.

6.2.1 Offerta casa

Analisi testuale

Il proprietario di un immobile può inserire un'offerta. L'offerta contiene il periodo in cui vuole scambiare la casa, l'indirizzo dell'immobile (comprensivo di via, numero civico, cap, città e stato), una descrizione completa dell'immobile, disponibilità di fornire in condivisione anche il proprio veicolo ed eventuali limitazioni. La descrizione di un immobile comprende tipologia dell'immobile, foto, una descrizione riguardante le composizioni della casa, una descrizione della zona circostante ed eventuali collegamenti con i mezzi pubblici. Tra le limitazioni sono presenti il divieto di fumare nell'immobile, il divieto di portare animali, eventuali vincoli sull'esecuzione di attività rumorose dopo le 10pm ed un suggerimento atto ad informare se la casa sia adatta o meno ad ospitare dei bambini.

Ancuni campi dell'offerta sono opzionali. Il servizio accetta un'offerta se tutti i campi obbligatori sono compilati. L'agente può richiedere la lista degli immobili presenti in un posto specifico. Dopo aver concluso uno scambio, l'agente comunica al proprietario che qualcuno è interessato all'immobile. Il proprietario può visualizzare lo stato della sua offerta per verificare se è stato individuato uno scambio o meno.

Casi d'uso

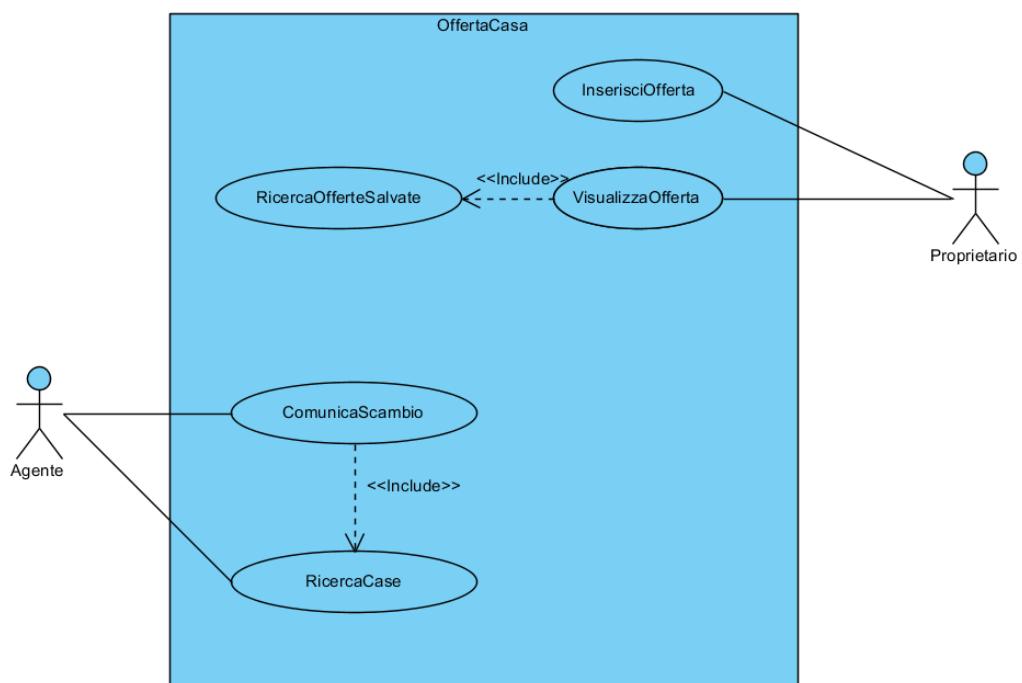


Figura 6.2: Caso d'uso per il modulo *Offerta Casa*

Wireframe

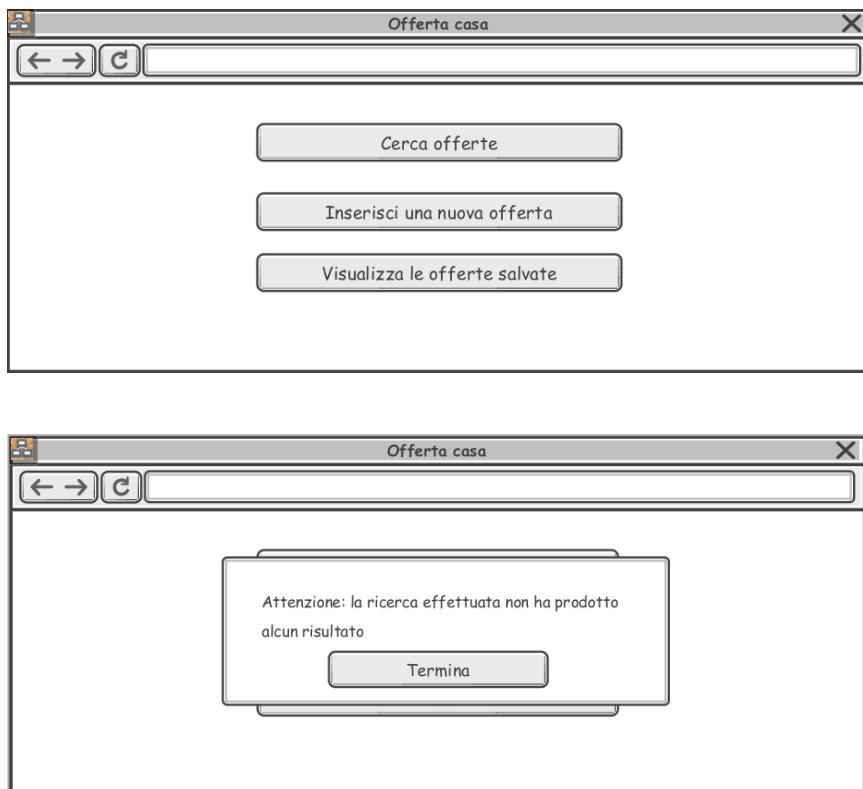


Figura 6.3 Homepage *Offerta Casa*



Figura 6.4 Lista di offerte

Inserisci la tua offerta

Effettua scambio da* 10/11/2017 a* 23/04/2018

Inserisci l'indirizzo dell'immobile

in* via/piazza n° civico* 12
città* Pisa cap* 56126 stato* Italia

Inserisci una descrizione esauritiva dell'immobile:

Tipologia d'immobile* Appartamento Inserisci una foto*

Descrivi la tua casa* [Large text area]

Descrivi la zona in cui si trova la tua casa [Large text area]

Elenco i mezzi pubblici accessibili dalla tua casa [Large text area]

Si fornisce la possibilità di utilizzare un veicolo Si No

Limitazioni:

- [] La casa è adatta ai bambini Si No
- [] È possibile fumare all'interno della casa Si No
- [] È possibile portare animali Si No
- [] È richiesto particolare silenzio in orari notturni Si No

Inserisci

Inserisci la tua offerta

Effettua scambio da* 10/11/2017 a* 23/04/2018

Inserisci l'indirizzo dell'immobile

in* via/piazza n° civico* 12
città* Milano cap* 20121 stato* Italia

Inserisci una descrizione esauritiva dell'immobile:

Tipologia d'immobile* Attico/mansarda Inserisci una foto*

Descrivi la tua casa* [Large text area]

Attenzione: Tutti i campi contrassegnati da un asterisco sono obbligatori e, pertanto devono essere compilati

Descrivi la zona in cui si trova la tua casa [Large text area]

Ho capito

Elenco i mezzi pubblici accessibili dalla tua casa [Large text area]

La dimora è collegata alla linea blu della metro, e alla linea dei pullman. La stazione ferroviaria, sebbene lontana è raggiungibile con facilità utilizzando i collegamenti esistenti

Si fornisce la possibilità di utilizzare un veicolo Si No

Limitazioni:

- [] La casa è adatta ai bambini Si No
- [] È possibile fumare all'interno della casa Si No
- [] È possibile portare animali Si No
- [] È richiesto particolare silenzio in orari notturni Si No

Inserisci

Inserisci la tua offerta

Effettua scambio da* 10/11/2017 a* 23/04/2018

Inserisci l'indirizzo dell'immobile

in* Corso Vittorio Emanuele II n° 12 n° civico* 12
città* Milano cap* 20121 stato* Italia

Inserisci una descrizione esauritiva dell'immobile:

Tipologia d'immobile* Attico/mansarda Inserisci una foto*

Descrivi la tua casa* [Large text area]

Ampio e luminoso bilocale composto da ingresso in ampio soggiorno / pranzo con zona cottura, ampio balcone. Disimpegno zona notte, una camera matrimoniale con piccola cabina e balcone con affaccio su zona area verde, bagno con finestra, vasca e doccia. L'appartamento è completamente e finalmente arredato con mobili moderni, completo di ogni confort. Presente una continua finestra ad uso esclusivo.

Descrivi la zona in cui si trova la tua casa [Large text area]

L'appartamento si trova in una zona centrale vicino al duomo, di teatro Gerolamo e al cinema The Space. Nelle immediate vicinanze sono presenti ristorante, pizzeria napoletana e risto-pub. L'appartamento, situato sul corso di Milano si trova nelle vicinanze di numerose attività commerciali.

Elenco i mezzi pubblici accessibili dalla tua casa [Large text area]

La dimora è collegata alla linea blu della metro, e alla linea dei pullman. La stazione ferroviaria, sebbene lontana è raggiungibile con facilità utilizzando i collegamenti esistenti

Si fornisce la possibilità di utilizzare un veicolo Si No

Limitazioni:

- [] La casa è adatta ai bambini Si No
- [] È possibile fumare all'interno della casa Si No
- [] È possibile portare animali Si No
- [] È richiesto particolare silenzio in orari notturni Si No

Inserisci

Inserisci la tua offerta

Effettua scambio da* 10/11/2017 a* 23/04/2018

Inserisci l'indirizzo dell'immobile

in* Corso Vittorio Emanuele II n° 12 n° civico* 12
città* Milano cap* 20121 stato* Italia

Inserisci una descrizione esauritiva dell'immobile:

Tipologia d'immobile* Attico/mansarda Inserisci una foto*

Descrivi la tua casa* [Large text area]

Ampio e luminoso bilocale composto da ingresso in ampio soggiorno / pranzo con zona cottura, ampio balcone. Disimpegno zona notte, una camera matrimoniale con piccola cabina e vasca e doccia. Mobili moderni, completo di ogni confort. Presente una continua finestra ad uso esclusivo.

Descrivi la zona in cui si trova la tua casa [Large text area]

L'operazione è stata completata con successo

Termina

Elenco i mezzi pubblici accessibili dalla tua casa [Large text area]

La dimora è collegata alla linea blu della metro, e alla linea dei pullman. La stazione ferroviaria, sebbene lontana è raggiungibile con facilità utilizzando i collegamenti esistenti

Si fornisce la possibilità di utilizzare un veicolo Si No

Limitazioni:

- [] La casa è adatta ai bambini Si No
- [] È possibile fumare all'interno della casa Si No
- [] È possibile portare animali Si No
- [] È richiesto particolare silenzio in orari notturni Si No

Inserisci

Figura 6.5 Il proprietario inserisce di un'offerta

Screenshot 1 (Left):

- Indirizzo***: via/piazza
- Composizione della casa**:
 - Numero minimo di posti letto:
 - Numero minimo di:
 - E' presente una piscina:
 - E' richiesto un garage:
- Collegamenti con i mezzi pubblici**:
 - Sono presenti collegamenti alle fermate degli autobus:
 - Sono presenti collegamenti alla stazione dei treni:
- Richieste accessorie:**
 - La casa è adatta ai bambini:
 - E' possibile fumare all'interno della casa:
 - E' possibile portare animali:
 - E' richiesto particolare silenzio in orari notturni:
- Ricerca offerte**

Screenshot 2 (Right):

- Indirizzo***: Milano
- Composizione della casa**:
 - Numero minimo di posti letto:
 - Numero minimo di:
 - E' presente una piscina:
 - E' richiesto un garage:
- Collegamenti con i mezzi pubblici**:
 - Sono presenti collegamenti alle fermate degli autobus:
 - Sono presenti collegamenti alla stazione dei treni:
- Richieste accessorie:**
 - La casa è adatta ai bambini:
 - E' possibile fumare all'interno della casa:
 - E' possibile portare animali:
 - E' richiesto particolare silenzio in orari notturni:
- Ricerca offerte**

Figura 6.6: L'agente ricerca le offerte

Screenshot 1 (Left):

Inserisci le informazioni personali del richiedente, esse permetteranno al proprietario di contattarlo.

| | |
|--------------------|----------------------|
| Nome | <input type="text"/> |
| Cognome | <input type="text"/> |
| Numero di telefono | <input type="text"/> |
| Cellulare | <input type="text"/> |
| Indirizzo e-mail | <input type="text"/> |

Pubblica

Screenshot 2 (Middle):

Inserisci le informazioni personali del richiedente, esse permetteranno al proprietario di contattarlo.

| | |
|--------------------|----------------------|
| Nome | <input type="text"/> |
| Cognome | <input type="text"/> |
| Numero di telefono | <input type="text"/> |
| Cellulare | <input type="text"/> |
| Indirizzo e-mail | <input type="text"/> |

I recapiti inseriti ha un formato non valido.
Inserisci almeno un recapito del richiedente

Ho capito

Screenshot 3 (Right):

Inserisci le informazioni personali del richiedente, esse permetteranno al proprietario di contattarlo.

| | |
|--------------------|--|
| Nome | <input type="text" value="Mario"/> |
| Cognome | <input type="text" value="Rossi"/> |
| Numero di telefono | <input type="text" value="0587-1212121"/> |
| Cellulare | <input type="text" value="334-1234567"/> |
| Indirizzo e-mail | <input type="text" value="mario.rossi@gmail.com"/> |

Pubblica

Figura 6.7 L'agente inserisce la proposta

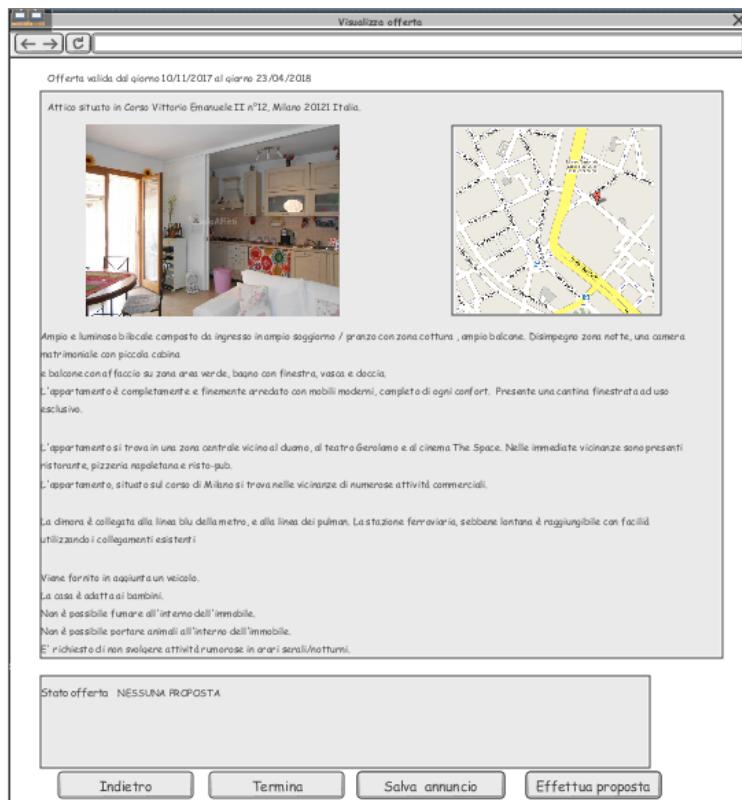


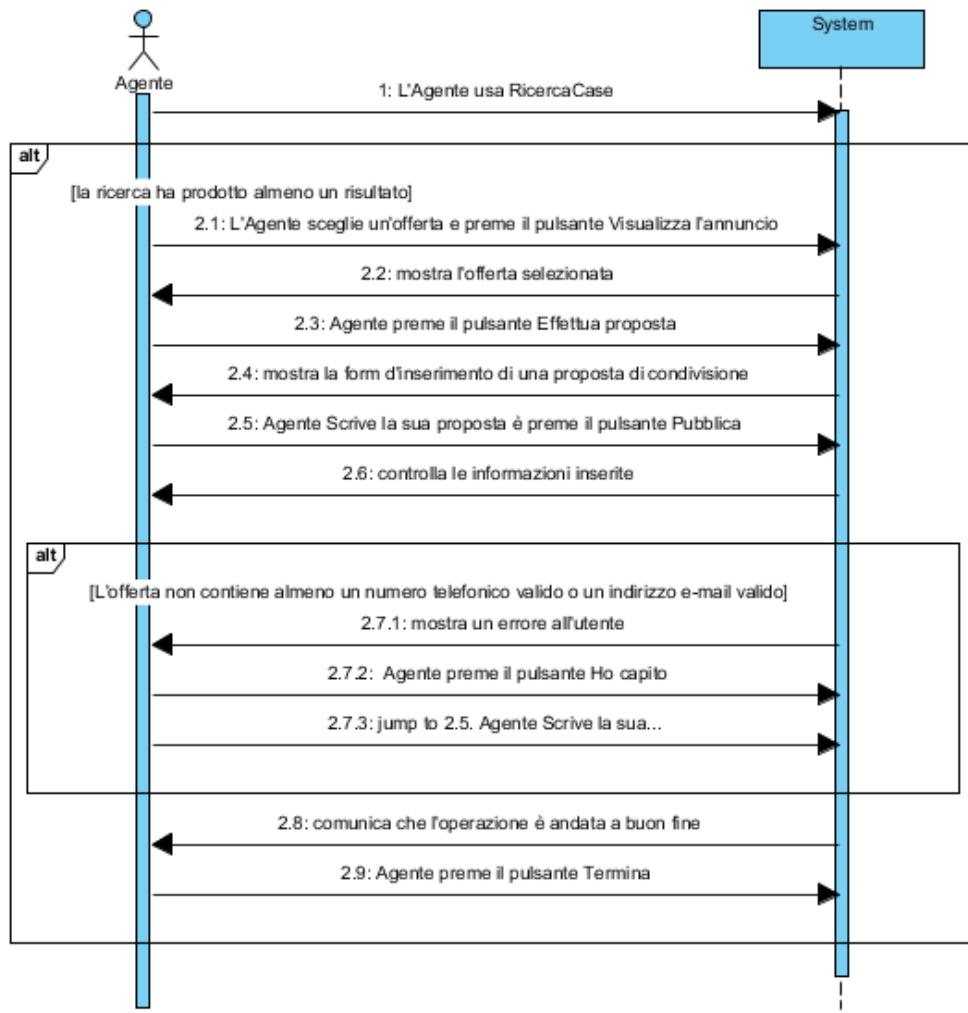
Figura 6.8: Visualizza offerta

Casi d'uso dettagliati e diagrammi di sequenza

ComunicaScambio

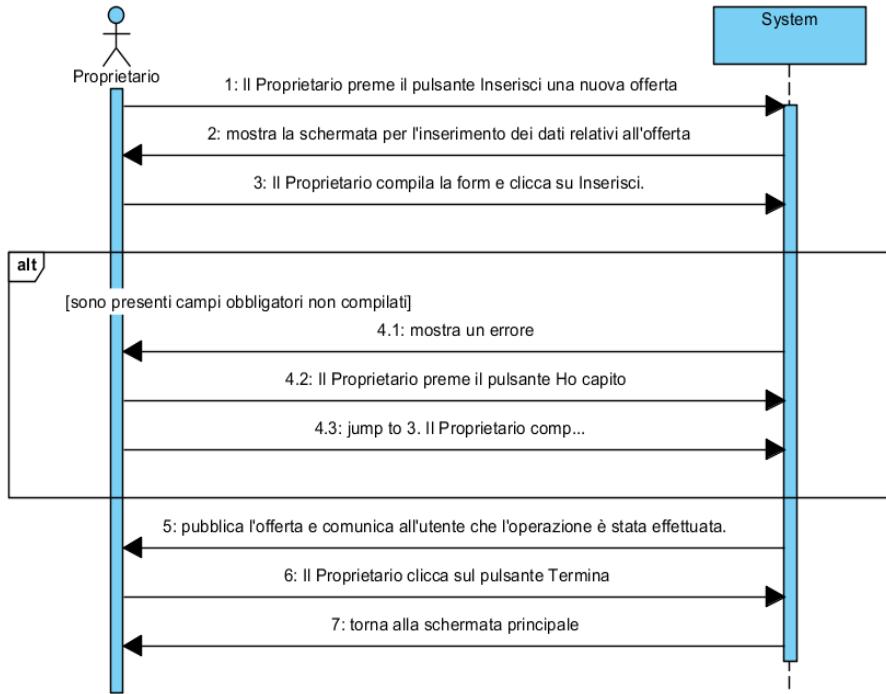
1. L' Agente usa RicercaCase
 2. **if** la ricerca ha prodotto almeno un risultato
 - 2.1. L' Agente sceglie un'offerta e preme il pulsante **Visualizza l'annuncio**
 - 2.2. **SYSTEM** mostra l'offerta selezionata
 - 2.3. Agente preme il pulsante **Effettua proposta**
 - 2.4. **SYSTEM** mostra la form d'inserimento di una proposta di condivisione
 - 2.5. Agente Scrive la sua proposta e preme il pulsante **Pubblica**
 - 2.6. **SYSTEM** controlla le informazioni inserite
 - 2.7. **if** L'offerta non contiene almeno un numero telefonico valido o un indirizzo e-mail valido
 - 2.7.1. **SYSTEM** mostra un errore all'utente
 - 2.7.2. Agente preme il pulsante **Ho capito**
 - 2.7.3. **jump to** 2.5. Agente Scrive la sua...

end if
 - 2.8. **SYSTEM** comunica che l'operazione è andata a buon fine
 - 2.9. Agente preme il pulsante **Termina**
- end if**



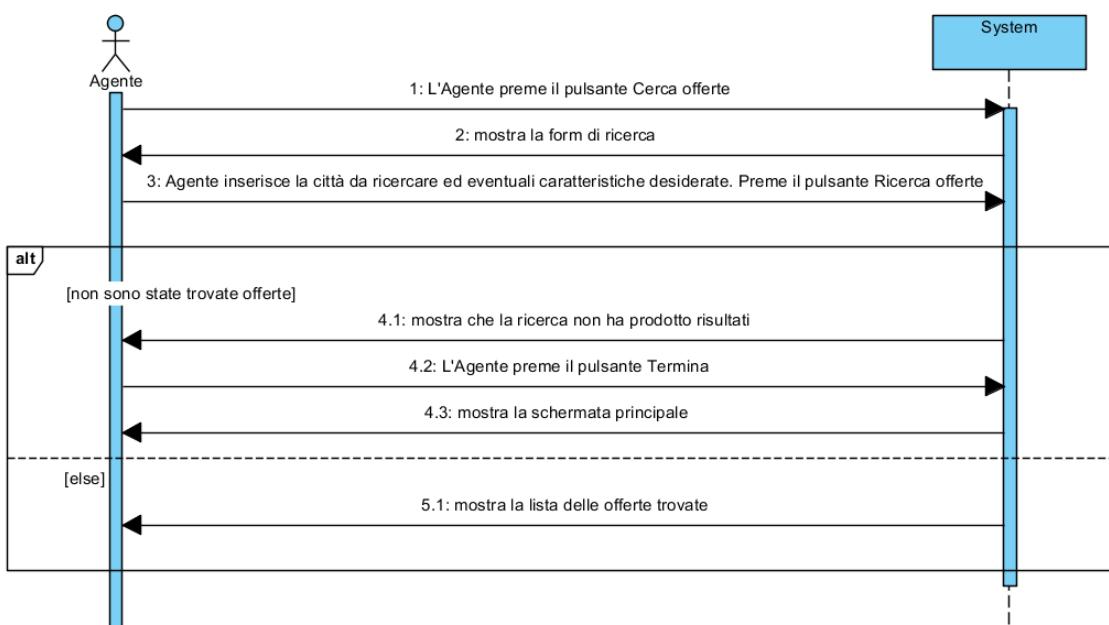
InserisciOfferta

1. Il **Proprietario** preme il pulsante **Inserisci una nuova offerta**
2. **SYSTEM** mostra la schermata per l'inserimento dei dati relativi all'offerta
3. Il **Proprietario** compila la form e clicca su **Inserisci**.
4. **if** sono presenti campi obbligatori non compilati
 - 4.1. **SYSTEM** mostra un errore
 - 4.2. Il **Proprietario** preme il pulsante **Ho capito**
 - 4.3. **jump to** 3. Il Proprietario comp...
5. **SYSTEM** pubblica l'offerta e comunica all'utente che l'operazione è stata effettuata.
6. Il **Proprietario** clicca sul pulsante **Termina**
7. **SYSTEM** torna alla schermata principale



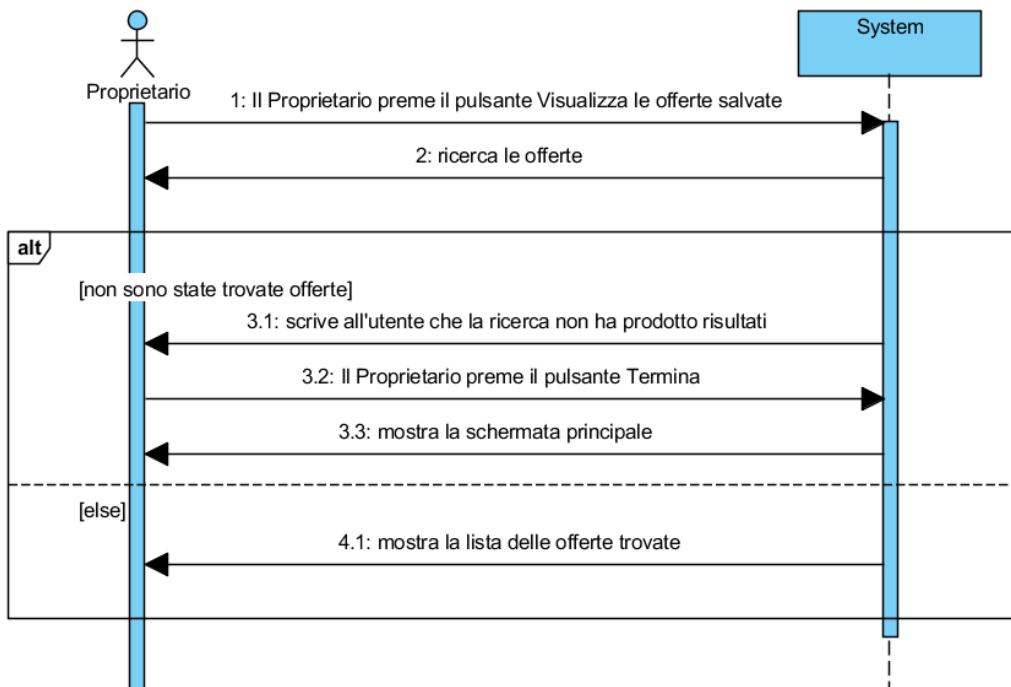
RicercaCase

1. L' **Agente** preme il pulsante **Cerca offerte**
 2. **SYSTEM** mostra la form di ricerca
 3. **Agente** inserisce la città da ricercare ed eventuali caratteristiche desiderate. Preme il pulsante **Ricerca offerte**
 4. **if** non sono state trovate offerte
 - 4.1. **SYSTEM** mostra che la ricerca non ha prodotto risultati
 - 4.2. L' **Agente** preme il pulsante **Termina**
 - 4.3. **SYSTEM** mostra la schermata principale
 5. **else**
 - 5.1. **SYSTEM** mostra la lista delle offerte trovate
- end if**



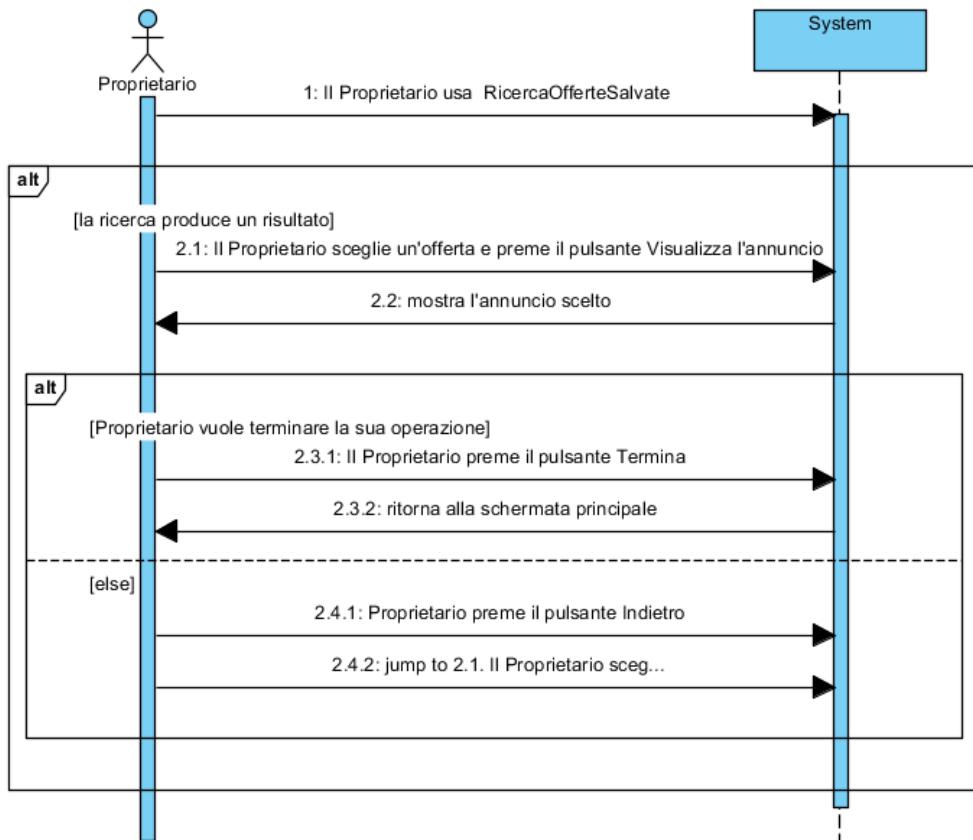
RicercaOfferteSalvate

1. Il Proprietario preme il pulsante **Visualizza le offerte salvate**
2. **SYSTEM** ricerca le offerte
3. **if** non sono state trovate offerte
 - 3.1. **SYSTEM** scrive all'utente che la ricerca non ha prodotto risultati
 - 3.2. Il Proprietario preme il pulsante **Termina**
 - 3.3. **SYSTEM** mostra la schermata principale
4. **else**
 - 4.1. **SYSTEM** mostra la lista delle offerte trovate**end if**



VisualizzaOfferta

1. Il Proprietario usa RicercaOfferteSalvate
2. **if** la ricerca produce un risultato
 - 2.1. Il Proprietario sceglie un'offerta e preme il pulsante **Visualizza l'annuncio**
 - 2.2. **SYSTEM** mostra l'annuncio scelto
 - 2.3. **if** Proprietario vuole terminare la sua operazione
 - 2.3.1. Il Proprietario preme il pulsante **Termina**
 - 2.3.2. **SYSTEM** ritorna alla schermata principale
 - 2.4. **else**
 - 2.4.1. Proprietario preme il pulsante **Indietro**
 - 2.4.2. **jump to** 2.1. Il Proprietario sceg...**end if****end if**



6.2.2 Richiesta casa

Analisi testuale

Il richiedente compila una richiesta. Una richiesta comprende la città, lo stato, l'inizio e la fine del periodo desiderato, la descrizione dell'immobile, delle classi di attrazioni, informazioni sul trasporto, delle limitazioni e informazioni sul richiedente. La descrizione dell'immobile contiene il numero di posti letto e di bagni, la presenza di garage, piscina e terrazzo. Le classi di attrazioni si riferiscono ai vari tipi di attrazioni come ristoranti, musei e parchi. Il trasporto comprende l'accessibilità ai mezzi pubblici e la condivisione dell'auto. Le limitazioni disponibili sono il divieto di fumare in casa, la possibilità di suonare oltre un certo orario, di portare bambini piccoli e di avere degli animali. L'agente può vedere le nuove richieste o quelle che gestisce e vederne i dettagli. La lista di richieste può contenere una o più richieste. L'agente può visionare le richieste e decidere se gestirle. L'agente può suggerire una proposta di condivisione al richiedente. La proposta di condivisione contiene l'indirizzo e il periodo di condivisione, una descrizione dell'immobile, le limitazioni, il trasporto, informazioni sul richiedente e infine un contesto. L'agente può visualizzare una proposta di condivisione. L'agente può modificare una proposta di condivisione. Il richiedente può vedere la sua proposta e vedere se accettarla o meno.

Casi d'uso

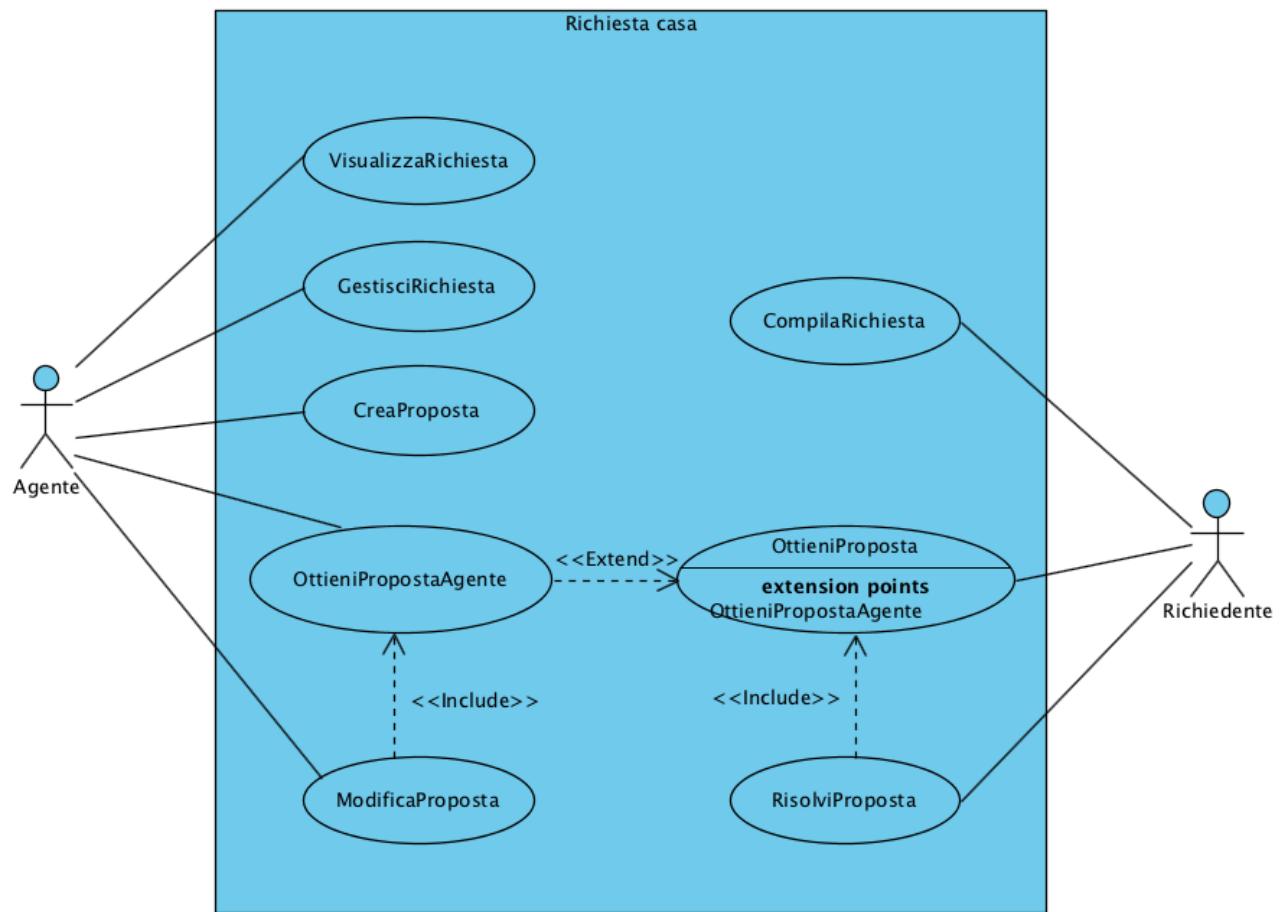


Figura 6.9: Casi d'uso per il modulo Richiesta Casa

Wireframe

The wireframe shows a web browser window titled 'Compila i campi per effettuare la tua richiesta'. The form includes fields for destination ('Nazione: Francia', 'Città: Nizza'), period ('Da: 05/04/2018', 'A: 05/06/2018'), and property details ('Nº postiletto: 2', 'Nº bagni: 1', 'Tipo Immobile: Nessuna preferenza'). It also includes checkboxes for 'Mezzi pubblici' and 'Condivisione auto'. Under 'Attrazioni', there are two columns of checkboxes: 'Concerti e spettacoli', 'Musei', 'Zoo e acquari', 'Attività all'aperto', 'Cibo e bevande', 'Shopping' in the left column, and 'Parchi e natura', 'Siti storici', 'Corsi e seminari', 'Giocchi d'azzardo', 'Parchi divertimenti', 'Spa e benessere' in the right column. At the bottom are buttons for 'Invia', 'Resetta', and 'Annulla'.



Figura 6.10: Il richiedente compila una richiesta

Compila l'annuncio

Immobile e periodo

- Stato: Svezia
- Città: Stoccolma
- Via: Jakobsgatan
- N° civico: 25
- CAP: 100 55
- Da: 17/05/2017
- A: 01/01/2018

Trasporto e restrizioni

Mezzi pubblici: No Sì

Condivisione auto: No Sì

Fumatore: No Sì

Bambini piccoli: No Sì

Animali: No Sì

Suonare in orari notturni: No Sì

Descrizione Immobile

Un cottage avente una cucina, salotto, 2 bagni, 4 posti letto, una piscina ed un terrazzo.

Inserisci contesto

Ci sono 2 ristoranti nel raggio di 500m, un parco giochi e una piscina comunale.

Compila l'annuncio

Immobile e periodo

- Stato: Svezia
- Città: Stoc
- Via: Jak
- N° civico: 25
- CAP: 100
- Da: 17/0
- A: 01/0

Trasporto e restrizioni

Mezzi pubblici: No Sì

Digita il nome della foto da aggiungere
search...

Descrizione Immobile

Un cottage avente un letto, una piscina ed u

Inserisci contesto

Ci sono 2 ristoranti nel raggio di 500m, un parco giochi e una piscina comunale

Proponi

Compila l'annuncio

Immobile e periodo

- Stato: Svezia
- Città: Stoccolma
- Via: Jakobsgatan
- N° civico: 25
- CAP: 100 55
- Da: 17/05/2017
- A: 01/01/2018

Trasporto e restrizioni

Mezzi pubblici: No Sì

Condivisione auto: No Sì

Fumatore: No Sì

Bambini piccoli: No Sì

Animali: No Sì

Suonare in orari notturni: No Sì

Descrizione Immobile

Un cottage avente una cucina, salotto, 2 bagni, 4 posti letto, una piscina ed un terrazzo.

Inserisci contesto

Ci sono 2 ristoranti nel raggio di 500m, un parco giochi e una piscina comunale

Foto

Carica

X X X

Proponi

Figura 6.11 L'agente crea una proposta

Selezione che tipi di richieste visualizzare.

| RichiestaID | Nome Cliente | Destinazione | Periodo | Seleziona |
|-------------|-------------------|--------------------|--------------------------|-------------------------------------|
| 0056 | Giuseppe La Russa | Parigi, Francia | 05/05/2017 05/07/2017 | <input type="checkbox"/> |
| 0075 | Mario Rossi | Barcellona, Spagna | 11/04/2017 21/04/2017 | <input checked="" type="checkbox"/> |
| 0101 | Samantha Foglia | Barcellona, Spagna | 24/04/2017 01/08/2017 | <input type="checkbox"/> |

Browser

Sei sicuro di voler gestire la richiesta?

Browser

Hai accettato di gestire la seguente richiesta

| | |
|--------------|--------------------------|
| ID | 0075 |
| Nome | Mario Rossi |
| Destinazione | Barcellona, Spagna |
| Periodo | 11/04/2017 21/04/2017 |

Figura 6.12: L'agente gestisce una richiesta

Inserisci l'annuncio

| | |
|-----------|-------------|
| Stato | Svezia |
| Città | Stoccolma |
| Via | Jakobsgatan |
| N° civico | 23 |
| CAP | 100 55 |
| Da | 17/05/2017 |
| A | 19/09/2017 |

Trasporto e restrizioni

| | |
|---------------------------|--|
| Mezzi pubblici | <input type="radio"/> No <input type="radio"/> Sì |
| Condivisione auto | <input type="radio"/> No <input checked="" type="radio"/> Sì |
| Fumatore | <input type="radio"/> No <input checked="" type="radio"/> Sì |
| Bambini piccoli | <input type="radio"/> No <input type="radio"/> Sì |
| Animali | <input type="radio"/> No <input checked="" type="radio"/> Sì |
| Suonare in orari notturni | <input type="radio"/> No <input type="radio"/> Sì |

Descrizione Immobile

Villa con 2 camere da letto, bagno, cucina, salotto, ripostiglio, terrazzo e giardino

Foto

Inserisci l'annuncio

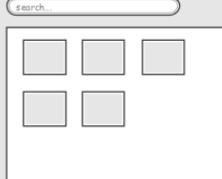
| | |
|-----------|-------------|
| Stato | Svezia |
| Città | Stoccolma |
| Via | Jakobsgatan |
| N° civico | 23 |
| CAP | 100 55 |
| Da | 17/05/2017 |
| A | 19/09/2017 |

Trasporto e restrizioni

| | |
|-------------------|--|
| Mezzi pubblici | <input type="radio"/> No <input type="radio"/> Sì |
| Condivisione auto | <input type="radio"/> No <input checked="" type="radio"/> Sì |

Digita il nome della foto da aggiungere

search...



Descrizione Immobile

Villa con 2 camere da letto, bagno, cucina, salotto, ripostiglio, terrazzo e giardino

Inserisci contesto

Gi sono dei ristoranti nel raggio di 250m, un parco giochi a 500m.

Figura 6.13: L'agente modifica una proposta

Figura 6.14: L'utente ottiene una proposta

Figura 6.15: l'Agente ottiene una proposta

Browser

| | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|--|-------------------------------|--------------------------------|-------------|-------|----------|--------------|-----------|-------------|-------|-------------------|---|-------|--------|-------|-----------|-----|-------------|-----------|----|-----|--------|----|------------|---|------------|---|----------------|----|-------------------|----|----------|---|-----------------|---|---------|----|---------------------------|----|---|
| Dati personali | Destinazione e periodo | Trasporto e restrizioni | Foto | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| <table border="1"> <tr><td>Nome</td><td>Mario</td></tr> <tr><td>Cognome</td><td>Rossi</td></tr> <tr><td>Telefono</td><td>0342 2343424</td></tr> <tr><td>Cellulare</td><td>333 3333333</td></tr> <tr><td>Email</td><td>mariorossi@msn.it</td></tr> </table> | Nome | Mario | Cognome | Rossi | Telefono | 0342 2343424 | Cellulare | 333 3333333 | Email | mariorossi@msn.it | <table border="1"> <tr><td>Stato</td><td>Svezia</td></tr> <tr><td>Città</td><td>Stoccolma</td></tr> <tr><td>Via</td><td>Jakobsgatan</td></tr> <tr><td>Nº civico</td><td>23</td></tr> <tr><td>CAP</td><td>100 55</td></tr> <tr><td>Da</td><td>13/05/2017</td></tr> <tr><td>A</td><td>17/09/2017</td></tr> </table> | Stato | Svezia | Città | Stoccolma | Via | Jakobsgatan | Nº civico | 23 | CAP | 100 55 | Da | 13/05/2017 | A | 17/09/2017 | <table border="1"> <tr><td>Mezzi pubblici</td><td>No</td></tr> <tr><td>Condivisione auto</td><td>Sì</td></tr> <tr><td>Fumatore</td><td>-</td></tr> <tr><td>Bambini piccoli</td><td>-</td></tr> <tr><td>Animali</td><td>Sì</td></tr> <tr><td>Suonare in orari notturni</td><td>Sì</td></tr> </table> | Mezzi pubblici | No | Condivisione auto | Sì | Fumatore | - | Bambini piccoli | - | Animali | Sì | Suonare in orari notturni | Sì |    |
| Nome | Mario | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Cognome | Rossi | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Telefono | 0342 2343424 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Cellulare | 333 3333333 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Email | mariorossi@msn.it | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Stato | Svezia | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Città | Stoccolma | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Via | Jakobsgatan | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Nº civico | 23 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| CAP | 100 55 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Da | 13/05/2017 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| A | 17/09/2017 | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Mezzi pubblici | No | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Condivisione auto | Sì | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Fumatore | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Bambini piccoli | - | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Animali | Sì | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| Suonare in orari notturni | Sì | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Descrizione Immobile

Baita con 3 bagni, cucina, salotto, giardino, terrazzo, garage, zona residenziale

Inserisci contesto

zone d'interesse, parchi, ...

Accetta **Prossimo**

Browser

Sei sicuro di accettare questa proposta?

Si **No**

Browser

Ottima scelta! Adesso verrà contattato il proprietario dell'immobile.

Indietro

Figura 6.16: Il richiedente risolve una proposta

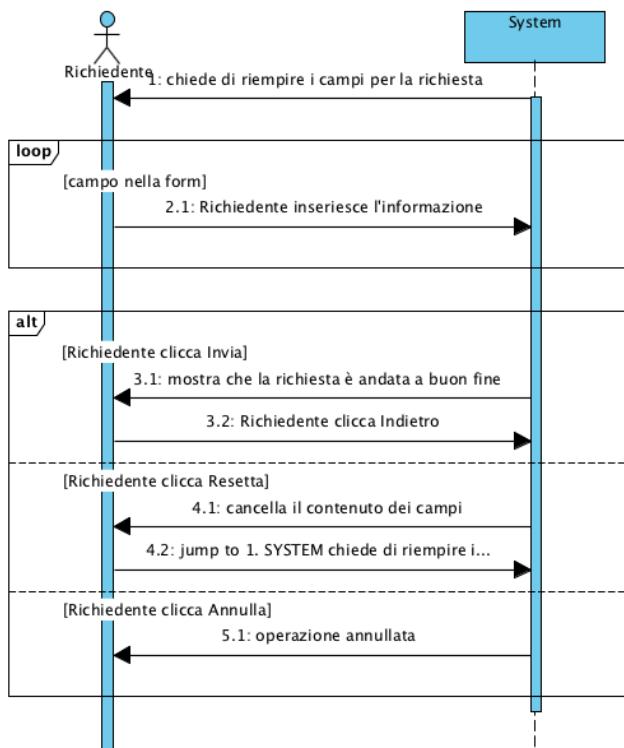
| Browser | Browser | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
|---|-------------------|--------------------|--------------------------|--------------------------|-----------|------|---------------|--------------|-------------------------|--------------------------|------|----------------|-----------------|-------------------------|--------------------------|---|-------------|--------------|--------------|---------|-----------|------|-------------------|-----------------|--------------------------|--------------------------|------|-------------|--------------------|--------------------------|--------------------------|------|-----------------|--------------------|--------------------------|--------------------------|
| <p>Selezione che tipi di richieste visualizzare</p> <p>Gestite</p> <table border="1"> <thead> <tr><th>RichiestaID</th><th>Nome Cliente</th><th>Destinazione</th><th>Periodo</th><th>Selezione</th></tr> </thead> <tbody> <tr><td>0003</td><td>Piero Falcone</td><td>Roma, Italia</td><td>15/01/2018 23/04/201</td><td><input type="checkbox"/></td></tr> <tr><td>0044</td><td>Lisa Navarrone</td><td>Parigi, Francia</td><td>15/01/2018 23/04/201</td><td><input type="checkbox"/></td></tr> </tbody> </table> <p>Visualizza</p> | RichiestaID | Nome Cliente | Destinazione | Periodo | Selezione | 0003 | Piero Falcone | Roma, Italia | 15/01/2018 23/04/201 | <input type="checkbox"/> | 0044 | Lisa Navarrone | Parigi, Francia | 15/01/2018 23/04/201 | <input type="checkbox"/> | <p>Selezione che tipi di richieste visualizzare</p> <p>Non gestite</p> <table border="1"> <thead> <tr><th>RichiestaID</th><th>Nome Cliente</th><th>Destinazione</th><th>Periodo</th><th>Selezione</th></tr> </thead> <tbody> <tr><td>0056</td><td>Giuseppe La Russa</td><td>Parigi, Francia</td><td>05/05/2017 05/07/2017</td><td><input type="checkbox"/></td></tr> <tr><td>0075</td><td>Mario Rossi</td><td>Barcellona, Spagna</td><td>11/04/2017 21/04/2017</td><td><input type="checkbox"/></td></tr> <tr><td>0101</td><td>Samantha Foglia</td><td>Barcellona, Spagna</td><td>24/04/2017 01/06/2017</td><td><input type="checkbox"/></td></tr> </tbody> </table> <p>Visualizza Gestisci</p> | RichiestaID | Nome Cliente | Destinazione | Periodo | Selezione | 0056 | Giuseppe La Russa | Parigi, Francia | 05/05/2017 05/07/2017 | <input type="checkbox"/> | 0075 | Mario Rossi | Barcellona, Spagna | 11/04/2017 21/04/2017 | <input type="checkbox"/> | 0101 | Samantha Foglia | Barcellona, Spagna | 24/04/2017 01/06/2017 | <input type="checkbox"/> |
| RichiestaID | Nome Cliente | Destinazione | Periodo | Selezione | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0003 | Piero Falcone | Roma, Italia | 15/01/2018 23/04/201 | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0044 | Lisa Navarrone | Parigi, Francia | 15/01/2018 23/04/201 | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| RichiestaID | Nome Cliente | Destinazione | Periodo | Selezione | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0056 | Giuseppe La Russa | Parigi, Francia | 05/05/2017 05/07/2017 | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0075 | Mario Rossi | Barcellona, Spagna | 11/04/2017 21/04/2017 | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |
| 0101 | Samantha Foglia | Barcellona, Spagna | 24/04/2017 01/06/2017 | <input type="checkbox"/> | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | | |

Figura 6.17: L'agente visualizza una richiesta

Casi d'uso dettagliati e diagrammi di sequenza

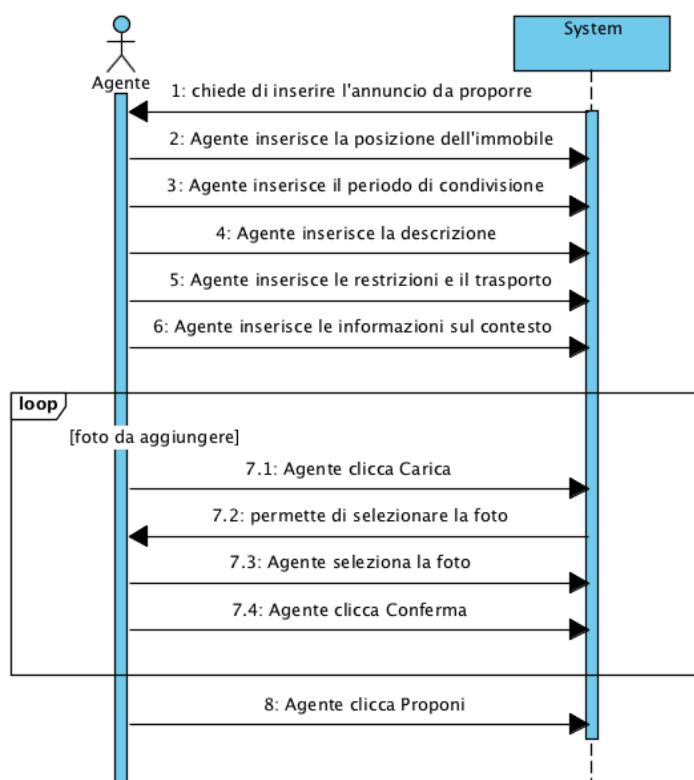
CompilaRichiesta

1. **SYSTEM** chiede di riempire i campi per la richiesta
2. **for each** campo nella form
 - 2.1. **Richiedente** inserisce l'informazione**end for each**
3. **if** **Richiedente** clicca **Invia**
 - 3.1. **SYSTEM** mostra che la richiesta è andata a buon fine
 - 3.2. **Richiedente** clicca **Indietro**
4. **else if** **Richiedente** clicca **Resetta**
 - 4.1. **SYSTEM** cancella il contenuto dei campi
 - 4.2. **jump to** 1. **SYSTEM** chiede di riempire i...
5. **else if** **Richiedente** clicca **Nulla**
 - 5.1. **SYSTEM** operazione annullata**end if**



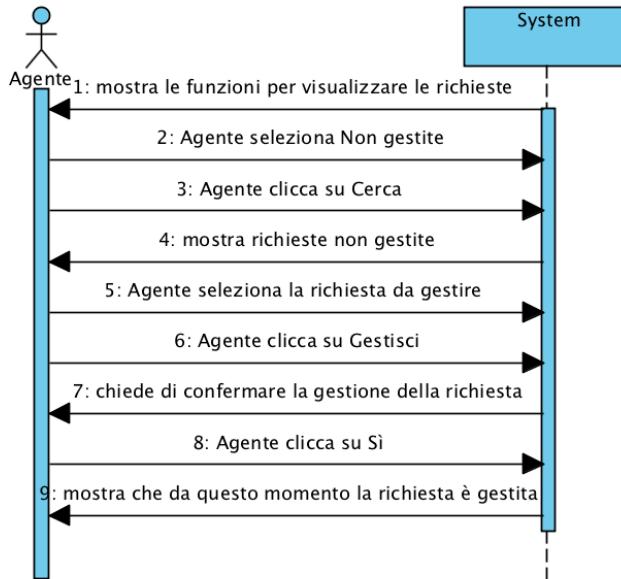
CreaProposta

1. **SYSTEM** chiede di inserire l'annuncio da proporre
2. **Agente** inserisce la posizione dell'immobile
3. **Agente** inserisce il periodo di condivisione
4. **Agente** inserisce la descrizione
5. **Agente** inserisce le restrizioni e il trasporto
6. **Agente** inserisce le informazioni sul contesto
7. **for each** foto da aggiungere
 - 7.1. **Agente** clicca **Carica**
 - 7.2. **SYSTEM** permette di selezionare la foto
 - 7.3. **Agente** seleziona la foto
 - 7.4. **Agente** clicca **Conferma****end for each**
8. **Agente** clicca **Proponi**



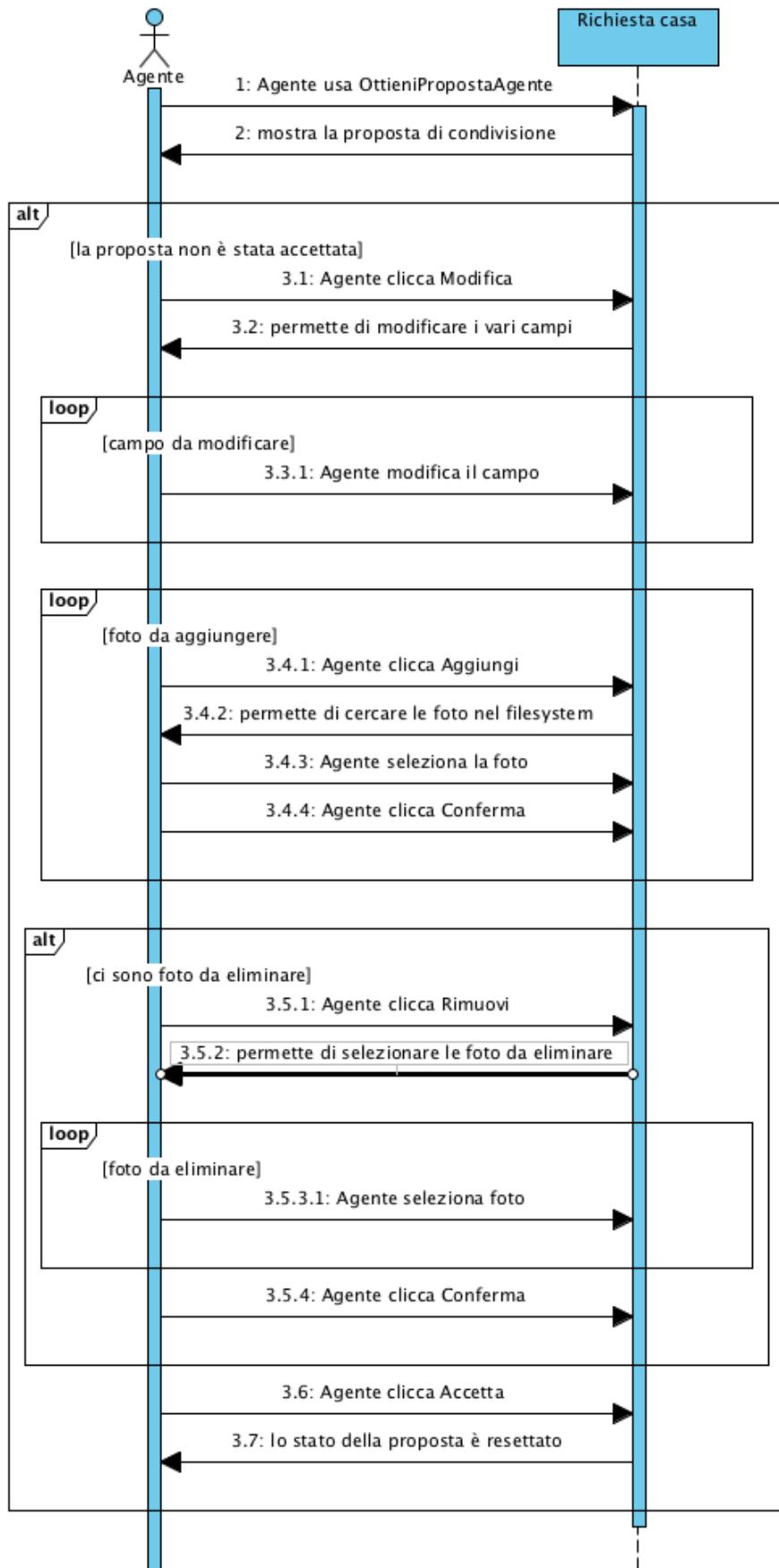
GestisciRichiesta

1. **SYSTEM** mostra le funzioni per visualizzare le richieste
2. **Agente** seleziona **Non gestite**
3. **Agente** clicca su **Cerca**
4. **SYSTEM** mostra richieste non gestite
5. **Agente** seleziona la richiesta da gestire
6. **Agente** clicca su **Gestisci**
7. **SYSTEM** chiede di confermare la gestione della richiesta
8. **Agente** clicca su **Sì**
9. **SYSTEM** mostra che da questo momento la richiesta è gestita



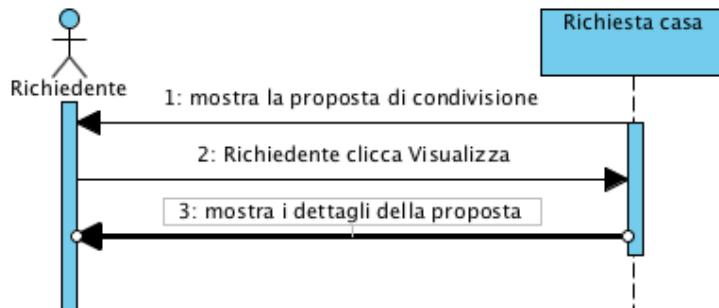
ModificaProposta

1. ⚡ Agente usa ⚡ OttieniPropostaAgente
 2. SYSTEM mostra la proposta di condivisione
 3. if la proposta non è stata accettata
 - 3.1. ⚡ Agente clicca Modifica
 - 3.2. SYSTEM permette di modificare i vari campi
 - 3.3. for each campo da modificare
 - 3.3.1. ⚡ Agente modifica il campo
 - end for each
 - 3.4. for each foto da aggiungere
 - 3.4.1. ⚡ Agente clicca Aggiungi
 - 3.4.2. SYSTEM permette di cercare le foto nel filesystem
 - 3.4.3. ⚡ Agente seleziona la foto
 - 3.4.4. ⚡ Agente clicca Conferma
 - end for each
 - 3.5. if ci sono foto da eliminare
 - 3.5.1. ⚡ Agente clicca Rimuovi
 - 3.5.2. SYSTEM permette di selezionare le foto da eliminare
 - 3.5.3. for each foto da eliminare
 - 3.5.3.1. ⚡ Agente seleziona foto
 - end for each
 - 3.5.4. ⚡ Agente clicca Conferma
 - end if
 - 3.6. ⚡ Agente clicca Accetta
 - 3.7. SYSTEM lo stato della proposta è resettato
- end if



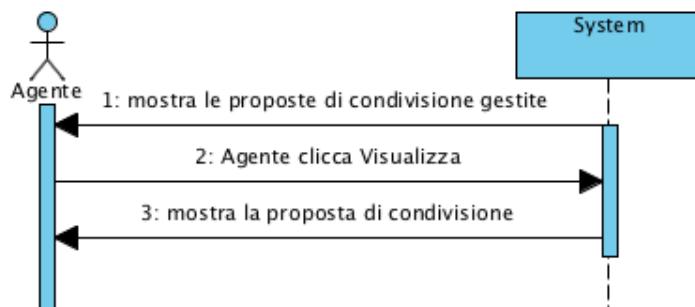
OttieniProposta

1. SYSTEM mostra la proposta di condivisione
2. ⚡ Richiedente clicca Visualizza
3. SYSTEM mostra i dettagli della proposta



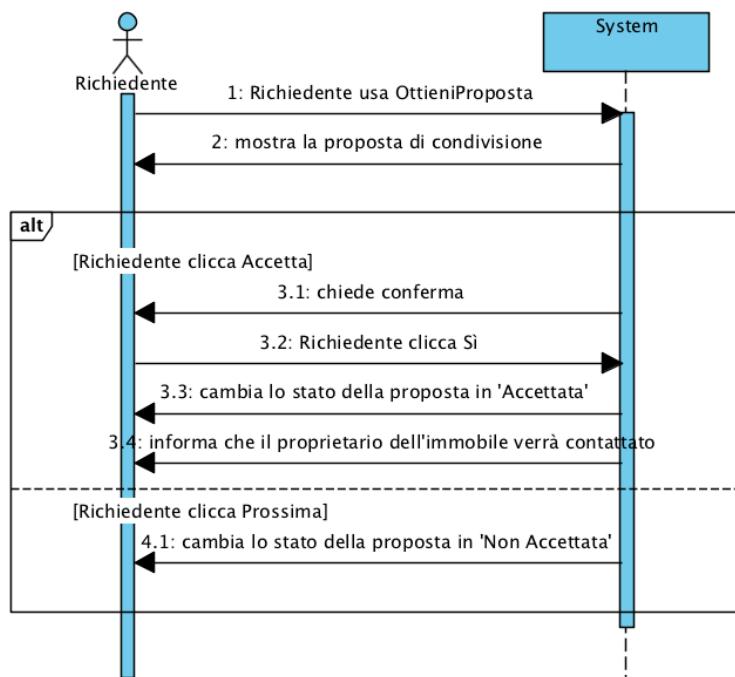
OttieniPropostaAgente

1. SYSTEM mostra le proposte di condivisione gestite
2. ⚡ Agente clicca Visualizza
3. SYSTEM mostra la proposta di condivisione



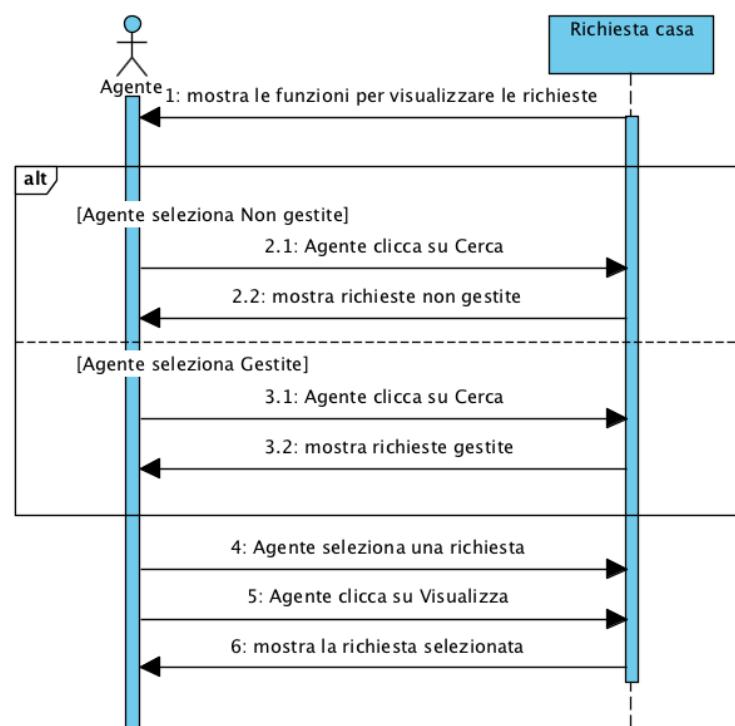
RisolviProposta

1. ⚡ Richiedente usa ⚡ OttieniProposta
2. SYSTEM mostra la proposta di condivisione
3. if ⚡ Richiedente clicca Accetta
 - 3.1. SYSTEM chiede conferma
 - 3.2. ⚡ Richiedente clicca Sì
 - 3.3. SYSTEM cambia lo stato della proposta in 'Accettata'
 - 3.4. SYSTEM informa che il proprietario dell'immobile verrà contattato
4. else if ⚡ Richiedente clicca Prossima
 - 4.1. SYSTEM cambia lo stato della proposta in 'Non Accettata'
- end if



VisualizzaRichiesta

1. SYSTEM mostra le funzioni per visualizzare le richieste
2. if ⚡ Agente seleziona Non gestite
 - 2.1. ⚡ Agente clicca su Cerca
 - 2.2. SYSTEM mostra richieste non gestite
3. else if ⚡ Agente seleziona Gestite
 - 3.1. ⚡ Agente clicca su Cerca
 - 3.2. SYSTEM mostra richieste gestite
- end if
- Agente seleziona una richiesta
- Agente clicca su Visualizza
6. SYSTEM mostra la richiesta selezionata



6.2.3 Attrazioni

Analisi testuale

Il servizio offre all'agente la possibilità di ottenere una lista di attrazioni accanto a un determinato indirizzo; per ogni attrazione sono specificate la classe, l'indirizzo e la distanza dall'indirizzo passato. L'agente può inoltre suggerire al richiedente una lista di possibili attività da svolgere, specificando per ogni attività il periodo, la zona e la classe di attrazioni (ristoranti, musei, parchi, locali notturni, ...) da includere. Ogni elemento della lista contiene un giorno all'interno del periodo, una fascia oraria e una lista di attrazioni che possono essere prese in considerazione.

Il richiedente consulta la lista di proposte e può visualizzare i dettagli di ogni attrazione inclusa e rimuoverla se non la gradisce. Il richiedente può accettare la lista di proposte di attività se la gradisce e non desidera che venga più modificata dall'agente.

L'agente può visualizzare la lista di proposte di attività che aveva presentato al cliente e decidere se aggiungere o rimuovere attrazioni dalla proposta, se il richiedente non l'ha ancora accettata.

Casi d'uso

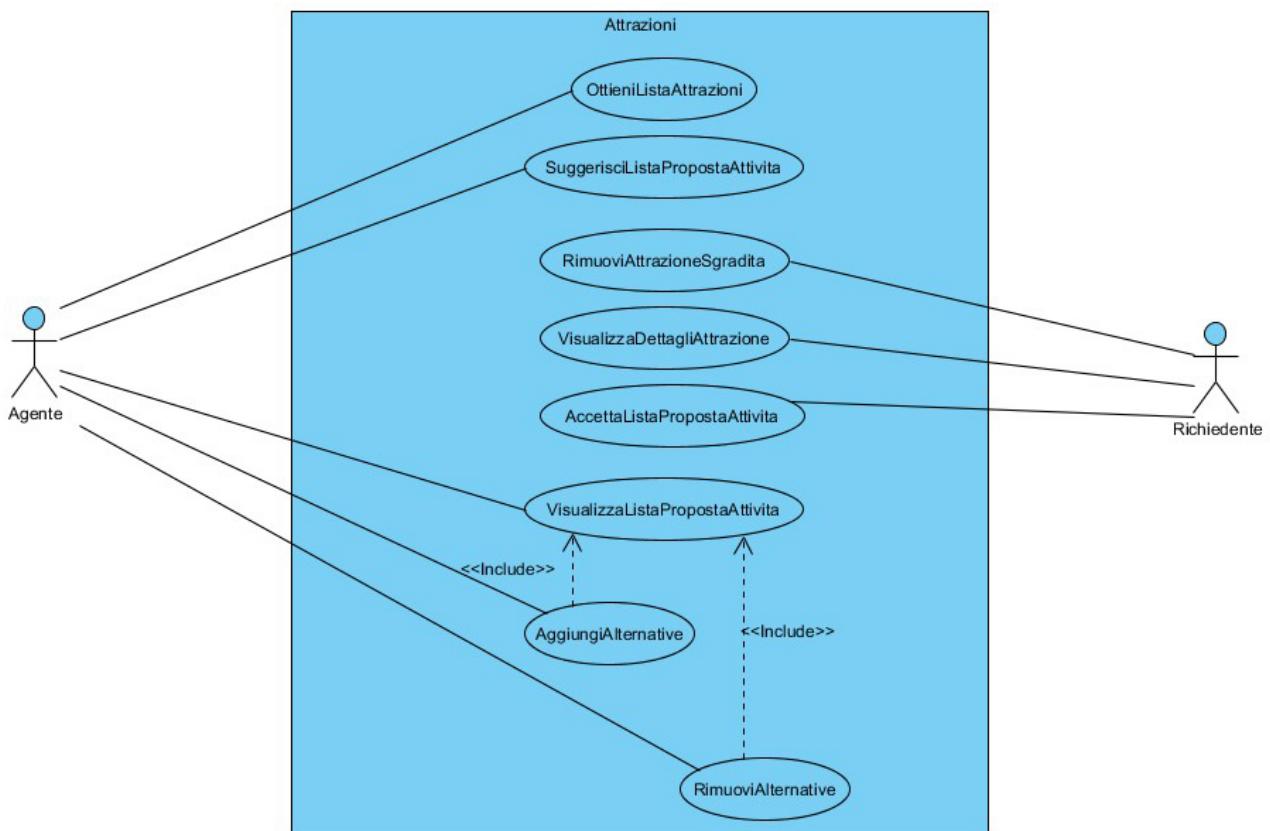


Figura 6.18: Casi d'uso per il modulo Attrazioni

Wireframe

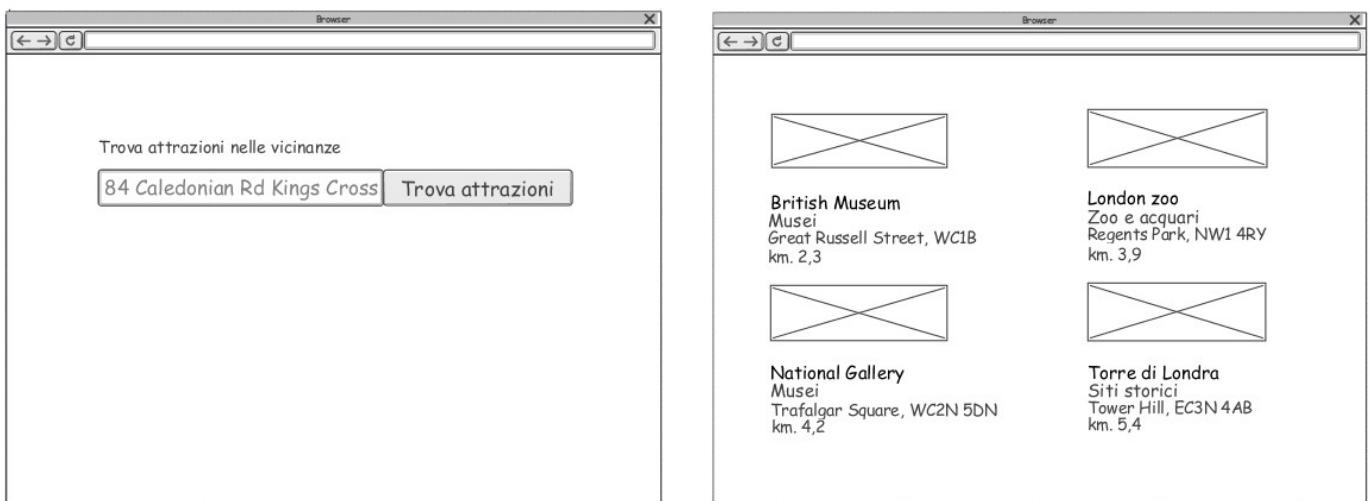


Figura 6.19: L'agente ottiene una lista d'attrazioni

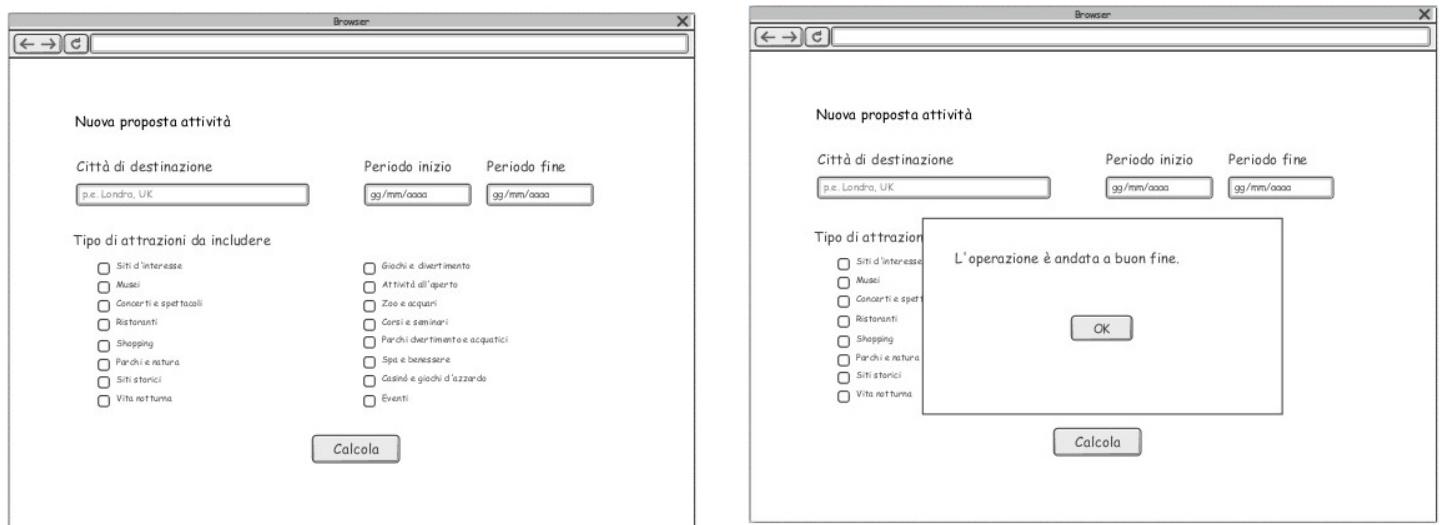


Figura 6.20: L'agente suggerisce una lista proposta di attività

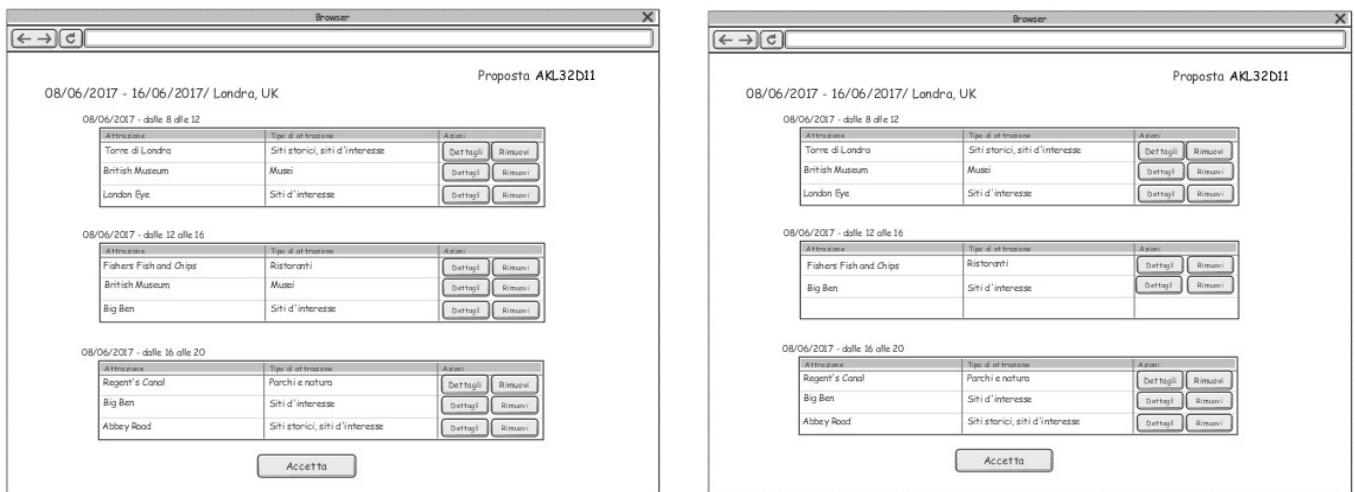


Figura 6.21: Il richiedente rimuove un'attrazione sgradita

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione | Azioni |
|-----------------|--------------------------------|--|
| Torre di Londra | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| London Eye | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione | Azioni |
|------------------------|--------------------|--|
| Fishers Fish and Chips | Ristoranti | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione | Azioni |
|----------------|--------------------------------|--|
| Regent's Canal | Parchi e natura | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Abbey Road | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione | Azioni |
|-----------------|--------------------------------|--|
| Torre di Londra | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| London Eye | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione | Azioni |
|------------------------|--------------------|--|
| Fishers Fish and Chips | Ristoranti | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione | Azioni |
|----------------|--------------------------------|--|
| Regent's Canal | Parchi e natura | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Abbey Road | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

Figura 6.22: Il richiedente visualizza i dettagli di un'attrazione

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione | Azioni |
|-----------------|--------------------------------|--|
| Torre di Londra | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| London Eye | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione | Azioni |
|------------------------|--------------------|--|
| Fishers Fish and Chips | Ristoranti | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione | Azioni |
|----------------|--------------------------------|--|
| Regent's Canal | Parchi e natura | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Abbey Road | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

Dopo aver accettato, non protraî più modificare la lista di proposte di attività. Continuare?

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione | Azioni |
|-----------------|--------------------------------|--|
| Torre di Londra | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| London Eye | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione | Azioni |
|------------------------|--------------------|--|
| Fishers Fish and Chips | Ristoranti | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| British Museum | Musei | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione | Azioni |
|----------------|--------------------------------|--|
| Regent's Canal | Parchi e natura | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Big Ben | Siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |
| Abbey Road | Siti storici, siti d'interesse | <input type="button" value="Dettagli"/> <input type="button" value="Rimuovi"/> |

Figura 6.23: Il richiedente accetta una lista di proposta di attività

Ricerca una proposta

Ricerca una proposta

Attenzione! La proposta da te selezionata non esiste.

Browser

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK
Stato: visualizzato

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione |
|-----------------|--------------------------------|
| Torre di Londra | Siti storici, siti d'interesse |
| British Museum | Musei |
| London Eye | Siti d'interesse |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione |
|------------|--------------------|
| Big Ben | Siti d'interesse |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione |
|----------------|--------------------------------|
| Regent's Canal | Parchi e natura |
| Abbey Road | Siti storici, siti d'interesse |

[Aggiungi alternative](#) [Rimuovi alternative](#)

Figura 6.24: L'agente visualizza una lista di proposta di attività

Browser

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK
Stato: visualizzato

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione |
|-----------------|--------------------------------|
| Torre di Londra | Siti storici, siti d'interesse |
| British Museum | Musei |
| London Eye | Siti d'interesse |

aggiungi altre

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione |
|------------|--------------------|
| Big Ben | Siti d'interesse |

aggiungi altre

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione |
|----------------|--------------------------------|
| Regent's Canal | Parchi e natura |
| Abbey Road | Siti storici, siti d'interesse |

aggiungi altre

[Aggiungi](#)

Browser

Proposta AKL32D11

08/06/2017 - 16/06/2017/ Londra, UK
Stato: visualizzato

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione |
|-----------------|--------------------------------|
| Torre di Londra | Siti storici, siti d'interesse |
| British Museum | Musei |
| London Eye | Siti d'interesse |

aggiungi altre

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione |
|------------|--------------------|
| Big Ben | Siti d'interesse |

aggiungi altre

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione |
|----------------|--------------------------------|
| Regent's Canal | Parchi e natura |
| Abbey Road | Siti storici, siti d'interesse |

aggiungi altre

[Aggiungi](#)

Figura 6.25 L'agente chiede di aggiungere alternative

Browser

Proposta AKL32D11

08/06/2017 - 16/06/2017 / Londra, UK

Stato: visualizzato

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione |
|-----------------|--------------------------------|
| Torre di Londra | Siti storici, siti d'interesse |
| British Museum | Musei |
| London Eye | Siti d'interesse |

da rimuovere
 da rimuovere
 da rimuovere

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione |
|------------|--------------------|
| Big Ben | Siti d'interesse |

da rimuovere
 da rimuovere
 da rimuovere

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione |
|----------------|--------------------------------|
| Regent's Canal | Parchi e natura |
| Abbey Road | Siti storici, siti d'interesse |

da rimuovere
 da rimuovere
 da rimuovere

Rimuovi

Browser

Proposta AKL32D11

08/06/2017 - 16/06/2017 / Londra, UK

Stato: da visualizzare

08/06/2017 - dalle 8 alle 12

| Attrazione | Tipo di attrazione |
|-----------------|--------------------------------|
| Torre di Londra | Siti storici, siti d'interesse |
| British Museum | Musei |

08/06/2017 - dalle 12 alle 16

| Attrazione | Tipo di attrazione |
|------------|--------------------|
| Big Ben | Siti d'interesse |

08/06/2017 - dalle 16 alle 20

| Attrazione | Tipo di attrazione |
|----------------|--------------------------------|
| Regent's Canal | Parchi e natura |
| Abbey Road | Siti storici, siti d'interesse |

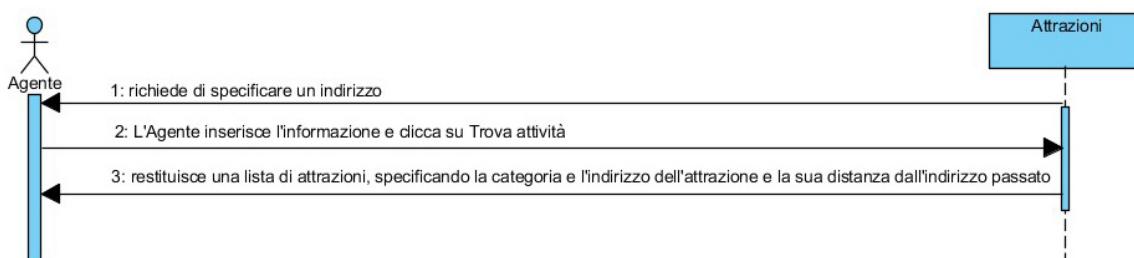
Aggiungi alternative Rimuovi alternative

Figura 6.26: L'agente rimuove delle alternative

Casi d'uso dettagliati e diagrammi di sequenza

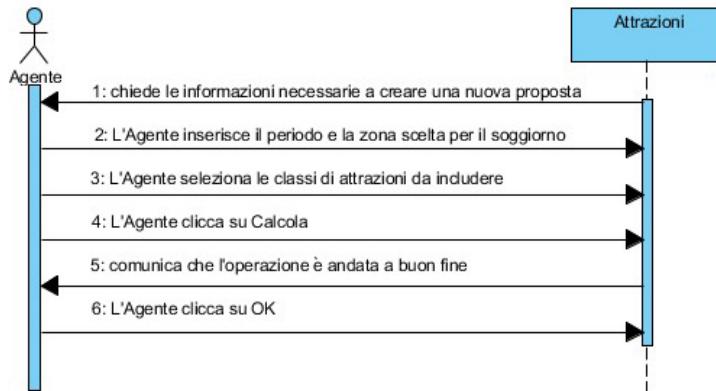
OttieniListaAttrazioni

1. **SYSTEM** richiede di specificare un indirizzo
2. L' **Agente** inserisce l'informazione e clicca su **Trova attività**
3. **SYSTEM** restituisce una lista di attrazioni, specificando la categoria e l'indirizzo dell'attrazione e la sua distanza dall'indirizzo passato



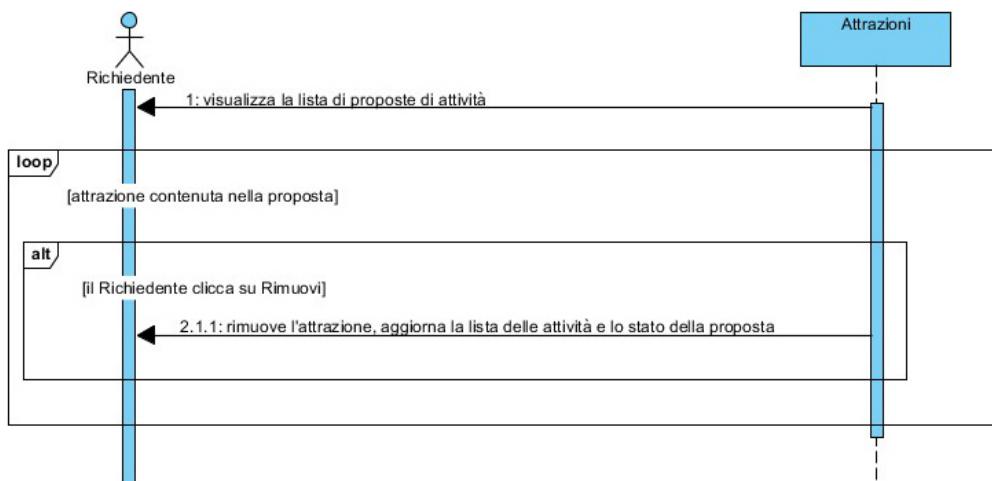
SuggerisciListaPropostaAttività

1. **SYSTEM** chiede le informazioni necessarie a creare una nuova proposta
2. L' **Agente** inserisce il periodo e la zona scelta per il soggiorno
3. L' **Agente** seleziona le classi di attrazioni da includere
4. L' **Agente** clicca su **Calcola**
5. **SYSTEM** comunica che l'operazione è andata a buon fine
6. L' **Agente** clicca su **OK**



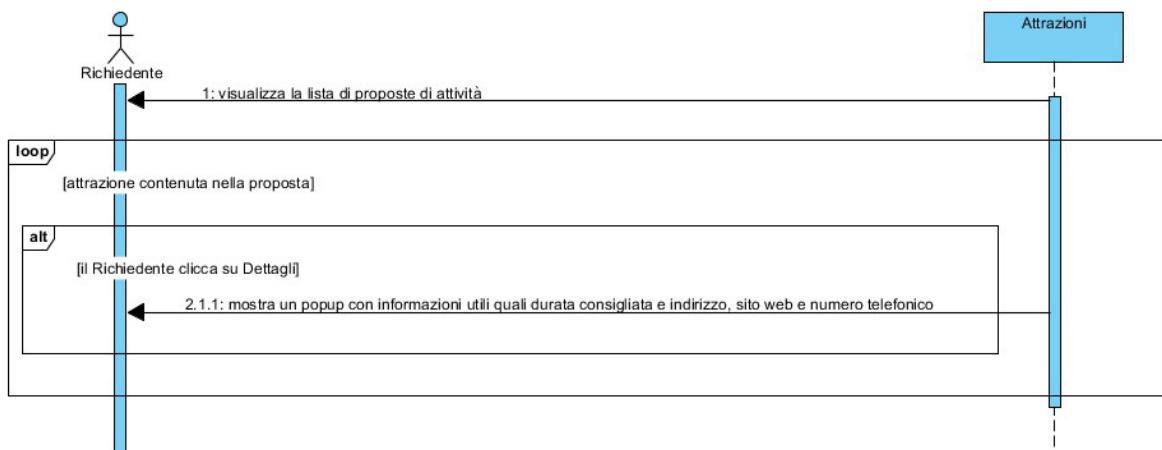
RimuoviAttrazioneSgradita

1. **SYSTEM** visualizza la lista di proposte di attività
2. **for each** attrazione contenuta nella proposta
 - 2.1. **if** il  **Richiedente** clicca su **Rimuovi**
 - 2.1.1. **SYSTEM** rimuove l'attrazione, aggiorna la lista delle attività e lo stato della proposta



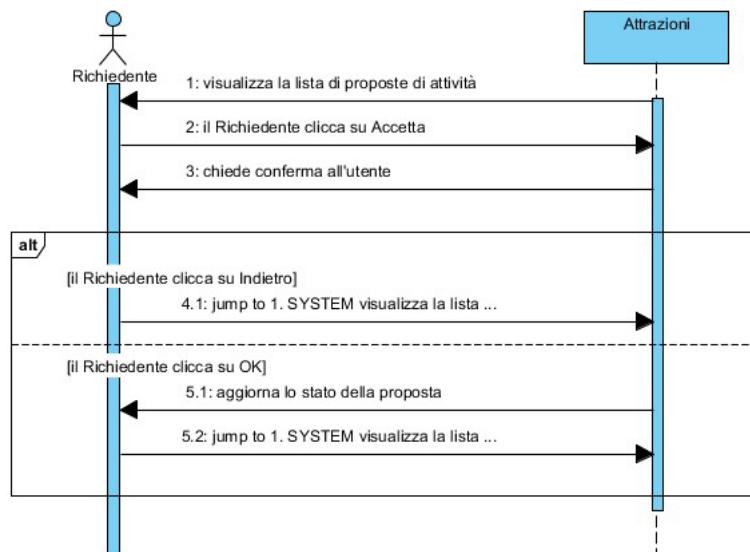
VisualizzaDettagliAttrazione

1. **SYSTEM** visualizza la lista di proposte di attività
2. **for each** attrazione contenuta nella proposta
 - 2.1. **if** il  **Richiedente** clicca su **Dettagli**
 - 2.1.1. **SYSTEM** mostra un popup con informazioni utili quali durata consigliata e indirizzo, sito web e numero telefonico



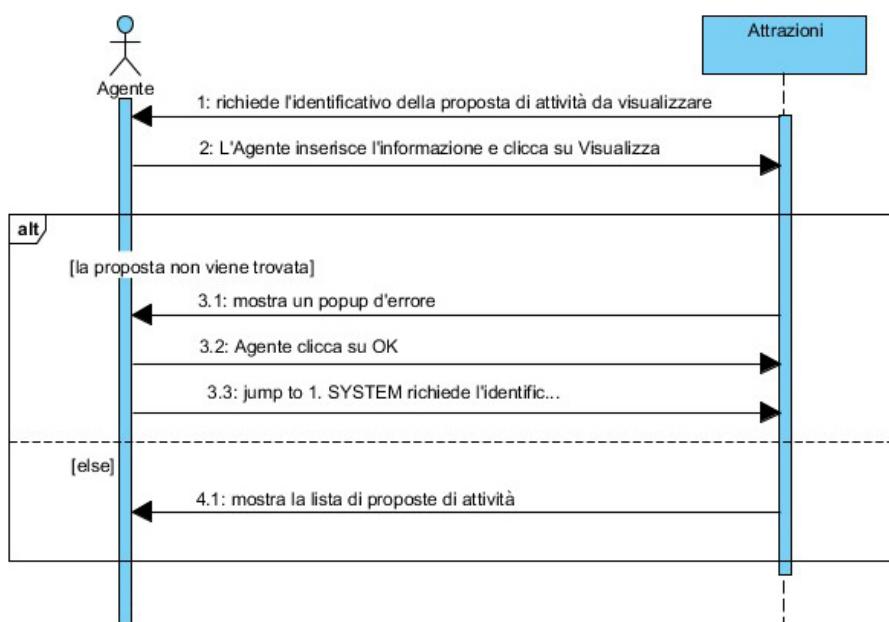
AccettaListaPropostaAttività

1. **SYSTEM** visualizza la lista di proposte di attività
 2. il **Richiedente** clicca su **Accetta**
 3. **SYSTEM** chiede conferma all'utente
 4. **if** il **Richiedente** clicca su **Indietro**
 - 4.1. **jump to** 1. **SYSTEM** visualizza la lista ...
 5. **else if** il **Richiedente** clicca su **OK**
 - 5.1. **SYSTEM** aggiorna lo stato della proposta
 - 5.2. **jump to** 1. **SYSTEM** visualizza la lista ...
- end if**



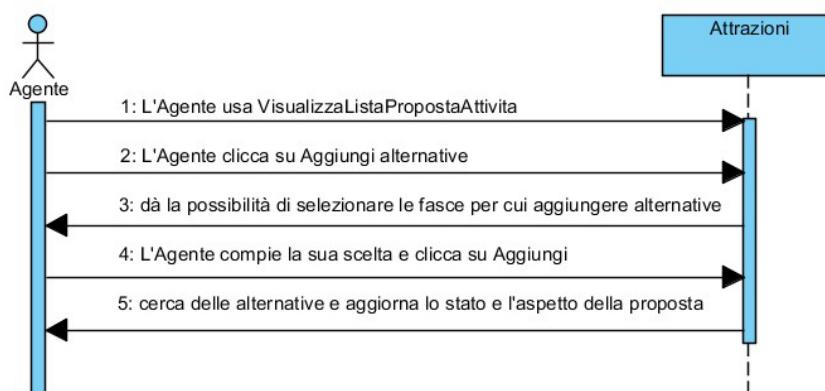
VisualizzaListaPropostaAttivita

1. **SYSTEM** richiede l'identificativo della proposta di attività da visualizzare
 2. L' **Agente** inserisce l'informazione e clicca su **Visualizza**
 3. **if** la proposta non viene trovata
 - 3.1. **SYSTEM** mostra un popup d'errore
 - 3.2.  **Agente** clicca su **OK**
 - 3.3. **jump to** 1. **SYSTEM** richiede l'identific...
 4. **else**
 - 4.1. **SYSTEM** mostra la lista di proposte di attività
- end if**



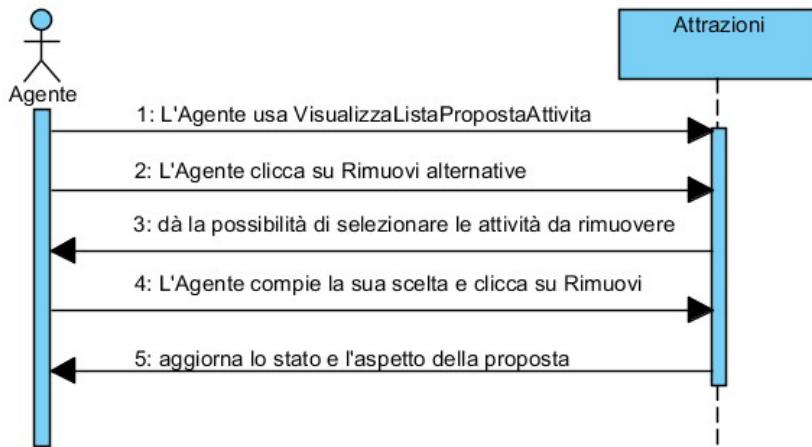
AggiungiAlternative

1. L' **Agente** usa  **VisualizzaListaPropostaAttivita**
2. L' **Agente** clicca su **Aggiungi alternative**
3. **SYSTEM** dà la possibilità di selezionare le fasce per cui aggiungere alternative
4. L' **Agente** compie la sua scelta e clicca su **Aggiungi**
5. **SYSTEM** cerca delle alternative e aggiorna lo stato e l'aspetto della proposta



Rimuovi Alternative

1. L'  **Agente** usa  **VisualizzaListaPropostaAttivita**
2. L'  **Agente** clicca su **Rimuovi alternative**
3. **SYSTEM** dà la possibilità di selezionare le attività da rimuovere
4. L'  **Agente** compie la sua scelta e clicca su **Rimuovi**
5. **SYSTEM** aggiorna lo stato e l'aspetto della proposta



6.2.4 Servizi accessori

Analisi testuale

L'agente può richiedere una lista di servizi accessori disponibili in una data zona. I servizi accessori comprendono giardinaggio, manutenzione della piscina, pulizia domestica, baby-sitting e catering. La lista di servizi comprende uno o più servizi accessori. Il richiedente può visualizzare la lista di servizi e selezionare i servizi a cui è interessato. L'agente visualizza la lista dei servizi scelti, immette le fasce orarie e richiede una pianificazione. Il richiedente può visualizzare e accettare la pianificazione mostrata dall'agente. La pianificazione è composta da una lista di servizi con il rispettivo orario.

Casi d'uso

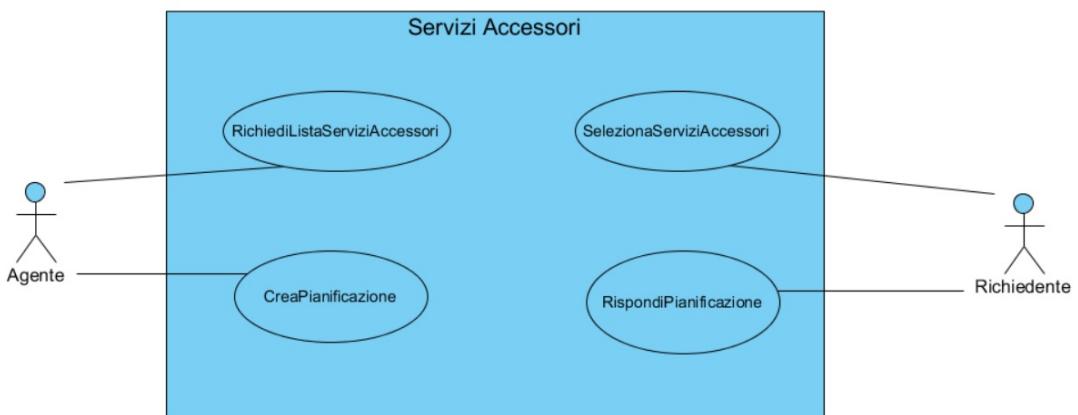


Figura 6.27: Casi d'uso per il modulo Servizi accessori

Wireframe

The four wireframes illustrate the process of creating a planning schedule:

- Step 1:** A table titled "Pianificazione" lists services: Giardino-ggio, Catering, Baby-sitting. Below the table is a button "Immetti Fasce Orarie".
- Step 2:** The same table is shown, but now includes time ranges: Giardino-ggio (D: 10:30, A: 12:30), Catering (D: 14:00, A: 15:00), Baby-sitting (D: 20:30, A: 21:00). Below the table is a button "Crea Pianificazione".
- Step 3:** The table shows the scheduled times. Below the table is a button "Crea Pianificazione".
- Step 4:** A confirmation message "L'operazione è andata a buon fine" with an "OK" button.

Figura 6.28: L'agente crea una pianificazione

The two wireframes show the search for additional services:

- Step 1:** An "Indirizzo" section contains fields for Via, N. Civico, Città, CAP, and Stato. To the right is a "Servizi Accessori" panel.
- Step 2:** The "Indirizzo" section now has populated values: Via [Fondello], N. Civico [23], Città [Trapani], CAP [91100], Stato [Italia]. The "Servizi Accessori" panel lists: Giardino-ggio, Catering, Baby-sitting, Pulizia Domestica. Below the address section is a "Cerca" button.

Figura 6.29: L'agente richiede una lista di servizi accessori

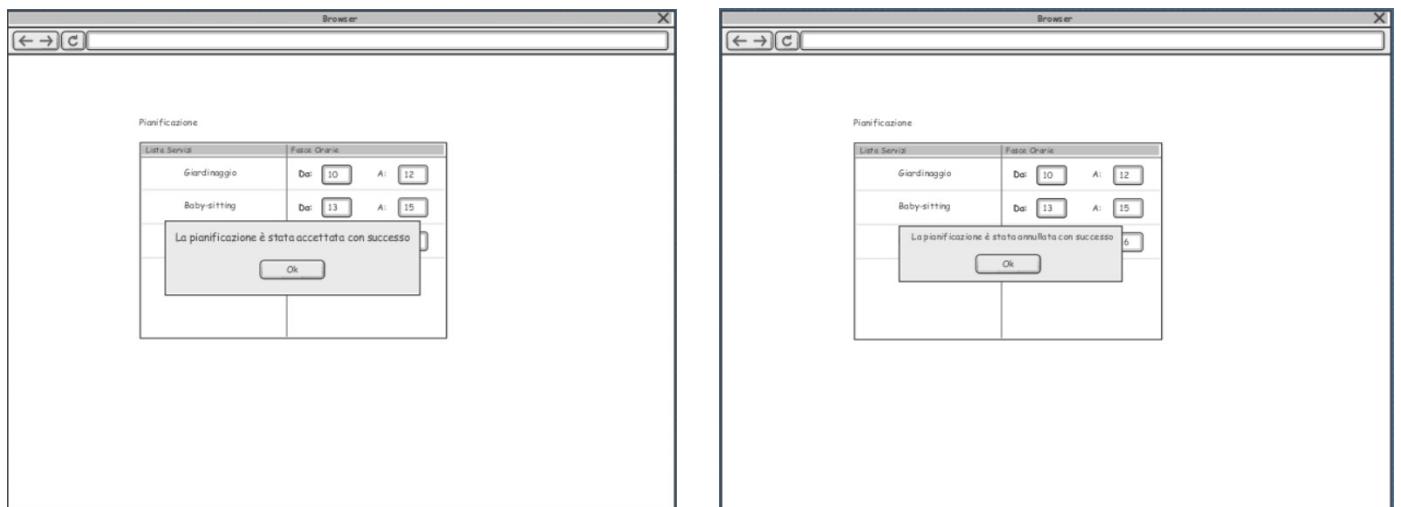
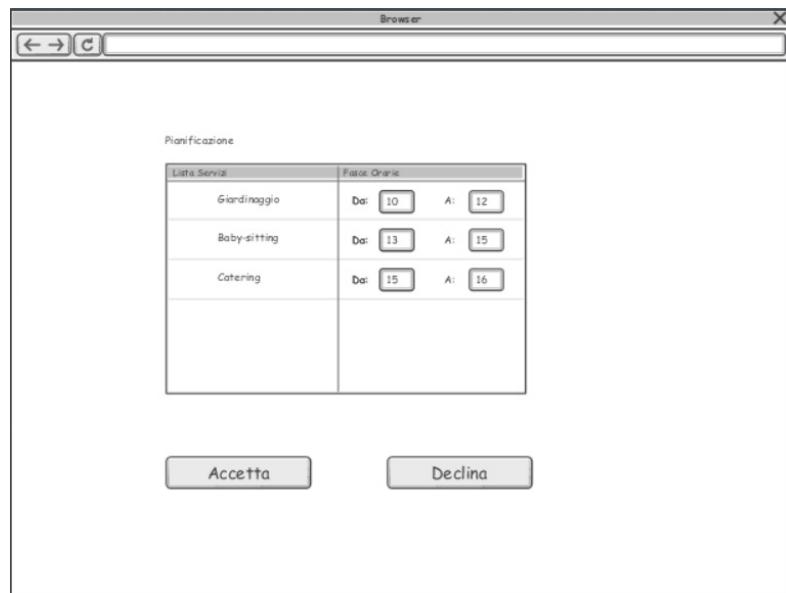
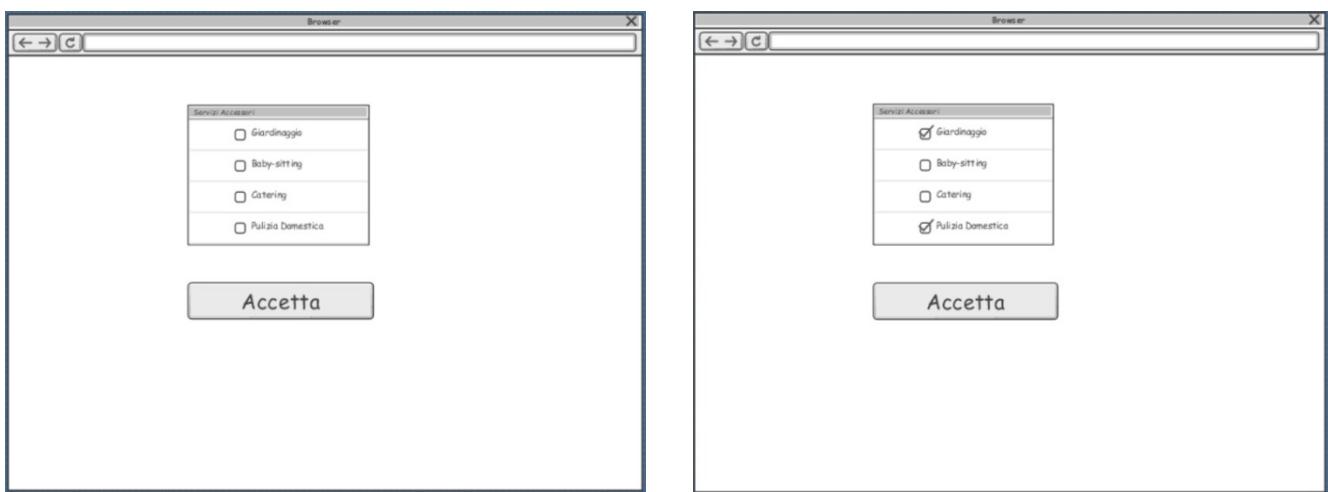


Figura 6.30: Il richiedente accetta o rifiuta una pianificazione



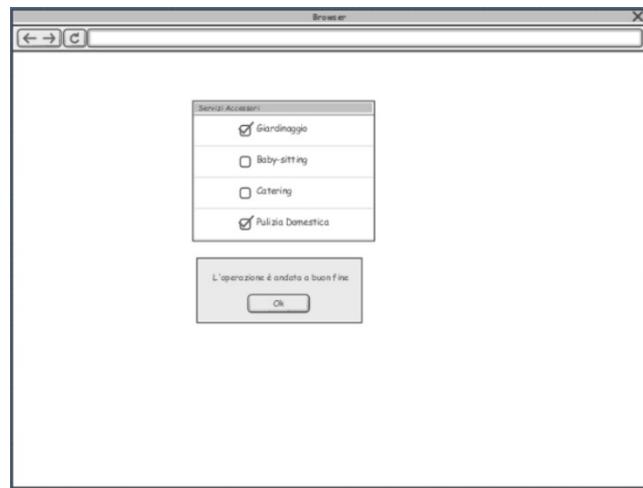
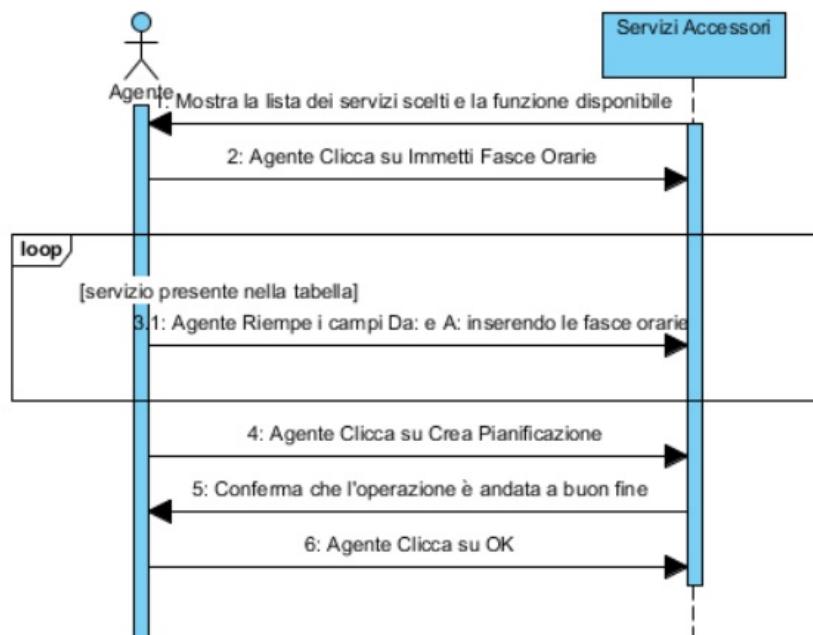


Figura 6.31 Il richiedente seleziona i servizi accessori

Casi d'uso dettagliati e diagrammi di sequenza

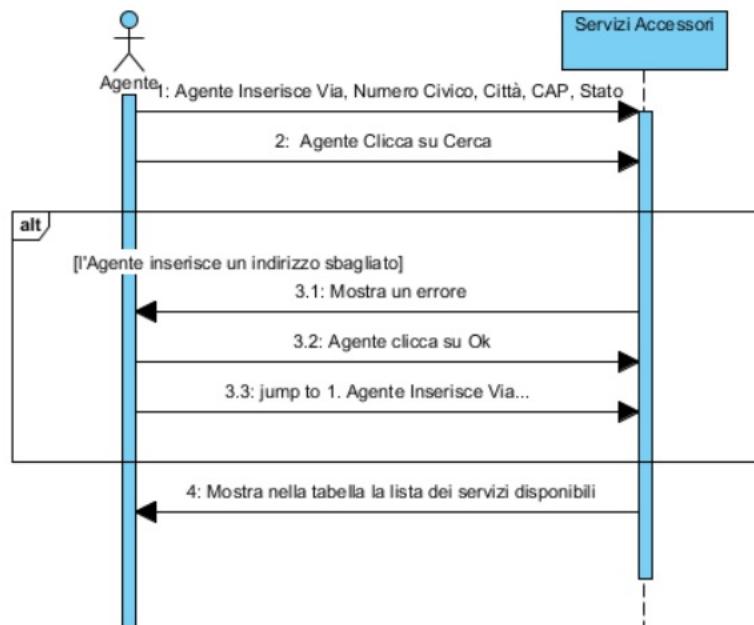
CreaPianificazione

1. **SYSTEM** Mostra la lista dei servizi scelti e la funzione disponibile
2. **Agente** Clicca su **Immetti Fasce Orarie**
3. **for each** servizio presente nella tabella
 - 3.1. **Agente** Riempe i campi Da: e A: inserendo le fasce orarie**end for each**
4. **Agente** Clicca su **Crea Pianificazione**
5. **SYSTEM** Conferma che l'operazione è andata a buon fine
6. **Agente** Clicca su **OK**



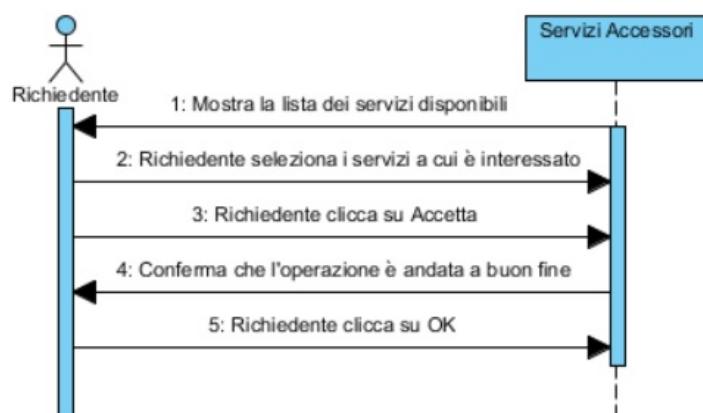
RichiediListaServiziAccessori

1. **Agente** Inserisce Via, Numero Civico, Città, CAP, Stato
2. **Agente** Clicca su **Cerca**
3. **if** l'Agente inserisce un indirizzo sbagliato
 - 3.1. **SYSTEM** Mostra un errore
 - 3.2. **Agente** clicca su **Ok**
 - 3.3. **jump to** 1. Agente Inserisce Via...**end if**
4. **SYSTEM** Mostra nella tabella la lista dei servizi disponibili



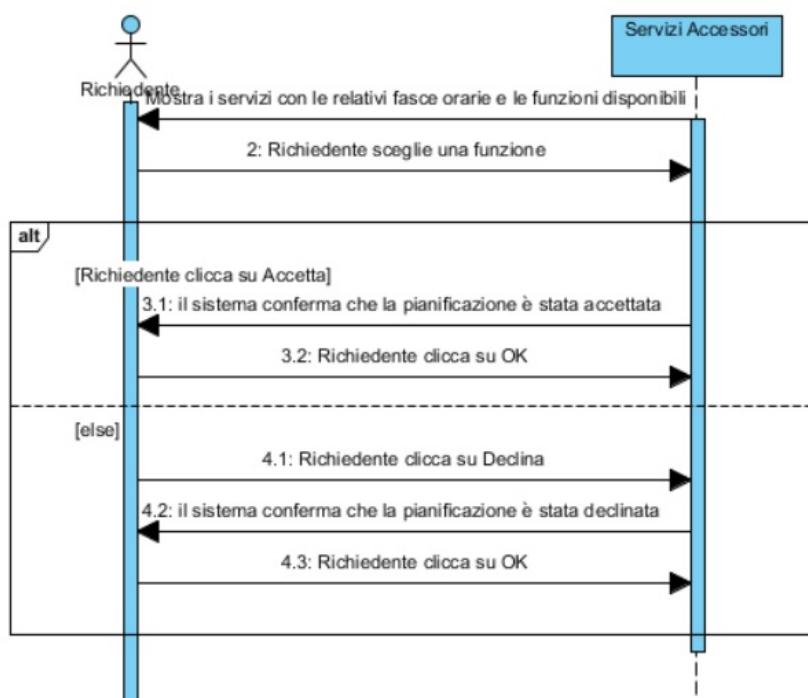
SelezionaServiziAccessori

1. **SYSTEM** Mostra la lista dei servizi disponibili
2. **Richiedente** seleziona i servizi a cui è interessato
3. **Richiedente** clicca su **Accetta**
4. **SYSTEM** Conferma che l'operazione è andata a buon fine
5. **Richiedente** clicca su **OK**



RispondiPianificazione

1. **SYSTEM** Mostra i servizi con le relativi fasce orarie e le funzioni disponibili
 2. **Richiedente** sceglie una funzione
 3. **if** **Richiedente** clicca su **Accetta**
 - 3.1. **SYSTEM** il sistema conferma che la pianificazione è stata accettata
 - 3.2. **Richiedente** clicca su **OK**
 4. **else**
 - 4.1. **Richiedente** clicca su **Declina**
 - 4.2. **SYSTEM** il sistema conferma che la pianificazione è stata declinata
 - 4.3. **Richiedente** clicca su **OK**
- end if**



6.2.5 SharingEverywhere

Caso d'uso

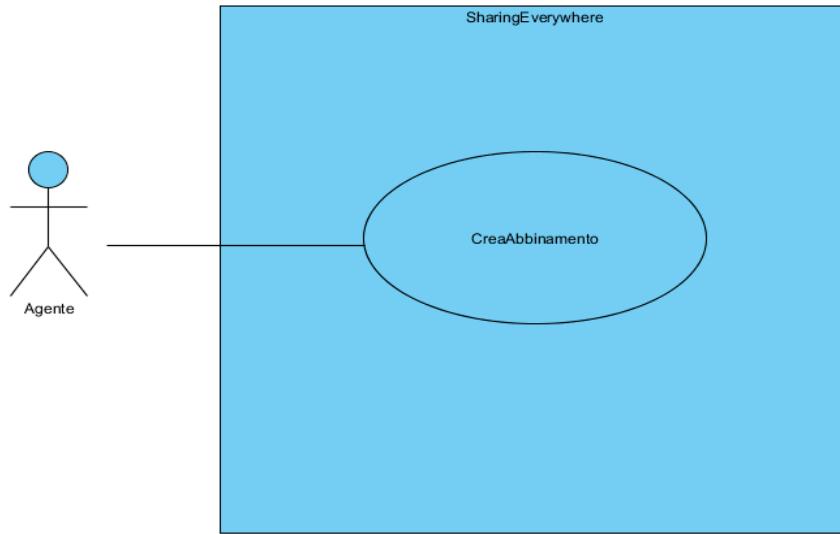


Figura 6.32: caso d'uso di SharingEverywhere

Caso d'uso dettagliato e diagrammi di sequenza

1. L' Agente usa VisualizzaRichiesta
2. **SYSTEM** mostra le richieste non gestite
3. L' Agente usa GestisciRichiesta
4. **SYSTEM** sposta la richiesta nella lista di richieste gestite
5. L' Agente usa VisualizzaRichiesta
6. **SYSTEM** mostra la richiesta
7. L' Agente usa RicercaCase
8. **SYSTEM** mostra una lista d'immobili ordinati in base al numero di campi che l'offerta soddisfa
9. **if** non ci sono offerte
 - 9.1. L' Agente rimuove alcune limitazioni
 - 9.2. **jump to** 7. L'Agente usa Ricerca...**end if**
10. **while** lista d'immobili contiene almeno un'offerta
 - 10.1. **for each** offerta
 - 10.1.1. L' Agente usa OttieniListaAttrazioni
 - 10.1.2. **SYSTEM** restituisce la lista d'attrazioni con le rispettive classi d'appartenenza, specificando la distanza di ogni attrazione
 - 10.1.3. L' Agente conta le diverse classi d'attrazioni nelle vicinanze
 - 10.1.4. L' Agente assegna un punteggio basato sul numero di classi trovate**end for each**
 - 10.2. L' Agente ordina le offerte sulla base del punteggio
 - 10.3. L' Agente preleva la prima offerta nella lista
 - 10.4. L' Agente usa SuggerisciListaPropostaAttivita
 - 10.5. **SYSTEM** crea una lista di proposte attività
 - 10.6. L' Agente usa RichiediListaServiziAccessori
 - 10.7. **SYSTEM** crea una lista di servizi accessori
 - 10.8. L' Agente usa CreaProposta
 - 10.9. **SYSTEM** pubblica la proposta di condivisione

- 10.10. **loop**
- 10.10.1. L' **Agente** estrae un'altra offerta dalla lista
 - 10.10.2. L' **Agente** usa **RichiediListaServiziAccessori**
 - 10.10.3. **SYSTEM** crea una lista di servizi accessori
 - 10.10.4. L' **Agente** usa **ModificaProposta**
 - 10.10.5. **SYSTEM** pubblica la proposta di condivisione
- until** il richiedente non accetta la proposta
- end while**
11. **if** il richiedente non ha accettato
- 11.1. L' **Agente** rimuove alcune limitazioni della richiesta
 - 11.2. **jump to** 7. L'Agente usa Ricerca...
- end if**
12. **Agente** usa **ComunicaScambio**
13. **SYSTEM** aggiorna lo stato dell'offerta, aggiungendo la proposta inserita
14. L' **Agente** usa **VisualizzaListaPropostaAttivita**
15. **SYSTEM** mostra la lista proposta attività
16. **while** la proposta di attività non è stata accettata dal richiedente
- 16.1. **loop**
 - 16.1.1. L' **Agente** usa **AggiungiAlternative**
 - 16.1.2. **SYSTEM** aggiunge le alternative alla lista proposta attività
 - 16.1.3. **for each** alternativa nella lista proposta delle attività
 - 16.1.3.1. **if** l'alternativa appartiene ad una classe non presente nella richiesta
 - 16.1.3.1.1. L' **Agente** usa **RimuoviAlternative**
 - 16.1.3.1.2. **SYSTEM** rimuove le alternative dalla lista proposta attività

end if

end for each

until c'è almeno una fascia oraria vuota

end while

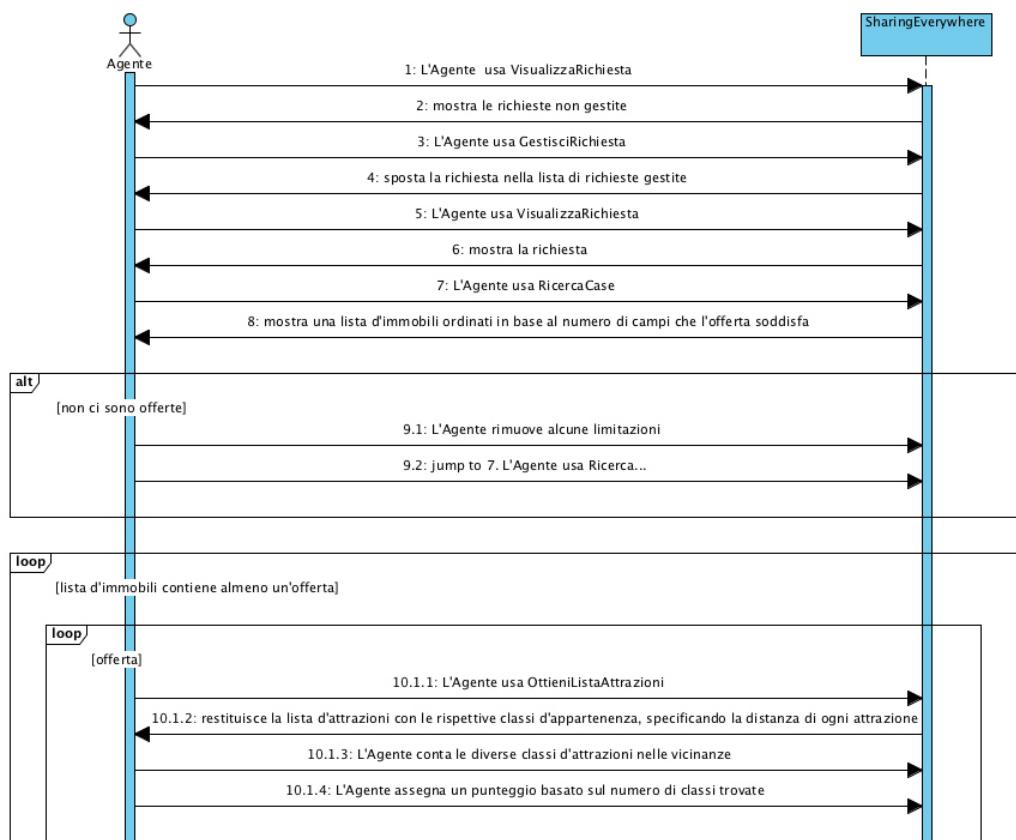
17. L' **Agente** usa **VisualizzaListaPropostaAttivita**

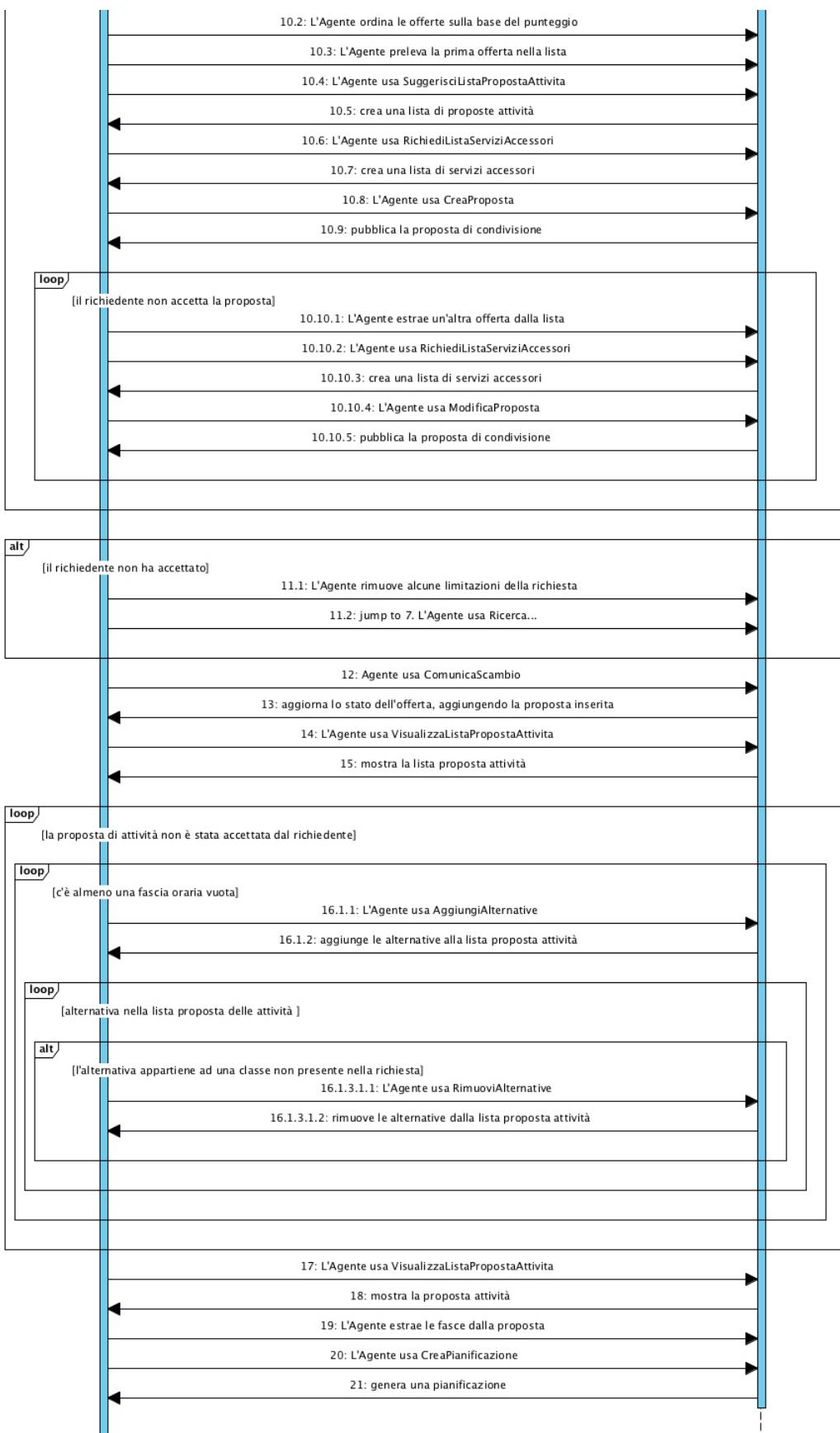
18. **SYSTEM** mostra la proposta attività

19. L' **Agente** estrae le fasce dalla proposta

20. L' **Agente** usa **CreaPianificazione**

21. **SYSTEM** genera una pianificazione





6.3 Classi

Una classe viene definita come "Il descrittore per un set di oggetti che condividono gli stessi attributi, operazioni, metodi, relazioni e comportamenti". Questo può essere facilmente riassunto dicendo che una classe è un descrittore per un insieme di oggetti che hanno caratteristiche simili. Le classi di analisi sono classi che rappresentano un'astrazione nel problema di dominio e che presentano un insieme di attributi ad "alto livello". Esse indicano gli attributi che probabilmente avranno le classi in fase di design. Si potrebbe quindi dire che le classi di analisi definiscono attributi che sono candidati per le classi di design.

6.3.1 Offerta casa

Diagramma della classe



Figura 6.33: Funzionalità per l'agente esposte da *Offerta casa*

Diagramma delle strutture dati complesse

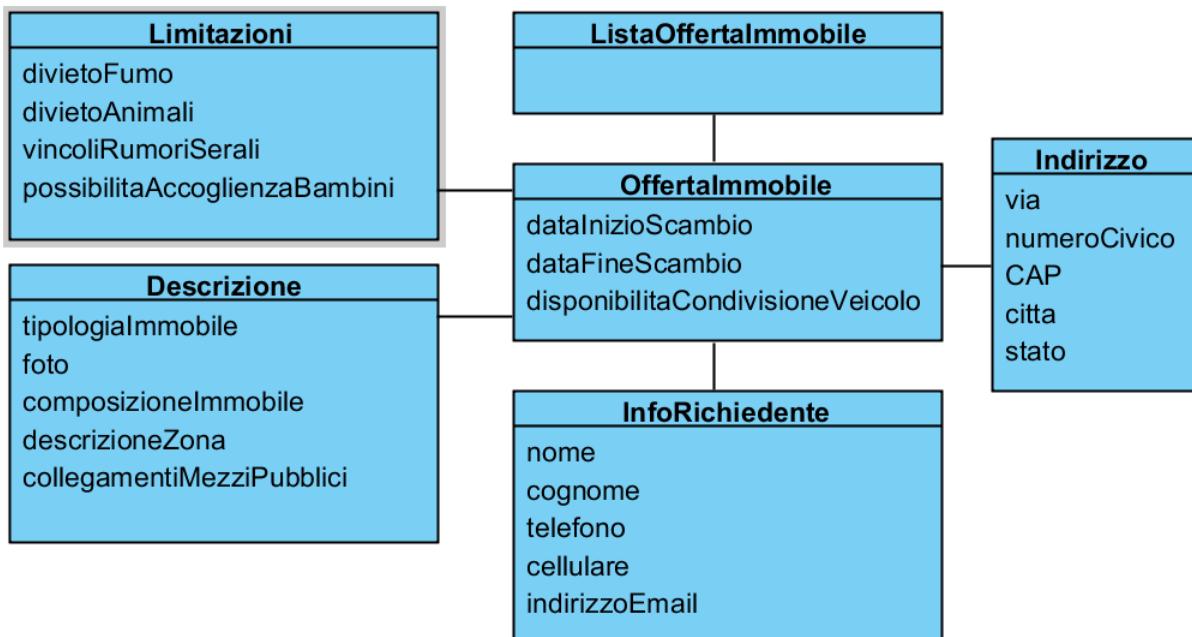


Figura 6.34: Strutture dati complesse in *Offerta casa*

6.3.2 Richiesta casa

Diagramma della classe



Figura 6.33: Funzionalità per l'agente esposte da *Richiesta casa*

Diagramma delle strutture dati complesse

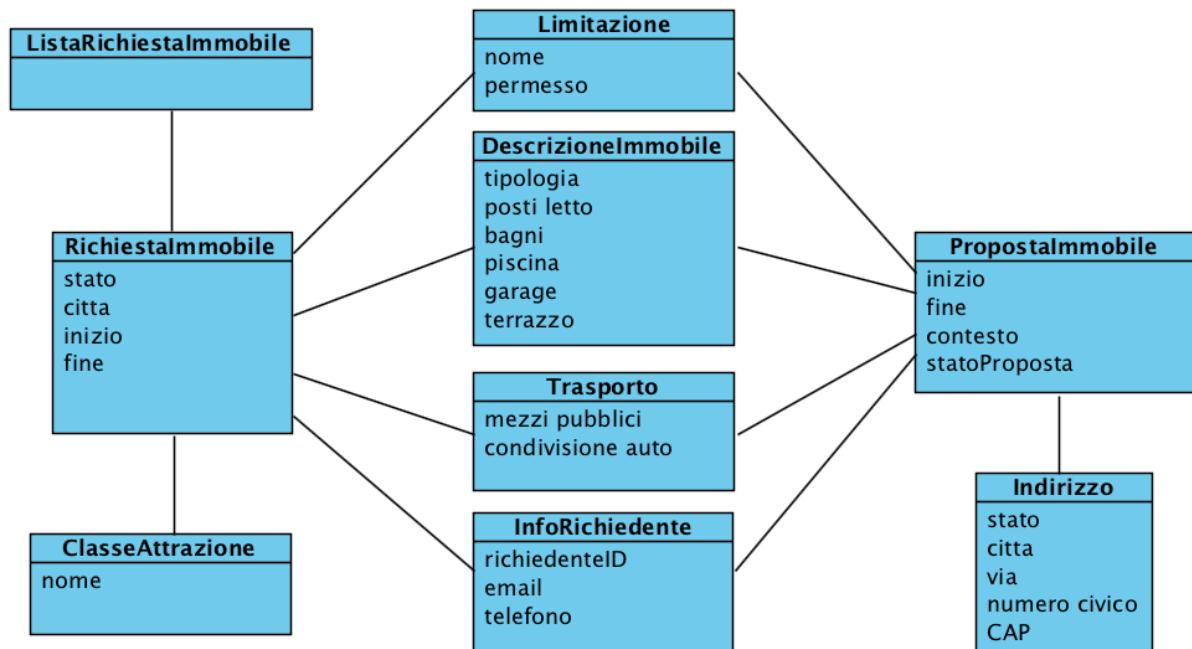


Figura 6.34: Strutture dati complesse in *Richiesta casa*

6.3.3 Attrazioni

Diagramma della classe



Figura 6.33: Funzionalità per l'agente esposte da *Attrazioni*

Diagramma delle strutture dati complesse

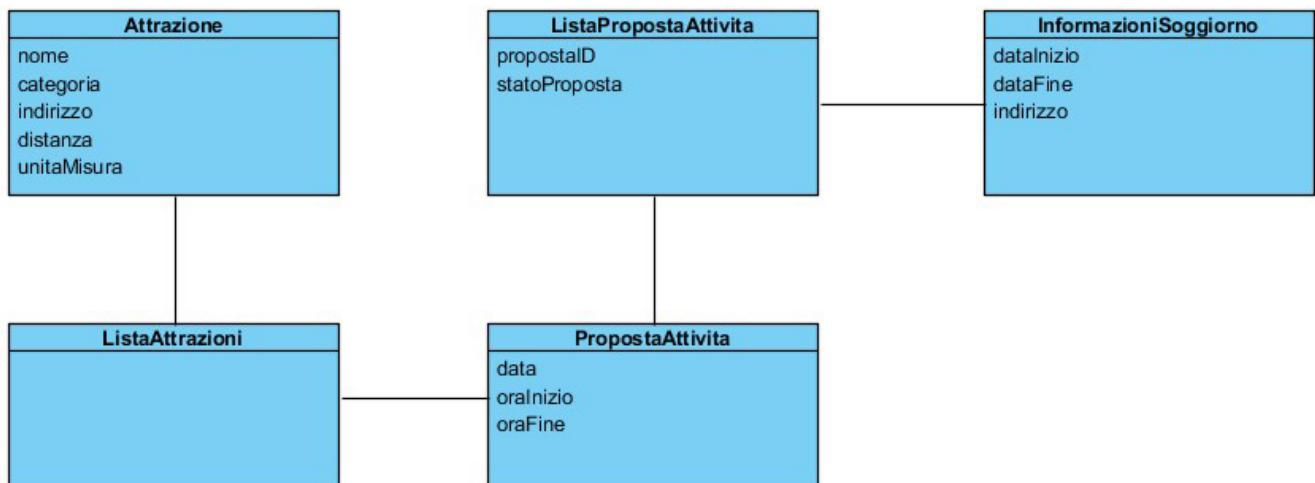


Figura 6.34: Strutture dati complesse in *Attrazioni*

6.3.4 Servizi accessori

Diagramma della classe



Figura 6.35: Funzionalità per l'agente esposte da *Servizi accessori*

Diagramma delle strutture dati complesse

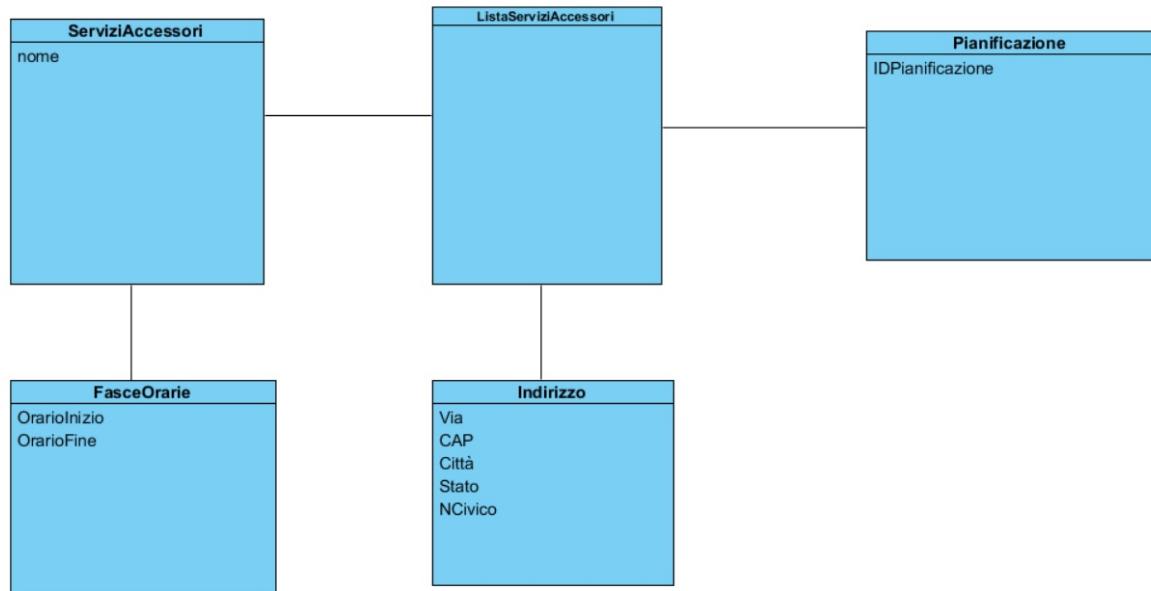


Figura 6.36: Strutture dati complesse in *Servizi accessori*

Capitolo 7

Design

7.1 Organizzazione della fase

Le attività riguardanti la fase di design sono state organizzate e pianificate tenendo conto del diagramma in Figura 7.1, il quale indica i documenti che devono essere elaborati:

J Diagramma SoaML delle interfacce di ogni applicativo.

K Diagramma di sequenza SoaML per il protocollo.

L Diagramma delle classi interface-like derivato da J.

M Diagramma delle classi/oggetti per i parametri complessi di ogni applicativo.

7.1.1 Tipi di attività

Attività individuali

Attività J richiede che ogni membro elabori le interfacce usate ed esposte dal proprio modulo. Non ha bisogno di alcuna informazione di come gli altri moduli siano sviluppati.

Attività L richiede il completamento dell'attività J per essere elaborata. Quindi L segue logicamente J e non necessita di alcuna sincronizzazione tra i membri.

Attività M segue l'attività L: produce un diagramma strutturale di classi/oggetti di un applicativo. È necessario specificare il tipo di classe usato nelle funzioni delle interfacce.

Attività cooperative

Attività K Ogni modulo deve fornire alcune funzionalità con le sue interfacce in modo da creare la logica di coordinamento. Richiede la cooperazione da parte di tutti i membri.

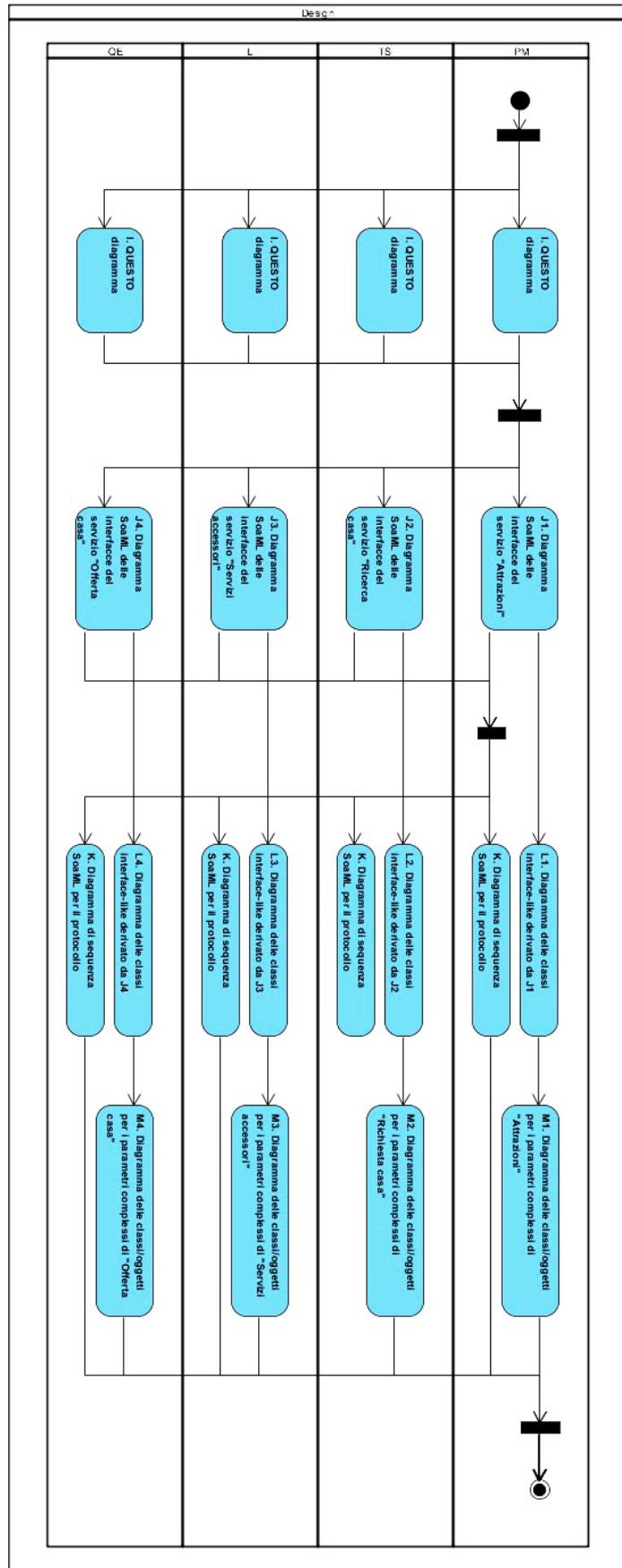


Figura 7.1: Diagramma di attività della fase di design

7.2 Diagrammi SoaML delle interfacce dei servizi

Il diagramma delle interfacce dei servizi è uno dei più importanti tipi di diagrammi SoaML. In SoaML, un servizio può essere specificato usando tre approcci: interfaccia semplice, interfaccia di servizio e interfaccia di contratto. Per noi l'interesse è andato alle interfacce di servizio poiché queste consentono comunicazione bidirezionale tra un produttore e un consumatore per portare a termine un servizio. Questo tipo di diagramma consente di modellare le specifiche del servizio.

7.2.1 Offerta casa

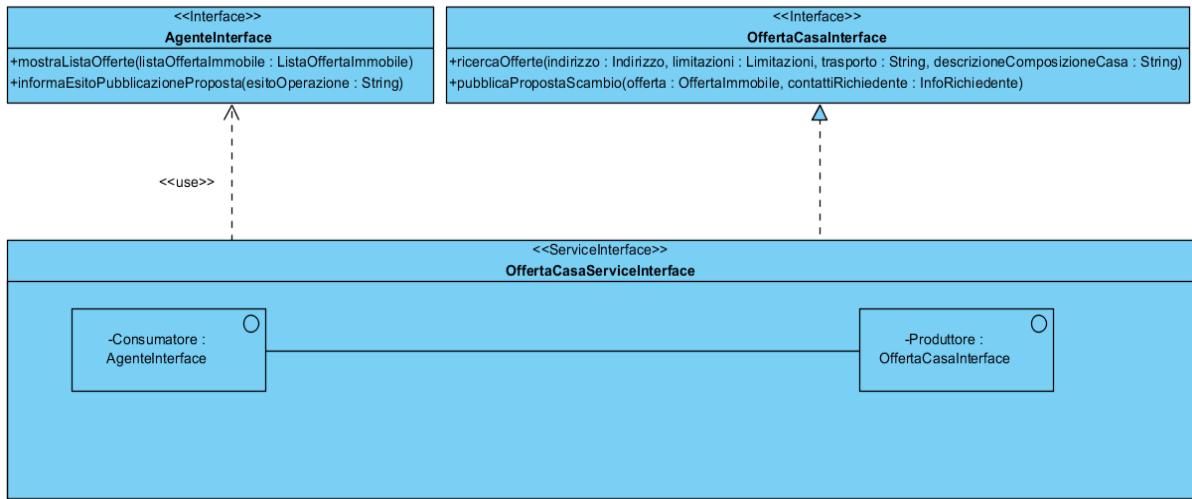


Figura 7.2: Diagramma SoaML dell'interfaccia di *Offerta casa*

7.2.2 Richiesta casa

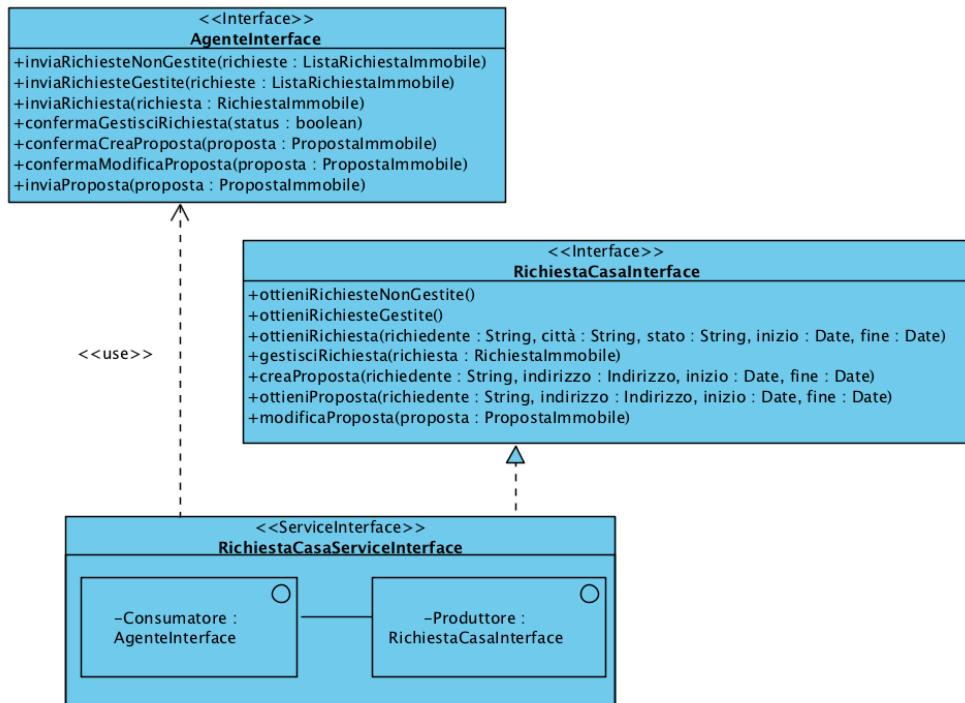


Figura 7.3: Diagramma SoaML dell'interfaccia di *Richiesta casa*

7.2.3 Attrazioni

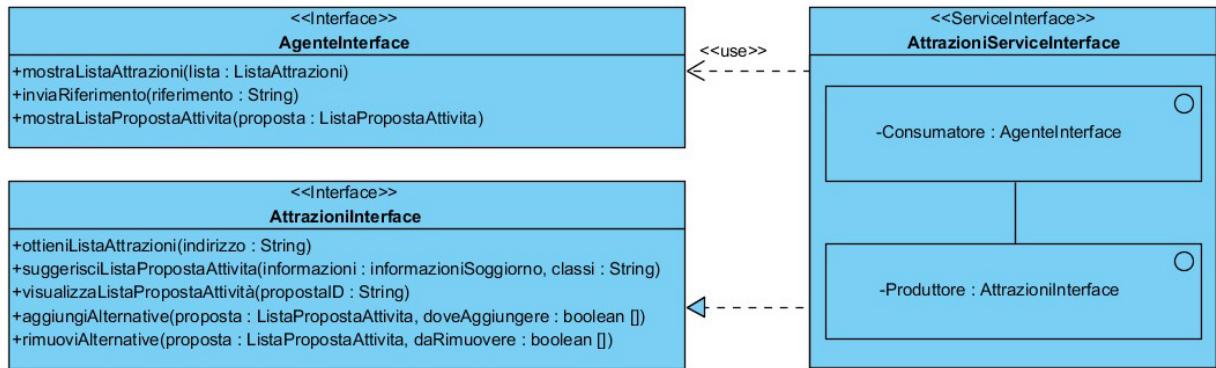


Figura 7.4: Diagramma SoaML dell’interfaccia di *Attrazioni*

7.2.4 Servizi accessori

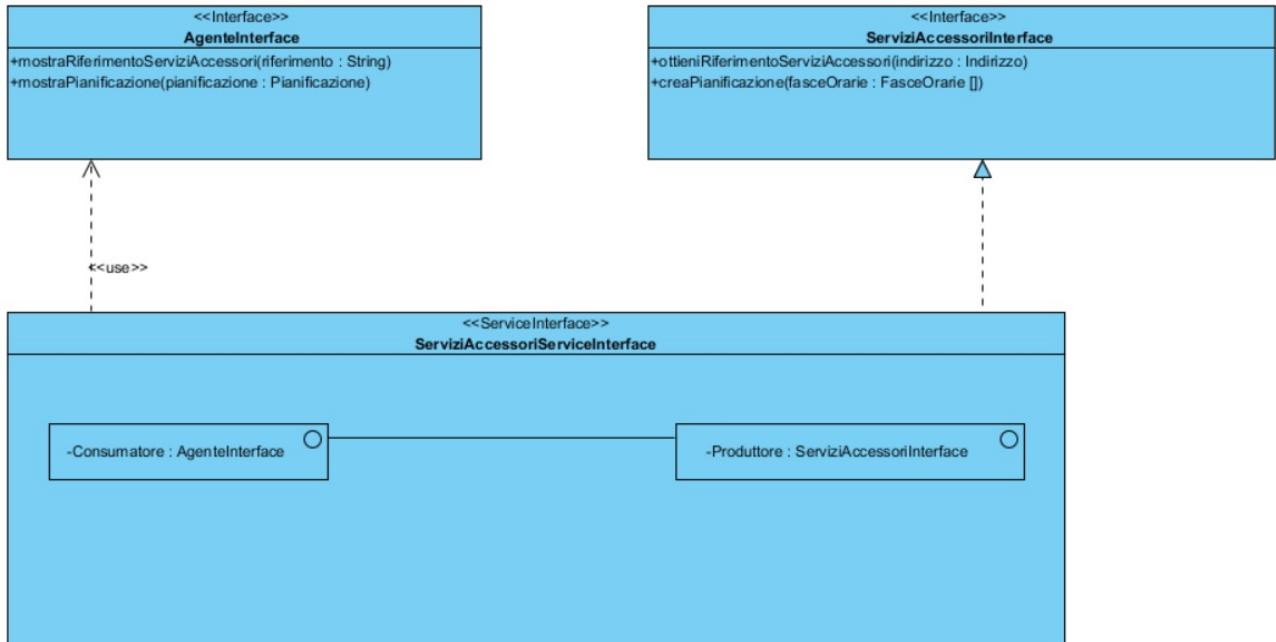
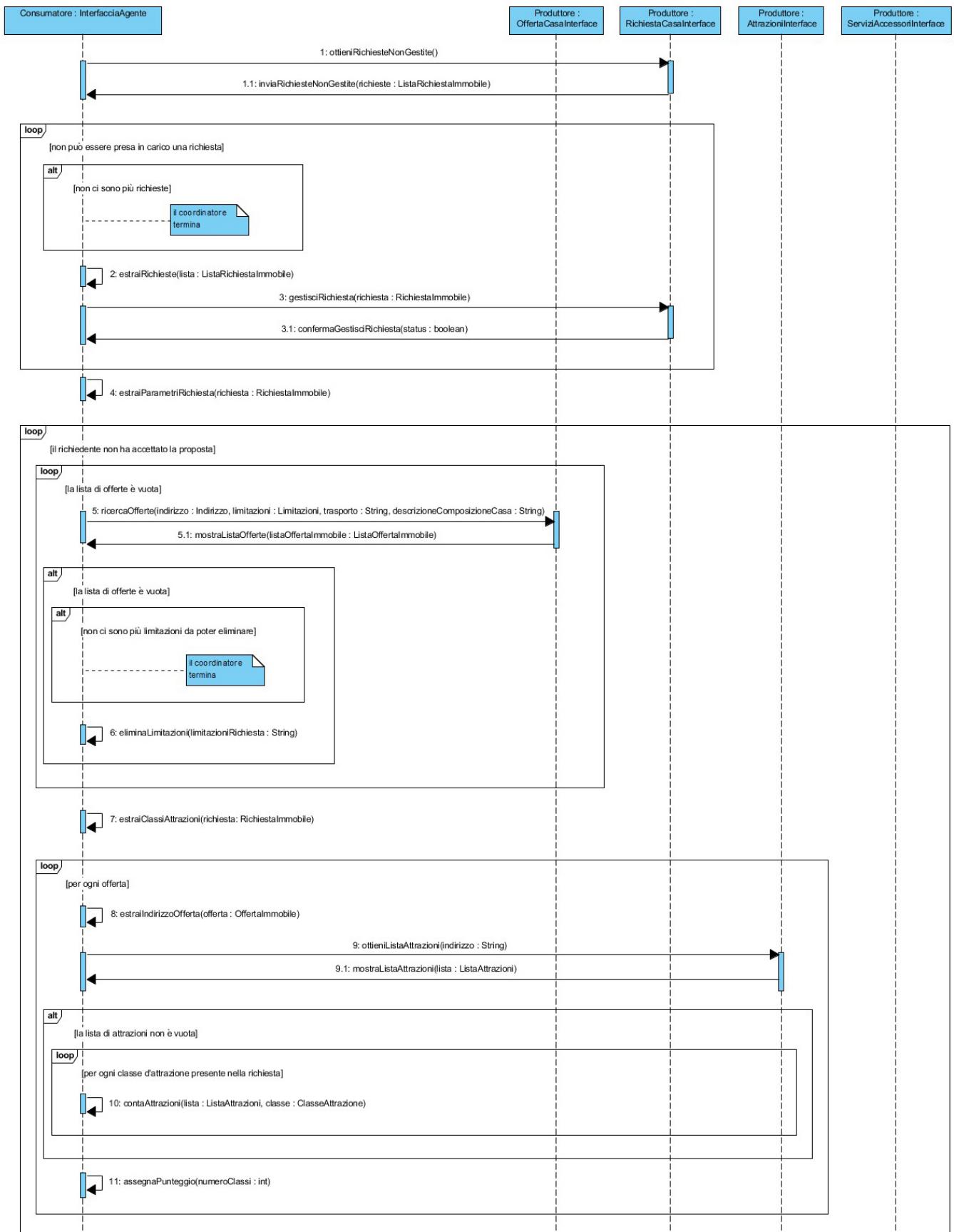
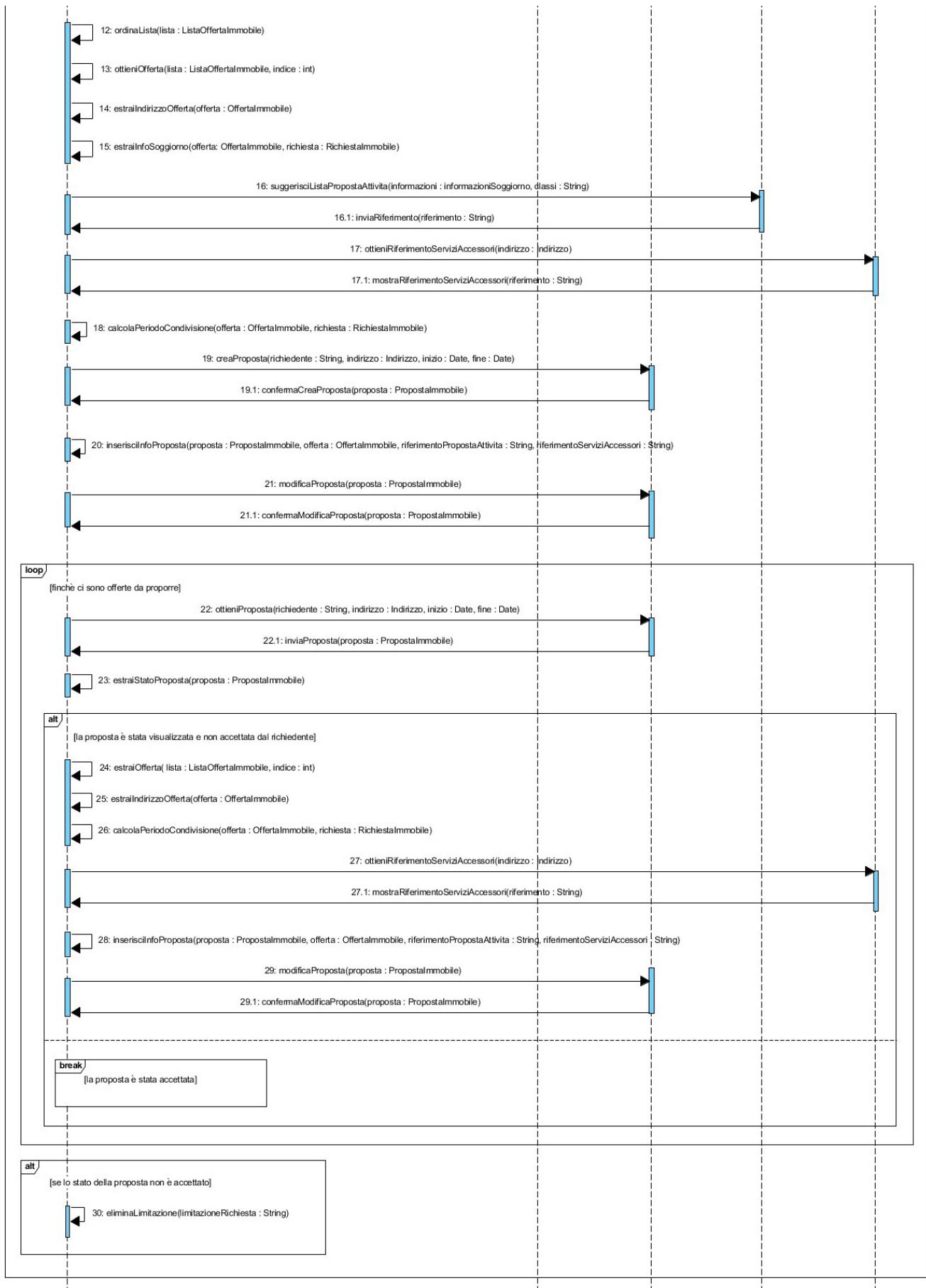


Figura 7.5: Diagramma SoaML dell’interfaccia di *Servizi accessori*

7.3 Diagramma di sequenza SoaML per il protocollo





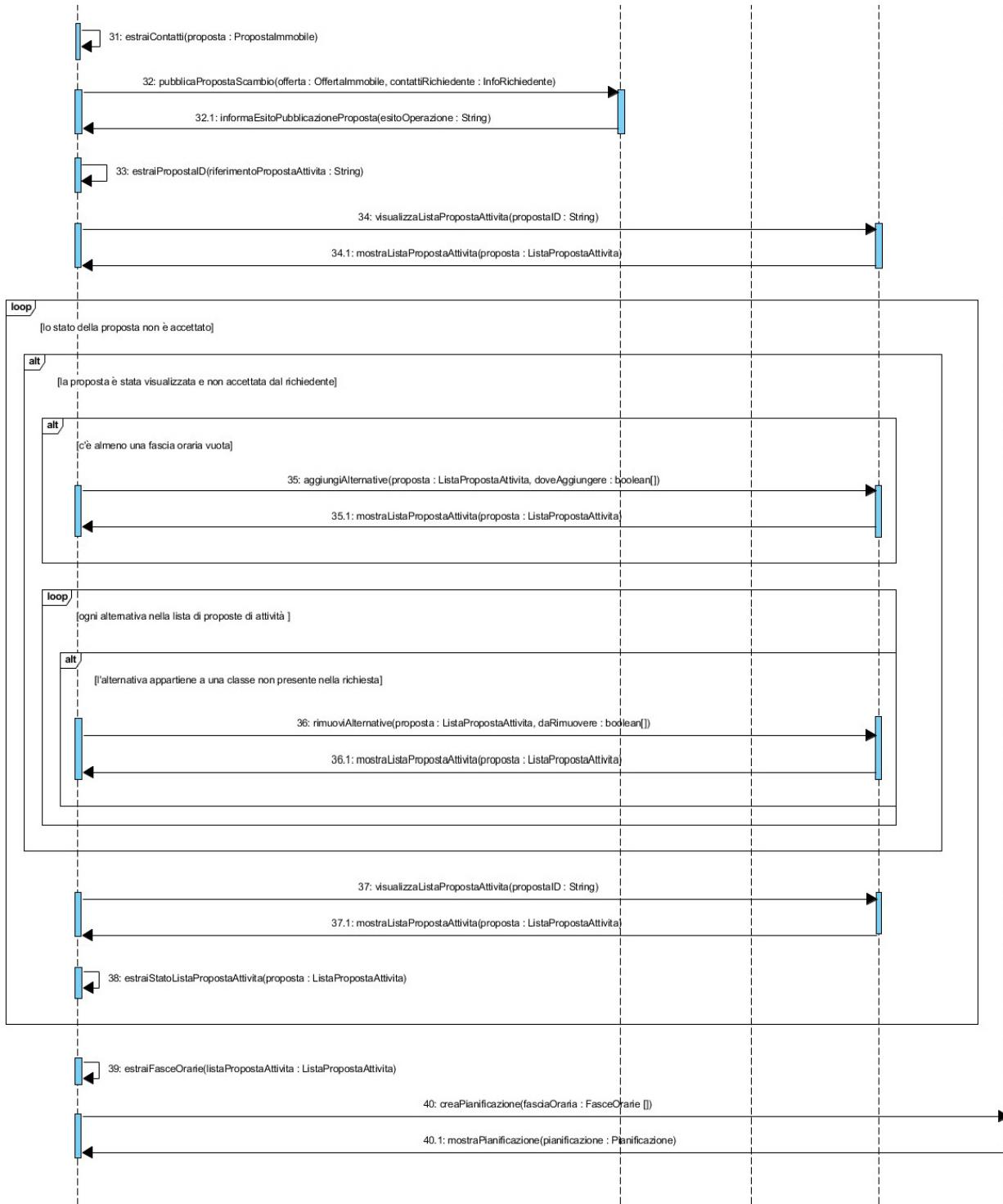


Figura 7.6: Diagramma di sequenza SoaML della logica

7.4 Diagrammi UML interface-like delle classi

7.4.1 Offerta casa

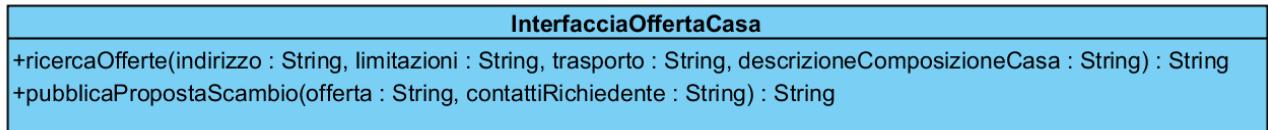


Figura 7.7: diagramma UML delle classi interface-like di *Offerta casa*

7.4.2 Richiesta casa



Figura 7.8: diagramma UML delle classi interface-like di *Richiesta casa*

7.4.3 Attrazioni



Figura 7.9: diagramma UML delle classi interface-like di *Attrazioni*

7.4.4 Servizi accessori



Figura 7.10: diagramma UML delle classi interface-like di *Servizi accessori*

7.5 Diagrammi strutturali delle classi e degli oggetti

7.5.1 Offerta casa

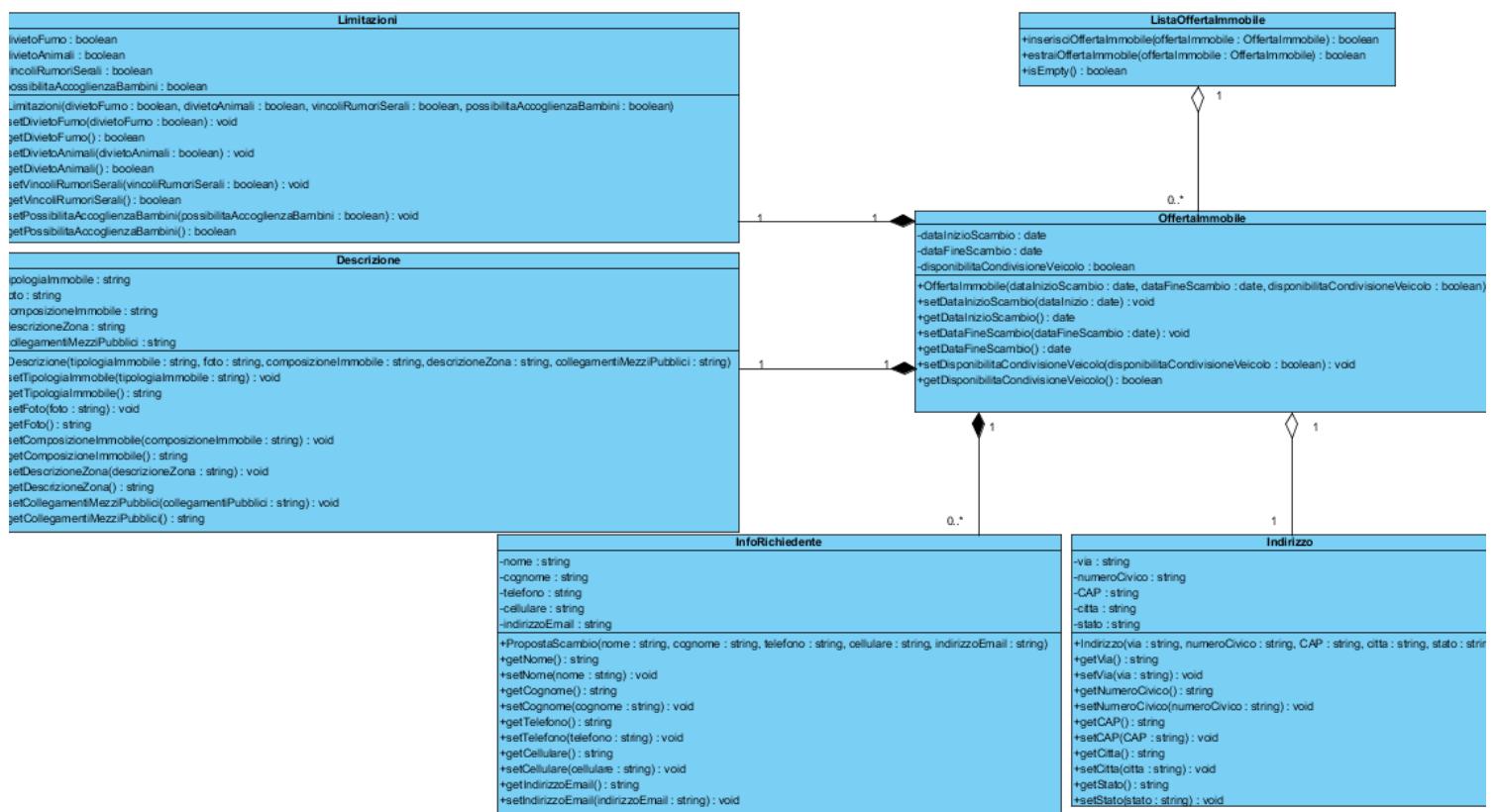


Figura 7.11: Diagramma UML delle classi del modulo *Offerta casa*

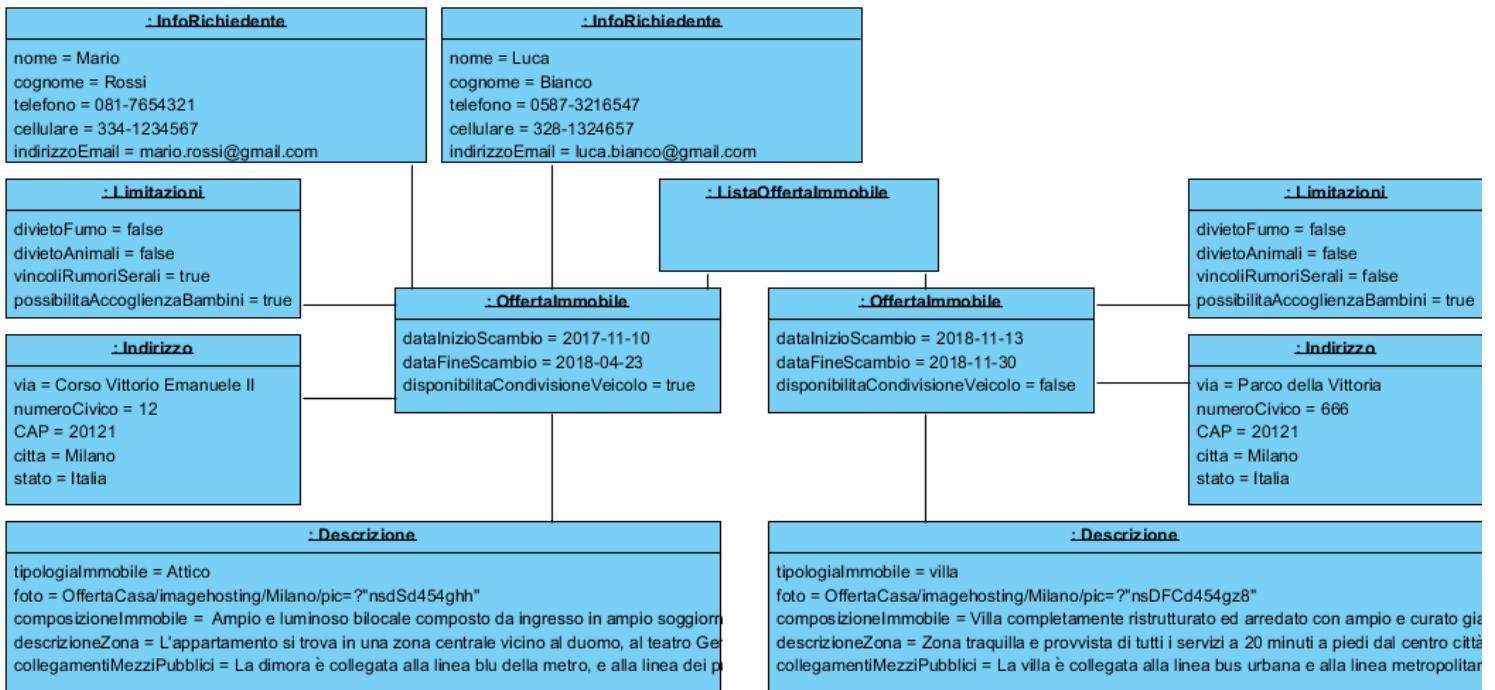


Figura 7.12: Diagramma UML degli oggetti del modulo *Offerta casa*

7.5.2 Richiesta casa

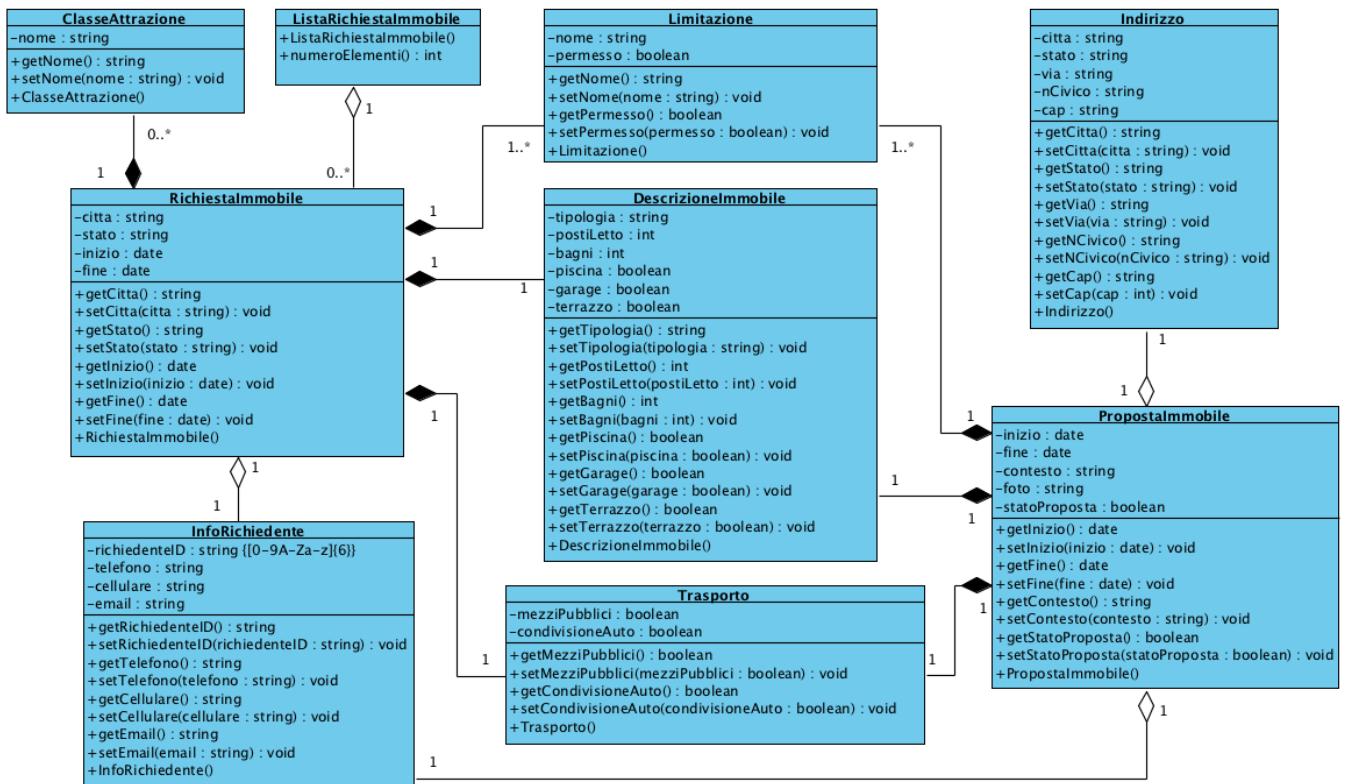
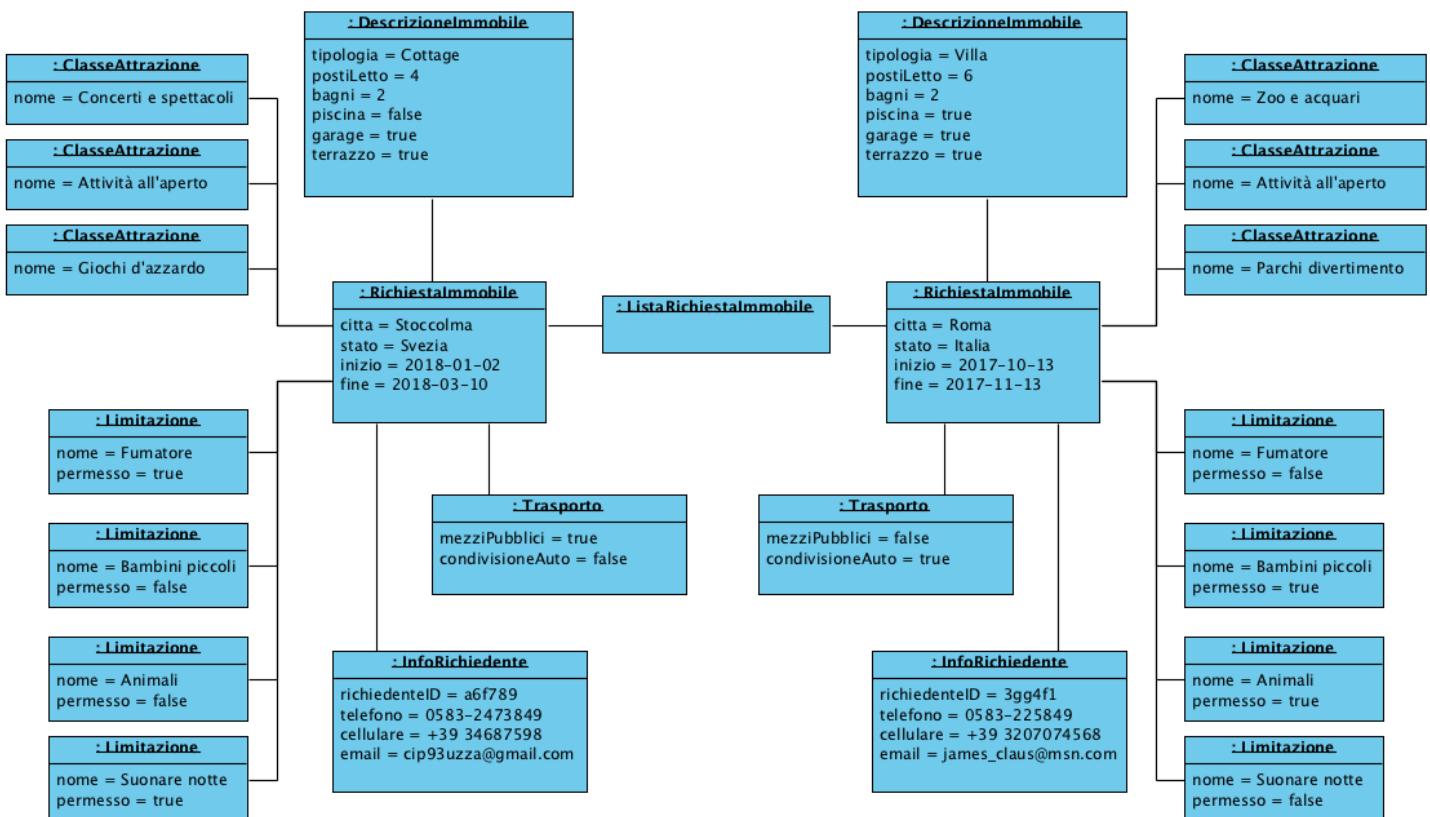


Figura 7.13: Diagramma UML delle classi del modulo *Richiesta casa*

Lo stato in PropostaImmobile è stato modellato come un intero perché la proposta può trovarsi in 3 stati infatti se il richiedente:

- visualizza la proposta ma non prende una decisione (0), l'agente è costretto ad aspettare;
- non accetta la proposta (1), l'agente può ricercare una nuova offerta con la relativa lista di servizi accessori e modificare la proposta con i nuovi dati;
- accetta la proposta (2), l'agente informa tramite l'applicativo che il proprietario dell'offerta verrà contattato.



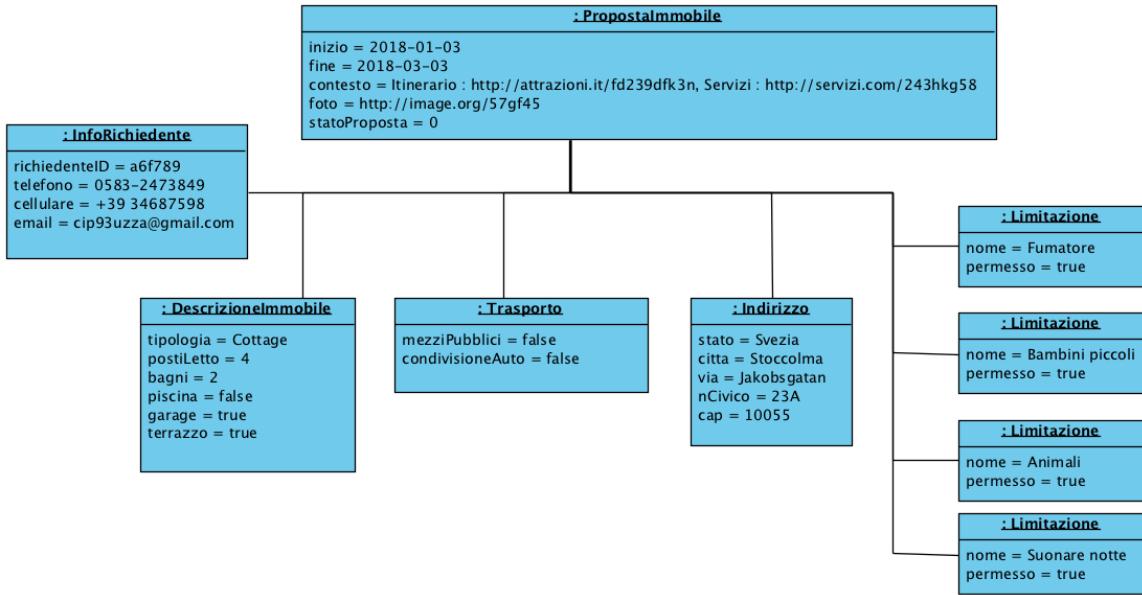


Figura 7.14: Diagramma UML degli oggetti del modulo *Richiesta casa*

7.5.3 Attrazioni

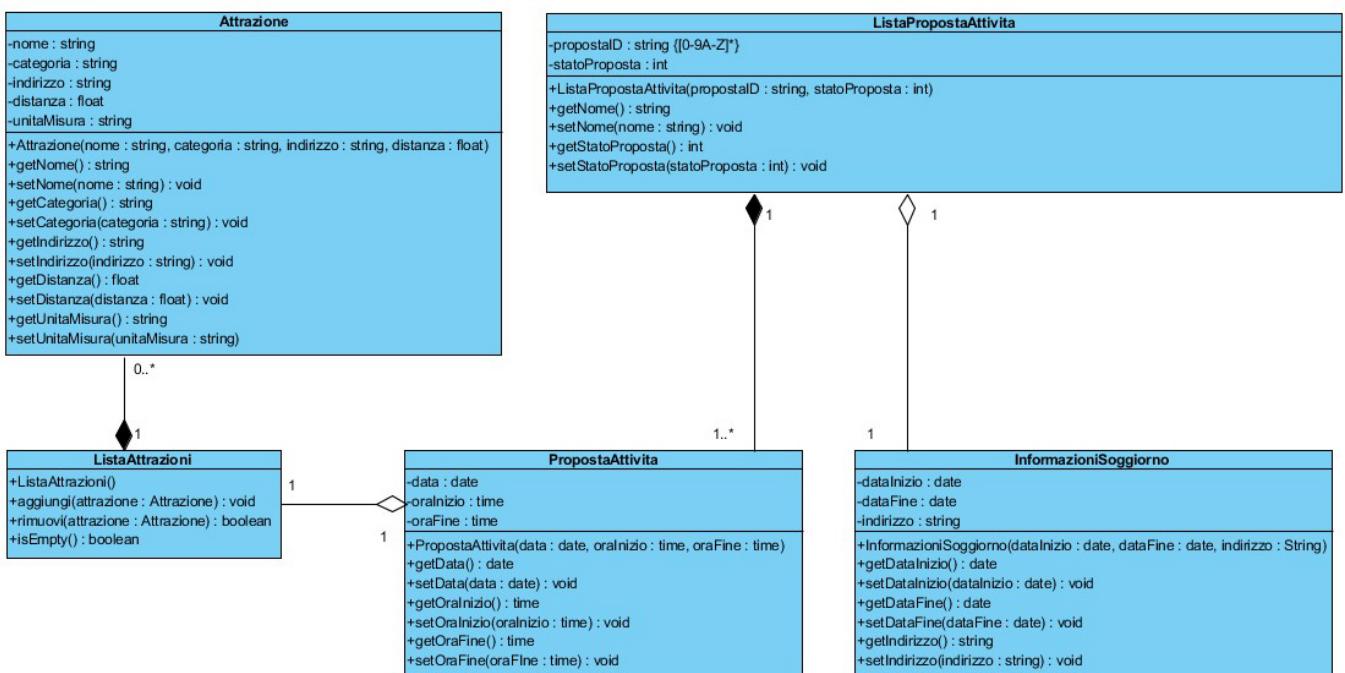


Figura 7.15: Diagramma UML delle classi del modulo *Attrazioni*

Lo stato della proposta in ListaPropostaAttivita è stato modellato con un intero perché la proposta può trovarsi in 3 stati:

- non è ancora stata visualizzata dal richiedente (0), pertanto l'agente deve solo aspettare;
- è stata visualizzata dal richiedente ma non accettata (1), pertanto l'agente può decidere se vuole aggiungere o rimuovere alcune delle alternative per vedere se il richiedente gradisce le modifiche;
- è stata accettata dal richiedente (2), pertanto l'agente non deve più interagire con la proposta.

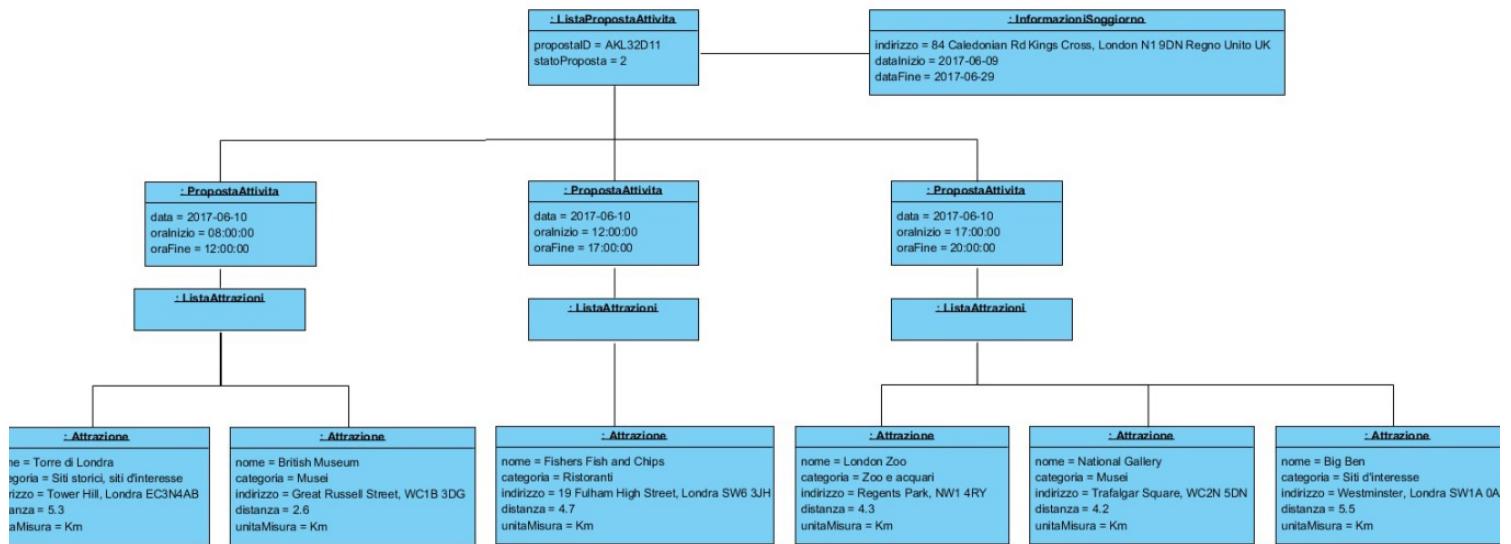
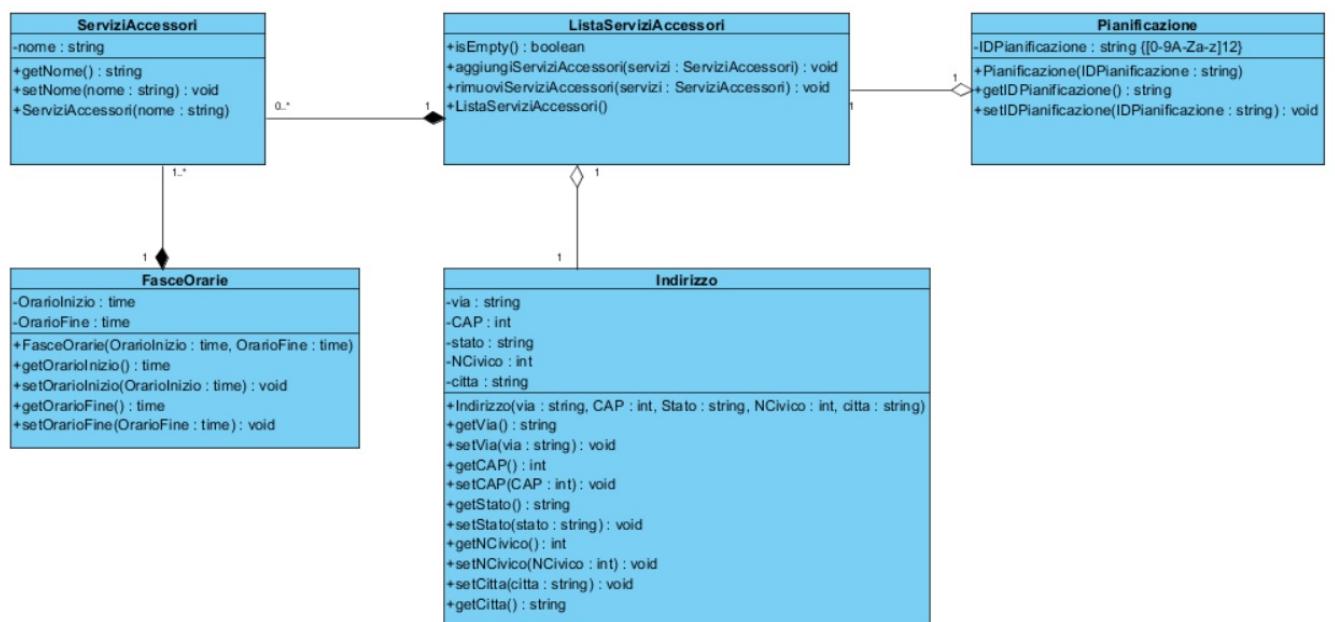


Figura 7.16: Diagramma UML degli oggetti del modulo *Attrazioni*

7.5.4 Servizi accessori



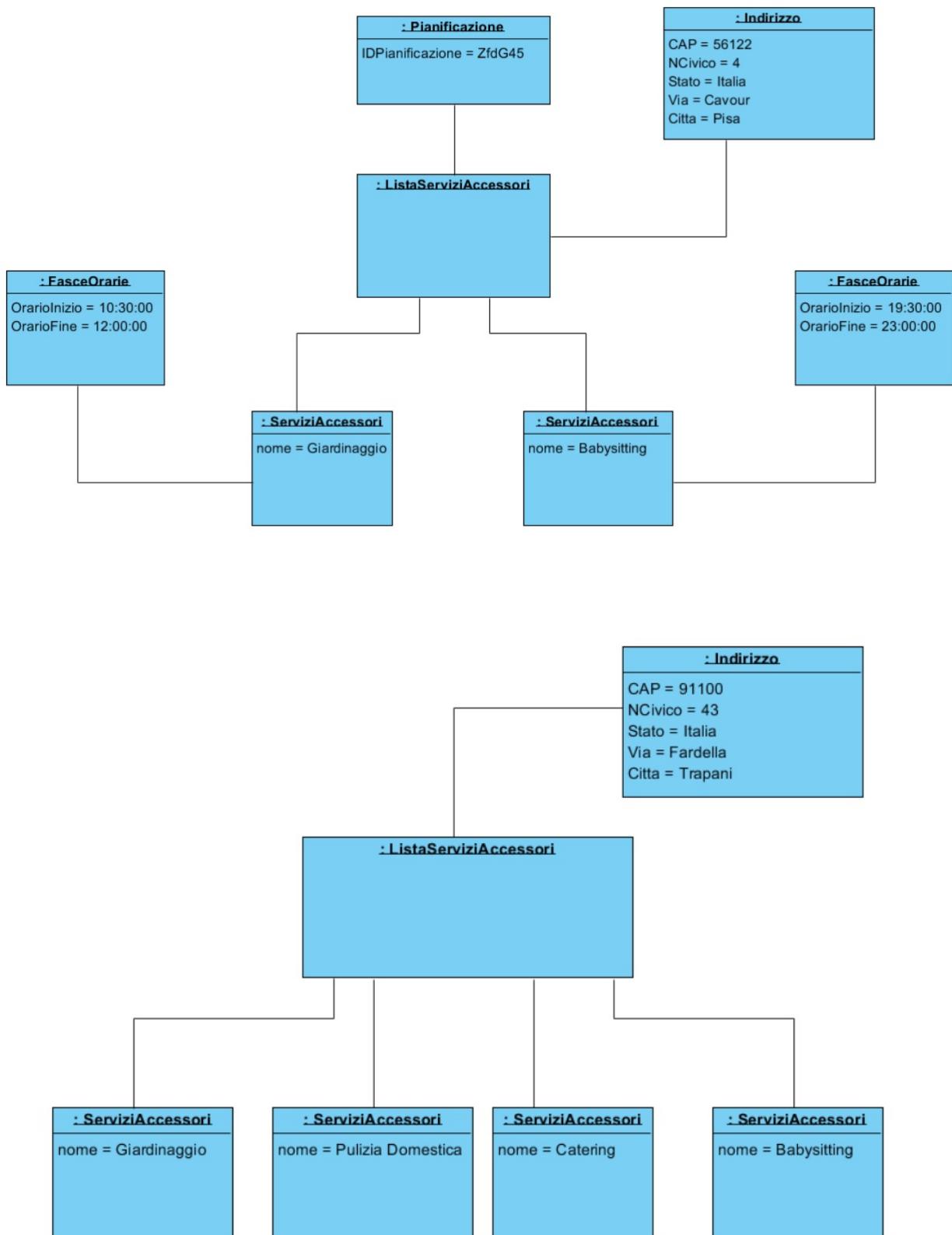


Figura 7.18: Diagramma UML degli oggetti del modulo *Servizi accessori*

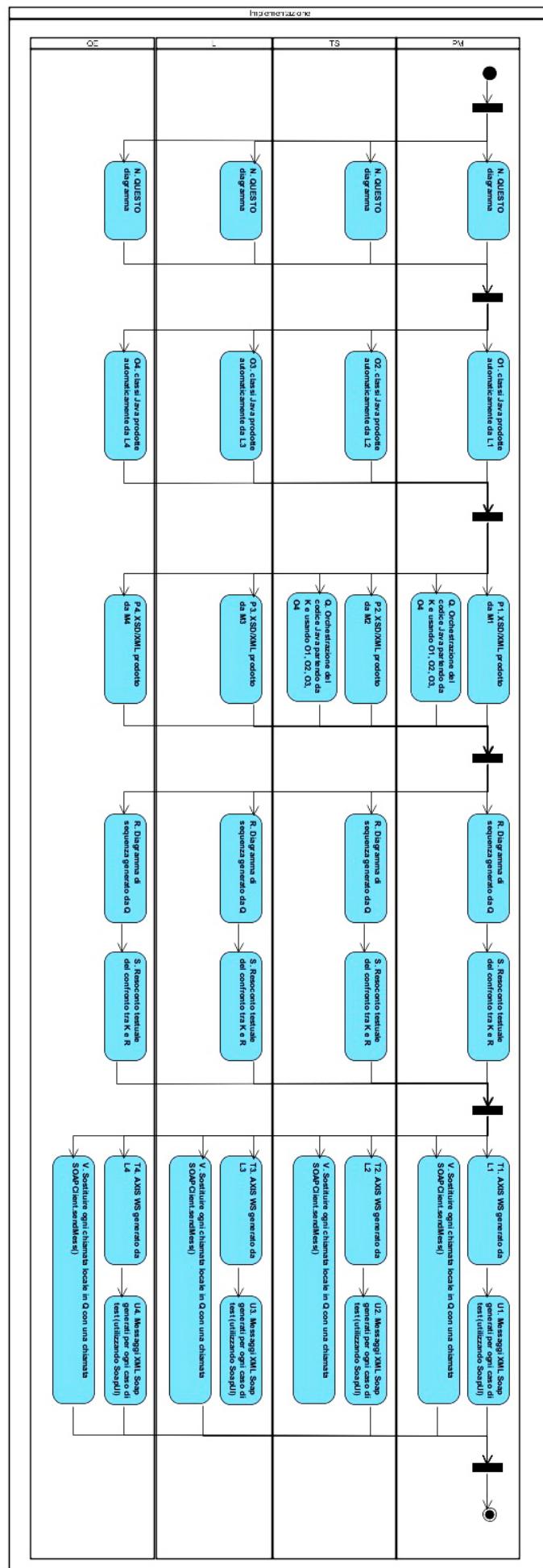
Capitolo 8

Implementazione

8.1 Organizzazione della fase

Le attività riguardanti la fase d'implementazione sono state organizzate e pianificate tenendo conto del diagramma in Figura 8.1, il quale indica i documenti che devono essere elaborati:

- O** classi Java prodotte automaticamente da L.
- P** XSD/XML prodotti da M.
- Q** Orchestrazione del codice Java partendo da K e usando O.
- R** Diagramma di sequenza generato da Q.
- S** Resoconto testuale del confronto tra K e R.
- T** AXIS WS generato dai vari L.
- U** Messaggi XML Soap generati per ogni caso di test.
- V** Sostituire ogni chiamata locale in Q con una chiamata SOAP.



8.1.1 Tipi di attività

Attività individuali

Attività O richiede che ogni membro produca le classi Java usate nel suo applicativo. Non ha bisogno di alcuno scambio di informazioni per sapere come sono stati sviluppati gli altri moduli.

Attività P richiede il completamento dell'attività O per essere elaborata. Quindi P segue logicamente O, dato che i file XSD e XML richiedono la conoscenza dei tipi usati dalle interfacce dei moduli.

Attività T considerata come attività individuale da eseguire in parallelo dato che nessun membro condivide le informazioni riguardo il proprio modulo per produrre il deliverable.

Attività U considerata come attività individuale da eseguire in parallelo dato che nessun membro condivide le informazioni riguardo il proprio modulo per produrre il deliverable.

Attività cooperative

Attività Q, svolta solo dal PM e dal TS poiché più indietro degli altri con le ore, viene eseguita in parallelo all'attività P.

Attività R richiede che tutti i membri diano un parere sul lavoro svolto da PM e TS e generino un diagramma consistente con tutti i quattro moduli prodotti.

Attività S richiede una sincronizzazione alla fine e la verifica e la validità da parte di tutti i membri.

Attività V richiede il contributo di ogni membro del gruppo per essere completata, perché ogni membro sa quali sono i messaggi Soap che il proprio modulo richiede.

8.2 Classi Java per le interfacce

8.2.1 Interfaccia Offerta casa

```
public class InterfacciaOffertaCasa {  
  
    public String ricercaOfferte(String aIndirizzo, String aLimitazioni, String aTrasporto, String  
aDescrizioneComposizioneCasa) {  
        return "Lista di offerte pertinenti con i parametri di ricerca.";  
    }  
  
    public String pubblicaPropostaScambio(String aOfferta, String aContattiRichiedente) {  
        return "La proposta e' stata effettuata con successo.";  
    }  
}
```

8.2.2 Interfaccia Richiesta casa

```
import java.lang.Math;  
import java.util.Date;  
  
public class InterfacciaRichiestaCasa {  
  
    public String ottieniRichiesteNonGestite() {  
        return "Richieste non gestite";  
    }  
  
    public String ottieniRichiesteGestite() {  
        return "Richieste gestite dall'agente";  
    }  
  
    public String ottieniRichiesta(String aRichiedenteID, String aCitta, String aStato, Date alnizio, Date  
aFine) {  
        return "Richiesta";  
    }  
  
    public boolean gestisciRichiesta(String aRichiesta) {  
        boolean result = (Math.random() < 0.05)? false : true;  
        return result;  
    }  
  
    public String creaProposta(String aRichiedenteID, String aDestinazione, Date alnizio, Date aFine) {  
        return "Nuova proposta";  
    }  
  
    public String ottieniProposta(String aRichiedenteID, String aDestinazione, Date alnizio, Date aFine) {  
        return "Proposta";  
    }  
  
    public String modificaProposta(String aProposta) {  
        return "Proposta modificata";  
    }  
}
```

```
    }  
}
```

8.2.3 Interfaccia Attrazioni

```
public class InterfacciaAttrazioni {  
  
    public String ottieniListaAttrazioni(String aIndirizzo) {  
        return "Lista di attrazioni ordinata per distanza";  
    }  
  
    public String suggerisciListaPropostaAttivita(String aInformazioni, String aClassiAttrazioni) {  
        return "Riferimento alla lista proposta attivita generata";  
    }  
  
    public String visualizzaListaPropostaAttivita(String aPropostaID) {  
        return "Lista proposta attivita";  
    }  
  
    public String aggiungiAlternative(String aListaPropostaAttivita, boolean[]  
aInQualiProposteAggiungere) {  
        return "Lista proposta attivita modificata";  
    }  
  
    public String rimuoviAlternative(String aListaPropostaAttivita, boolean[] aAlternativeDaRimuovere)  
{  
    return "Lista proposta attivita modificata";  
}  
}
```

8.2.4 Interfaccia Servizi accessori

```
public class InterfacciaServiziAccessori {  
  
    public String ottieniRiferimentoServiziAccessori(String aIndirizzo) {  
        return "riferimento servizi accessori";  
    }  
  
    public String creaPianificazione(String aFasceOrarie) {  
        return "pianificazione";  
    }  
}
```

8.3 File XSD e XML

I file XSD prodotti in automatico sono stati rivisti e modificati applicando le cinque regole seguenti:

1. Gli elementi sono usati per encapsulare dati strutturati e dati auto-contenuti;
2. Gli attributi sono usati per fornire informazioni riguardo a un elemento;
3. Gli attributi sono usati quando l'informazione richiede dati di tipo semplice e
 - a. È un valore di default o quasi sempre fisso;
 - b. È un metadata strutturato o descrittivo per un elemento esistente;
 - c. La dimensione del file XML è un problema;
4. Gli elementi sono usati quando l'ordinamento è significativo (gli attributi possono apparire in qualsiasi ordine);
5. Gli elementi possono avere più di un'occorrenza, gli attributi possono avere una sola occorrenza.

Abbiamo cercato di applicare le regole inserendo gli attributi dove necessario, evitando forzature: w3schools consiglia di utilizzare gli elementi per descrivere i dati e di utilizzare gli attributi solo per fornire informazioni di contesto. In questo modo è più facile leggere e aggiornare i documenti XML.

8.3.1 Offerta casa

XSD

```
<?xml version="1.0" encoding="UTF-8"?>
<!-- OffertaCasa.xsd --&gt;
&lt;xsschema xmlns:xss="http://www.w3.org/2001/XMLSchema"&gt;

    &lt;xss:element name="ListaOffertalmmobile"&gt;
        &lt;xss:complexType&gt;
            &lt;xss:sequence&gt;
                &lt;xss:element ref="Offertalmmobile" minOccurs="0"
                           maxOccurs="unbounded"&gt;
                &lt;/xss:element&gt;
            &lt;/xss:sequence&gt;
        &lt;/xss:complexType&gt;
    &lt;/xss:element&gt;

    &lt;xss:element name="Offertalmmobile"&gt;
        &lt;xss:complexType&gt;
            &lt;xss:sequence&gt;
                &lt;xss:element name="dataInizioScambio" type="xs:date"&gt;
                &lt;/xss:element&gt;
                &lt;xss:element name="dataFineScambio" type="xs:date"&gt;
                &lt;/xss:element&gt;
                &lt;xss:element name="disponibilitaCondivisioneVeicolo" type="xs:boolean"&gt;
                &lt;/xss:element&gt;
                &lt;xss:element ref="Limitazioni" minOccurs="1" maxOccurs="1"&gt;
                &lt;/xss:element&gt;
                &lt;xss:element ref="Descrizione" minOccurs="1" maxOccurs="1"&gt;
</pre>
```

```

        </xs:element>
        <xs:element ref="Indirizzo" minOccurs="1" maxOccurs="1">
        </xs:element>
        <xs:element ref="InfoRichiedente" minOccurs="0"
maxOccurs="unbounded">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Limitazioni">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="divietoFumo" type="xs:boolean">
            </xs:element>
            <xs:element name="divietoAnimali" type="xs:boolean">
            </xs:element>
            <xs:element name="vincoliRumoriSerali" type="xs:boolean">
            </xs:element>
            <xs:element name="possibilitaAccoglienzaBambini" type="xs:boolean">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="Descrizione">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="tipologiaImmobile" type="xs:string">
            </xs:element>
            <xs:element name="foto" type="xs:string">
            </xs:element>
            <xs:element name="composizioneImmobile" type="xs:string">
            </xs:element>
            <xs:element name="descrizioneZona" type="xs:string">
            </xs:element>
            <xs:element name="collegamentiMezziPubblici" type="xs:string">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="Indirizzo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="via" type="xs:string">
            </xs:element>
            <xs:element name="numeroCivico" type="xs:string">
            </xs:element>
        </xs:sequence>
        <xs:attribute name="CAP" type="xs:string" use="required">
        </xs:attribute>

```

```

<xs:attribute name="citta" type="xs:string" use="required">
</xs:attribute>
<xs:attribute name="stato" type="xs:string" use="required">
</xs:attribute>
</xs:complexType>
</xs:element>

<xs:element name="InfoRichiedente">
<xs:complexType>
<xs:sequence>
<xs:element name="nome" type="xs:string">
</xs:element>
<xs:element name="cognome" type="xs:string">
</xs:element>
<xs:element name="telefono" type="xs:string">
</xs:element>
<xs:element name="cellulare" type="xs:string">
</xs:element>
<xs:element name="indirizzoEmail" type="xs:string">
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

XML

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- OffertaCasa.xml -->
<ListaOffertalImmobile xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="OffertaCasa.xsd">
<OffertalImmobile>
<dataInizioScambio>2017-11-10</dataInizioScambio>
<dataFineScambio>2018-04-23</dataFineScambio>
<disponibilitaCondivisioneVeicolo>true</disponibilitaCondivisioneVeicolo>
<Limitazioni>
<divietoFumo>false</divietoFumo>
<divietoAnimali>false</divietoAnimali>
<vincoliRumoriSerali>true</vincoliRumoriSerali>
<possibilitaAccoglienzaBambini>true</possibilitaAccoglienzaBambini>
</Limitazioni>
<Descrizione>
<tipologialImmobile>Attico</tipologialImmobile>
<foto>OffertaCasa/imagehosting/Milano/pic=?"nsdSd454ghh"</foto>
<composizioneImmobile> Ampio e luminoso bilocale composto da ingresso
in ampio soggiorno/pranzo con zona cottura, ampio balcone. Disimpegno zona notte, una
camera matrimoniale con piccola cabina e balcone con affaccio su zona area verde, bagno
con finestra, vasca e doccia. L'appartamento e' completamente e finemente arredato con
mobili moderni, completo di ogni confort. Presente una cantina finestrata ad uso
esclusivo</composizioneImmobile>
<descrizioneZona>L'appartamento si trova in una zona centrale vicino al duomo, al
teatro Gerolamo e al cinema The Space. Nelle immediate vicinanze sono presenti

```

ristorante, pizzeria napoletana e risto-pub. L'appartamento, situato sul corso di Milano si trova nelle vicinanze di numerose attivita' commerciali.</descrizioneZona>

<collegamentiMezziPubblici>La dimora e' collegata alla linea blu della metro, e alla linea dei pulman. La stazione ferroviaria, sebbene lontana e' raggiungibile con facilita' utilizzando i collegamenti esistenti</collegamentiMezziPubblici>

</Descrizione>

<Indirizzo CAP="20121" citta="Milano" stato="Italia">

<via>Corso Vittorio Emanuele II</via>

<numeroCivico>12</numeroCivico>

</Indirizzo>

<InfoRichiedente>

<nome>Mario</nome>

<cognome>Rossi</cognome>

<telefono>081-7654321</telefono>

<cellulare>334-1234567</cellulare>

<indirizzoEmail>mario.rossi@gmail.com</indirizzoEmail>

</InfoRichiedente>

<InfoRichiedente>

<nome>Luca</nome>

<cognome>Bianco</cognome>

<telefono>0587-3216547</telefono>

<cellulare>328-1324657</cellulare>

<indirizzoEmail>lucabianco@gmail.com</indirizzoEmail>

</InfoRichiedente>

</OffertalImmobile>

<OffertalImmobile>

<dataInizioScambio>2018-11-13</dataInizioScambio>

<dataFineScambio>2018-11-30</dataFineScambio>

<disponibilitaCondivisioneVeicolo>false</disponibilitaCondivisioneVeicolo>

<Limitazioni>

<divietoFumo>false</divietoFumo>

<divietoAnimali>false</divietoAnimali>

<vincoliRumoriSerali>false</vincoliRumoriSerali>

<possibilitaAccoglienzaBambini>true</possibilitaAccoglienzaBambini>

</Limitazioni>

<Descrizione>

<tipologialImmobile>villa</tipologialImmobile>

<foto>OffertaCasa/imagehosting/Milano/pic=?"nsDFCd454gz8"</foto>

<composizioneImmobile>Villa completamente ristrutturato ed arredato con ampio e curato giardino pavimentato. La villa e' composto da ingresso, salone, cucina (completamente arredata e completa di tutti gli elettrodomestici), camera, servizio, piccola dependance adibita a magazzino/lavanderia ed ampio giardino semiperimetrale.</composizioneImmobile>

<descrizioneZona>Zona traquilla e provvista di tutti i servizi a 20 minuti a piedi dal centro citta'.</descrizioneZona>

<collegamentiMezziPubblici>La villa e' collegata alla linea bus urbana e alla linea metropolitana. La stazione ed il centro storico sono facilmente raggiungibili utilizzando i mezzi pubblici</collegamentiMezziPubblici>

</Descrizione>

<Indirizzo CAP="20121" citta="Milano" stato="Italia" >

<via>Parco della Vittoria</via>

<numeroCivico>666</numeroCivico>

```

        </Indirizzo>
    </OffertalImmobile>
</ListaOffertalImmobile>
```

8.3.2 Richiesta casa

XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
    <xs:element name="RichiestaCasa">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="ListaRichiestalImmobile" minOccurs="0"
maxOccurs="unbounded" />
                <xs:element ref="RichiestalImmobile" minOccurs="0"
maxOccurs="unbounded" />
                <xs:element ref="PropostalImmobile" minOccurs="0"
maxOccurs="unbounded" />
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="ListaRichiestalImmobile">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="RichiestalImmobile" minOccurs="0"
maxOccurs="unbounded"></xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>

    <xs:element name="RichiestalImmobile">
        <xs:complexType>
            <xs:sequence>
                <xs:element ref="InfoRichiedente" minOccurs="1"
maxOccurs="1"></xs:element>
                <xs:element name="citta" type="xs:string"></xs:element>
                <xs:element name="stato" type="xs:string"></xs:element>
                <xs:element name="inizio" type="xs:date"></xs:element>
                <xs:element name="fine" type="xs:date"></xs:element>
                <xs:element ref="DescrizioneImmobile" minOccurs="1"
maxOccurs="1"></xs:element>
                <xs:element ref="Limitazione" minOccurs="1"
maxOccurs="unbounded"></xs:element>
                <xs:element ref="Trasporto" minOccurs="1" maxOccurs="1"></xs:element>
                <xs:element ref="ClasseAttrazione" minOccurs="0"
maxOccurs="unbounded"></xs:element>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
```

```

<xs:element name="PropostaImmobile">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="InfoRichiedente" minOccurs="1"
maxOccurs="1"></xs:element>
            <xs:element name="inizio" type="xs:date"></xs:element>
            <xs:element name="fine" type="xs:date"></xs:element>
            <xs:element name="contesto" type="xs:string"></xs:element>
            <xs:element name="foto" type="xs:string"></xs:element>
            <xs:element ref="DescrizioneImmobile" minOccurs="1"
maxOccurs="1"></xs:element>
            <xs:element ref="Limitazione" minOccurs="1"
maxOccurs="unbounded"></xs:element>
                <xs:element ref="Trasporto" minOccurs="1" maxOccurs="1"></xs:element>
                <xs:element ref="Indirizzo" minOccurs="1" maxOccurs="1"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="InfoRichiedente">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="richiedenteID">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[a-zA-Z0-9]{6}" />
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="telefono" type="xs:string"></xs:element>
            <xs:element name="cellulare" type="xs:string"></xs:element>
            <xs:element name="email" type="xs:string"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="DescrizioneImmobile">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="tipologia" type="xs:string"></xs:element>
            <xs:element name="postiLetto" type="xs:int"></xs:element>
            <xs:element name="bagni" type="xs:int"></xs:element>
            <xs:element name="piscina" type="xs:boolean"></xs:element>
            <xs:element name="garage" type="xs:boolean"></xs:element>
            <xs:element name="terrazzo" type="xs:boolean"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="ClasseAttrazione">
    <xs:complexType >

```

```

<xs:sequence>
    <xs:element name="nome" type="xs:string"></xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="Limitazione">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nome" type="xs:string"></xs:element>
            <xs:element name="permesso" type="xs:boolean"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="Trasporto">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="mezziPubblici" type="xs:boolean">
            </xs:element>
            <xs:element name="condivisioneAuto" type="xs:boolean">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="Indirizzo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="citta" type="xs:string"></xs:element>
            <xs:element name="stato" type="xs:string"></xs:element>
            <xs:element name="via" type="xs:string"></xs:element>
            <xs:element name="nCivico" type="xs:string"></xs:element>
            <xs:element name="cap" type="xs:string"></xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

</xs:schema>

```

XML

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- RichiestaCasa.xml -->
<RichiestaCasa xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="RichiestaCasa.xsd">
    <ListaRichiestalImmobile>
        <RichiestalImmobile>
            <InfoRichiedente>
                <richiedenteID>a6f789</richiedenteID>
                <telefono>0583-2473849</telefono>
                <cellulare>+39 34687598</cellulare>

```

```

<email>cip93uzza@gmail.com</email>
</InfoRichiedente>
<citta>Stoccolma</citta>
<stato>Svezia</stato>
<inizio>2018-01-02</inizio>
<fine>2018-03-10</fine>
<DescrizioneImmobile>
    <tipologia>Cottage</tipologia>
    <postiLetto>4</postiLetto>
    <bagni>2</bagni>
    <piscina>false</piscina>
    <garage>true</garage>
    <terrazzo>true</terrazzo>
</DescrizioneImmobile>
<Limitazione>
    <nome>Fumatore</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Bambini piccoli</nome>
    <permesso>false</permesso>
</Limitazione>
<Limitazione>
    <nome>Animali</nome>
    <permesso>false</permesso>
</Limitazione>
<Limitazione>
    <nome>Suonare notte</nome>
    <permesso>true</permesso>
</Limitazione>
<Trasporto>
    <mezziPubblici>true</mezziPubblici>
    <condivisioneAuto>false</condivisioneAuto>
</Trasporto>
<ClasseAttrazione>
    <nome>Concerti e spettacoli</nome>
</ClasseAttrazione>
<ClasseAttrazione>
    <nome>Attività all'aperto'</nome>
</ClasseAttrazione>
<ClasseAttrazione>
    <nome>Giocchi d'azzardo</nome>
</ClasseAttrazione>
</RichiestaImmobile>
<RichiestaImmobile>
    <InfoRichiedente>
        <richiedenteID>3gg4f1</richiedenteID>
        <telefono>0583-225849</telefono>
        <cellulare>+39 3207074568</cellulare>
        <email>james_claus@msn.com</email>
    </InfoRichiedente>
    <citta>Roma</citta>

```

```

<stato>Italia</stato>
<inizio>2017-10-13</inizio>
<fine>2017-11-13</fine>
<DescrizioneImmobile>
    <tipologia>Villa</tipologia>
    <postiLetto>6</postiLetto>
    <bagni>2</bagni>
    <piscina>true</piscina>
    <garage>true</garage>
    <terrazzo>true</terrazzo>
</DescrizioneImmobile>
<Limitazione>
    <nome>Fumatore</nome>
    <permesso>false</permesso>
</Limitazione>
<Limitazione>
    <nome>Bambini piccoli</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Animali</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Suonare notte</nome>
    <permesso>false</permesso>
</Limitazione>
<Trasporto>
    <mezziPubblici>false</mezziPubblici>
    <condivisioneAuto>true</condivisioneAuto>
</Trasporto>
<ClasseAttrazione>
    <nome>Zoo e acquari</nome>
</ClasseAttrazione>
<ClasseAttrazione>
    <nome>Attività all'aperto'</nome>
</ClasseAttrazione>
<ClasseAttrazione>
    <nome>Parchi divertimento</nome>
</ClasseAttrazione>
</RichiestaImmobile>
</ListaRichiestalImmobile>
<PropostaImmobile>
    <InfoRichiedente>
        <richiedenteID>a6f789</richiedenteID>
        <telefono>0583-2473849</telefono>
        <cellulare>+39 34687598</cellulare>
        <email>cip93uzza@gmail.com</email>
    </InfoRichiedente>
    <inizio>2018-01-03</inizio>
    <fine>2018-03-03</fine>

```

```

<contesto>Itinerario: http://attrazioni.it/fd239dfk3n, Servizi:
http://servizi.com/243hkg58</contesto>
<foto>http://image.org/57gf45</foto>
<DescrizioneImmobile>
    <tipologia>Cottage</tipologia>
    <postiLetto>4</postiLetto>
    <bagni>2</bagni>
    <piscina>true</piscina>
    <garage>false</garage>
    <terrazzo>true</terrazzo>
</DescrizioneImmobile>
<Limitazione>
    <nome>Fumatore</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Bambini piccoli</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Animali</nome>
    <permesso>true</permesso>
</Limitazione>
<Limitazione>
    <nome>Suonare notte</nome>
    <permesso>true</permesso>
</Limitazione>
<Trasporto>
    <mezziPubblici>false</mezziPubblici>
    <condivisioneAuto>false</condivisioneAuto>
</Trasporto>
<Indirizzo>
    <citta>Stoccolma</citta>
    <stato>Svezia</stato>
    <via>Jakobsgatan</via>
    <nCivico>23A</nCivico>
    <cap>10055</cap>
</Indirizzo>
</PropostaImmobile>
</RichiestaCasa>

```

8.3.3 Attrazioni

XSD

```

<?xml version="1.0" encoding="UTF-8"?>
<xss:schema
    xmlns:xss="http://www.w3.org/2001/XMLSchema">
    <xss:element name="Attrazioni">
        <xss:complexType>
            <xss:sequence>

```

```

        <xs:element ref="ListaPropostaAttivita" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ListaPropostaAttivita">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="proposalID">
                <xs:simpleType>
                    <xs:restriction base="xs:string">
                        <xs:pattern value="[0-9A-Z]*"/>
                    </xs:restriction>
                </xs:simpleType>
            </xs:element>
            <xs:element name="statoProposta" type="xs:int">
            </xs:element>
            <xs:element ref="InformazioniSoggiorno" minOccurs="1" maxOccurs="1">
            </xs:element>
            <xs:element ref="PropostaAttivita" minOccurs="1"
maxOccurs="unbounded">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="InformazioniSoggiorno">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="indirizzo" type="xs:string">
            </xs:element>
            <xs:element name="dataInizio" type="xs:date">
            </xs:element>
            <xs:element name="dataFine" type="xs:date">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="PropostaAttivita">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="data" type="xs:date">
            </xs:element>
            <xs:element name="oraInizio" type="xs:time">
            </xs:element>
            <xs:element name="oraFine" type="xs:time">
            </xs:element>
            <xs:element ref="ListaAttrazioni" minOccurs="1" maxOccurs="1">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ListaAttrazioni">

```

```

<xs:complexType>
    <xs:sequence>
        <xs:element ref="Attrazione" minOccurs="0" maxOccurs="unbounded">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="Attrazione">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nome" type="xs:string">
            </xs:element>
            <xs:element name="categoria" type="xs:string">
            </xs:element>
            <xs:element name="indirizzo" type="xs:string">
            </xs:element>
            <xs:element name="distanza">
                <xs:complexType>
                    <xs:simpleContent>
                        <xs:extension base="xs:float">
                            <xs:attribute name="unitaMisura"
                            type="xs:string" use="required"/>
                        </xs:extension>
                    </xs:simpleContent>
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>

```

XML

```

<?xml version="1.0" encoding="UTF-8"?>
<Attrazioni xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
xsi:noNamespaceSchemaLocation="Attrazioni_.xsd">

<ListaPropostaAttivita>
    <propostaID>AKL32D11</propostaID>
    <statoProposta>2</statoProposta>

    <InformazioniSoggiorno>
        <indirizzo>84 Caledonian Rd Kings Cross, London N1 9DN Regno Unito
        UK</indirizzo>
        <dataInizio>2017-06-09</dataInizio>
        <dataFine>2017-06-29</dataFine>
    </InformazioniSoggiorno>

    <PropostaAttivita>
        <data>2017-06-10</data>
        <oralnizio>08:00:00</oralnizio>
        <oraFine>12:00:00</oraFine>

```

```

<ListaAttrazioni>

    <Attrazione>
        <nome>Torre di Londra</nome>
        <categoria>Siti storici, siti d'interesse</categoria>
        <indirizzo>Tower Hill, Londra EC3N4AB</indirizzo>
        <distanza unitaMisura="Km">5.3</distanza>
    </Attrazione>

    <Attrazione>
        <nome>British Museum</nome>
        <categoria>Musei</categoria>
        <indirizzo>Great Russell Street, WC1B 3DG</indirizzo>
        <distanza unitaMisura="Km">2.6</distanza>
    </Attrazione>

</ListaAttrazioni>

</PropostaAttivita>

<PropostaAttivita>
    <data>2017-06-10</data>
    <oralnizio>12:00:00</oralnizio>
    <oraFine>17:00:00</oraFine>

    <ListaAttrazioni>

        <Attrazione>
            <nome>Fishers Fish and Chips</nome>
            <categoria>Ristoranti</categoria>
            <indirizzo>19 Fulham High Street, Londra SW6 3JH</indirizzo>
            <distanza unitaMisura="Km">4.7</distanza>
        </Attrazione>

    </ListaAttrazioni>

</PropostaAttivita>

<PropostaAttivita>
    <data>2017-06-10</data>
    <oralnizio>17:00:00</oralnizio>
    <oraFine>20:00:00</oraFine>

    <ListaAttrazioni>

        <Attrazione>
            <nome>London Zoo</nome>
            <categoria>Zoo e acquari</categoria>
            <indirizzo>Regents Park, NW1 4RY</indirizzo>
            <distanza unitaMisura="Km">4.3</distanza>
        </Attrazione>

```

```

<Attrazione>
    <nome>National Gallery</nome>
    <categoria>Musei</categoria>
    <indirizzo>Trafalgar Square, WC2N 5DN</indirizzo>
    <distanza unitaMisura="Km">4.2</distanza>
</Attrazione>

<Attrazione>
    <nome>Big Ben</nome>
    <categoria>Siti d'interesse</categoria>
    <indirizzo>Westminster, Londra SW1A 0AA</indirizzo>
    <distanza unitaMisura="Km">5.5</distanza>
</Attrazione>

</ListaAttrazioni>

</PropostaAttivita>

</ListaPropostaAttivita>

</Attrazioni>

```

8.3.4 Servizi accessori

XSD

```

<?xml version="1.0" encoding="UTF-8" ?>
<xss:schema
    xmlns:xss="http://www.w3.org/2001/XMLSchema">

    <xss:element name="ServiziAccessoriRoot">
        <xss:complexType>
            <xss:sequence>
                <xss:element ref="Pianificazione" minOccurs="1" maxOccurs="1">
                </xss:element>
                <xss:element ref="FasceOrarie" minOccurs="1" maxOccurs="unbounded">
                </xss:element>
            </xss:sequence>
        </xss:complexType>
    </xss:element>

    <xss:element name="Pianificazione">
        <xss:complexType>
            <xss:sequence>
                <xss:element name="IDPianificazione">
                    <xss:simpleType>
                        <xss:restriction base="xs:string">
                            <xss:pattern value="[0-9A-Za-z]*"/>
                        </xss:restriction>
                    </xss:simpleType>
                </xss:element>
            </xss:sequence>
        </xss:complexType>
    </xss:element>

```

```

        </xs:element>
        <xs:element ref="ListaServiziAccessori" minOccurs="1" maxOccurs="1">
        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>

<xs:element name="ListaServiziAccessori">
    <xs:complexType>
        <xs:sequence>
            <xs:element ref="Indirizzo" minOccurs="1" maxOccurs="1">
            </xs:element>
            <xs:element ref="ServiziAccessori" minOccurs="0"
maxOccurs="unbounded">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="Indirizzo">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="via" type="xs:string"/>
            <xs:element name="CAP" type="xs:int"/>
            <xs:element name="stato" type="xs:string"/>
            <xs:element name="NCivico" type="xs:int"/>
            <xs:element name="citta" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="ServiziAccessori">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="nome" type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>

<xs:element name="FasceOrarie">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="OrarioInizio" type="xs:time"/>
            <xs:element name="OrarioFine" type="xs:time"/>
            <xs:element ref="ServiziAccessori" minOccurs="1"
maxOccurs="unbounded">
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>

```

XML

```
<?xml version="1.0" encoding="UTF-8"?>
<ServiziAccessoriRoot xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation="ServiziAccessori.xsd">
    <Pianificazione>
        <IDPianificazione>ZfdG45</IDPianificazione>
        <ListaServiziAccessori>
            <Indirizzo>
                <via>Cavour</via>
                <CAP>56122</CAP>
                <stato>Italia</stato>
                <NCivico>4</NCivico>
                <citta>Pisa</citta>
            </Indirizzo>
            <ServiziAccessori>
                <nome>Giardinaggio</nome>
            </ServiziAccessori>
            <ServiziAccessori>
                <nome>Babysitting</nome>
            </ServiziAccessori>
        </ListaServiziAccessori>
    </Pianificazione>

    <FasceOrarie>
        <OrarioInizio>10:30:00</OrarioInizio>
        <OrarioFine>12:00:00</OrarioFine>
        <ServiziAccessori>
            <nome>Giardinaggio</nome>
        </ServiziAccessori>
    </FasceOrarie>
    <FasceOrarie>
        <OrarioInizio>19:30:00</OrarioInizio>
        <OrarioFine>23:00:00</OrarioFine>
        <ServiziAccessori>
            <nome>Babysitting</nome>
        </ServiziAccessori>
    </FasceOrarie>
</ServiziAccessoriRoot>
```

8.4 Orchestrazione locale in codice Java

```
import java.lang.Math;
import java.util.Date;

public class Coordinatore {

    public static void main(String[] args) {
        InterfacciaOffertaCasa offertaCasa = new InterfacciaOffertaCasa();
```

```

InterfacciaRichiestaCasa richiestaCasa = new InterfacciaRichiestaCasa();
InterfacciaAttrazioni attrazioni = new InterfacciaAttrazioni();
InterfacciaServiziAccessori serviziAccessori = new InterfacciaServiziAccessori();

int maxAttivita = 10;
String richiesteNonGestite;
boolean status = false;
String esito;
String richiesta, offerta, pianificazione;
String proposta, propostaCompilata, propostaModificata;
String indirizzoRichiesta, limitazioniRichiesta, trasportoRichiesta, descrizioneRichiesta;
String classiAttrazioniRichiesta;
String listaOfferte, listaOfferteOrdinata;
String indirizzoOfferta;
String listaAttrazioni, classeAttrazione;
String riferimentoPropostaAttivita, riferimentoServiziAccessori;
String infoSoggiorno;
String richiedente, contattiRichiedente;
Date inizioCondivisione, fineCondivisione;
int statoProposta;
String propostaID, listaPropostaAttivita, propostaAttivita;
String fasceOrarie;

richiesteNonGestite = richiestaCasa.ottieniRichiesteNonGestite();
int lunghezzaRichiesteNonGestite = estraiLunghezzaLista(richiesteNonGestite);

do{
    if(lunghezzaRichiesteNonGestite == 0)
        return;

    richiesta = estraiRichiesta(richiesteNonGestite);
    lunghezzaRichiesteNonGestite--;
    for(int i=0; i<3 || (!status); i++)
        status = richiestaCasa.gestisciRichiesta(richiesta);
}while(!status);

indirizzoRichiesta = estraiIndirizzoRichiesta(richiesta);
limitazioniRichiesta = estraiLimitazioniRichiesta(richiesta);
trasportoRichiesta = estraiTrasportoRichiesta(richiesta);
descrizioneRichiesta = estraiDescrizioneRichiesta(richiesta);

do{
    int lunghezzaOfferte;
    do{
        listaOfferte = offertaCasa.ricercaOfferte(indirizzoRichiesta,
        limitazioniRichiesta, trasportoRichiesta, descrizioneRichiesta);
        lunghezzaOfferte = estraiLunghezzaLista(listaOfferte);
        if(lunghezzaOfferte == 0) {
            if(limitazioniRichiesta.equals(""))
                return;
            limitazioniRichiesta = eliminaLimitazione(limitazioniRichiesta);
        }
    }
}

```

```

}while(lunghezzaOfferte == 0);

classiAttrazioniRichiesta = estraiClassiAttrazioniRichiesta(richiesta);
int lunghezzaClassiAttrazioni = estraiLunghezzaLista(classiAttrazioniRichiesta);
float[] punteggi = new float[lunghezzaOfferte];

for (int i=0; i<lunghezzaOfferte; i++) {
    offerta = ottieniOfferta(listaOfferte, i);
    indirizzoOfferta = estraiIndirizzoOfferta(offerta);

    listaAttrazioni = attrazioni.ottieniListaAttrazioni(indirizzoOfferta);
    int lunghezzaListaAttrazioni = estraiLunghezzaLista(listaAttrazioni);

    int numeroClassi = 0;
    for (int j=0; j<lunghezzaClassiAttrazioni && lunghezzaListaAttrazioni
!= 0; j++) {
        classeAttrazione = ottieniClasseAttrazione(classiAttrazioniRichiesta,
j);
        int numeroAttrazioni = contaAttrazioni(listaAttrazioni,
classeAttrazione);
        if (numeroAttrazioni > 0)
            numeroClassi++;
    }
    punteggi[i] = assegnaPunteggio(numeroClassi);
}

listaOfferteOrdinata = ordinaLista(listaOfferte, punteggi);

offerta = ottieniOfferta(listaOfferteOrdinata, 0);
indirizzoOfferta = estraiIndirizzoOfferta(offerta);
infoSoggiorno = estraiInfoSoggiorno(offerta, richiesta);

riferimentoPropostaAttivita =
attrazioni.suggerisciListaPropostaAttivita(infoSoggiorno, classiAttrazioniRichiesta);
riferimentoServiziAccessori =
serviziAccessori.ottieniRiferimentoServiziAccessori(indirizzoOfferta);

richiedente = estraiRichiedente(richiesta);
inizioCondivisione = calcolaInizioCondivisione(offerta, richiesta);
fineCondivisione = calcolaFineCondivisione(offerta, richiesta);

proposta = richiestaCasa.creaProposta(richiedente, indirizzoOfferta,
inizioCondivisione, fineCondivisione);
propostaCompilata = inserisciInfoProposta(proposta, offerta,
riferimentoPropostaAttivita, riferimentoServiziAccessori);
propostaModificata = richiestaCasa.modificaProposta(propostaCompilata);

int indice = 1;
statoProposta = 0;
while(indice < lunghezzaOfferte) {
    proposta = richiestaCasa.ottieniProposta(richiedente, indirizzoOfferta,
inizioCondivisione, fineCondivisione);
}

```

```

        statoProposta = estraiStatoProposta(proposta);
        if(statoProposta == 1) {
            offerta = ottieniOfferta(listaOfferteOrdinata, indice);
            indirizzoOfferta = estraiIndirizzoOfferta(offerta);
            inizioCondivisione = calcolaInizioCondivisione(offerta, richiesta);
            fineCondivisione = calcolaFineCondivisione(offerta, richiesta);
            riferimentoServiziAccessori =
                serviziAccessori.ottieniRiferimentoServiziAccessori(indirizzoOfferta);
                propostaCompilata = inserisciInfoProposta(proposta, offerta,
                riferimentoPropostaAttivita, riferimentoServiziAccessori);
                propostaModificata =
                    richiestaCasa.modificaProposta(propostaCompilata);
                    indice++;
            }
            else if(statoProposta == 2)
                break;
        }

        if(statoProposta != 2)
            limitazioniRichiesta = eliminaLimitazione(limitazioniRichiesta);

    }while(statoProposta != 2);

    contattiRichiedente = estraiContatti(proposta);
    esito = offertaCasa.pubblicaPropostaScambio(offerta, contattiRichiedente);

    propostaID = estraiPropostaID(riferimentoPropostaAttivita);
    listaPropostaAttivita = attrazioni.visualizzaListaPropostaAttivita(propostaID);
    statoProposta = estraiStatoListaPropostaAttivita(listaPropostaAttivita);
    while(statoProposta != 2) {
        if(statoProposta == 1) {
            int lunghezzaPropostaAttivita = estraiLunghezzaLista(listaPropostaAttivita);
            boolean[] fasceVuote = cercaFasceOrarieVuote(listaPropostaAttivita,
            lunghezzaPropostaAttivita);
            listaPropostaAttivita = attrazioni.aggiungiAlternative(listaPropostaAttivita,
            fasceVuote);

            int lunghezzaAlternative = lunghezzaPropostaAttivita * maxAttivita;
            boolean[] daRimuovere = new boolean[lunghezzaAlternative];
            daRimuovere = selezionaAttivitaDaRimuovere(listaPropostaAttivita,
            classiAttrazioniRichiesta, lunghezzaAlternative);
            listaPropostaAttivita = attrazioni.rimuoviAlternative(listaPropostaAttivita,
            daRimuovere);
        }
        listaPropostaAttivita = attrazioni.visualizzaListaPropostaAttivita(propostaID);
        statoProposta = estraiStatoListaPropostaAttivita(listaPropostaAttivita);
    }

    fasceOrarie = estraiFasceOrarie(listaPropostaAttivita);
    pianificazione = serviziAccessori.creaPianificazione(fasceOrarie);
}

```

```

public static String estraiRichiesta(String listaRichieste) {
    return "Prima richiesta";
}

public static String estraiIndirizzoRichiesta(String richiesta) {
    return "Indirizzo estratto da richiesta";
}

public static String estraiLimitazioniRichiesta(String richiesta) {
    return "Limitazioni estratte da richiesta";
}

public static String estraiTrasportoRichiesta(String richiesta) {
    return "Trasporto estratto da richiesta";
}

public static String estraiDescrizioneRichiesta(String richiesta) {
    return "Descrizione estratta da richiesta";
}

public static int estraiLunghezzaLista(String lista){
    if(lista.equals("Lista di attrazioni ordinata per distanza"))
        return (int)(Math.random()*10);
    if(lista.equals("Lista di offerte pertinenti con i parametri di ricerca."))
        return (int)(Math.random()*20);
    if(lista.equals("Richieste non gestite"))
        return (int)(Math.random()*20);
    if(lista.equals("Lista proposta attivita"))
        return (int)(Math.random()*100);

    return (int)(Math.random()*10);
}

public static String estraiClassiAttrazioniRichiesta(String richiesta) {
    return "Lista classi attrazioni estratte da richiesta";
}

public static String ottieniOfferta(String lista, int indice) {
    return "Offerta all'indice " + indice;
}

public static String estraiIndirizzoOfferta(String offerta) {
    return "Indirizzo estatto da offerta";
}

public static String ottieniClasseAttrazione(String classiAttrazioniRichiesta, int indice) {
    return "Classe d'attrazione all'indice " + indice;
}

public static int contaAttrazioni(String listaAttrazioni, String classe) {

```

```

        return (int)(Math.random()*5);
    }

    public static float assegnaPunteggio(int numeroClassi) {
        return numeroClassi * (float) Math.random() * 10;
    }

    public static String ordinaLista(String listaDaOrdinare, float[] punteggio) {
        return "Lista offerte ordinata per punteggio decrescente";
    }

    public static String estraiInfoSoggiorno(String offerta, String richiesta) {
        return "Informazioni soggiorno";
    }

    public static String estraiRichiedente(String richiesta){
        return "Richiedente";
    }

    public static Date calcolaInizioCondivisione(String offerta, String richiesta){
        return new Date(2017, (int)(Math.random()*12), (int)(Math.random()*27+1));
    }

    public static Date calcolaFineCondivisione(String offerta, String richiesta){
        return new Date(2018, (int)(Math.random()*12), (int)(Math.random()*27+1));
    }

    public static String inserisciInfoProposta(String proposta, String offerta, String
riferimentoPropostaAttivita, String riferimentoServiziAccessori){
        return "Proposta compilata";
    }

    public static int estraiStatoProposta(String richiesta){
        return (int)(Math.random()*3);
    }

    public static String eliminaLimitazione(String limitazioniRichiesta){
        double r = Math.random();
        if(r<=0.05)
            return "";
        return "Limitazioni ridotte";
    }

    public static String estraiContatti(String proposta){
        return "Contatti del richiedente";
    }

    public static String estraiPropostaID(String riferimentoPropostaAttivita) {
        return "ID della proposta";
    }

    public static int estraiStatoListaPropostaAttivita(String listaPropostaAttivita){

```

```

        return (int)(Math.random()*3);
    }

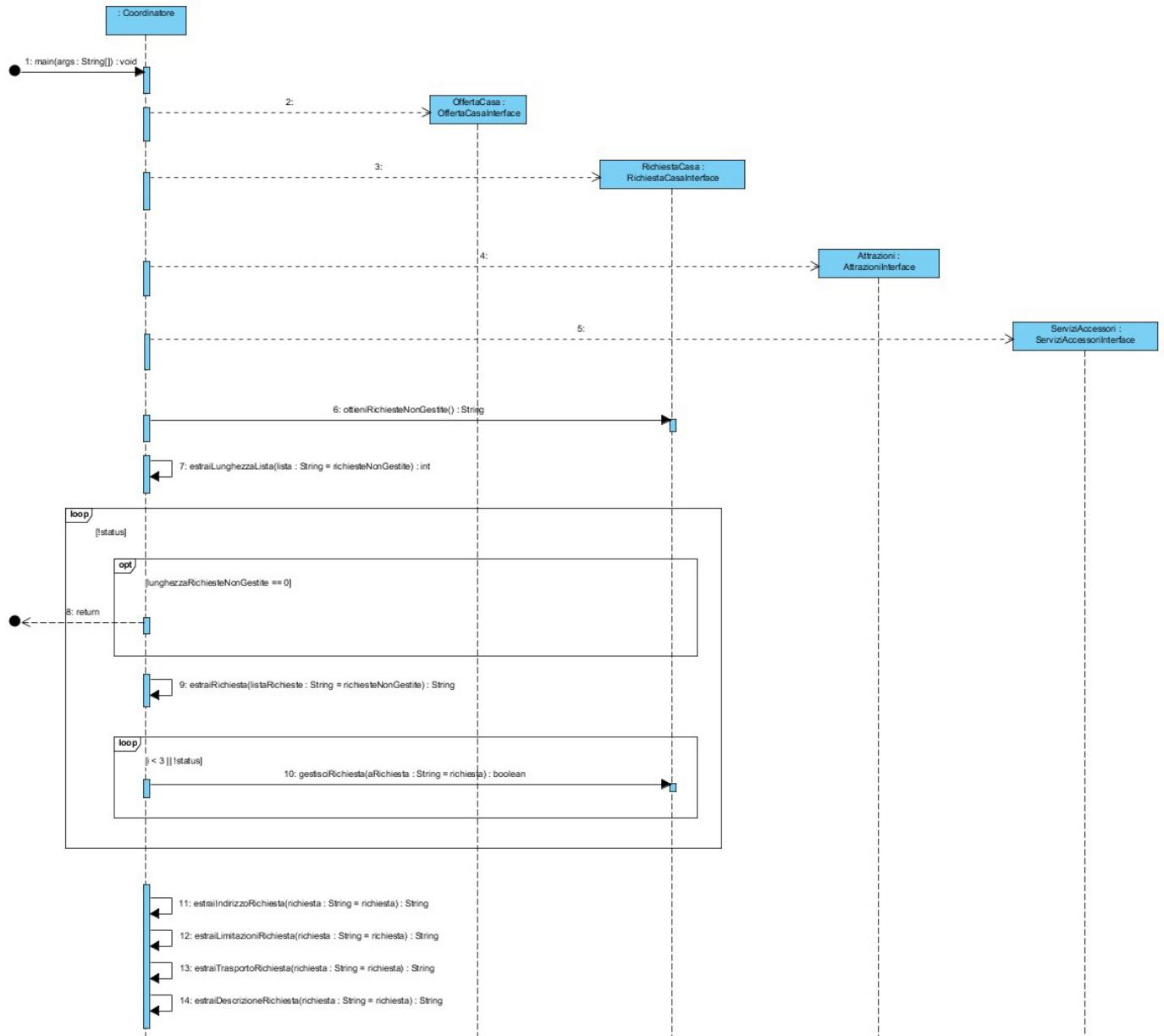
public static boolean[] cercaFasceOrarieVuote(String listaPropostaAttivita, int lunghezza){
    boolean[] temp = new boolean[lunghezza];
    for(int i = 0; i < lunghezza; i++ )
        temp[i] = (Math.random() < 0.3)? true : false;
    return temp;
}

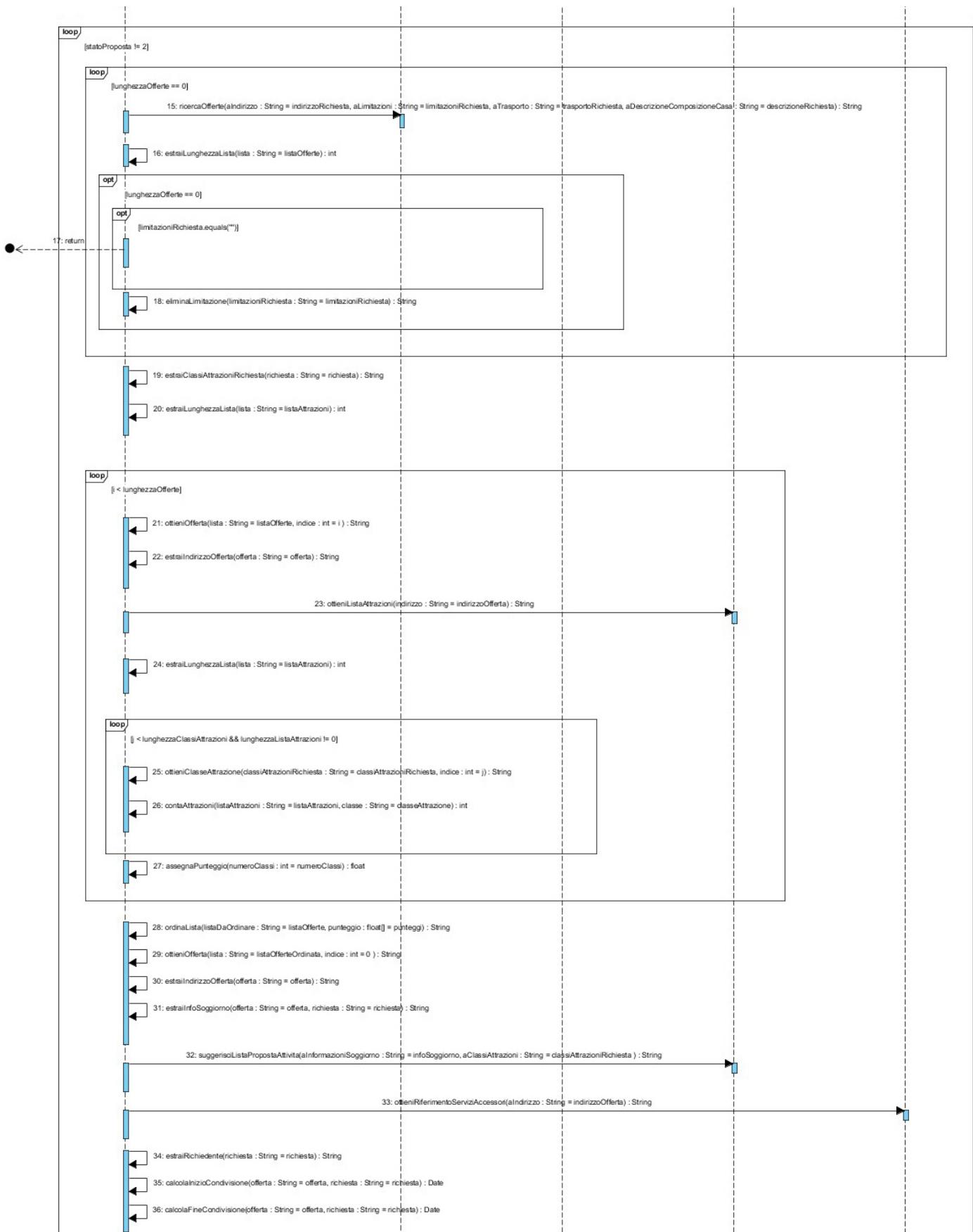
public static boolean[] selezionaAttivitaDaRimuovere(String listaPropostaAttivita, String
classiAttrazioni, int lunghezza){
    boolean[] temp = new boolean[lunghezza];
    for(int i = 0; i < lunghezza; i++ )
        temp[i] = (Math.random() < 0.2)? true : false;
    return temp;
}

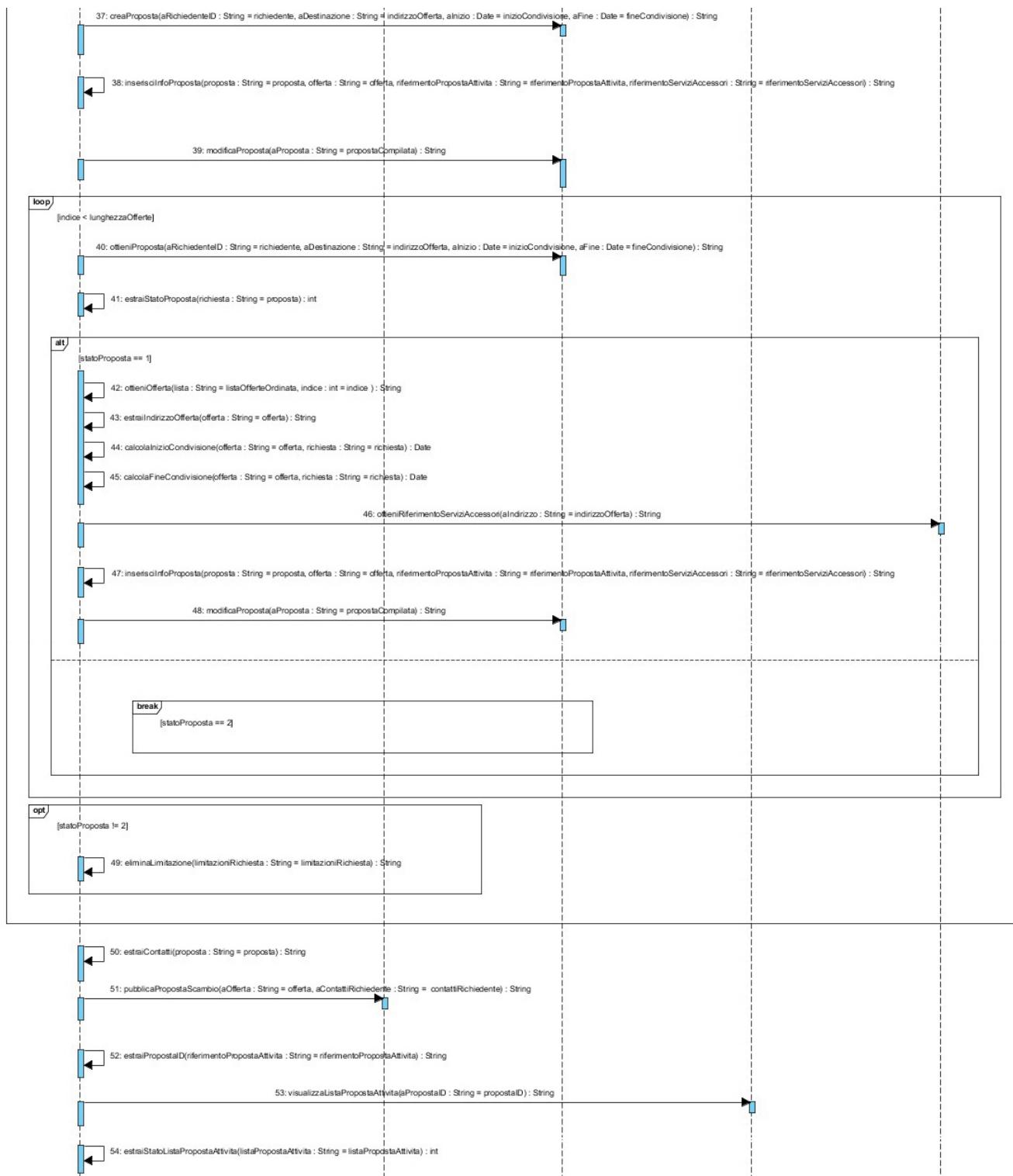
public static String estraiFasceOrarie(String listaPropostaAttivita){
    return "Fasce orarie estratte";
}
}

```

8.5 Diagramma di sequenza generato dal codice Java dell'orchestrazione







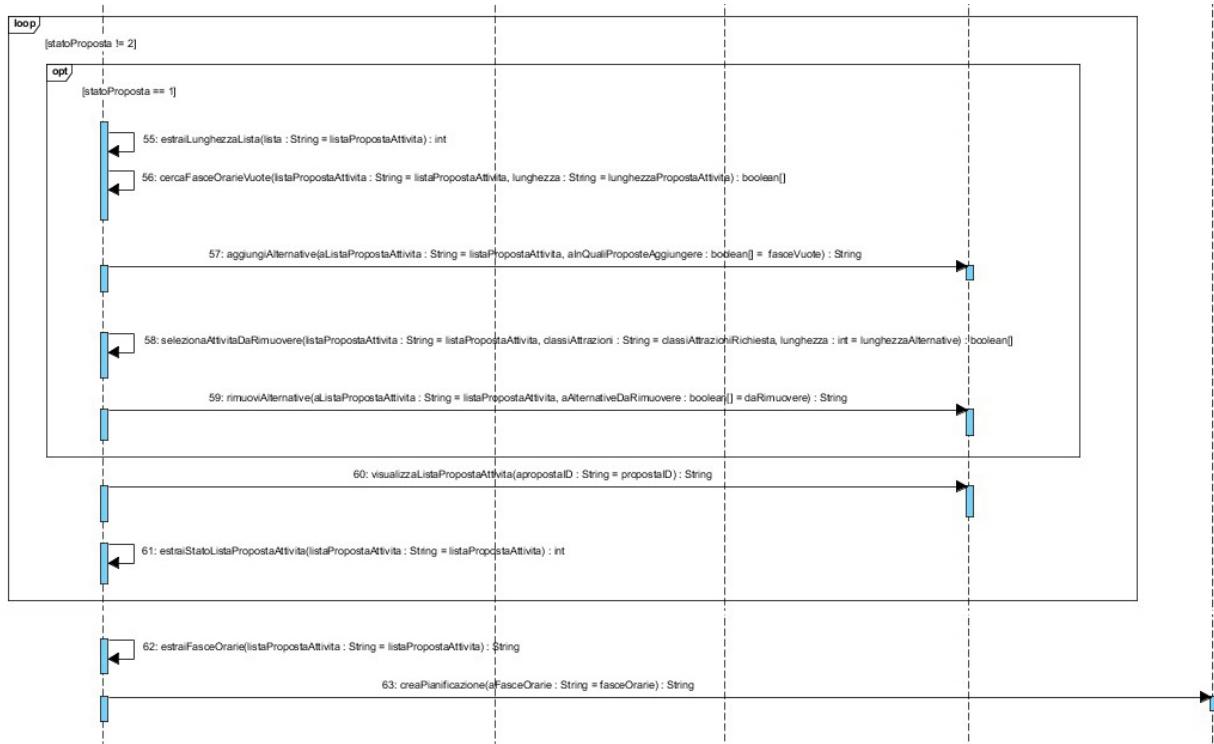


Figura 8.2: Diagramma di sequenza UML del coordinatore generato dal codice Java

8.5.1 Confronto tra R e K

Dal confronto tra i due diagrammi, non abbiamo notato tantissime differenze: l'implementazione del codice non ha richiesto particolari modifiche rispetto a quanto era stato progettato in fase di design. Tuttavia, dato che R è stato generato da codice Java, ci sono state delle espansioni alle azioni effettuate dall'agente:

- estraiParametriRichiesta è stata scomposta in quattro diverse funzioni (estraiIndirizzoRichiesta, estraiLimitazioniRichiesta, estraiTrasportoRichiesta e estraiDescrizioneRichiesta), ognuna delle quali è stata utilizzata per prelevare una specifica informazione dalla richiesta.
- È stata implementata una funzione estraiLunghezzaLista che restituisce la lunghezza della lista che viene passata come parametro: la dimensione viene generata con un numero random compreso in un intervallo che varia a seconda del tipo di lista.
- calcolaPeriodoCondivisione è stata scomposta in due diverse funzioni perché creaProposta richiede che vengano passate le date di inizio e fine. Scomponendo in calcolainizioCondivisione e calcolaFineCondivisione,abbiamo evitato di dover restituire come ritorno un vettore di Date e abbiamo realizzato due funzioni più semplici e leggibili.
- In R è stato aggiunto un ciclo per provare a gestire la stessa richiesta fino a un massimo di tre volte. gestisciRichiesta può avere esito negativo per ragioni che non riguardano la logica di coordinamento essendo legata solo all'applicativo *Richiesta casa* e allo stato in cui esso si trova, quindi è una scelta che riguarda l'implementazione.

- Le condizioni in K sono ancora espresse in linguaggio naturale ma poi in R sono state formalizzate: in Java sono state introdotte delle variabili, definendo il significato da attribuire a ognuno dei valori che queste possono assumere. Pertanto le condizioni così espresse sono state sostituite dalle corrispondenti condizioni espresse in linguaggio di programmazione.
- Nella parte finale del protocollo, la logica che permetteva al coordinatore di comunicare con il modulo “Attrazioni” è stata inglobata dentro le due funzioni cercaFasceOrarieVuote() e selezionaAttivitàDaRimuovere() perché i cicli pensati in fase di design potevano essere inclusi all’interno di queste funzioni senza compromettere la correttezza del protocollo. In questo modo, il codice è stato reso più lineare e comprensibile perché si sono ridotte le chiamate a funzione.

8.6 Axis WS

Le classi Java sono state caricate sul server Apache come web service Axis. Sono stati quindi compilate ed eseguite. Di seguito mostriamo per ogni modulo i relativi file WSDL, generati in automatico da Axis.

8.6.1 WSDL Offerta casa

```
<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
  xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://ws.apache.org/axis2"
  xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
  xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
  xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
  xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
  targetNamespace="http://ws.apache.org/axis2">
  <wsdl:documentation>InterfacciaOffertaCasa</wsdl:documentation>
  <wsdl:types>
    <xsschema attributeFormDefault="qualified" elementFormDefault="qualified"
      targetNamespace="http://ws.apache.org/axis2">
      <xss:element name="ricercaOfferte">
        <xss:complexType>
          <xss:sequence>
            <xss:element minOccurs="0" name="args0" nillable="true"
              type="xss:string"/>
            <xss:element minOccurs="0" name="args1" nillable="true"
              type="xss:string"/>
            <xss:element minOccurs="0" name="args2" nillable="true"
              type="xss:string"/>
            <xss:element minOccurs="0" name="args3" nillable="true"
              type="xss:string"/>
          </xss:sequence>
        </xss:complexType>
      </xss:element>
      <xss:element name="ricercaOfferteResponse">
        <xss:complexType>
          <xss:sequence>
```

```

                <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="pubblicaPropostaScambio">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
                    <xs:element minOccurs="0" name="args1" nillable="true"
type="xs:string"/>
                </xs:sequence>
            </xs:complexType>
        </xs:element>
    <xs:element name="pubblicaPropostaScambioResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="ricercaOfferteRequest">
    <wsdl:part name="parameters" element="ns:ricercaOfferte"/>
</wsdl:message>
<wsdl:message name="ricercaOfferteResponse">
    <wsdl:part name="parameters" element="ns:ricercaOfferteResponse"/>
</wsdl:message>
<wsdl:message name="pubblicaPropostaScambioRequest">
    <wsdl:part name="parameters" element="ns:pubblicaPropostaScambio"/>
</wsdl:message>
<wsdl:message name="pubblicaPropostaScambioResponse">
    <wsdl:part name="parameters" element="ns:pubblicaPropostaScambioResponse"/>
</wsdl:message>
<wsdl:portType name="InterfacciaOffertaCasaPortType">
    <wsdl:operation name="ricercaOfferte">
        <wsdl:input message="ns:ricercaOfferteRequest"
wsaw:Action="urn:ricercaOfferte"/>
        <wsdl:output message="ns:ricercaOfferteResponse"
wsaw:Action="urn:ricercaOfferteResponse"/>
    </wsdl:operation>
    <wsdl:operation name="pubblicaPropostaScambio">
        <wsdl:input message="ns:pubblicaPropostaScambioRequest"
wsaw:Action="urn:pubblicaPropostaScambio"/>
        <wsdl:output message="ns:pubblicaPropostaScambioResponse"
wsaw:Action="urn:pubblicaPropostaScambioResponse"/>
    </wsdl:operation>
</wsdl:portType>

```

```

<wsdl:binding name="InterfacciaOffertaCasaSoap11Binding"
type="ns:InterfacciaOffertaCasaPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="ricercaOfferte">
        <soap:operation soapAction="urn:ricercaOfferte" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="pubblicaPropostaScambio">
        <soap:operation soapAction="urn:pubblicaPropostaScambio" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="InterfacciaOffertaCasaSoap12Binding"
type="ns:InterfacciaOffertaCasaPortType">
        <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
        <wsdl:operation name="ricercaOfferte">
            <soap12:operation soapAction="urn:ricercaOfferte" style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="pubblicaPropostaScambio">
            <soap12:operation soapAction="urn:pubblicaPropostaScambio"
style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="InterfacciaOffertaCasaHttpBinding"
type="ns:InterfacciaOffertaCasaPortType">
        <http:binding verb="POST"/>
        <wsdl:operation name="ricercaOfferte">
            <http:operation location="ricercaOfferte"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>

```

```

        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="pubblicaPropostaScambio">
        <http:operation location="pubblicaPropostaScambio"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="InterfacciaOffertaCasa">
    <wsdl:port name="InterfacciaOffertaCasaHttpSoap11Endpoint"
binding="ns:InterfacciaOffertaCasaSoap11Binding">
        <soap:address
location="http://localhost:8080/axis2/services/InterfacciaOffertaCasa.InterfacciaOffertaCasaHttpS
oap11Endpoint"/>
    </wsdl:port>
    <wsdl:port name="InterfacciaOffertaCasaHttpSoap12Endpoint"
binding="ns:InterfacciaOffertaCasaSoap12Binding">
        <soap12:address
location="http://localhost:8080/axis2/services/InterfacciaOffertaCasa.InterfacciaOffertaCasaHttpS
oap12Endpoint"/>
    </wsdl:port>
    <wsdl:port name="InterfacciaOffertaCasaHttpEndpoint"
binding="ns:InterfacciaOffertaCasaHttpBinding">
        <http:address
location="http://localhost:8080/axis2/services/InterfacciaOffertaCasa.InterfacciaOffertaCasaHttpE
ndpoint"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

8.6.2 WSDL Richiesta casa

```

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://ws.apache.org/axis2"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.apache.org/axis2">
    <wsdl:documentation>InterfacciaRichiestaCasa</wsdl:documentation>
    <wsdl:types>
        <xsschema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://ws.apache.org/axis2">
            <xselement name="creaProposta">

```

```

<xs:complexType>
    <xs:sequence>
        <xs:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" name="args1" nillable="true"
type="xs:string"/>
        <xs:element minOccurs="0" name="args2" nillable="true"
type="xs:date"/>
        <xs:element minOccurs="0" name="args3" nillable="true"
type="xs:date"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="creaPropostaResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ottieniRichiesta">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
            <xs:element minOccurs="0" name="args1" nillable="true"
type="xs:string"/>
            <xs:element minOccurs="0" name="args2" nillable="true"
type="xs:string"/>
            <xs:element minOccurs="0" name="args3" nillable="true"
type="xs:date"/>
            <xs:element minOccurs="0" name="args4" nillable="true"
type="xs:date"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ottieniRichiestaResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="gestisciRichiesta">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
        </xs:sequence>
</xs:complexType>

```

```

</xs:element>
<xs:element name="gestisciRichiestaResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return"
type="xs:boolean"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniRichiesteGestite">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniRichiesteGestiteResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniProposta">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
            <xs:element minOccurs="0" name="args1" nillable="true"
type="xs:string"/>
            <xs:element minOccurs="0" name="args2" nillable="true"
type="xs:date"/>
            <xs:element minOccurs="0" name="args3" nillable="true"
type="xs:date"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniPropostaResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniRichiesteNonGestite">
    <xs:complexType>
        <xs:sequence/>
    </xs:complexType>
</xs:element>
<xs:element name="ottieniRichiesteNonGestiteResponse">
    <xs:complexType>
        <xs:sequence>

```

```

                <xs:element minOccurs="0" name="return" nillable="true"
                  type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="modificaProposta">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="args0" nillable="true"
                  type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
    <xs:element name="modificaPropostaResponse">
        <xs:complexType>
            <xs:sequence>
                <xs:element minOccurs="0" name="return" nillable="true"
                  type="xs:string"/>
            </xs:sequence>
        </xs:complexType>
    </xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="ottieniRichiesteGestiteRequest">
    <wsdl:part name="parameters" element="ns:ottieniRichiesteGestite"/>
</wsdl:message>
<wsdl:message name="ottieniRichiesteGestiteResponse">
    <wsdl:part name="parameters" element="ns:ottieniRichiesteGestiteResponse"/>
</wsdl:message>
<wsdl:message name="ottieniRichiestaRequest">
    <wsdl:part name="parameters" element="ns:ottieniRichiesta"/>
</wsdl:message>
<wsdl:message name="ottieniRichiestaResponse">
    <wsdl:part name="parameters" element="ns:ottieniRichiestaResponse"/>
</wsdl:message>
<wsdl:message name="ottieniRichiesteNonGestiteRequest">
    <wsdl:part name="parameters" element="ns:ottieniRichiesteNonGestite"/>
</wsdl:message>
<wsdl:message name="ottieniRichiesteNonGestiteResponse">
    <wsdl:part name="parameters" element="ns:ottieniRichiesteNonGestiteResponse"/>
</wsdl:message>
<wsdl:message name="creaPropostaRequest">
    <wsdl:part name="parameters" element="ns:creaProposta"/>
</wsdl:message>
<wsdl:message name="creaPropostaResponse">
    <wsdl:part name="parameters" element="ns:creaPropostaResponse"/>
</wsdl:message>
<wsdl:message name="gestisciRichiestaRequest">
    <wsdl:part name="parameters" element="ns:gestisciRichiesta"/>
</wsdl:message>
<wsdl:message name="gestisciRichiestaResponse">
    <wsdl:part name="parameters" element="ns:gestisciRichiestaResponse"/>

```

```

    </wsdl:message>
    <wsdl:message name="ottieniPropostaRequest">
        <wsdl:part name="parameters" element="ns:ottieniProposta"/>
    </wsdl:message>
    <wsdl:message name="ottieniPropostaResponse">
        <wsdl:part name="parameters" element="ns:ottieniPropostaResponse"/>
    </wsdl:message>
    <wsdl:message name="modificaPropostaRequest">
        <wsdl:part name="parameters" element="ns:modificaProposta"/>
    </wsdl:message>
    <wsdl:message name="modificaPropostaResponse">
        <wsdl:part name="parameters" element="ns:modificaPropostaResponse"/>
    </wsdl:message>
    <wsdl:portType name="InterfacciaRichiestaCasaPortType">
        <wsdl:operation name="ottieniRichiesteGestite">
            <wsdl:input message="ns:ottieniRichiesteGestiteRequest"
                wsaw:Action="urn:ottieniRichiesteGestite"/>
            <wsdl:output message="ns:ottieniRichiesteGestiteResponse"
                wsaw:Action="urn:ottieniRichiesteGestiteResponse"/>
        </wsdl:operation>
        <wsdl:operation name="ottieniRichiesta">
            <wsdl:input message="ns:ottieniRichiestaRequest"
                wsaw:Action="urn:ottieniRichiesta"/>
            <wsdl:output message="ns:ottieniRichiestaResponse"
                wsaw:Action="urn:ottieniRichiestaResponse"/>
        </wsdl:operation>
        <wsdl:operation name="ottieniRichiesteNonGestite">
            <wsdl:input message="ns:ottieniRichiesteNonGestiteRequest"
                wsaw:Action="urn:ottieniRichiesteNonGestite"/>
            <wsdl:output message="ns:ottieniRichiesteNonGestiteResponse"
                wsaw:Action="urn:ottieniRichiesteNonGestiteResponse"/>
        </wsdl:operation>
        <wsdl:operation name="creaProposta">
            <wsdl:input message="ns:creaPropostaRequest"
                wsaw:Action="urn:creaProposta"/>
            <wsdl:output message="ns:creaPropostaResponse"
                wsaw:Action="urn:creaPropostaResponse"/>
        </wsdl:operation>
        <wsdl:operation name="gestisciRichiesta">
            <wsdl:input message="ns:gestisciRichiestaRequest"
                wsaw:Action="urn:gestisciRichiesta"/>
            <wsdl:output message="ns:gestisciRichiestaResponse"
                wsaw:Action="urn:gestisciRichiestaResponse"/>
        </wsdl:operation>
        <wsdl:operation name="ottieniProposta">
            <wsdl:input message="ns:ottieniPropostaRequest"
                wsaw:Action="urn:ottieniProposta"/>
            <wsdl:output message="ns:ottieniPropostaResponse"
                wsaw:Action="urn:ottieniPropostaResponse"/>
        </wsdl:operation>
        <wsdl:operation name="modificaProposta">
            <wsdl:input message="ns:modificaPropostaRequest"

```

```

        wsaw:Action="urn:modificaProposta"/>
        <wsdl:output message="ns:modificaPropostaResponse"
        wsaw:Action="urn:modificaPropostaResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="InterfacciaRichiestaCasaSoap11Binding"
type="ns:InterfacciaRichiestaCasaPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="ottieniRichiesteGestite">
        <soap:operation soapAction="urn:ottieniRichiesteGestite" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesta">
        <soap:operation soapAction="urn:ottieniRichiesta" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesteNonGestite">
        <soap:operation soapAction="urn:ottieniRichiesteNonGestite" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="creaProposta">
        <soap:operation soapAction="urn:creaProposta" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="gestisciRichiesta">
        <soap:operation soapAction="urn:gestisciRichiesta" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

</wsdl:operation>
<wsdl:operation name="ottieniProposta">
    <soap:operation soapAction="urn:ottieniProposta" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="modificaProposta">
    <soap:operation soapAction="urn:modificaProposta" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterfacciaRichiestaCasaSoap12Binding"
type="ns:InterfacciaRichiestaCasaPortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="ottieniRichiesteGestite">
        <soap12:operation soapAction="urn:ottieniRichiesteGestite" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesta">
        <soap12:operation soapAction="urn:ottieniRichiesta" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesteNonGestite">
        <soap12:operation soapAction="urn:ottieniRichiesteNonGestite"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="creaProposta">

```

```

<soap12:operation soapAction="urn:creaProposta" style="document"/>
<wsdl:input>
    <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
    <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="gestisciRichiesta">
    <soap12:operation soapAction="urn:gestisciRichiesta" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="ottieniProposta">
    <soap12:operation soapAction="urn:ottieniProposta" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="modificaProposta">
    <soap12:operation soapAction="urn:modificaProposta" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterfacciaRichiestaCasaHttpBinding"
type="ns:InterfacciaRichiestaCasaPortType">
    <http:binding verb="POST"/>
    <wsdl:operation name="ottieniRichiesteGestite">
        <http:operation location="ottieniRichiesteGestite"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesta">
        <http:operation location="ottieniRichiesta"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>

```

```

        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRichiesteNonGestite">
        <http:operation location="ottieniRichiesteNonGestite"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="creaProposta">
        <http:operation location="creaProposta"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="gestisciRichiesta">
        <http:operation location="gestisciRichiesta"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniProposta">
        <http:operation location="ottieniProposta"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="modificaProposta">
        <http:operation location="modificaProposta"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="InterfacciaRichiestaCasa">
```

```

<wsdl:port name="InterfacciaRichiestaCasaHttpSoap11Endpoint"
binding="ns:InterfacciaRichiestaCasaSoap11Binding">
    <soap:address
location="http://localhost:8080/axis2/services/InterfacciaRichiestaCasa.InterfacciaRichiestaCasaHttpSoap11Endpoint"/>
</wsdl:port>
<wsdl:port name="InterfacciaRichiestaCasaHttpSoap12Endpoint"
binding="ns:InterfacciaRichiestaCasaSoap12Binding">
    <soap12:address
location="http://localhost:8080/axis2/services/InterfacciaRichiestaCasa.InterfacciaRichiestaCasaHttpSoap12Endpoint"/>
</wsdl:port>
<wsdl:port name="InterfacciaRichiestaCasaHttpEndpoint"
binding="ns:InterfacciaRichiestaCasaHttpBinding">
    <http:address
location="http://localhost:8080/axis2/services/InterfacciaRichiestaCasa.InterfacciaRichiestaCasaHttpEndpoint"/>
</wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

8.6.3 WSDL Attrazioni

```

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://ws.apache.org/axis2"
xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
xmlns:http="http://schemas.xmlsoap.org/wsdl/http/"
xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.apache.org/axis2">
    <wsdl:documentation>InterfacciaAttrazioni</wsdl:documentation>
    <wsdl:types>
        <xsschema attributeFormDefault="qualified" elementFormDefault="qualified"
targetNamespace="http://ws.apache.org/axis2">
            <xss:element name="visualizzaListaPropostaAttivita">
                <xss:complexType>
                    <xss:sequence>
                        <xss:element minOccurs="0" name="args0" nillable="true"
type="xs:string"/>
                    </xss:sequence>
                </xss:complexType>
            </xss:element>
            <xss:element name="visualizzaListaPropostaAttivitaResponse">
                <xss:complexType>
                    <xss:sequence>
                        <xss:element minOccurs="0" name="return" nillable="true"
type="xs:string"/>
                    </xss:sequence>
                </xss:complexType>
            </xss:element>
            <xss:element name="suggerisciListaPropostaAttivita">

```

```

<xs:complexType>
    <xs:sequence>
        <xs:element minOccurs="0" name="args0" nillable="true"
            type="xs:string"/>
        <xs:element minOccurs="0" name="args1" nillable="true"
            type="xs:string"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="suggerisciListaPropostaAttivitaResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
                type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ottieniListaAttrazioni">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="args0" nillable="true"
                type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="ottieniListaAttrazioniResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
                type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="aggiungiAlternative">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="args0" nillable="true"
                type="xs:string"/>
            <xs:element maxOccurs="unbounded" minOccurs="0"
                name="args1" type="xs:boolean"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="aggiungiAlternativeResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
                type="xs:string"/>
        </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rimuoviAlternative">

```

```

<xs:complexType>
    <xs:sequence>
        <xs:element minOccurs="0" name="args0" nillable="true"
        type="xs:string"/>
        <xs:element maxOccurs="unbounded" minOccurs="0"
        name="args1" type="xs:boolean"/>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="rimuoviAlternativeResponse">
    <xs:complexType>
        <xs:sequence>
            <xs:element minOccurs="0" name="return" nillable="true"
            type="xs:string"/>
        </xs:sequence>
    </xs:complexType>
</xs:element>
</xs:schema>
</wsdl:types>
<wsdl:message name="suggerisciListaPropostaAttivitaRequest">
    <wsdl:part name="parameters" element="ns:suggerisciListaPropostaAttivita"/>
</wsdl:message>
<wsdl:message name="suggerisciListaPropostaAttivitaResponse">
    <wsdl:part name="parameters" element="ns:suggerisciListaPropostaAttivitaResponse"/>
</wsdl:message>
<wsdl:message name="rimuoviAlternativeRequest">
    <wsdl:part name="parameters" element="ns:rimuoviAlternative"/>
</wsdl:message>
<wsdl:message name="rimuoviAlternativeResponse">
    <wsdl:part name="parameters" element="ns:rimuoviAlternativeResponse"/>
</wsdl:message>
<wsdl:message name="ottieniListaAttrazioniRequest">
    <wsdl:part name="parameters" element="ns:ottieniListaAttrazioni"/>
</wsdl:message>
<wsdl:message name="ottieniListaAttrazioniResponse">
    <wsdl:part name="parameters" element="ns:ottieniListaAttrazioniResponse"/>
</wsdl:message>
<wsdl:message name="visualizzaListaPropostaAttivitaRequest">
    <wsdl:part name="parameters" element="ns:visualizzaListaPropostaAttivita"/>
</wsdl:message>
<wsdl:message name="visualizzaListaPropostaAttivitaResponse">
    <wsdl:part name="parameters" element="ns:visualizzaListaPropostaAttivitaResponse"/>
</wsdl:message>
<wsdl:message name="aggiungiAlternativeRequest">
    <wsdl:part name="parameters" element="ns:aggiungiAlternative"/>
</wsdl:message>
<wsdl:message name="aggiungiAlternativeResponse">
    <wsdl:part name="parameters" element="ns:aggiungiAlternativeResponse"/>
</wsdl:message>
<wsdl:portType name="InterfacciaAttrazioniPortType">
    <wsdl:operation name="suggerisciListaPropostaAttivita">
        <wsdl:input message="ns:suggerisciListaPropostaAttivitaRequest">

```

```

        wsaw:Action="urn:suggerisciListaPropostaAttivita"/>
        <wsdl:output message="ns:suggerisciListaPropostaAttivitaResponse"
        wsaw:Action="urn:suggerisciListaPropostaAttivitaResponse"/>
    </wsdl:operation>
    <wsdl:operation name="rimuoviAlternative">
        <wsdl:input message="ns:rimuoviAlternativeRequest"
        wsaw:Action="urn:rimuoviAlternative"/>
        <wsdl:output message="ns:rimuoviAlternativeResponse"
        wsaw:Action="urn:rimuoviAlternativeResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ottieniListaAttrazioni">
        <wsdl:input message="ns:ottieniListaAttrazioniRequest"
        wsaw:Action="urn:ottieniListaAttrazioni"/>
        <wsdl:output message="ns:ottieniListaAttrazioniResponse"
        wsaw:Action="urn:ottieniListaAttrazioniResponse"/>
    </wsdl:operation>
    <wsdl:operation name="visualizzaListaPropostaAttivita">
        <wsdl:input message="ns:visualizzaListaPropostaAttivitaRequest"
        wsaw:Action="urn:visualizzaListaPropostaAttivita"/>
        <wsdl:output message="ns:visualizzaListaPropostaAttivitaResponse"
        wsaw:Action="urn:visualizzaListaPropostaAttivitaResponse"/>
    </wsdl:operation>
    <wsdl:operation name="aggiungiAlternative">
        <wsdl:input message="ns:aggiungiAlternativeRequest"
        wsaw:Action="urn:aggiungiAlternative"/>
        <wsdl:output message="ns:aggiungiAlternativeResponse"
        wsaw:Action="urn:aggiungiAlternativeResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="InterfacciaAttrazioniSoap11Binding"
type="ns:InterfacciaAttrazioniPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="suggerisciListaPropostaAttivita">
        <soap:operation soapAction="urn:suggerisciListaPropostaAttivita"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="rimuoviAlternative">
        <soap:operation soapAction="urn:rimuoviAlternative" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniListaAttrazioni">

```

```

<soap:operation soapAction="urn:ottieneListaAttrazioni" style="document"/>
<wsdl:input>
    <soap:body use="literal"/>
</wsdl:input>
<wsdl:output>
    <soap:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="visualizzaListaPropostaAttivita">
    <soap:operation soapAction="urn:visualizzaListaPropostaAttivita"
style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="aggiungiAlternative">
    <soap:operation soapAction="urn:aggiungiAlternative" style="document"/>
    <wsdl:input>
        <soap:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterfacciaAttrazioniSoap12Binding"
type="ns:InterfacciaAttrazioniPortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="suggerisciListaPropostaAttivita">
        <soap12:operation soapAction="urn:suggerisciListaPropostaAttivita"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="rimuoviAlternative">
        <soap12:operation soapAction="urn:rimuoviAlternative"
style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieneListaAttrazioni">

```

```

<soap12:operation soapAction="urn:ottieniListaAttrazioni" style="document"/>
<wsdl:input>
    <soap12:body use="literal"/>
</wsdl:input>
<wsdl:output>
    <soap12:body use="literal"/>
</wsdl:output>
</wsdl:operation>
<wsdl:operation name="visualizzaListaPropostaAttivita">
    <soap12:operation soapAction="urn:visualizzaListaPropostaAttivita"
style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
<wsdl:operation name="aggiungiAlternative">
    <soap12:operation soapAction="urn:aggiungiAlternative" style="document"/>
    <wsdl:input>
        <soap12:body use="literal"/>
    </wsdl:input>
    <wsdl:output>
        <soap12:body use="literal"/>
    </wsdl:output>
</wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterfacciaAttrazioniHttpBinding" type="ns:InterfacciaAttrazioniPortType">
    <http:binding verb="POST"/>
    <wsdl:operation name="suggerisciListaPropostaAttivita">
        <http:operation location="suggerisciListaPropostaAttivita"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="rimuoviAlternative">
        <http:operation location="rimuoviAlternative"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniListaAttrazioni">
        <http:operation location="ottieniListaAttrazioni"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>

```

```

        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="visualizzaListaPropostaAttivita">
        <http:operation location="visualizzaListaPropostaAttivita"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="aggiungiAlternative">
        <http:operation location="aggiungiAlternative"/>
        <wsdl:input>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:input>
        <wsdl:output>
            <mime:content type="application/xml" part="parameters"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:service name="InterfacciaAttrazioni">
    <wsdl:port name="InterfacciaAttrazioniHttpSoap11Endpoint"
binding="ns:InterfacciaAttrazioniSoap11Binding">
        <soap:address
            location="http://localhost:8080/axis2/services/InterfacciaAttrazioni.InterfacciaAttrazioniHttpSoap11Endpoint"/>
    </wsdl:port>
    <wsdl:port name="InterfacciaAttrazioniHttpSoap12Endpoint"
binding="ns:InterfacciaAttrazioniSoap12Binding">
        <soap12:address
            location="http://localhost:8080/axis2/services/InterfacciaAttrazioni.InterfacciaAttrazioniHttpSoap12Endpoint"/>
    </wsdl:port>
    <wsdl:port name="InterfacciaAttrazioniHttpEndpoint"
binding="ns:InterfacciaAttrazioniHttpBinding">
        <http:address
            location="http://localhost:8080/axis2/services/InterfacciaAttrazioni.InterfacciaAttrazioniHttpEndpoint"/>
    </wsdl:port>
</wsdl:service>
</wsdl:definitions>

```

8.6.4 WSDL Servizi accessori

```

<wsdl:definitions xmlns:wsdl="http://schemas.xmlsoap.org/wsdl/"
    xmlns:ns1="http://org.apache.axis2/xsd" xmlns:ns="http://ws.apache.org/axis2"
    xmlns:wsaw="http://www.w3.org/2006/05/addressing/wsdl"
    xmlns:http="http://schemas.xmlsoap.org/wsdl/http/">

```

```

xmlns:xs="http://www.w3.org/2001/XMLSchema"
xmlns:mime="http://schemas.xmlsoap.org/wsdl/mime/"
xmlns:soap="http://schemas.xmlsoap.org/wsdl/soap/"
xmlns:soap12="http://schemas.xmlsoap.org/wsdl/soap12/"
targetNamespace="http://ws.apache.org/axis2"
    <wsdl:documentation>InterfacciaServiziAccessori</wsdl:documentation>
    <wsdl:types>
        <xs:schema attributeFormDefault="qualified" elementFormDefault="qualified"
            targetNamespace="http://ws.apache.org/axis2">
            <xs:element name="creaPianificazione">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="args0" nillable="true"
                            type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="creaPianificazioneResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
                            type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="ottieniRiferimentoServiziAccessori">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="args0" nillable="true"
                            type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
            <xs:element name="ottieniRiferimentoServiziAccessoriResponse">
                <xs:complexType>
                    <xs:sequence>
                        <xs:element minOccurs="0" name="return" nillable="true"
                            type="xs:string"/>
                    </xs:sequence>
                </xs:complexType>
            </xs:element>
        </xs:schema>
    </wsdl:types>
    <wsdl:message name="creaPianificazioneRequest">
        <wsdl:part name="parameters" element="ns:creaPianificazione"/>
    </wsdl:message>
    <wsdl:message name="creaPianificazioneResponse">
        <wsdl:part name="parameters" element="ns:creaPianificazioneResponse"/>
    </wsdl:message>
    <wsdl:message name="ottieniRiferimentoServiziAccessoriRequest">
        <wsdl:part name="parameters" element="ns:ottieniRiferimentoServiziAccessori"/>
    </wsdl:message>

```

```

<wsdl:message name="ottieniRiferimentoServiziAccessoriResponse">
    <wsdl:part name="parameters"
element="ns:ottieniRiferimentoServiziAccessoriResponse"/>
</wsdl:message>
<wsdl:portType name="InterfacciaServiziAccessoriPortType">
    <wsdl:operation name="creaPianificazione">
        <wsdl:input message="ns:creaPianificazioneRequest"
wsaw:Action="urn:creaPianificazione"/>
        <wsdl:output message="ns:creaPianificazioneResponse"
wsaw:Action="urn:creaPianificazioneResponse"/>
    </wsdl:operation>
    <wsdl:operation name="ottieniRiferimentoServiziAccessori">
        <wsdl:input message="ns:ottieniRiferimentoServiziAccessoriRequest"
wsaw:Action="urn:ottieniRiferimentoServiziAccessori"/>
        <wsdl:output message="ns:ottieniRiferimentoServiziAccessoriResponse"
wsaw:Action="urn:ottieniRiferimentoServiziAccessoriResponse"/>
    </wsdl:operation>
</wsdl:portType>
<wsdl:binding name="InterfacciaServiziAccessoriSoap11Binding"
type="ns:InterfacciaServiziAccessoriPortType">
    <soap:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="creaPianificazione">
        <soap:operation soapAction="urn:creaPianificazione" style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
    <wsdl:operation name="ottieniRiferimentoServiziAccessori">
        <soap:operation soapAction="urn:ottieniRiferimentoServiziAccessori"
style="document"/>
        <wsdl:input>
            <soap:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>
</wsdl:binding>
<wsdl:binding name="InterfacciaServiziAccessoriSoap12Binding"
type="ns:InterfacciaServiziAccessoriPortType">
    <soap12:binding transport="http://schemas.xmlsoap.org/soap/http" style="document"/>
    <wsdl:operation name="creaPianificazione">
        <soap12:operation soapAction="urn:creaPianificazione" style="document"/>
        <wsdl:input>
            <soap12:body use="literal"/>
        </wsdl:input>
        <wsdl:output>
            <soap12:body use="literal"/>
        </wsdl:output>
    </wsdl:operation>

```

```

        </wsdl:operation>
        <wsdl:operation name="ottieneRiferimentoServiziAccessori">
            <soap12:operation soapAction="urn:ottieneRiferimentoServiziAccessori"
style="document"/>
            <wsdl:input>
                <soap12:body use="literal"/>
            </wsdl:input>
            <wsdl:output>
                <soap12:body use="literal"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:binding name="InterfacciaServiziAccessoriHttpBinding"
type="ns:InterfacciaServiziAccessoriPortType">
        <http:binding verb="POST"/>
        <wsdl:operation name="creaPianificazione">
            <http:operation location="creaPianificazione"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
        <wsdl:operation name="ottieneRiferimentoServiziAccessori">
            <http:operation location="ottieneRiferimentoServiziAccessori"/>
            <wsdl:input>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:input>
            <wsdl:output>
                <mime:content type="application/xml" part="parameters"/>
            </wsdl:output>
        </wsdl:operation>
    </wsdl:binding>
    <wsdl:service name="InterfacciaServiziAccessori">
        <wsdl:port name="InterfacciaServiziAccessoriHttpSoap11Endpoint"
binding="ns:InterfacciaServiziAccessoriSoap11Binding">
            <soap:address
location="http://localhost:8080/axis2/services/InterfacciaServiziAccessori.InterfacciaServizi
AccessoriHttpSoap11Endpoint/">
        </wsdl:port>
        <wsdl:port name="InterfacciaServiziAccessoriHttpSoap12Endpoint"
binding="ns:InterfacciaServiziAccessoriSoap12Binding">
            <soap12:address
location="http://localhost:8080/axis2/services/InterfacciaServiziAccessori.InterfacciaServizi
AccessoriHttpSoap12Endpoint/">
        </wsdl:port>
        <wsdl:port name="InterfacciaServiziAccessoriHttpEndpoint"
binding="ns:InterfacciaServiziAccessoriHttpBinding">
            <http:address
location="http://localhost:8080/axis2/services/InterfacciaServiziAccessori.InterfacciaServizi
AccessoriHttpEndpoint/">
        </wsdl:port>
    </wsdl:service>

```

```

        </wsdl:port>
    </wsdl:service>
</wsdl:definitions>
```

8.7 Messaggi XML Soap generati per ogni caso di test

8.7.1 Offerta casa

```

<--! richiesta pubblicaPropostaScambio -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
        <axis:pubblicaPropostaScambio>
            <axis:args0>Questo è un oggetto Offerta</axis:args0>
            <axis:args1>Questo è un oggetto ContattiRichiedente</axis:args1>
        </axis:pubblicaPropostaScambio>
    </soapenv:Body>
</soapenv:Envelope>

<--! risposta pubblicaPropostaScambio -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <ns:pubblicaPropostaScambioResponse xmlns:ns="http://ws.apache.org/axis2">
            <ns:return>La proposta e' stata effettuata con successo.</ns:return>
        </ns:pubblicaPropostaScambioResponse>
    </soapenv:Body>
</soapenv:Envelope>

<--!richiesta ricercaOfferte -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
        <axis:ricercaOfferte>
            <axis:args0>Questo è l'indirizzo</axis:args0>
            <axis:args1>Questo è una limitazione</axis:args1>
            <axis:args2>Questa è un oggetto di classe Trasporto</axis:args2>
            <axis:args3>Questa è la descrizione della composizione della casa</axis:args3>
        </axis:ricercaOfferte>
    </soapenv:Body>
</soapenv:Envelope>

<--! risposta ricercaOfferte-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
    <soapenv:Body>
        <ns:ricercaOfferteResponse xmlns:ns="http://ws.apache.org/axis2">
            <ns:return>Lista di offerte pertinenti con i parametri di ricerca.</ns:return>
        </ns:ricercaOfferteResponse>
    </soapenv:Body>
</soapenv:Envelope>
```

8.7.2 Richiesta casa

```
<!-- richiesta creaProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:axis="http://ws.apache.org/axis2">
  <soapenv:Header/>
  <soapenv:Body>
    <axis:creaProposta>
      <axis:args0>3gg89e</axis:args0>
      <axis:args1>indirizzo</axis:args1>
      <axis:args2>2017-04-09</axis:args2>
      <axis:args3>2018-01-10</axis:args3>
    </axis:creaProposta>
  </soapenv:Body>
</soapenv:Envelope>

<!-- risposta creaProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:creaPropostaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Nuova proposta</ns:return>
    </ns:creaPropostaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta gestisciProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:axis="http://ws.apache.org/axis2">
  <soapenv:Header/>
  <soapenv:Body>
    <axis:gestisciRichiesta>
      <axis:args0>Proposta</axis:args0>
    </axis:gestisciRichiesta>
  </soapenv:Body>
</soapenv:Envelope>

<!-- risposta gestisciProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:gestisciRichiestaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>true</ns:return>
    </ns:gestisciRichiestaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta modificaProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/" xmlns:axis="http://ws.apache.org/axis2">
  <soapenv:Header/>
  <soapenv:Body>
    <axis:modificaProposta>
      <axis:args0>Proposta</axis:args0>
```

```

</axis:modificaProposta>
</soapenv:Body>
</soapenv:Envelope>

<!-- risposta modificaProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:modificaPropostaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Proposta modificata</ns:return>
    </ns:modificaPropostaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta ottieniProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:ottieniProposta>
        <axis:args0>3gg57i</axis:args0>
        <axis:args1>indirizzo</axis:args1>
        <axis:args2>2017-10-01</axis:args2>
        <axis:args3>2018-03-02</axis:args3>
      </axis:ottieniProposta>
    </soapenv:Body>
  </soapenv:Envelope>

<!-- risposta ottieniProposta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:ottieniPropostaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Proposta</ns:return>
    </ns:ottieniPropostaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta ottieniRichiesta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:ottieniRichiesta>
        <axis:args0>56rh83</axis:args0>
        <axis:args1>Stoccolma</axis:args1>
        <axis:args2>Svezia</axis:args2>
        <axis:args3>2017-09-23</axis:args3>
        <axis:args4>2018-11-02</axis:args4>
      </axis:ottieniRichiesta>
    </soapenv:Body>
  </soapenv:Envelope>

```

```

<!-- risposta ottieniRichiesta -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:ottieniRichiestaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Richiesta</ns:return>
    </ns:ottieniRichiestaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta ottieniRichiesteGestite -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:ottieniRichiesteGestite/>
    </soapenv:Body>
  </soapenv:Envelope>

<!-- risposta ottieniRichiesteGestite -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:ottieniRichiesteGestiteResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Richieste gestite dall'agente</ns:return>
    </ns:ottieniRichiesteGestiteResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta ottieniRichiesteNonGestite -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:ottieniRichiesteNonGestite/>
    </soapenv:Body>
  </soapenv:Envelope>

<!-- risposta ottieniRichiesteNonGestite -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:ottieniRichiesteNonGestiteResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Richieste non gestite</ns:return>
    </ns:ottieniRichiesteNonGestiteResponse>
  </soapenv:Body>
</soapenv:Envelope>

```

8.7.3 Attrazioni

```

<!-- richiesta ottieniListaAttrazioni -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>

```

```

<soapenv:Body>
  <axis:ottieniListaAttrazioni>
    <axis:args0>Indirizzo offerta</axis:args0>
  </axis:ottieniListaAttrazioni>
</soapenv:Body>
</soapenv:Envelope>

<!-- risposta ottieniListaAttrazioni -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:ottieniListaAttrazioniResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Lista di attrazioni ordinata per distanza</ns:return>
    </ns:ottieniListaAttrazioniResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta suggerisciListaPropostaAttivita-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:suggerisciListaPropostaAttivita>
        <axis:args0>Informazioni soggiorno</axis:args0>
        <axis:args1>Classi attrazioni</axis:args1>
      </axis:suggerisciListaPropostaAttivita>
    </soapenv:Body>
  </soapenv:Envelope>

<!-- risposta suggerisciListaPropostaAttivita-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:suggerisciListaPropostaAttivitaResponse xmlns:ns="http://ws.apache.org/axis2">
      <ns:return>Riferimento alla lista proposta attivita generata</ns:return>
    </ns:suggerisciListaPropostaAttivitaResponse>
  </soapenv:Body>
</soapenv:Envelope>

<!-- richiesta visualizzaListaPropostaAttivita-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <ns:axis="http://ws.apache.org/axis2">
    <soapenv:Header/>
    <soapenv:Body>
      <axis:visualizzaListaPropostaAttivita>
        <axis:args0>AKL32D11</axis:args0>
      </axis:visualizzaListaPropostaAttivita>
    </soapenv:Body>
  </soapenv:Envelope>

<!-- risposta visualizzaListaPropostaAttivita -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
  <soapenv:Body>
    <ns:visualizzaListaPropostaAttivitaResponse xmlns:ns="http://ws.apache.org/axis2">

```

```

<ns:return>Lista proposta attivita</ns:return>
</ns:visualizzaListaPropostaAttivitaResponse>
</soapenv:Body>
</soapenv:Envelope>

<!-- richiesta aggiungiAlternative-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<axis:aggiungiAlternative>
<axis:args0>Lista proposta attivita</axis:args0>
<!--Zero or more repetitions:-->
<axis:args1>false</axis:args1>
<axis:args1>false</axis:args1>
    <axis:args1>true</axis:args1>
    <axis:args1>false</axis:args1>
</axis:aggiungiAlternative>
</soapenv:Body>
</soapenv:Envelope>

<!-- risposta aggiungiAlternative -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<soapenv:Body>
<ns:aggiungiAlternativeResponse xmlns:ns="http://ws.apache.org/axis2">
<ns:return>Lista proposta attivita modificata</ns:return>
</ns:aggiungiAlternativeResponse>
</soapenv:Body>
</soapenv:Envelope>

<!-- richiesta rimuoviAlternative -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<axis:rimuoviAlternative>
<axis:args0>Lista proposta attivita</axis:args0>
<!--Zero or more repetitions:-->
<axis:args1>true</axis:args1>
<axis:args1>false</axis:args1>
<axis:args1>false</axis:args1>
<axis:args1>false</axis:args1>
<axis:args1>true</axis:args1>
<axis:args1>false</axis:args1>
<axis:args1>false</axis:args1>
<axis:args1>false</axis:args1>
</axis:rimuoviAlternative>
</soapenv:Body>
</soapenv:Envelope>

<!-- risposta rimuoviAlternative -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">

```

```

<soapenv:Body>
  <ns:rimuoviAlternativeResponse xmlns:ns="http://ws.apache.org/axis2">
    <ns:return>Lista proposta attivita modificata</ns:return>
  </ns:rimuoviAlternativeResponse>
</soapenv:Body>
</soapenv:Envelope>

```

8.7.4 Servizi accessori

```

<!-- richiesta creaPianificazione -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:creaPianificazioneResponse xmlns:ns="http://ws.apache.org/axis2">
  <ns:return>pianificazione</ns:return>
</ns:creaPianificazioneResponse>
</soapenv:Envelope>

<!-- risposta creaPianificazione -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:creaPianificazioneResponse xmlns:ns="http://ws.apache.org/axis2">
  <ns:return>pianificazione</ns:return>
</ns:creaPianificazioneResponse>
</soapenv:Envelope>
</soapenv:Envelope>

<!-- richiesta ottieniRiferimentoServiziAccessori-->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:ottieniRiferimentoServiziAccessoriResponse xmlns:ns="http://ws.apache.org/axis2">
  <ns:return>riferimento servizi accessori</ns:return>
</ns:ottieniRiferimentoServiziAccessoriResponse>
</soapenv:Envelope>
</soapenv:Envelope>

<!-- risposta ottieniRiferimentoServiziAccessori -->
<soapenv:Envelope xmlns:soapenv="http://schemas.xmlsoap.org/soap/envelope/">
<ns:ottieniRiferimentoServiziAccessoriResponse xmlns:ns="http://ws.apache.org/axis2">
  <ns:return>riferimento servizi accessori</ns:return>
</ns:ottieniRiferimentoServiziAccessoriResponse>
</soapenv:Envelope>
</soapenv:Envelope>

```

Capitolo 9

Piano di test

9.1 Introduzione

9.1.1 Scopo

Lo scopo di questo documento è di raccogliere tutte le informazioni necessarie per pianificare e controllare il testing di SharingEverywhere. Il documento descrive l'approccio da seguire per testare il software e rappresenta il piano di alto livello utilizzato dai collaudatori.

9.2 Oggetto del test

Saranno testati i seguenti servizi web:

- Offerta casa
- Richiesta casa
- Attrazioni
- Servizi accessori

dopodiché verrà testata la logica di coordinamento SharingEverywhere.

9.3 Tipi e tecniche di testing

9.3.1 Test d'unità

9.3.1.1 Test black-box: Correttezza del risultato

Abbiamo deciso di testare le funzionalità dei quattro differenti applicativi con un approccio di tipo black-box.

Il seguente test studia il comportamento dei servizi definendo, dato uno specifico input, l'output desiderato e comparando il risultato con l'output effettivamente prodotto dal web service. Ogni servizio verrà testato fornendo in input sia parametri corretti che parametri errati.

| Obiettivo della tecnica | Verificare la correttezza delle singole funzionalità fornite dai servizi web |
|-------------------------|--|
| Tecnica | Caricare gli input e verificare che l'output generato sia quello desiderato |
| Oracoli | Il collaudatore |
| Strumenti richiesti | SoapUI 5.2.1, Apache 6.0 |
| Criterio di successo | Tutti i servizi sono stati provati ed i test effettuati hanno mostrato che i servizi si comportano come previsto |

9.3.1.2 Test black-box: Disponibilità dei servizi

Per ogni servizio è stata settata una disponibilità del 99,9%: questo numero è stato scelto poiché nella realtà i servizi web hanno, più o meno, un così alto valore di disponibilità. Analizzando il codice dei servizi utilizzati è possibile notare che tutti i metodi hanno una implementazione simile, quindi è ragionevole assumere che siano tutti raggiungibili in egual misura.

La disponibilità viene calcolata eseguendo un programma di testing che invoca i metodi di ogni servizio 3500 volte, suddividendo le chiamate tra i metodi di cui è composto:

| Applicativo | Numero di metodi invocabili | Numero d'invocazioni per metodo |
|-------------------|-----------------------------|---------------------------------|
| OffertaCasa | 2 | 1750 |
| Richiesta casa | 7 | 500 |
| Attrazioni | 5 | 700 |
| Servizi accessori | 2 | 1750 |

Questo programma dovrà essere eseguito in sequenza 10 volte. I dati risultanti ed il loro valor medio dovranno essere inseriti nel report del test d'unità insieme ai risultati ottenuti al paragrafo 9.3.1.1.

9.3.2 Test d'integrazione

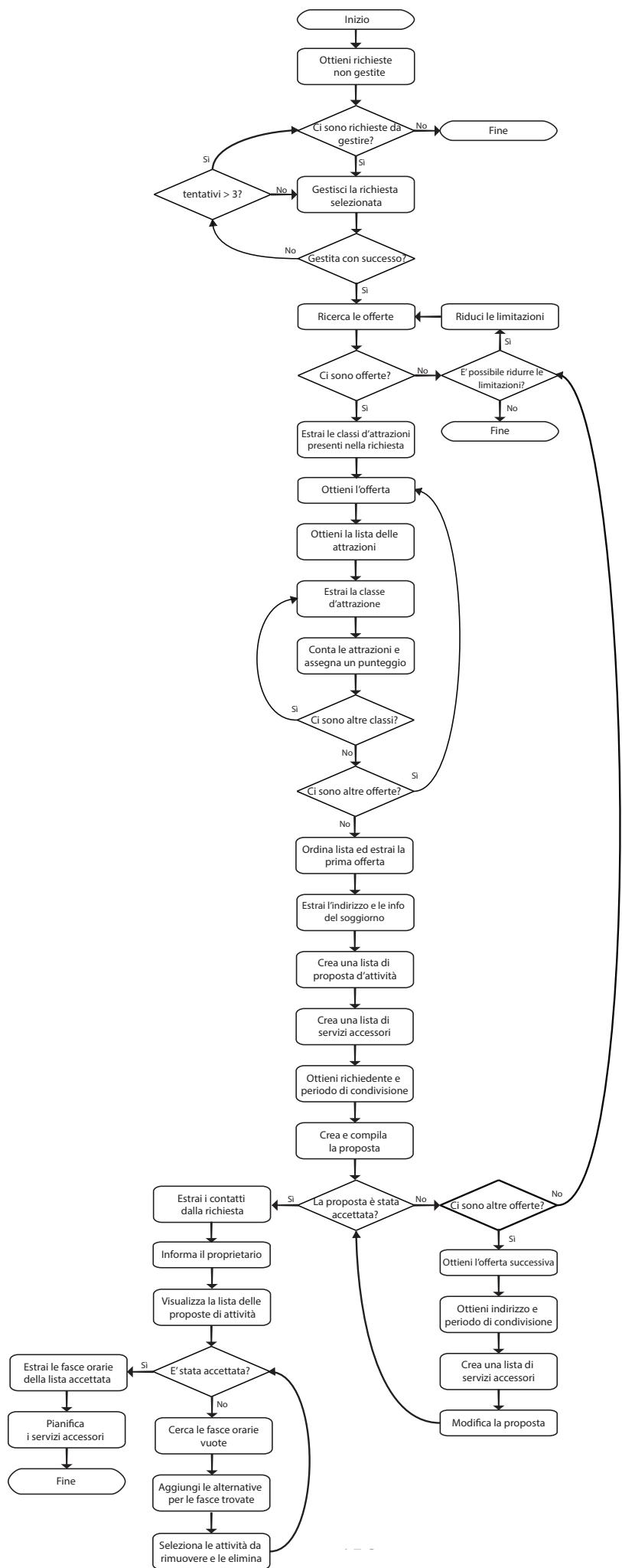
Il test d'integrazione verrà effettuato effettuando sia un test black-box che un test strutturale white-box.

9.3.2.1 Test black-box della logica: disponibilità del servizio

Il test calcola la disponibilità complessiva del servizio SharingEverywhere. Il singolo servizio si affida ad altri servizi web ognuno dei quali, come detto precedentemente, ha una disponibilità del 99,9%.

Per effettuare questo test è stato scritto un semplice algoritmo che esegue l'intero protocollo per 1000 volte. L'algoritmo viene ripetuto per 10 volte, in ogni esecuzione viene tenuto conto di quanto volte ciascun applicativo è stato causa dell'indisponibilità del servizio.

| Obiettivo della tecnica | Quantificare la disponibilità di SharingEverywhere |
|-------------------------|---|
| Tecnica | Eseguire in loop il protocollo di coordinamento |
| Oracoli | I collaudatori |
| Strumenti richiesti | Java 1.8, Apache 6.0 |
| Criterio di successo | SharingEverywhere ha una disponibilità sopra il 95% ed i singoli applicativi non causano più del 3% dell'assenza del servizio |



9.3.2.2 Test white-box della logica

Dall'analisi del codice del protocollo è possibile fare l'analisi strutturale. L'idea principale è quella di trovare degli input che coprono tutti i percorsi del grafo del flusso di controllo (CFG) riportato nella figura 9.1.

I percorsi verranno scelti dal team in maniera congiunta e riportati in un report separato, insieme ai risultati del test.

9.4 Criteri d'inizio e di fine test

9.4.1 Piano di test

Il testing inizia alla fine della fase d'implementazione ed il codice è stato revisionato da tutti i componenti del team in maniera informale.

Il testing termina quando tutto gli obiettivi in esso racchiusi sono stati raggiunti.

Il testing viene sospeso nel caso in cui vengano individuati errori critici o gravi in fase di design o quando le ore assegnate per il testing sono terminate. Il testing viene ripreso durante il ciclo successivo, dopo la fase d'implementazione.

9.5 Deliverable

I risultati dei test verranno documentati in report separati e successivamente inseriti nel report finale.

9.6 Responsabilità

| Compito | Responsabile | | Dettagli delle responsabilità |
|--------------------------------------|---|---|--|
| Stesura del piano di test | Chiara Caiazza Gionatan Gallo | | Definire le tecniche di testing e gli approcci da seguire |
| Test di unità (BB) ¹ | Chiara Caiazza Gionatan Gallo Daniele Modica Martina Troscia | Offerta casa Richiesta casa Servizi accessori Attrazioni | Eseguire i test per l'applicativo assegnatogli |
| Test d'integrazione(BB) ¹ | Chiara Caiazza Gionatan Gallo Daniele Modica Martina Troscia | | Eseguire i test della logica di coordinamento |
| Test d'integrazione(WB) ² | Chiara Caiazza Gionatan Gallo Daniele Modica Martina Troscia | | Analizzare la struttura della logica di coordinamento e verificare la correttezza dei percorsi individuati |

⁽¹⁾ BB = Black-Box

⁽²⁾ WB = White-Box

9.7 Rischi

Nella tabella seguente sono elencati i rischi che potrebbero influenzare l'esecuzione di questo piano di test. Sono indicate alcune strategie per mitigare l'occorrenza di questi rischi e piani di contingenza per contenere gli effetti avversi qualora si verifichi un evento negativo.

| Rischio | Strategia di mitigazione | Piano di contingenza |
|-------------------------------|--|---|
| Casi di test inadeguati | I collaudatori s'impegneranno per assicurare la generazione di casi di test adeguati | <ul style="list-style-type: none">• Ridefinire i dati dei test• Rivedere il piano di test e modificarlo• Riscrivere i casi di test e farli visionare ai membri del team |
| Fine del tempo per il testing | Lavorare incessantemente | <ul style="list-style-type: none">• Game over• Sperare che tutto funzioni |

Capitolo 10

Testing

10.1 Organizzazione della fase

Le attività riguardanti la fase di design sono state organizzate e pianificate tenendo conto del diagramma in Figura 10.1, il quale indica i documenti che devono essere elaborati:

- X** Emula ogni servizio come un “lancio di moneta” rispondendo a secondo del caso e delle statistiche di disponibilità di servizio.
- Y** Calcola il tasso di disponibilità del servizio orchestrato.

10.1.1 Tipi di attività

Attività individuali

Attività X richiede che ogni membro produca un codice Java in modo da avere un risultato casuale per le funzioni del loro modulo. Non richiede alcun tipo di informazione riguardo gli altri applicativi sviluppati.

Attività cooperative

Attività Y Raccoglie i dati da tutti i moduli and necessita delle informazioni di tutti i quattro applicativi.

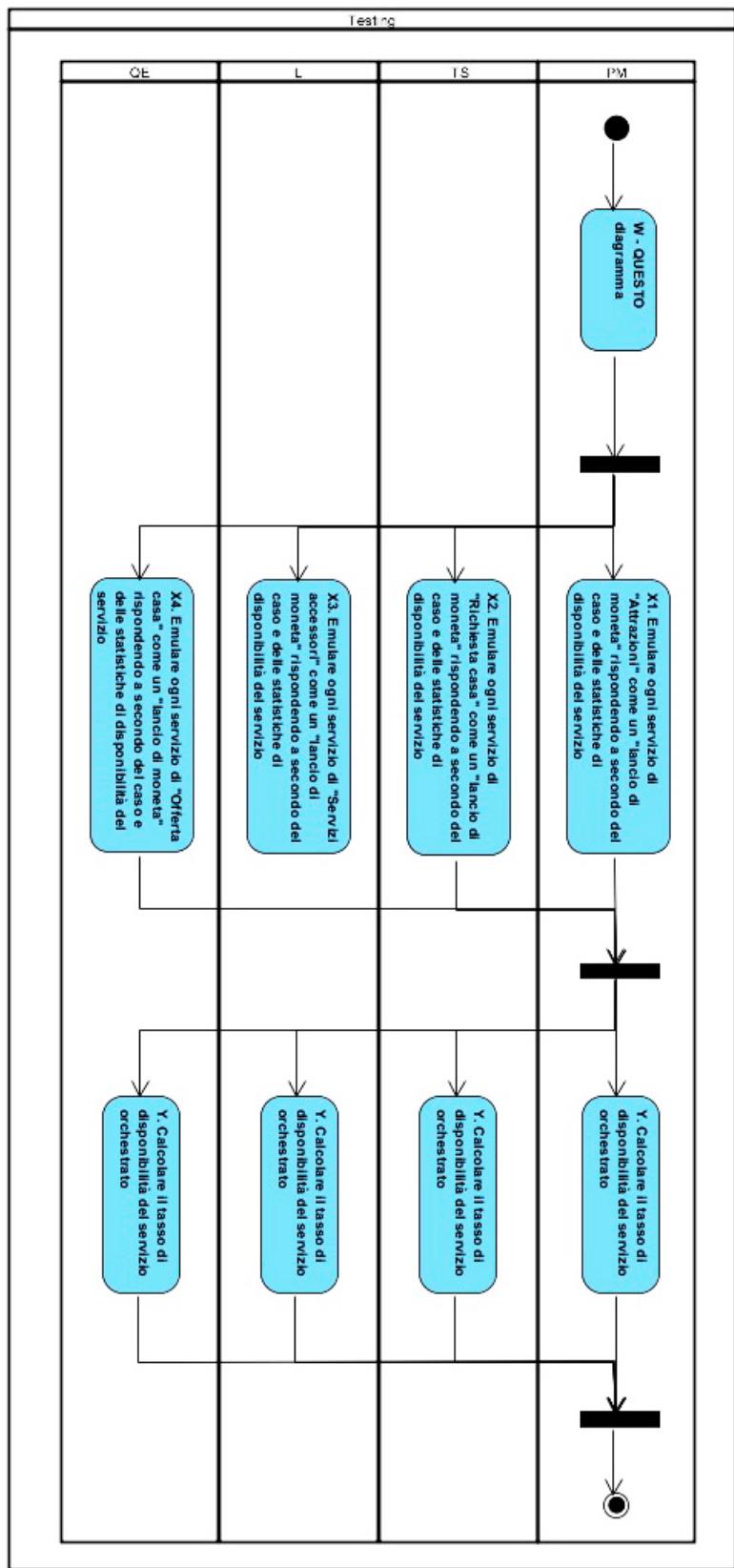


Figura 10.1: Diagramma di attività della fase di testing

10.2 Test d'unità

Questo test è stato eseguito osservando la risposta del sistema a differenti input, usando lo strumento SoapUI. Sono stati provati sia gli input corretti che quelli sbagliati e per tutti i casi il comportamento osservato è stato quello aspettato. Per ognuno dei quattro moduli è stato implementato un codice Java per l'interfaccia e uno per eseguire il test.

10.2.1 Offerta casa

InterfacciaCoinFlip

```
public class InterfacciaOffertaCasaCoinFlip {

    public String ricercaOfferte(String aIndirizzo, String aLimitazioni, String aTrasporto, String
aDescrizioneComposizioneCasa) {
        if ((Math.random())<=0.001)
            return "Servizio non disponibile";

        if (aIndirizzo.equals(""))
            return "Parametri errati: indirizzo non specificato";
        if (aLimitazioni.equals(""))
            return "Parametri errati: limitazioni non specificate";
        if (aTrasporto.equals(""))
            return "Parametri errati: informazioni di trasporto non specificate";
        if (aDescrizioneComposizioneCasa.equals(""))
            return "Parametri errati: Descrizione della casa non specificata";

        return "Lista di offerte pertinenti con i parametri di ricerca.";
    }

    public String pubblicaPropostaScambio(String aOfferta, String aContattiRichiedente) {
        if ((Math.random()) <= 0.001)
            return "Servizio non disponibile";

        if (aContattiRichiedente.equals(""))
            return "Parametri errati: contatti del richiedente errati";

        if (aOfferta.equals(""))
            return "Parametri errati: errore nell'offerta";

        return "La proposta e' stata effettuata con successo.";
    }
}
```

Test

```
import java.lang.Math;

public class Test {
    public static void main(String[] args) {
```

```

String url = "http://localhost:8080/axis2/services/InterfacciaOffertaCasaCoinFlip";

String ricercaOfferte = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ricercaOfferte
> <axis:args0>indirizzo richiesta </axis:args0> <axis:args1>limitazioni richiesta </axis:args1>
<axis:args2>trasporto richiesta </axis:args2> <axis:args3>descrizione richiesta </axis:args3>
</axis:ricercaOfferte></soapenv:Body></soapenv:Envelope>";

String pubblicaPropostaScambio = "<?xml version=\"1.0\" encoding=\"utf-
8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:pubblicaProp
ostaScambio><axis:args0>offerta</axis:args0><axis:args1>contattiRichiedente</axis:args1></axis:p
ubblicaPropostaScambio></soapenv:Body></soapenv:Envelope>";

String listaOfferte, esitoPropostaScambio;
double k = 0.00;
double totale = 0.00;
double availability = 0.00;

try {
    for (int i = 0; i<1750; i++){
        listaOfferte = SOAPClient.sendMess(url, ricercaOfferte);
        if (listaOfferte.contains("Servizio non disponibile"))
            k++;
    }

    availability = (double)(1-(k/1750))*100.0;
    System.out.println("Availability ricercaOfferte():\t" + availability + "%\n");

    totale += k;
    k = 0.00;
    for (int i = 0; i<1750; i++){
        esitoPropostaScambio = SOAPClient.sendMess(url,
pubblicaPropostaScambio);

        if (esitoPropostaScambio.contains("Servizio non disponibile"))
            k++;
    }

    availability = (double)(1-(k/1750))*100.0;
    System.out.println("Availability pubblicaPropostaScambio():\t" + availability
+ "%\n");

    totale += k;
    availability = (double)(1-(totale/3500))*100.0;
    System.out.println("Availability di OffertaCasa:\t" + availability + "%\n");
}

} catch(Exception e) {System.out.println("Eccezione\n");}
}
}

```

Report

ricercaOfferte

| Input | Output previsto | Output ottenuto |
|--|---|---|
| Indirizzo Limitazioni Trasporto descrizione della composizione della casa | Lista di offerte pertinenti con i parametri di ricerca. | Lista di offerte pertinenti con i parametri di ricerca. |
| - Limitazioni Trasporto descrizione della composizione della casa | Parametri errati: indirizzo non specificato | Parametri errati: indirizzo non specificato |
| Indirizzo - Trasporto descrizione della composizione della casa | Parametri errati: limitazioni non specificate | Parametri errati: limitazioni non specificate |
| Indirizzo Limitazioni - descrizione della composizione della casa | Parametri errati: informazioni di trasporto non specificate | Parametri errati: informazioni di trasporto non specificate |
| Indirizzo Limitazioni Trasporto - | Parametri errati: Descrizione della casa non specificata | Parametri errati: Descrizione della casa non specificata |

pubblicaPropostaScambio

| Input | Output previsto | Output ottenuto |
|-------------------------------------|---|---|
| offerta contatti del richiedente | La proposta è stata effettuata con successo. | La proposta è stata effettuata con successo. |
| - Contatti del richiedente | Parametri errati: errore nell'offerta | Parametri errati: errore nell'offerta |
| offerta - | Parametri errati: contatti del richiedente errati | Parametri errati: contatti del richiedente errati |

Disponibilità

| | Offerta casa | |
|--------------|---------------|---------------|
| | Ciclo 1 | Ciclo 2 |
| Test 1 | 99,86% | 99,97% |
| Test 2 | 99,89% | 99,91% |
| Test 3 | 99,83% | 99,97% |
| Test 4 | 99,86% | 99,94% |
| Test 5 | 99,94% | 99,89% |
| Test 6 | 99,91% | 99,86% |
| Test 7 | 99,80% | 99,91% |
| Test 8 | 99,83% | 99,83% |
| Test 9 | 99,91% | 99,94% |
| Test 10 | 99,89% | 99,91% |
| Media | 99,87% | 99,91% |

10.2.2 Richiesta casa

InterfacciaCoinFlip

```

import java.util.Date;
import java.util.regex.Pattern;
import java.lang.Math;

public class InterfacciaRichiestaCasaCoinFlip {

    public String ottieniRichiesteNonGestite() {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        return "Richieste non gestite";
    }

    public String ottieniRichiesteGestite() {
        double coin = Math.random();
        if(coin <= 0.001) return "Servizio non disponibile";

        return "Richieste gestite dall'agente";
    }

    public String ottieniRichiesta(String aRichiedenteID, String aCitta, String aStato, Date alnizio, Date
aFine) {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        if(aRichiedenteID.length() == 0) return "Richiedente non specificato";
        if(aCitta.length() == 0) return "Città non specificata";
        if(aStato.length() == 0) return "Stato non specificato";
        if(alnizio == null) return "Data d'inizio non specificata";
        if(aFine == null) return "Data di fine non specificata";
    }
}

```

```

        if(!Pattern.matches("[0-9A-Za-z]{6}", aRichiedenteID))
            return "L'ID richiedente non è di 6 caratteri alfanumerici";
        if(alnizio.after(aFine)) return "Il periodo di condivisione è errato.";
            return "Richiesta";
    }

    public String gestisciRichiesta(String aRichiesta) {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        if(aRichiesta.length() == 0) return "Richiesta non specificata";
        if(Math.random() < 0.05) return "false";
            return "true";
    }

    public String creaProposta(String aRichiedenteID, String aDestinazione, Date alnizio, Date aFine) {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        if(aRichiedenteID.length() == 0) return "Richiedente non specificato";
        if(aDestinazione.length() == 0) return "Destinazione non specificata";
        if(alnizio == null) return "Data d'inizio non specificata";
        if(aFine == null) return "Data di fine non specificata";

        if(!Pattern.matches("[0-9A-Za-z]{6}", aRichiedenteID))
            return "L'ID richiedente non è di 6 caratteri alfanumerici";
        if(alnizio.after(aFine)) return "Il periodo di condivisione è errato.";
            return "Nuova proposta";
    }

    public String ottieniProposta(String aRichiedenteID, String aDestinazione, Date alnizio, Date aFine) {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        if(aRichiedenteID.length() == 0) return "Richiedente non specificato";
        if(aDestinazione.length() == 0) return "Destinazione non specificata";
        if(alnizio == null) return "Data d'inizio non specificata";
        if(aFine == null) return "Data di fine non specificata";

        if(!Pattern.matches("[0-9A-Za-z]{6}", aRichiedenteID))
            return "L'ID richiedente non è di 6 caratteri alfanumerici";
        if(alnizio.after(aFine)) return "Il periodo di condivisione è errato.";
            return "Proposta";
    }

    public String modificaProposta(String aProposta) {
        if(Math.random() <= 0.001) return "Servizio non disponibile";

        if(aProposta.length() == 0) return "Proposta non specificata";
        return "Proposta modificata";
    }
}

```

Test

```
import java.lang.Math;
import java.util.Date;

public class Test {

    public static void main(String[] args)
    {
        String url = "http://localhost:8080/axis2/services/InterfacciaRichiestaCasaCoinFlip";

        String ottieniRichiesteGestite = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottieniRichiesteGestite/></soapenv:Body></soapenv:Envelope>";

        String ottieniRichiesteNonGestite = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottieniRichiesteNonGestite/></axis:ottieniRichiesteNonGestite></soapenv:Body></soapenv:Envelope>";

        String ottieniRichiesta = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottieniRichiesta><axis:args0>56rh83</axis:args0><axis:args1>Stoccolma</axis:args1><axis:args2>Svezia</axis:args2><axis:args3>2017-09-23</axis:args3><axis:args4>2018-11-02</axis:args4></axis:ottieniRichiesta></soapenv:Body></soapenv:Envelope>";

        String gestisciRichiesta = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:gestisciRichiesta><axis:args0>richiesta</axis:args0></axis:gestisciRichiesta></soapenv:Body></soapenv:Envelope>";

        String creaProposta = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:creaProposta><axis:args0>uj37hd</axis:args0><axis:args1>indirizzo</axis:args1><axis:args2>2017-01-01</axis:args2><axis:args3>2018-01-01</axis:args3></axis:creaProposta></soapenv:Body></soapenv:Envelope>";

        String modificaProposta = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:modificaProposta><axis:args0>proposta</axis:args0></axis:modificaProposta></soapenv:Body></soapenv:Envelope>";

        String ottieniProposta = "<?xml version=\"1.0\" encoding=\"utf-8\"?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\" xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottieniProposta><axis:args0>56rh83</axis:args0><axis:args1>indirizzo</axis:args1><axis:args2>2017-01-01</axis:args2><axis:args3>2018-01-01</axis:args3></axis:ottieniProposta></soapenv:Body></soapenv:Envelope>";
```

```

try{
    int servizioNonDisp = 0;

    String risposta;
    CharSequence noDisp = "Servizio non disponibile";
    for (int i = 0; i < 500; i++)
    {
        risposta = SOAPClient.sendMess(url, ottieniRichiesteGestite);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, ottieniRichiesteNonGestite);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, ottieniRichiesta);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, gestisciRichiesta);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, creaProposta);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, modificaProposta);
        if(risposta.contains(noDisp)) servizioNonDisp++;
        risposta = SOAPClient.sendMess(url, ottieniProposta);
        if(risposta.contains(noDisp)) servizioNonDisp++;
    }

    double disponibilita = (1.0 - ((double)servizioNonDisp / 3500.0)) * 100.0;
    System.out.println("La disponibilità complessiva è: " + disponibilita);
}catch(Exception e){System.out.println("Eccezione\n");}
}
}

```

Report

OttieniRichiesteGestite

| Input | Output atteso | Output ottenuto |
|--------------|-------------------------------|-------------------------------|
| - | Richieste gestite dall'agente | Richieste gestite dall'agente |

OttieniRichiesteNonGestite

| Input | Output atteso | Output ottenuto |
|--------------|-----------------------|------------------------|
| - | Richieste non gestite | Richieste non gestite |

GestisciRichiesta

| Input | Output atteso | Output ottenuto |
|--------------|---------------------------|---------------------------|
| - | Richiesta non specificata | Richiesta non specificata |
| Richiesta | Richiesta gestita | Richiesta gestita |

OttieniRichiesta

| Input | Output atteso | Output ottenuto |
|---|--|--|
| 3ggthd Stoccolma Svezia 2018-01-02 2018-03-14 | Richiesta | Richiesta |
| - Stoccolma Svezia 2018-01-02 2018-03-14 | Richiedente non specificato | Richiedente non specificato |
| 3ggthd - Svezia 2018-01-02 2018-03-14 | Città non specificata | Città non specificata |
| 3ggthd Stoccolma - 2018-01-02 2018-03-14 | Stato non specificato | Stato non specificato |
| 3ggthd Stoccolma Svezia - 2018-03-14 | Data d'inizio non specificata | Data d'inizio non specificata |
| 3ggthd Stoccolma Svezia 2018-01-02 - | Data di fine non specificata | Data di fine non specificata |
| 3ggthdfr54 Stoccolma Svezia 2018-01-02 2018-03-14 | L'ID richiedente non è di 6 caratteri alfanumerici | L'ID richiedente non è di 6 caratteri alfanumerici |
| 3ggthd Stoccolma Svezia 2018-01-02 14-03-2018 | Formato data invalido | invalid date format (14-03-2018) with out - s at correct place |
| 3ggthd Stoccolma Svezia 2018-03-14 2018-01-02 | Il periodo di condivisione è errato | Il periodo di condivisione è errato |

CreaProposta

| Input | Output atteso | Output ottenuto |
|---|---|---|
| d53khd indirizzo 2018-01-02 2018-02-02 | Nuova proposta | Nuova proposta |
| - indirizzo 2018-01-02 2018-02-02 | Richiedente non specificato | Richiedente non specificato |
| d53khd - 2018-01-02 2018-02-02 | Destinazione non specificata | Destinazione non specificata |
| d53khd indirizzo - 2018-02-02 | Data d'inizio non specificata | Data d'inizio non specificata |
| d53khd indirizzo 2018-01-02 - | Data di fine non specificata | Data di fine non specificata |
| gh?&_% indirizzo 2018-01-02 2018-02-02 | L'ID richiedente non è di 6 caratteri alfanumerici | L'ID richiedente non è di 6 caratteri alfanumerici |
| d53khd indirizzo 02-01-2018 02-04-2018 | Formato data invalido | invalid date format (02-01-2018) with out - s at correct place |
| d53khd indirizzo 2019-01-02 2018-02-02 | Il periodo di condivisione è errato | Il periodo di condivisione è errato |

OttieniProposta

| Input | Output atteso | Output ottenuto |
|---|-----------------------------|-----------------------------|
| d0l69d indirizzo 2019-06-02 2020-09-02 | Proposta | Proposta |
| - indirizzo 2019-06-02 2020-09-02 | Richiedente non specificato | Richiedente non specificato |

| | | |
|---|--|--|
| d0l69d - 2019-06-02 2020-09-02 | Destinazione non specificata | Destinazione non specificata |
| d0l69d indirizzo - 2020-09-02 | Data d'inizio non specificata | Data d'inizio non specificata |
| d0l69d indirizzo 2019-06-02 - | Data di fine non specificata | Data di fine non specificata |
| d09d indirizzo 2019-06-02 2020-09-02 | L'ID richiedente non è di 6 caratteri alfanumerici | L'ID richiedente non è di 6 caratteri alfanumerici |
| d0l69d indirizzo 19-06-02 20-09-02 | Formato data invalido | invalid date format (19-06-02) with out - s at correct place |
| d53khd indirizzo 2020-09-02 2019-06-02 | Il periodo di condivisione è errato | Il periodo di condivisione è errato |

ModificaProposta

| Input | Output atteso | Output ottenuto |
|----------|--------------------------|--------------------------|
| - | Proposta non specificata | Proposta non specificata |
| proposta | Proposta modificata | Proposta modificata |

Al secondo ciclo è stato ripetuto il test cambiando solamente l'output atteso in GestisciRichiesta.

GestisciRichiesta

| Input | Output atteso | Output ottenuto |
|-----------|---------------------------|---------------------------|
| - | Richiesta non specificata | Richiesta non specificata |
| richiesta | true o false | true |

Disponibilità

| | Richiesta casa | |
|--------------|----------------|---------------|
| | Ciclo 1 | Ciclo 2 |
| Test 1 | 99,86% | 99,89% |
| Test 2 | 99,97% | 99,89% |
| Test 3 | 99,83% | 99,94% |
| Test 4 | 100,00% | 99,86% |
| Test 5 | 99,80% | 99,86% |
| Test 6 | 99,91% | 99,94% |
| Test 7 | 99,91% | 99,89% |
| Test 8 | 99,94% | 99,86% |
| Test 9 | 99,89% | 99,91% |
| Test 10 | 99,91% | 99,86% |
| Media | 99,90% | 99,89% |

10.2.3 Attrazioni

InterfacciaCoinFlip

```
import java.util.regex.Pattern;
import java.util.Arrays;

public class InterfacciaAttrazioniCoinFlip {

    double disponibilita = 0.001;

    public String ottieniListaAttrazioni(String alndirizzo) {
        if (alndirizzo.equals(""))
            return "Indirizzo non specificato";
        double r = Math.random();
        if (r <= disponibilita)
            return "Servizio non disponibile";
        return "Lista di attrazioni ordinata per distanza";
    }

    public String suggerisciListaPropostaAttivita(String aInformazioni, String aClassiAttrazioni) {
        if (aInformazioni.equals(""))
            return "Informazioni soggiorno non specificate";
        if (aClassiAttrazioni.equals(""))
            return "Classi d'attrazioni non specificate";
        double r = Math.random();
        if (r <= disponibilita)
            return "Servizio non disponibile";
        return "Riferimento alla lista proposta attivita generata";
    }
}
```

```

public String visualizzaListaPropostaAttivita(String aPropostaID) {
    if (aPropostaID.equals(""))
        return "ID non specificato";
    if (!Pattern.matches("[A-Z0-9]*", aPropostaID))
        return "ID non valido";
    double r = Math.random();
    if (r <= disponibilita)
        return "Servizio non disponibile";
    return "Lista proposta attivita";
}

public String aggiungiAlternative(String aListaPropostaAttivita, boolean[] aInQualiProposteAggiungere) {
    if (aListaPropostaAttivita.equals(""))
        return "Lista proposta attivita non specificata";
    if (aInQualiProposteAggiungere.length==0)
        return "Dove aggiungere non specificato";
    double r = Math.random();
    if (r <= disponibilita)
        return "Servizio non disponibile";
    return "Lista proposta attivita modificata";
}

public String rimuoviAlternative(String aListaPropostaAttivita, boolean[] aAlternativeDaRimuovere){
    if (aListaPropostaAttivita.equals(""))
        return "Lista proposta attivita non specificata";
    if (aAlternativeDaRimuovere.length==0)
        return "Cosa rimuovere non specificato";
    double r = Math.random();
    if (r <= disponibilita)
        return "Servizio non disponibile";
    return "Lista proposta attivita modificata";
}
}

```

Test

```

class Test {

    public static void main(String[] args) {
        int conta = 0;

        String url = "http://localhost:8080/axis2/services/InterfacciaAttrazioniCoinFlip";

        String[] mess = new String[5];
        mess[0] = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottieniListaAttrazioni><axis:args0>indirizzo offerta</axis:args0>
</axis:ottieniListaAttrazioni></soapenv:Body></soapenv:Envelope>";
        mess[1] = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope\"

```

```

xmlns:axis=\"http://ws.apache.org/axis2\">><soapenv:Header/><soapenv:Body><axis:suggerisciList
aPropostaAttivita><axis:args0>Informazioni
soggiorno</axis:args0><axis:args1>Classi</axis:args1></axis:suggerisciListaPropostaAttivita></soap
env:Body></soapenv:Envelope>";
mess[2] = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope\"
xmlns:axis=\"http://ws.apache.org/axis2\">><soapenv:Header/><soapenv:Body><axis:visualizzaLista
PropostaAttivita><axis:args0>AKL32D11</axis:args0></axis:visualizzaListaPropostaAttivita></soape
nv:Body></soapenv:Envelope>";
mess[3] = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope\"
xmlns:axis=\"http://ws.apache.org/axis2\">><soapenv:Header/><soapenv:Body><axis:aggiungiAlter
native><axis:args0>Lista proposta attivita</axis:args0> <axis:args1>false</axis:args1>
<axis:args1>false</axis:args1> <axis:args1>true</axis:args1> <axis:args1>false</axis:args1>
</axis:aggiungiAlternative></soapenv:Body></soapenv:Envelope>";
mess[4] = "<?xml version=\"1.0\" encoding=\"utf-8\" ?><soapenv:Envelope
xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope\"
xmlns:axis=\"http://ws.apache.org/axis2\">><soapenv:Header/><soapenv:Body><axis:rimuoviAltern
ative> <axis:args0>Lista proposta attivita</axis:args0> <axis:args1>true</axis:args1>
<axis:args1>false</axis:args1> <axis:args1>false</axis:args1> <axis:args1>false</axis:args1>
<axis:args1>true</axis:args1> <axis:args1>false</axis:args1> <axis:args1>false</axis:args1>
<axis:args1>false</axis:args1> </axis:rimuoviAlternative></soapenv:Body></soapenv:Envelope>";

try {
    for (int i=0; i<700; i++) {
        for (String s : mess) {
            String ret = SOAPClient.sendMess(url, s);
            if (ret.contains("Servizio non disponibile"))
                conta++;
        }
    }
} catch (Exception e){}
System.out.println("Disponibilità "+(double)(3500-conta)/3500*100+"%");
}
}

```

Report

ottieniListaAttrazioni

| Input | Output atteso | Output ottenuto |
|-------------------|---|---|
| Indirizzo offerta | Lista di attrazioni ordinata per distanza | Lista di attrazioni ordinata per distanza |
| - | Indirizzo non specificato | Indirizzo non specificato |

suggerisciListaPropostaAttivita

| Input | Output atteso | Output ottenuto |
|----------------------------------|---|---|
| Informazioni soggiorno Classi | Riferimento alla lista proposta attivita generata | Riferimento alla lista proposta attivita generata |
| - Classi | Informazioni soggiorno non specificate | Informazioni soggiorno non specificate |
| Informazioni soggiorno - | Classi d'attrazioni non specificate | Classi d'attrazioni non specificate |

visualizzaListaPropostaAttivita

| Input | Output atteso | Output ottenuto |
|--------------|-------------------------|-------------------------|
| AKL32D11 | Lista proposta attivita | Lista proposta attivita |
| - | ID non specificato | ID non specificato |
| P ??234 | ID non valido | ID non valido |

aggiungiAlternative

| Input | Output atteso | Output ottenuto |
|--|---|---|
| Lista proposta attivita <false, false, true, false> | Lista proposta attivita modificata | Lista proposta attivita modificata |
| - <false, false, true, false> | Lista proposta attivita non specificata | Lista proposta attivita non specificata |
| Lista proposta attivita - | Dove aggiungere non specificato | Dove aggiungere non specificato |
| Lista proposta attivita s | Tipo non booleano | invalid string -s for boolean value |

rimuoviAlternative

| Input | Output atteso | Output ottenuto |
|--|---|---|
| Lista proposta attivita <false, false, true, false> | Lista proposta attivita modificata | Lista proposta attivita modificata |
| - <false, false, true, false> | Lista proposta attivita non specificata | Lista proposta attivita non specificata |
| Lista proposta attivita - | Cosa rimuovere non specificato | Cosa rimuovere non specificato |
| Lista proposta attivita s | Tipo non booleano | invalid string -s for boolean value |

Disponibilità

| Attrazioni | | |
|--------------|---------------|---------------|
| | Ciclo 1 | Ciclo 2 |
| Test 1 | 99,91% | 99,91% |
| Test 2 | 99,86% | 99,97% |
| Test 3 | 99,97% | 99,86% |
| Test 4 | 99,94% | 99,91% |
| Test 5 | 99,97% | 99,83% |
| Test 6 | 99,94% | 99,89% |
| Test 7 | 100,00% | 99,94% |
| Test 8 | 99,83% | 99,89% |
| Test 9 | 99,86% | 99,86% |
| Test 10 | 99,86% | 99,91% |
| Media | 99,91% | 99,90% |

10.2.4 Servizi accessori

InterfacciaCoinFlip

```

import java.lang.Math;

public class InterfacciaServiziAccessoriCoinFlip {
    double disponibilita = 0.001;                                //disponibilità 99.9%

    public String ottieniRiferimentoServiziAccessori(String aIndirizzo) {

        if (aIndirizzo.equals(""))
            return "Indirizzo sbagliato";
        double r = Math.random();
        if( r <= disponibilita)
            return "Servizio non disponibile";

        return "Riferimento ai servizi accessori";
    }

    public String creaPianificazione(String aFasceOrarie) {

        if (aFasceOrarie.equals(""))
            return "Fascia oraria non inserita";
        double r = Math.random();
        if( r <= disponibilita)
            return "Servizio non disponibile";

        return "Pianificazione";
    }
}

```

Test

```
import java.lang.Math;

public class Test {

    public static void main(String[] args) {
        String url = "http://localhost:8080/axis2/services/InterfacciaServiziAccessoriCoinFlip";

        String[] messaggio = new String[2];

        messaggio[0] = "<soapenv:Envelope
xmlns:soapenv=\\"http://schemas.xmlsoap.org/soap/envelope\\\"
xmlns:axis=\\"http://ws.apache.org/axis2\\\"><soapenv:Header/><soapenv:Body><axis:creaPianificaz
ione><axis:args0>pianificazione</axis:args0></axis:creaPianificazione></soapenv:Body></soapenv:
Envelope>";

        messaggio[1] = "<soapenv:Envelope
xmlns:soapenv=\\"http://schemas.xmlsoap.org/soap/envelope\\\"
xmlns:axis=\\"http://ws.apache.org/axis2\\\"><soapenv:Header/><soapenv:Body><axis:ottieniRiferim
entoServiziAccessori><axis:args0>cavour</axis:args0></axis:ottieniRiferimentoServiziAccessori></s
oapenv:Body></soapenv:Envelope>";

        try{
            int contatore = 0;
            String risultato;

            for(int i=0; i<2; i++) {
                for(int j=0; j<1750; j++) {
                    risultato = SOAPClient.sendMess(url,messaggio[i]);
                    if (risultato.contains("Servizio non disponibile"))
                        contatore++;
                }
            }
            System.out.println("Disponibilità:" +(double)((3500 - contatore)*100)/3500 + "% \n");

        } catch(Exception e) {System.out.println("Eccezione\n");}
    }
}
```

Report

OttieniRiferimentoServiziAccessori

| Input | Output atteso | Output ottenuto |
|-----------|----------------------------------|----------------------------------|
| Indirizzo | Riferimento ai servizi accessori | Riferimento ai servizi accessori |
| - | Indirizzo sbagliato | Indirizzo sbagliato |

creaPianificazione

| Input | Output atteso | Output ottenuto |
|--------------|----------------------------|----------------------------|
| Fasce orarie | Pianificazione | Pianificazione |
| - | Fascia oraria non inserita | Fascia oraria non inserita |

Disponibilità

| | Servizi Accessori | |
|--------------|-------------------|---------------|
| | Ciclo 1 | Ciclo 2 |
| Test 1 | 99,91% | 99,97% |
| Test 2 | 99,88% | 99,94% |
| Test 3 | 99,91% | 100,00% |
| Test 4 | 99,85% | 99,85% |
| Test 5 | 99,94% | 99,94% |
| Test 6 | 99,94% | 99,91% |
| Test 7 | 99,94% | 99,94% |
| Test 8 | 99,94% | 99,94% |
| Test 9 | 99,91% | 99,88% |
| Test 10 | 99,94% | 99,87% |
| Media | 99,92% | 99,92% |

10.2.5 Commenti ai test d'unità

Nel TestPlan era stato definito il seguente criterio di successo:

| | |
|----------------------|--|
| Criterio di successo | Tutti i servizi sono stati provati ed i test effettuati hanno mostrato che i servizi si comportano come previsto |
|----------------------|--|

Tutti i casi di test generati per i quattro applicativi hanno prodotto l'output previsto, pertanto possiamo considerare questa fase di test superata.

Come possiamo notare, le disponibilità medie dei servizi si discostano lievemente dal valore 99,9%. Tuttavia è importante notare non solo che 10 test non rappresentano un campione abbastanza ampio per effettuare valutazioni statistiche ma anche che la funzione di libreria Math.random() non fornisce valori casuali totalmente indipendenti e quindi ne è sconsigliato l'uso.

10.3 Logica di Coordinamento: Black-box testing

10.3.1 Black-box testing

CoordinatoreBB

```
import java.lang.Math;
import java.util.Date;

public class CoordinatoreBB {
    static int[] count = new int[4];

    public static void main(String[] args) {
        count[0] = 0;
        count[1] = 0;
        count[2] = 0;
        count[3] = 0;
        try {
            String url1 = "http://localhost:8080/axis2/services/InterfacciaOffertaCasaCoinFlip";
            String url2 =
                "http://localhost:8080/axis2/services/InterfacciaRichiestaCasaCoinFlip";
            String url3 = "http://localhost:8080/axis2/services/InterfacciaAttrazioniCoinFlip";
            String url4 =
                "http://localhost:8080/axis2/services/InterfacciaServiziAccessoriCoinFlip";

            String ottieniRichiesteNonGestite = "<?xml version=\"1.0\" encoding=\"utf-8\"?
?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottien
iRichiesteNonGestite></axis:ottieniRichiesteNonGestite></soapenv:Body></soapenv:Envel
ope>";

            String gestisciRichiesta = "<?xml version=\"1.0\" encoding=\"utf-8\"?
?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:gestis
ciRichiesta><axis:args0>richiesta</axis:args0></axis:gestisciRichiesta></soapenv:Body></so
apenv:Envelope>";

            String ricercaOfferte = "<?xml version=\"1.0\" encoding=\"utf-8\"?
?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ricerc
aOfferte> <axis:args0>indirizzo richiesta</axis:args0> <axis:args1>limitazioni
richiesta</axis:args1> <axis:args2>trasporto richiesta</axis:args2> <axis:args3>descrizione
richiesta</axis:args3> </axis:ricercaOfferte></soapenv:Body></soapenv:Envelope>";

            String ottieniListaAttrazioni = "<?xml version=\"1.0\" encoding=\"utf-8\"?
?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"
xmlns:axis=\"http://ws.apache.org/axis2\"><soapenv:Header/><soapenv:Body><axis:ottien
iListaAttrazioni> <axis:args0>indirizzo offerta</axis:args0>
</axis:ottieniListaAttrazioni></soapenv:Body></soapenv:Envelope>";

            String suggerisciListaPropostaAttivita = "<?xml version=\"1.0\" encoding=\"utf-8\"?
?><soapenv:Envelope xmlns:soapenv=\"http://schemas.xmlsoap.org/soap/envelope/\"

```

```
xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:suggerisciListaPropostaAttivita><axis:args0>Informazioni soggiorno</axis:args0><axis:args1>Classi</axis:args1></axis:suggerisciListaPropostaAttivita></soapenv:Body></soapenv:Envelope>;
```

```
String ottieniRiferimentoServiziAccessori = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:ottieniRiferimentoServiziAccessori><axis:args0>indirizzoOfferta</axis:args0></axis:ottieniRiferimentoServiziAccessori></soapenv:Body></soapenv:Envelope>";
```

```
String creaProposta = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:creaProposta><axis:args0>456hg3</axis:args0><axis:args1>indirizzo</axis:args1><axis:args2>2017-03-02</axis:args2><axis:args3>2018-02-01</axis:args3></axis:creaProposta></soapenv:Body></soapenv:Envelope>";
```

```
String modificaProposta = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:modificaProposta><axis:args0>proposta</axis:args0></axis:modificaProposta></soapenv:Body></soapenv:Envelope>";
```

```
String ottieniProposta = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:ottieniProposta><axis:args0>456hg3</axis:args0><axis:args1>indirizzo</axis:args1><axis:args2>2017-03-02</axis:args2><axis:args3>2018-02-01</axis:args3></axis:ottieniProposta></soapenv:Body></soapenv:Envelope>";
```

```
String pubblicaPropostaScambio = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:pubblicaPropostaScambio><axis:args0>offerta</axis:args0><axis:args1>contattiRichiedente</axis:args1></axis:pubblicaPropostaScambio></soapenv:Body></soapenv:Envelope>";
```

```
String visualizzaListaPropostaAttivita = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:visualizzaListaPropostaAttivita><axis:args0>AKL32D11</axis:args0></axis:visualizzaListaPropostaAttivita></soapenv:Body></soapenv:Envelope>";
```

```
String aggiungiAlternative = "<?xml version=\\\"1.0\\\" encoding=\\\"utf-8\\\"?><soapenv:Envelope xml�ns:soapenv=\\\"http://schemas.xmlsoap.org/soap/envelope\\\" xml�ns:axis=\\\"http://ws.apache.org/axis2\\\"><soapenv:Header><soapenv:Body><axis:aggiungiAlternative><axis:args0>Lista proposta attivita</axis:args0><axis:args1>false</axis:args1><axis:args1>false</axis:args1><axis:args1>true</axis:args1><axis:args1>false</axis:args1></axis:aggiungiAlternative></soapenv:Body></soapenv:Envelope>";
```

```

String rimuoviAlternative = "<?xml version=\\"1.0\\" encoding=\\"utf-8\\"?
><soapenv:Envelope xmlns:soapenv=\\"http://schemas.xmlsoap.org/soap/envelope\\\"
xmlns:axis=\\"http://ws.apache.org/axis2\\\"><soapenv:Header/><soapenv:Body><axis:rimuo
viAlternative> <axis:args0>Lista proposta attivita</axis:args0>
<axis:args1>true</axis:args1> <axis:args1>false</axis:args1> <axis:args1>false</axis:args1>
<axis:args1>false</axis:args1> <axis:args1>true</axis:args1> <axis:args1>false</axis:args1>
<axis:args1>false</axis:args1> <axis:args1>false</axis:args1>
</axis:rimuoviAlternative></soapenv:Body></soapenv:Envelope>";

String creaPianificazione = "<?xml version=\\"1.0\\" encoding=\\"utf-8\\"?
><soapenv:Envelope xmlns:soapenv=\\"http://schemas.xmlsoap.org/soap/envelope\\\"
xmlns:axis=\\"http://ws.apache.org/axis2\\\"><soapenv:Header/><soapenv:Body><axis:creaPi
anificazione><axis:args0>Fasce orarie</axis:args0>
</axis:creaPianificazione></soapenv:Body></soapenv:Envelope>";

int maxAttivita = 10;
String richiesteNonGestite;
String status;
String esito;
String richiesta, offerta, pianificazione;
String proposta, propostaCompilata, propostaModificata;
String indirizzoRichiesta, limitazioniRichiesta, trasportoRichiesta,
descrizioneRichiesta;
String classiAttrazioniRichiesta;
String listaOfferte, listaOfferteOrdinata;
String indirizzoOfferta;
String listaAttrazioni, classeAttrazione;
String riferimentoPropostaAttivita, riferimentoServiziAccessori;
String infoSoggiorno;
String richiedente, contattiRichiedente;
Date inizioCondivisione, fineCondivisione;
int statoProposta;
String propostaID, listaPropostaAttivita, propostaAttivita;
String fasceOrarie;

cicloEsterno: for(int k = 0; k < 1000; k++){
    System.out.println("ciclo " + (k+1) );

    richiesteNonGestite = SOAPClient.sendMess(url2,
ottieniRichiesteNonGestite);
    verificaDisponibilita(richiesteNonGestite, 1);
    int lunghezzaRichiesteNonGestite =
estraiLunghezzaLista(richiesteNonGestite);

    do{
        if(lunghezzaRichiesteNonGestite == 0) {
            continue cicloEsterno;
        }
        richiesta = estraiRichiesta(richiesteNonGestite);
        lunghezzaRichiesteNonGestite--;
        status = "false";
    }
}

```

```

        for(int i=0; i<3; i++){
            if(status.contains("false")){
                status = SOAPClient.sendMess(url2,
                gestisciRichiesta);
                verificaDisponibilita(status, 1);
            }
        }while(status.contains("false"));

        indirizzoRichiesta = estraiIndirizzoRichiesta(richiesta);
        limitazioniRichiesta = estraiLimitazioniRichiesta(richiesta);
        trasportoRichiesta = estraiTrasportoRichiesta(richiesta);
        descrizioneRichiesta = estraiDescrizioneRichiesta(richiesta);

        do {
            int lunghezzaOfferte;
            do{
                listaOfferte = SOAPClient.sendMess(url1, ricercaOfferte);
                verificaDisponibilita(listaOfferte, 0);
                lunghezzaOfferte = estraiLunghezzaLista(listaOfferte);
                if(lunghezzaOfferte == 0) {
                    if(limitazioniRichiesta.equals("")) {
                        continue cicloEsterno;
                    }
                    limitazioniRichiesta =
                    eliminaLimitazione(limitazioniRichiesta);
                }
            }while(lunghezzaOfferte == 0);

            classiAttrazioniRichiesta = estraiClassiAttrazioniRichiesta(richiesta);
            int lunghezzaClassiAttrazioni =
            estraiLunghezzaLista(classiAttrazioniRichiesta);
            float[] punteggi = new float[lunghezzaOfferte];

            for (int i=0; i<lunghezzaOfferte; i++) {
                offerta = ottieniOfferta(listaOfferte, i);
                indirizzoOfferta = estraiIndirizzoOfferta(offerta);

                listaAttrazioni = SOAPClient.sendMess(url3,
                ottieniListaAttrazioni);
                verificaDisponibilita(listaAttrazioni, 2);
                int lunghezzaListaAttrazioni =
                estraiLunghezzaLista(listaAttrazioni);

                int numeroClassi = 0;
                for (int j=0; j<lunghezzaClassiAttrazioni &&
                lunghezzaListaAttrazioni != 0; j++) {
                    classeAttrazione =
                    ottieniClasseAttrazione(classiAttrazioniRichiesta, j);
                    int numeroAttrazioni = contaAttrazioni(listaAttrazioni,
                    classeAttrazione);
                    if (numeroAttrazioni > 0)
                        numeroClassi++;
                }
            }
        }
    }
}

```

```

        }
        punteggi[i] = assegnaPunteggio(numeroClassi);
    }

    listaOfferteOrdinata = ordinaLista(listaOfferte, punteggi);

    offerta = ottieniOfferta(listaOfferteOrdinata, 0);
    indirizzoOfferta = estraiIndirizzoOfferta(offerta);
    infoSoggiorno = estraiInfoSoggiorno(offerta, richiesta);

    riferimentoPropostaAttivita = SOAPClient.sendMess(url3,
suggerisciListaPropostaAttivita);
    verificaDisponibilita(riferimentoPropostaAttivita, 2);

    riferimentoServiziAccessori = SOAPClient.sendMess(url4,
ottieniRiferimentoServiziAccessori);
    verificaDisponibilita(riferimentoServiziAccessori, 3);

    richiedente = estraiRichiedente(richiesta);
    inizioCondivisione = calcolaInizioCondivisione(offerta, richiesta);
    fineCondivisione = calcolaFineCondivisione(offerta, richiesta);

    proposta = SOAPClient.sendMess(url2, creaProposta);
    verificaDisponibilita(proposta, 1);

    propostaCompilata = inserisciInfoProposta(proposta, offerta,
riferimentoPropostaAttivita, riferimentoServiziAccessori);
    propostaModificata = SOAPClient.sendMess(url2,
modificaProposta);
    verificaDisponibilita(propostaModificata, 1);

    int indice = 1;
    statoProposta = 0;
    while(indice < lunghezzaOfferte) {
        proposta = SOAPClient.sendMess(url2, ottieniProposta);
        verificaDisponibilita(proposta, 1);

        statoProposta = estraiStatoProposta(proposta);
        if(statoProposta == 1) {
            offerta = ottieniOfferta(listaOfferteOrdinata,
indice);
            indirizzoOfferta = estraiIndirizzoOfferta(offerta);
            inizioCondivisione =
calcolaInizioCondivisione(offerta, richiesta);
            fineCondivisione = calcolaFineCondivisione(offerta,
richiesta);

            riferimentoServiziAccessori =
SOAPClient.sendMess(url4,
ottieniRiferimentoServiziAccessori);
            verificaDisponibilita(riferimentoServiziAccessori, 3);

```

```

        propostaCompilata =
inserisciInfoProposta(proposta, offerta,
riferimentoPropostaAttivita, riferimentoServiziAccessori);

        propostaModificata = SOAPClient.sendMess(url2,
modificaProposta);
verificaDisponibilita(propostaModificata, 1);

        indice++;
}
else if(statoProposta == 2)
break;
}

if(statoProposta != 2)
limitazioniRichiesta =
eliminaLimitazione(limitazioniRichiesta);

} while(statoProposta != 2);

contattiRichiedente = estraiContatti(proposta);
esito = SOAPClient.sendMess(url1, pubblicaPropostaScambio);
verificaDisponibilita(esito, 0);

propostaID = estraiPropostaID(riferimentoPropostaAttivita);
listaPropostaAttivita = SOAPClient.sendMess(url3,
visualizzaListaPropostaAttivita);
verificaDisponibilita(listaPropostaAttivita, 2);

statoProposta = estraiStatoListaPropostaAttivita(listaPropostaAttivita);
while(statoProposta != 2) {
    if(statoProposta == 1) {
        int lunghezzaPropostaAttivita =
estraiLunghezzaLista(listaPropostaAttivita);
        boolean[] fasceVuote =
cercaFasceOrarieVuote(listaPropostaAttivita,
lunghezzaPropostaAttivita);
        listaPropostaAttivita = SOAPClient.sendMess(url3,
aggiungiAlternative);
        verificaDisponibilita(listaPropostaAttivita, 2);

        int lunghezzaAlternative = lunghezzaPropostaAttivita *
maxAttivita;
        boolean[] daRimuovere = new
boolean[lunghezzaAlternative];
        daRimuovere =
selezionaAttivitaDaRimuovere(listaPropostaAttivita,
classiAttrazioniRichiesta, lunghezzaAlternative);
        listaPropostaAttivita = SOAPClient.sendMess(url3,
rimuoviAlternative);
        verificaDisponibilita(listaPropostaAttivita, 2);
    }
}

```

```

        listaPropostaAttivita = SOAPClient.sendMess(url3,
visualizzaListaPropostaAttivita);
        verificaDisponibilita(listaPropostaAttivita, 2);
        statoProposta =
estraiStatoListaPropostaAttivita(listaPropostaAttivita);
    }

fasceOrarie = estraiFasceOrarie(listaPropostaAttivita);
pianificazione = SOAPClient.sendMess(url4, creaPianificazione);
verificaDisponibilita(pianificazione, 3);
}

double disponibilita = (1.0 - (count[0]+count[1]+count[2]+count[3])/1000.0)*100.0;
for(int m = 0; m < 4; m++) {
    System.out.println("Servizio " + m + " non disponibile per " + count[m] + " volte");
}
System.out.println("Disponibilita' totale: " + disponibilita + "%");
} catch(Exception e){System.out.println("Eccezione");}
}

public static void verificaDisponibilita(String risposta, int indice)
{
    if(risposta.contains("Servizio non disponibile"))
        count[indice]++;
}

public static String estraiRichiesta(String listaRichieste) {
    return "Prima richiesta";
}

public static String estrailIndirizzoRichiesta(String richiesta) {
    return "Indirizzo estratto da richiesta";
}

public static String estraiLimitazioniRichiesta(String richiesta) {
    return "Limitazioni estratte da richiesta";
}

public static String estraiTrasportoRichiesta(String richiesta) {
    return "Trasporto estratto da richiesta";
}

public static String estraiDescrizioneRichiesta(String richiesta) {
    return "Descrizione estratta da richiesta";
}

public static int estraiLunghezzaLista(String lista){

if(lista.contains("Lista di attrazioni ordinata per distanza"))
    return (int)(Math.random()*10);
if(lista.contains("Lista di offerte pertinenti con i parametri di ricerca."))

```

```

        return (int)(Math.random()*20);
    if(lista.contains("Richieste non gestite"))
        return (int)(Math.random()*20);
    if(lista.contains("Lista proposta attivita"))
        return (int)(Math.random()*100);

    return (int)(Math.random()*10);
}

public static String estraiClassiAttrazioniRichiesta(String richiesta) {
    return "Lista classi attrazioni estratte da richiesta";
}

public static String ottieniOfferta(String lista, int indice) {
    return "Offerta all'indice "+indice;
}

public static String estrailIndirizzoOfferta(String offerta) {
    return "Indirizzo estatto da offerta";
}

public static String ottieniClasseAttrazione(String classiAttrazioniRichiesta, int indice) {
    return "Classe d'attrazione all'indice "+indice;
}

public static int contaAttrazioni(String listaAttrazioni, String classe) {
    return (int)(Math.random()*5);
}

public static float assegnaPunteggio(int numeroClassi) {
    return numeroClassi * (float)Math.random() * 10;
}

public static String ordinaLista(String listaDaOrdinare, float[] punteggio) {
    return "Lista offerte ordinata per punteggio decrescente";
}

public static String estrailInfoSoggiorno(String offerta, String richiesta) {
    return "Informazioni soggiorno";
}

public static String estraiRichiedente(String richiesta){
    return "Richiedente";
}

public static Date calcolaInizioCondivisione(String offerta, String richiesta){
    return new Date(2017, (int)(Math.random()*12), (int)(Math.random()*27+1));
}

public static Date calcolaFineCondivisione(String offerta, String richiesta){
    return new Date(2018, (int)(Math.random()*12), (int)(Math.random()*27+1));
}

```

```

public static String inserisciInfoProposta(String proposta, String offerta, String
riferimentoPropostaAttivita, String riferimentoServiziAccessori){
    return "Proposta compilata";
}

public static int estraiStatoProposta(String richiesta){
    return (int)(Math.random()*3);
}

public static String eliminaLimitazione(String limitazioniRichiesta){
    return "Limitazioni ridotte";
}

public static String estraiContatti(String proposta){
    return "Contatti del richiedente";
}

public static String estraiPropostaID(String riferimentoPropostaAttivita) {
    return "ID della proposta";
}

public static int estraiStatoListaPropostaAttivita(String listaPropostaAttivita){
    return (int)(Math.random()*3);
}

public static boolean[] cercaFasceOrarieVuote(String listaPropostaAttivita, int lunghezza){
    boolean[] temp = new boolean[lunghezza];
    for(int i = 0; i < lunghezza; i++ )
        temp[i] = (Math.random() < 0.3)? true : false;
    return temp;
}

public static boolean[] selezionaAttivitaDaRimuovere(String listaPropostaAttivita, String
classiAttrazioni, int lunghezza){
    boolean[] temp = new boolean[lunghezza];
    for(int i = 0; i < lunghezza; i++ )
        temp[i] = (Math.random() < 0.2)? true : false;
    return temp;
}

public static String estraiFasceOrarie(String listaPropostaAttivita){
    return "Fasce orarie estratte";
}
}

```

Report

Disponibilità del servizio al primo ciclo

Abbiamo lanciato 10 volte un programma che esegue 1000 volte il coordinatore, tenendo conto di quale servizio si blocca.

La disponibilità di tutti e quattro i servizi è stata impostata al 99,9%.

| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 | MEDIA |
|----------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------------|
| Offerta casa | 1 | 1 | 1 | 3 | 1 | 3 | 1 | 1 | 4 | 3 | 1,90 |
| Richiesta casa | 2 | 9 | 10 | 6 | 18 | 8 | 10 | 14 | 5 | 10 | 9,20 |
| Attrazioni | 14 | 14 | 17 | 17 | 16 | 11 | 16 | 10 | 22 | 17 | 15,40 |
| Servizi accessori | 3 | 2 | 0 | 2 | 6 | 2 | 2 | 8 | 6 | 2 | 3,30 |
| Disponibilità media | 98,00% | 97,40% | 97,20% | 97,20% | 95,90% | 97,60% | 97,10% | 96,70% | 96,30% | 96,80% | 97,02% |

Come era stato previsto, i servizi *Richiesta casa* ed *Attrazioni* hanno un maggior numero di fallimenti, infatti i loro metodi sono invocati più frequentemente. Questo comportamento era prevedibile da un'accurata analisi del diagramma di sequenza della logica di coordinamento prodotto in fase d'implementazione.

Anche se la probabilità che il richiedente non accetti una proposta in *Richiesta casa* e in *Attrazioni* è uguale, il numero delle operazioni necessarie in caso di rifiuto è maggiore per *Attrazioni* (3 chiamate contro 2). Per questo motivo sono maggiori i casi in cui quest'ultimo servizio non è disponibile, essendo testata più frequentemente la sua disponibilità.

Nonostante entrambi i moduli *Offerta casa* e *Servizi accessori* si trovino all'interno di un ciclo, il maggior numero di fallimenti a cui quest'ultimo è soggetto dipende dalle soglie scelte per le variabili aleatorie: *Offerta casa* ha un numero di offerte pari a 0 nel 5% dei casi mentre *Servizi accessori* calcola un nuovo riferimento ogni volta che la proposta non è accettata (quest'evento avviene nel 33% dei casi).

Disponibilità del servizio al secondo ciclo

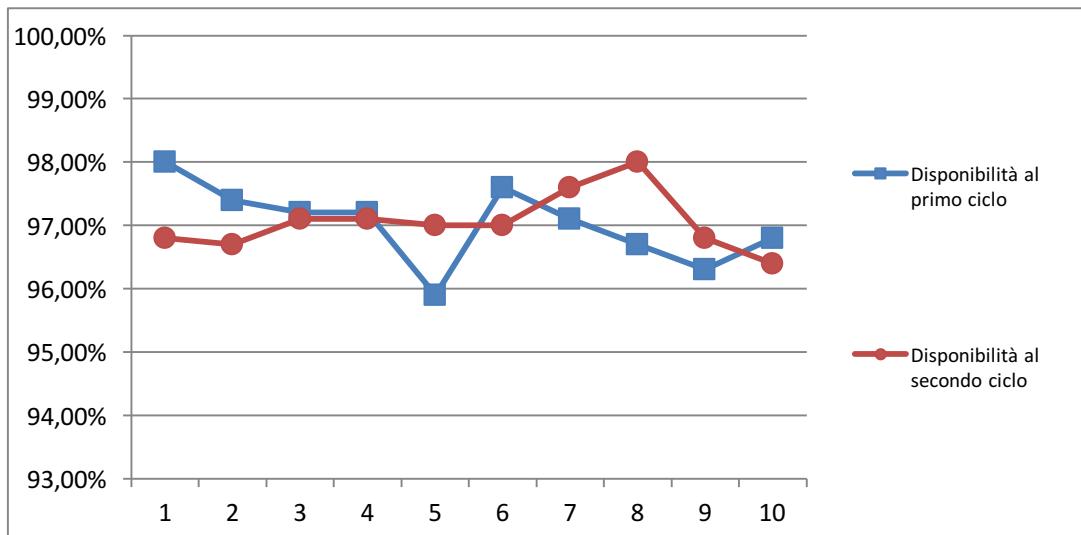
| | Test 1 | Test 2 | Test 3 | Test 4 | Test 5 | Test 6 | Test 7 | Test 8 | Test 9 | Test 10 | MEDIA |
|----------------------------|--------|--------|--------|--------|--------|--------|--------|--------|--------|---------|---------------|
| OffertaCasa | 4 | 8 | 0 | 1 | 2 | 4 | 4 | 2 | 1 | 3 | 2,90 |
| RichiestaCasa | 7 | 7 | 11 | 9 | 11 | 10 | 8 | 6 | 9 | 6 | 8,40 |
| Attrazioni | 19 | 15 | 16 | 17 | 12 | 15 | 10 | 10 | 19 | 23 | 15,60 |
| ServiziAccessori | 2 | 3 | 2 | 2 | 5 | 1 | 2 | 2 | 3 | 4 | 2,60 |
| Disponibilità media | 96,80% | 96,70% | 97,10% | 97,10% | 97,00% | 97,00% | 97,60% | 98,00% | 96,80% | 96,40% | 97,05% |

Successo del test

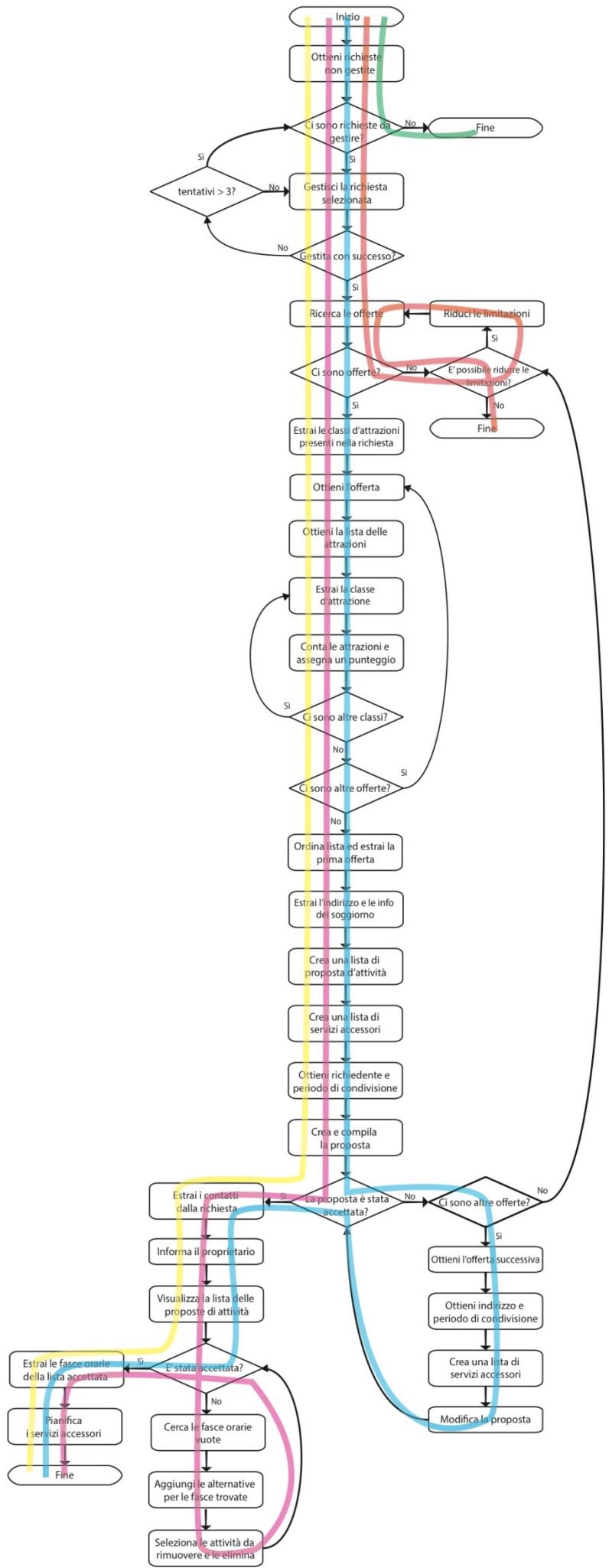
Il TestPlan specifica il seguente criterio di successo:

| | |
|----------------------|--|
| Criterio di successo | SharingEverywhere ha una disponibilità sopra il 95% ed i singoli applicativi non causano più del 3% dell'assenza del servizio. |
|----------------------|--|

Il 3% di 1000 casi di test equivale a 30 fallimenti. Come si può vedere dalla tabella, nessun applicativo raggiunge quella soglia.



Come è possibile vedere dal grafico, la disponibilità di SharingEverywhere nel suo complesso si assesta intorno al 97% con picchi del 98%. Pertanto possiamo dire che il test ha avuto esito positivo.



10.3.2 White-box testing

Report

Scelta dei percorsi d'analizzare

Il team ha analizzato il grafo CFG riportato in Figura 9.2 individuando cinque percorsi principali tali che ogni istruzione della logica di coordinamento venga eseguita almeno una volta. I risultati di questa fase d'analisi sono riportati in Figura 10.2.

Basandosi sui percorsi individuati sono stati generati i seguenti cinque casi di test:

Test 1 (verde):

| | |
|--------------------------|----------------------------------|
| Input | lunghezzaRichiesteNonGestite = 0 |
| Output desiderato | Non ci sono richieste da gestire |
| Output ottenuto | Non ci sono richieste da gestire |
| Esito | POSITIVO |

Test 2 (rosso):

| | |
|--------------------------|--|
| Input | lunghezzaRichiesteNonGestite = 5 lunghezzaOfferte = 0 limitazioniRichiesta ≠ "" (prima iterazione) limitazioniRichiesta = "" (seconda iterazione) |
| Output desiderato | Non ci sono più offerte da proporre |
| Output ottenuto | Non ci sono più offerte da proporre |
| Esito | POSITIVO |

Test 3 (azzurro):

| | |
|--------------------------|--|
| Input | lunghezzaRichiesteNonGestite = 5 lunghezzaOfferte = 5 statoProposta = 1 (prima offerta) statoProposta = 2 (seconda offerta) statoListaPropostaAttivita = 2 |
| Output desiderato | È stata proposta una nuova offerta al richiedente La condivisione è andata a buon fine |
| Output ottenuto | È stata proposta una nuova offerta al richiedente La condivisione è andata a buon fine |
| Esito | POSITIVO |

Test 4 (fucsia):

| | |
|--------------|---|
| Input | lunghezzaRichiesteNonGestite = 5 lunghezzaOfferte = 5 statoProposta = 2 statoListaPropostaAttivita = 1 (prima iterazione) statoListaPropostaAttivita = 2 (seconda iterazione) |
|--------------|---|

| | |
|--------------------------|---|
| Output desiderato | Aggiornata la lista proposta attività La condivisione è andata a buon fine |
| Output ottenuto | Aggiornata la lista delle proposte d'attività La condivisione è andata a buon fine |
| Esito | POSITIVO |

Test 5 (giallo):

| | |
|--------------------------|---|
| Input | lunghezzaRichiesteNonGestite = 5 lunghezzaOfferte = 5 statoProposta = 2 statoListaPropostaAttivita = 2 |
| Output desiderato | La condivisione è andata a buon fine |
| Output ottenuto | La condivisione è andata a buon fine |
| Esito | POSITIVO |

Capitolo 11

Manutenibilità

11.1 Manutenzione del progetto

Per manutenzione si intendono i cambiamenti che devono essere fatti al progetto dopo che è già stato consegnato ai committenti. Ci sono diversi motivi per cui ciò può essere necessario:

- Ogni sistema ha la necessità di adattarsi all'ambiente circostante, seguendone l'evoluzione
- Possono essere necessarie risorse aggiuntive per preservare la semantica del sistema e semplificarne la struttura, oppure per migliorarne le prestazioni
- In alcuni casi, è meglio correggere subito dei difetti notati dopo la consegna del lavoro piuttosto che lasciare che siano i committenti a accorgersi dei malfunzionamenti

La manutenzione è un'attività importante e sicuramente non va sottovalutata, in quanto può influenzare fortemente i costi di sviluppo. Per avere una stima di tali costi, è necessario valutare la complessità delle componenti del sistema.

11.2 Indice di gestibilità

È possibile utilizzare un unico indice che riassume le caratteristiche di complessità del sistema intese come

- Numero di operatori e operandi del programma (metrica di complessità di Halstead);
- Complessità della logica, ovvero numero dei percorsi di esecuzione indipendenti del codice (complessità ciclomatica);
- Dimensione del codice espressa come numero di linee di codice.

11.2.1 Metrica di complessità di Halstead

Per il calcolo di questa metrica abbiamo utilizzato il software Testwell CMTJava perché un calcolo manuale di tutti gli indici ci avrebbe portato via troppo tempo.

Siano

- n_1 il numero di operatori distinti
- n_2 il numero di operandi distinti
- N_1 il numero di occorrenze di operatori
- N_2 il numero di occorrenze di operandi

intendendo per operatore ogni simbolo o parola chiave che specifica un'azione e per operando ogni simbolo che rappresenta un dato. Il software ci ha restituito i seguenti valori:

| | |
|-------|-----|
| n_1 | 237 |
| n_2 | 290 |
| N_1 | 691 |
| N_2 | 627 |

e ha calcolato le seguenti misurazioni, pesando i risultati ottenuti in precedenza:

| Misurazione | Formula | Risultato ottenuto |
|---------------------------|---------------------------|--------------------|
| Dimensione del programma | $N = N_1 + N_2$ | 1118 |
| Vocabolario del programma | $n = n_1 + n_2$ | 327 |
| Occupazione di memoria | $V = N \cdot \log_2 n$ | 9977 |
| Difficoltà | $D = n_1/2 \cdot N_2/n_2$ | 198 |
| Sforzo | $E = D \cdot V$ | 253941 |
| Tempo per lo sviluppo | $T = E/15$ | 4h circa |

11.2.2 Complessità ciclomatica di Mc Cabe

Per il calcolo di questo indice, abbiamo sempre utilizzato il software Testwell CMTJava poiché con l'analisi in white box non abbiamo analizzato tutti i possibili percorsi di esecuzione ma solo i più significativi. Un'analisi completa del codice per identificarli tutti e poter applicare la formula

$$G = e - n + 2 \cdot p$$

(con e numero di “confini” dei CFG, n numero di nodi dei CFG e p numero di sotto-grafi) avrebbe richiesto troppo tempo.

Abbiamo ottenuto che questo indice ha valore $G = 28$, poiché sono molti i cicli annidati e le condizioni nel nostro codice. Ciò significa che il rischio di errori nel codice è piuttosto alto ma non tale da impedire un corretto debugging.

11.2.3 Numero di linee di codice

Testwell CMTJava ci ha anche fornito informazioni riguardo le linee del codice del programma e di come sono ripartite.

| | |
|-------------------------------------|----------------|
| Numero di righe totali | $LOCphy = 269$ |
| Numero di righe vuote | $LOCbl = 51$ |
| Numero di righe effettive di codice | $LOCpro = 218$ |
| Numero di righe di commenti | $LOCcom = 0$ |
| Numero di istruzioni | $stat = 142$ |

Le linee di codice prodotte per il progetto sono $LOCpro = 218$, meno di quelle stimate con l'utilizzo di SystemStar.

11.3 Risultato

L'indice di gestibilità calcola un valore compreso tra 0 e 100 che rappresenta la facilità con cui si può effettuare la manutenzione del codice. A un valore più alto dell'indice corrisponde un livello più alto di manutenibilità.

Chiamando

- V l'occupazione di memoria media in accordo alla definizione di Halstead
- G la complessità ciclomatica
- LOC il numero di righe di codice

e applicando la formula

$$MI = 171 - 5,2 \cdot \ln(V) - 0,23 \cdot G - 16 \cdot \ln(LOC)$$

Testwell CMTJava ci ha fornito degli indici di gestibilità in termini assoluti che poi abbiamo normalizzato per avere valori nell'intervallo [0,100]:

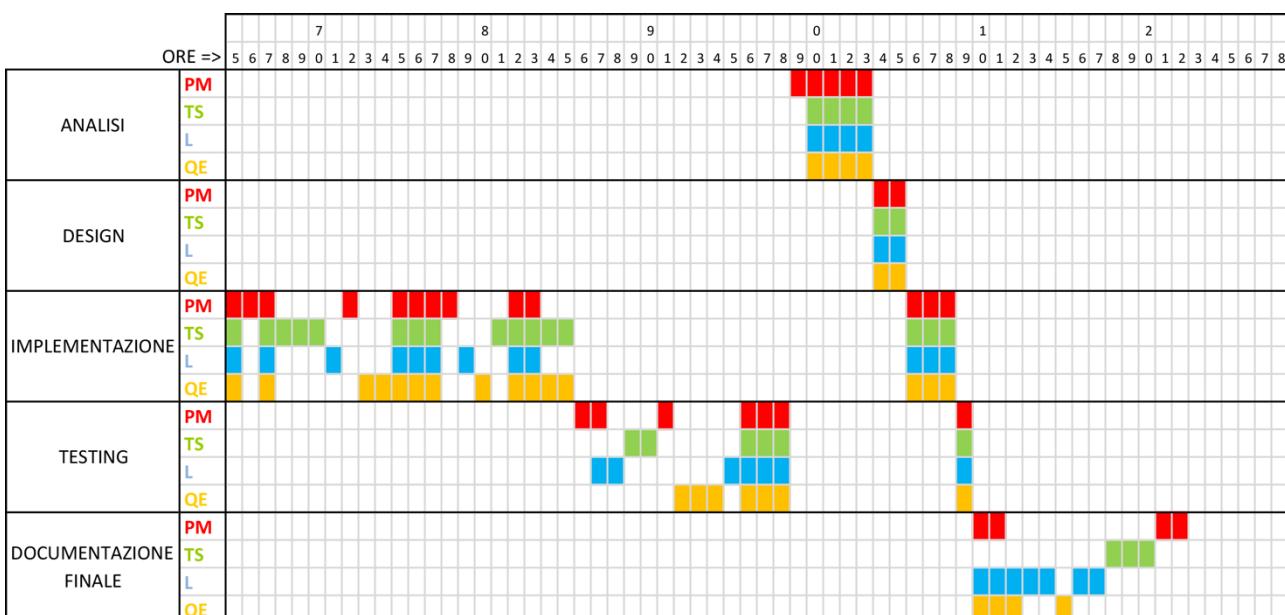
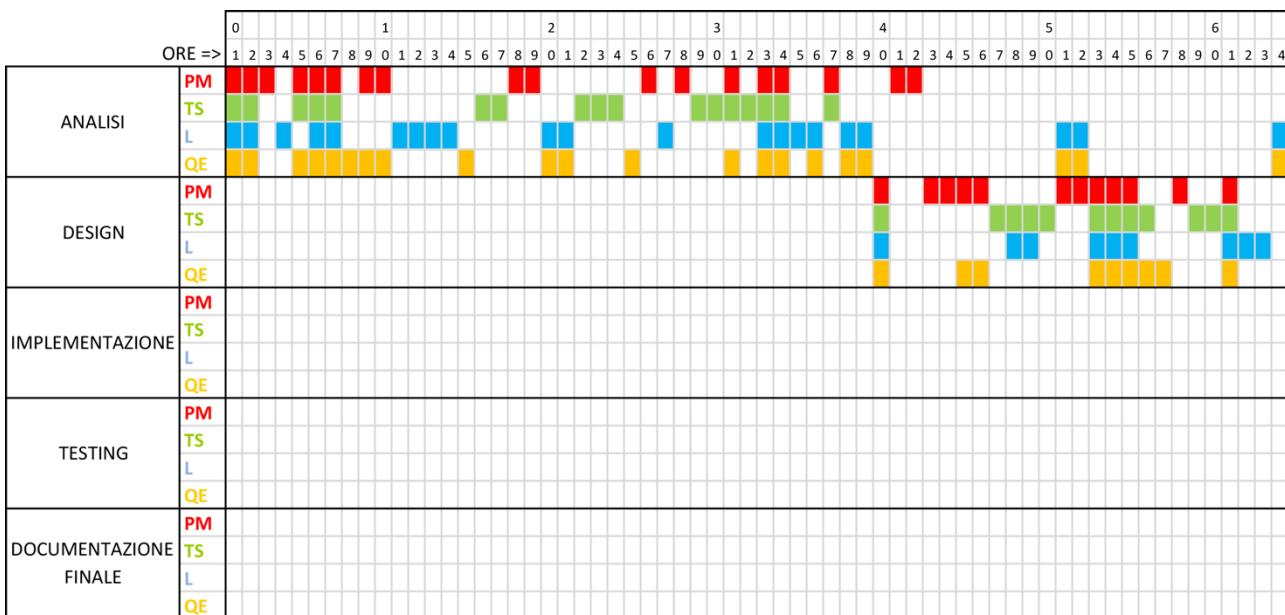
| Funzione | MI | MI normalizzato |
|--------------------------------|--------|-----------------|
| Funzione main() | 42 | 24,56 |
| Media per le altre 26 funzioni | 128,73 | 75,28 |

Come previsto, la funzione main è più complessa e per questo l'indice è più basso, considerando anche che il software da noi sviluppato non è stato pensato per essere efficiente ma solo per orchestrare in maniera semplice gli applicativi. Tutte le altre funzioni ritornano delle stringhe o compiono semplici computazioni quindi è sensato che abbiamo un indice alto.

Nel complesso $MI = 73,40$ che è un buon valore e indica una manutenibilità molto alta.

Capitolo 12

Diagramma di Gantt



12.1 Commenti al diagramma di Gantt

Dall'analisi del diagramma di Gantt, il modello del processo di sviluppo scelto è stato seguito per la maggior parte del tempo. Le uniche differenze si hanno con delle attività di documentazione. Quando abbiamo finito la fase di analisi non ci era chiaro che alcuni documenti dovessero essere consegnati prima dell'inizio della fase successiva di design. Dopo esserci chiariti con i committenti, abbiamo registrato le seguenti ore:

1. PM (2 ore): riguarda la registrazione della stima delle ore, della scelta del modello e dell'analisi dei rischi. Il modello era già stato definito all'inizio dell'analisi e anche la stima delle ore era già stata calcolata, ma queste informazioni non erano state messe per iscritto in maniera completa e non era stata registrata l'attività relativa.
2. QE e L (2+1 ore): riguarda la conclusione dell'SRS e del "Piano sulla Qualità". Come nel caso precedente, le decisioni sono state prese prima della fine della fase di analisi ma per motivi di incomprensioni la fase di design è stata iniziata prima che questi documenti fossero registrati.

Nel template del Gantt è stata eliminata la fascia delle specifiche poiché non è necessario registrare le relative ore come suggerito dai committenti e, dopo che queste sono state definite, non occorre modificarle ma sono state considerate come il punto di partenza per le decisioni successive. La fascia della documentazione finale è stata espansa perché i diversi componenti del team sono responsabili di una quantità differente di documenti, quindi abbiamo ritenuto importante differenziare le ore.

Inoltre, è possibile notare che le attività individuali sono state registrate in orari quasi sempre differenti tra loro, questo per sottolineare che il livello d'indipendenza tra i membri è molto alto. Il fenomeno visibile nelle ultime ore è dovuto ad un maggior coordinamento del lavoro e dall'approssimarsi della scadenza della consegna del progetto, quindi le attività compaiono nelle stesse fasce orarie.

Infine, il diagramma non tiene conto che il PM sicuramente farà una revisione finale della documentazione e creerà l'archivio contenente il codice sorgente e binario.