

A&R Rental Book

SSE Final Report

Gruppo di lavoro:

Cascone Livio
Gallina Cristina
Peraz Giuliano
Tanferna Valerio



Università di Pisa
Master Degree in Computer Engineering

Indice

Indice	2
Premessa	5
1.Specifiche dei Requisiti ed Analisi	6
1.1 Introduzione	6
1.1.1 Purpose	6
1.1.2 Scope	6
1.1.3 Overview	6
1.2 Descrizione generale	6
1.2.1 Product perspective	6
1.3 Interfacce software	7
1.3.1 Offerta & Stato Libri	7
1.3.2 Ricerca & Prestito	16
1.3.3 Appuntamento Per Consegna	22
1.3.4 Recensione Libro/Utente	27
1.4 Product Functions	31
1.5 User characteristics	32
1.6 Constraints	32
1.7 Assumptions and dependencies	32
1.8 Specifica dei requisiti	32
1.8.1 Interfaccia utente	32
1.8.2 Requisiti funzionali	33
1.8.3 Interfaccia software	34
1.8.4 Requisiti di prestazioni	34
1.8.5 Multiutenza	34
1.8.6 Sicurezza	34
1.8.7 Attributi del sistema software	34
1.8.8 Altri requisiti	35
2.Software Project Management Plan	35
2.1 Introduzione	35
2.1.1 Descrizione generale di progetto	35
2.1.2 Project deliverables	35
2.1.3 Definizioni	35
2.2 Organizzazione di progetto	35
2.2.1 Gruppo di lavoro	36
2.2.2 Modello di processo	36
2.2.3 Responsabilità di progetto	36
2.2.4 Strumenti Utilizzati	37

2.2.5 Funzionalità di supporto al progetto	37
2.2.6 Obiettivi e priorità	38
2.2.7 Assunzioni, Dipendenze e Vincoli	38
2.3 Risk Management	38
2.3.1 Identificazione dei rischi	38
2.3.2 Analisi dei rischi	39
2.3.3 Pianificazione dei rischi	39
2.3.4 Monitoraggio dei rischi	40
2.4 Analisi costi e tempi di sviluppo	41
2.4.1 Function Point technique	41
2.4.2 Function Point analisi	44
2.4.3 COCOMO Analisi	46
2.4.3 Risultati COSTAR	47
3.Quality Plan	48
3.1 Introduzione	48
3.2 Funzionalità	49
3.3 Affidabilità	49
3.4 Usabilità	49
3.5 Efficienza	50
3.6 Manutenibilità	50
3.7 Portabilità	50
3.8 Process quality	50
4.Design	52
4.1 Offerta & Stato Libri	52
4.1.1 Class Diagrams	52
4.1.2 SoaML Service Interface	53
4.2 Ricerca & Prestito	53
4.2.1 Class Diagrams	53
4.2.2 SoaML Service Interface Diagram	55
4.3 Appuntamento Per Consegna	55
4.3.1 Class Diagrams	55
4.3.2 SoaML Service Interface Diagram	56
4.4 Recensione Libro/Utente	57
4.4.1 Class Diagrams	57
4.4.2 SoaML Service Interface Diagram	58
4.5 Agent	59
4.5.1 SoaML Service Interface Diagram	59
4.5.2 SoaML Sequence Diagram	60
5.Implementazione	61
5.1 Offerta & Stato Libri	61

5.1.1 Service Code	61
5.1.2 XML instance e XSD schema	62
5.2 Ricerca & Prestito	63
5.2.1 Service Code	63
5.3.2 XML instance e XSD schema	64
5.3 Appuntamento Per Consegna	66
5.3.1 Service Code	66
5.3.2 XML instance e XSD schema	67
5.4 Recensione Libro/Utente	70
5.4.1 Service Code	70
5.4.2 XML instance e XSD schema	71
5.5 Agent	73
5.5.1 Agent.java Class	73
5.5.2 Messages.java Class	79
5.5.3 SOAPClient.java Class	79
6.Test Plan	80
6.1 Introduzione	80
6.2 Functional testing	80
6.2.1 Funzionalità da testare	80
6.2.2 Approccio	80
6.2.3 Pass Criteria	81
6.2.4 Fail Criteria	81
6.2.5 Test Deliverables	81
6.3 Structural testing	89
6.3.1 Funzionalità da testare	89
6.3.2 Approach	89
6.3.3 Pass Criteria	89
6.3.4 Fail Criteria	89
6.3.5 Test Deliverables	89

Premessa

Il servizio A&R Rental Book fornisce la possibilità ad un proprietario di libri (owner) di condividerli all'interno della propria libreria personale, e ad un lettore (reader) di poter ricercare e richiedere in prestito un libro tra quelli condivisi. Il servizio consente di proporre ad un lettore un libro condiviso dal suo proprietario sulla base di alcune preferenze espresse dal lettore stesso. Per effettuare lo scambio del libro il sistema mette a disposizione una funzionalità di consegna del libro sia tramite corriere sia direttamente di persona tra i due utenti; al termine dell'esperienza di prestito è possibile fornire una recensione sia sul libro letto che sull'esperienza di prestito. Per fornire queste funzionalità il sistema si compone dei seguenti moduli applicativi:

- A. Offerta & Stato libri**
- B. Richiesta & prestito**
- C. Appuntamento per consegna**
- D. Recensione libro/utente**

Documentazione

I documenti prodotti durante il progetto e riportati in questo documento di sintesi finale sono i seguenti:

- **Software Requirements Specification (SRS):** scritto secondo lo standard IEEE 830-1998, descrive precisamente ogni requisito essenziale (funzioni, prestazioni, vincoli di progettazione) del software. Ogni requisito dovrà essere verificato e validato
- **Software Project Management Plan (SPMP):** scritto secondo lo standard IEEE1058-1998, definisce i processi tecnici e manageriali necessari per lo sviluppo di un prodotto software che soddisfa tutti i requisiti
- **Software Quality Assurance Plan (SQAP):** scritto secondo lo standard IEEE 12207 definisce lo standard di qualità che il software deve raggiungere. Inoltre definisce i metodi e le tecniche che saranno usare per raggiungere la qualità desiderata
- **Software Test Plan (STP):** scritto secondo lo standard IEEE 829, definisce la strategia che devono essere usate per verificare che il sistema rispetti i requisiti e le specifiche di progettazione

1. Specifiche dei Requisiti ed Analisi

1.1 Introduzione

Questa sezione contiene una descrizione generale di tutto il Sistema, gli applicativi di cui è composto, il loro comportamento, i requisiti funzionali e non funzionali. Questo capitolo è conforme allo standard IEEE 830.

1.1.1 Purpose

Lo scopo di questo documento è dare una descrizione dettagliata dei requisiti per il software A&R. Verrà illustrata una dichiarazione completa per sviluppo del sistema, i vincoli, interfacce ed eventuali interazioni con applicazioni esterne. Questo documento deve essere proposto al consumatore per la sua approvazione e vuole essere un riferimento per lo sviluppo della prima versione del sistema per il team di sviluppo.

1.1.2 Scope

A&R è un servizio che ha come obiettivo rendere più semplice lo scambio e la condivisione dei libri.

Vantaggi:

- Sistema di gestione interno per la ricerca di tutti i libri disponibili
- Libreria personale dove l'utente può inserire e rimuovere i libri
- Facilità di utilizzo, interfaccia semplice e veloce
- Sistema di prenotazione, prestito e reso veloce ed intuitivo

1.1.3 Overview

Il paragrafo 1.2 contiene una descrizione delle funzionalità del sistema e le interazioni con gli altri sistemi. Nel terzo paragrafo sono descritte le specifiche dei requisiti in maniera dettagliata ed una descrizione delle differenti interfacce del sistema.

1.2 Descrizione generale

In questa sezione è presente una panoramica dell'intero sistema, come A&R interagisce con gli altri sistemi, una descrizione delle funzionalità base.

1.2.1 Product perspective

Il servizio A&R interagisce con i seguenti 4 servizi fondamentali:

- Offerta & Stato Libri (OSL)
- Richiesta & Prestito (REP)
- Appuntamento Per Consegna (APC)
- Recensione Libro Utente (RLU)

1.3 Interfacce software

Questa sezione descrive quali sono le funzionalità dei 4 servizi usati e le interfacce che il servizio A&R utilizza per comunicare con i servizi sopra citati.

Per ognuno dei servizi si darà una breve descrizione degli Use Case, il dettaglio di questi, l'interfaccia utente del servizio e i sequence diagram estratti dal dettaglio degli use case.

1.3.1 Offerta & Stato Libri

Il servizio di Offerta e Stato Libri contiene un catalogo di Libri inseriti dagli utenti che vogliono condividere la propria libreria con altre persone e permette agli utenti di gestire il prestito per ogni libro di loro proprietà. Lo stato del libro può essere disponibile, in prestito o prenotato. Il servizio offre le funzionalità di:

- cambio di stato di un libro
- ricerca di un libro tramite parametri come titolo, autore e genere
- cambio manuale della data di inizio del prestito

Il servizio OSL imposta automaticamente la data di fine del prestito a 30 giorni dalla data di inizio prestito.

Use Case

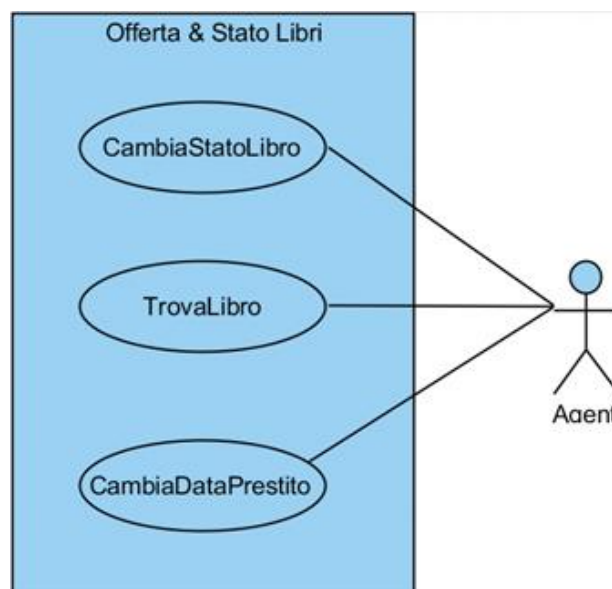


Figure 1: use case Offerta & Stato Libri

USE CASE DETTAGLIATI

Nelle sezioni seguenti sono mostrati i dettagli dei casi d'uso, ovverosia quali sono le azioni dell'Agent quando usa i servizi di OSL. Oltre al dettaglio dei casi d'uso sono aggiunti i wireframe dell'applicazione, che emulano la user interface delle funzionalità usate dall'Agent.

USE CASE DETAILS: CambiaStatoLibro












1.  **Agent** inserisce **Titolo**
2.  **Agent** inserisce **Proprietario**
3.  **Agent** preme Trova Libro
4. **SYSTEM** ritorna le informazioni del Libro
5. **if** lo stato del Libro è *disponibile*
 - 5.1.  **Agent** cambia stato da *disponibile* a *in prestito*
 - 5.2. **SYSTEM** chiede chi è il lettore che ha chiesto il libro in prestito
 - 5.3.  **Agent** inserisce il Lettore
6. **else if** il prestito è scaduto
 - 6.1.  **Agent** cambia stato da *in prestito* a *disponibile*
 - 6.2. **SYSTEM** conferma il cambiamento di stato
7. **else if** lo stato del Libro è *in prestito*
 - 7.1.  **Agent** cambia stato da *in prestito* a *prenotato*
 - 7.2. **SYSTEM** chiede chi è il Lettore che ha prenotato
 - 7.3.  **Agent** inserisce il Lettore
8. **else if** lo stato del Libro è *prenotato*
 - 8.1. **if** è scaduto il prestito
 - 8.1.1.  **Agent** cambia stato da *prenotato* a *in prestito*
 - 8.1.2. **SYSTEM** chiede chi è il Lettore che ha preso in prestito il libro
 - 8.1.3.  **Agent** inserisce il Lettore
 - 8.2. **else**
 - 8.2.1. **SYSTEM** avvisa che non è terminato il prestito**end if**
9.  **Agent** preme Conferma

Figure 2: use case dettagliato CambiaStatoLibro

WIREFRAME: CambiaStatoLibro

Cambia Stato del Libro


Titolo

Proprietario

Trova Libro

Figure 3: wireframe 1 CambiaStatoLibro

12:00

Cambia Stato del Libro 

Titolo	I promessi Sposi
Autore	Alessandro Manzoni
Genere	Narrativa
Proprietario	Mario Rossi

Stato del libro

☐ Disponibile

☒ In Prestito a

☐ Prenotato da

Conferma






Figure 4: wireframe 2 CambiaStatoLibro

SEQUENCE DIAGRAM: CambiaStatoLibro

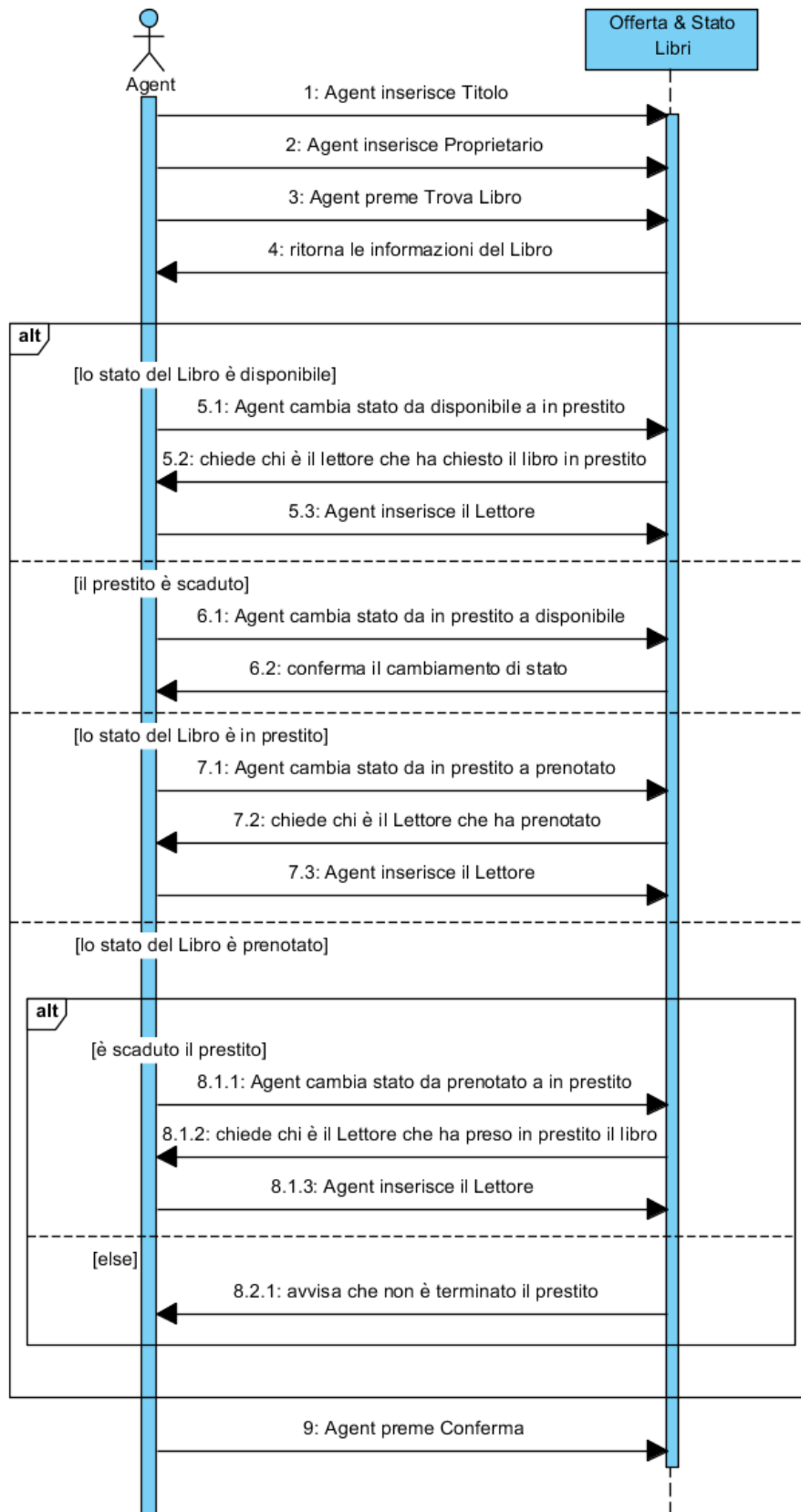


Figure 5: sequence diagram CambiaStatoLibro

USE CASE DETAILS: TrovaLibro






1.  [Agent](#) inserisce **Titolo**
2.  [Agent](#) inserisce **Genere**
3.  [Agent](#) inserisce **Autore**
4.  [Agent](#) preme Cerca
5. **SYSTEM**  ritorna la lista di libri corrispondente ai termini di ricerca e ordinati per stato (solo disponibile e in prestito)

Figure 6: use case dettagliato TrovaLibro

WIREFRAME: TrovaLibro



Figure 7: wireframe 1 TrovaLibro

Lista dei Libri Cercati 	
Titolo	Stato
I promessi sposi Alessandro Manzoni	Disponibile
I promessi sposi Alessandro Manzoni	Disponibile
La vita dei promessi sposi Andrea Bellavita	Disponibile
I promessi sposi Alessandro Manzoni	In Prestito
La vita dei promessi sposi Andrea Bellavita	In Prestito

Figure 8: wireframe 2 TrovaLibro

SEQUENCE DIAGRAM: TrovaLibro

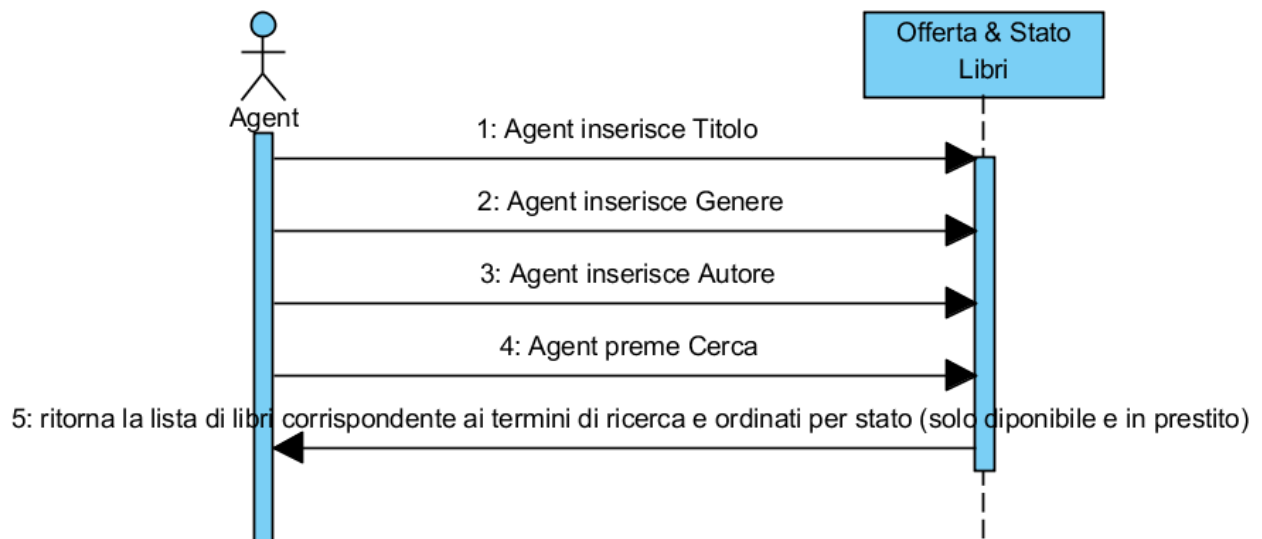


Figure 9: sequence diagram TrovaLibro

USE CASE DETAILS: CambiaDataPrestito

1. **Agent** inserisce **Titolo**
2. **Agent** inserisce **Proprietario**
3. **Agent** preme **Trova Libro**
4. **SYSTEM** ritorna il Libro richiesto
5. **SYSTEM** chiede quali sono le date di inizio e di fine prestito
6. **Agent** inserisce le date di inizio e di fine del prestito
7. **Agent** preme **Aggiorna date**
8. **SYSTEM** conferma l'aggiornamento

Figure 10: use case dettagliato CambiaDataPrestito

WIREFRAME: CambiaDataPrestito

The wireframe shows a mobile application interface for changing loan data. At the top is a status bar with icons for signal, Wi-Fi, battery, and the time 12:03. Below this is a header bar with the title "Cambia Data del Prestito" and a clock icon. The main content area contains a table with loan details, followed by input fields for the start and end dates, and an "Aggiorna Date" button. At the bottom is a navigation bar with three icons: a back arrow, a home icon, and a document icon.

Cambia Data del Prestito	
Titolo	I promessi Sposi
Autore	Alessandro Manzoni
Genere	Narrativa
Proprietario	Mario Rossi

Data di inizio del Prestito

Data di fine del Prestito

Aggiorna Date

Figure 11: wireframe CambiaDataPrestito

SEQUENCE DIAGRAM: CambiaDataPrestito

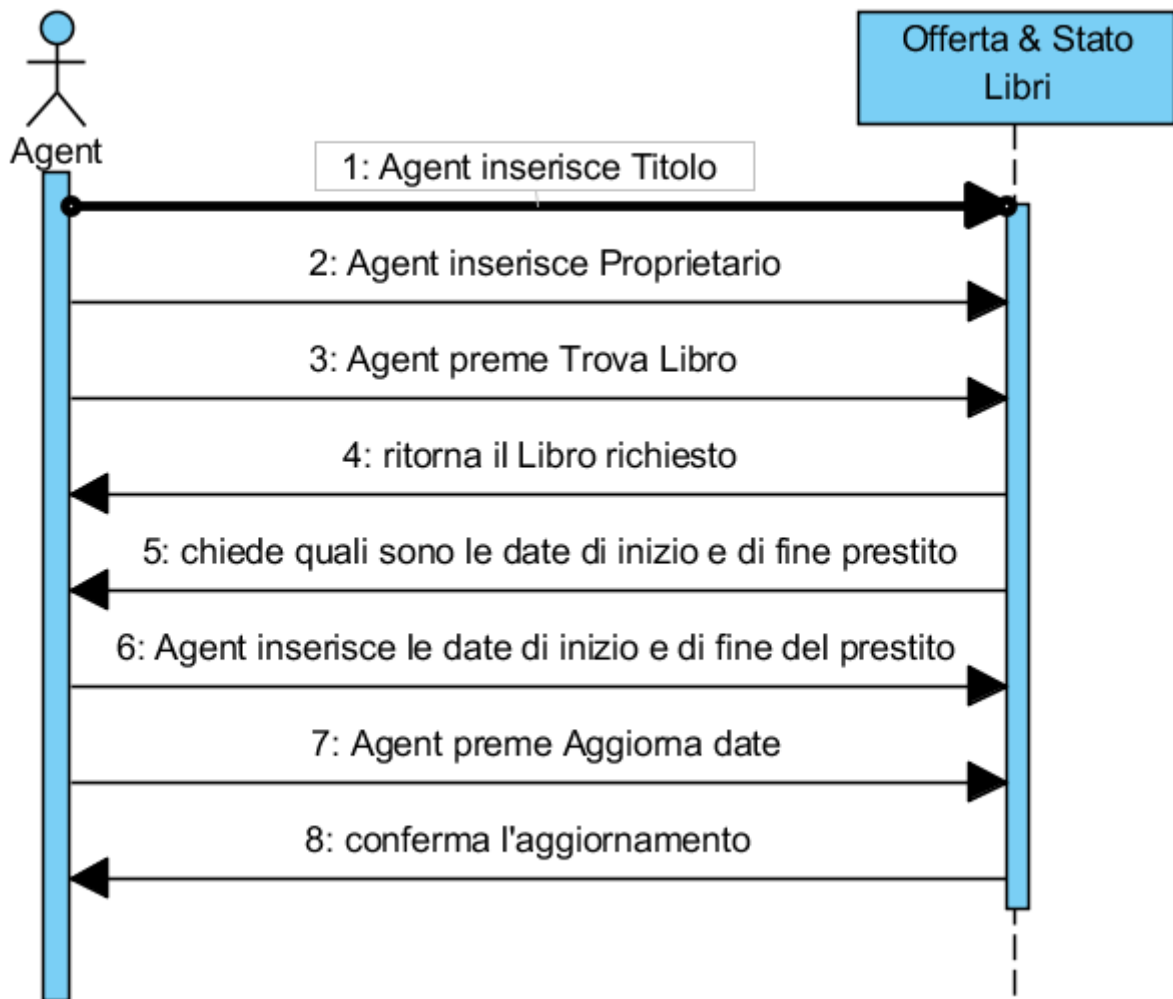


Figure 12: sequence diagram CambiaDataPrestito

1.3.2 Ricerca & Prestito

Il servizio Richiesta&Prestito si occupa di ricevere e salvare le richieste effettuate dagli utenti che possono richiedere un determinato titolo, un determinato autore o uno specifico genere letterario. Inoltre permette di filtrare i risultati ottenuti dalla ricerca tramite i parametri inseriti nella richiesta in modo tale da fornire un solo risultato al lettore. Il filtraggio è effettuato in base all'affidabilità del proprietario del libro, dalla vicinanza del libro al lettore o alla disponibilità del libro.

Le funzionalità offerte dal servizio risultano quindi essere:

- inserimento di una nuova richiesta di libro
- filtraggio dei risultati in base a una determinata priorità

Use Case

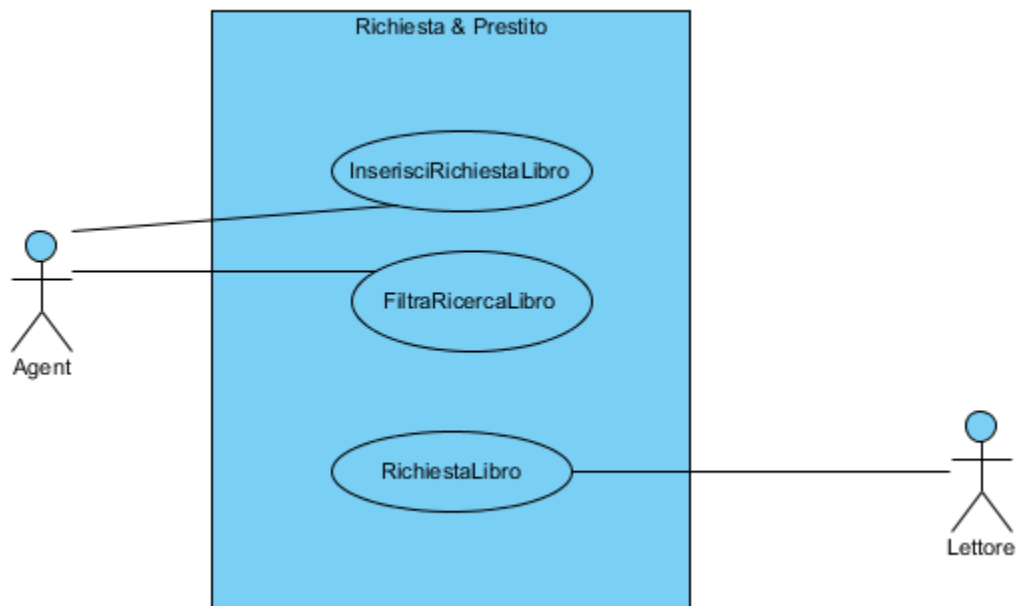


Figure 13: use case Richiesta e Prestito

USE CASE DETTAGLIATI

Nelle sezioni seguenti sono mostrati i dettagli dei casi d'uso, ovverosia quali sono le azioni dell'Agent quando usa i servizi di REP. Oltre al dettaglio dei casi d'uso sono aggiunti i wireframe dell'applicazione, che emulano la user interface delle funzionalità usate dall'Agent.

USE CASE DETAILS: InserisciRichiestaLibro

1. **Agent** inserisce **Autore**
2. **Agent** inserisce **Titolo**
3. **Agent** inserisce **Genere**
4. **Agent** inserisce **Priorità**
5. **if** Priorità="Vicinanza"
 - 5.1. **Agent** inserisce **Luogo****end if**
6. **Agent** preme **Inserisci Richiesta**
7. **SYSTEM** mostra **ID Richiesta**

Figure 14: use case dettagliato InserisciRichiestaLibro

WIREFRAME: InserisciRichiestaLibro

The wireframe illustrates a mobile application screen for submitting a book request. It features a series of input fields for the following information: Autore, Titolo, Genere, Priorità, Vicinanza, and Luogo. A button labeled 'Inserisci Richiesta' is positioned below the input fields. At the bottom of the screen, a status bar displays 'ID RICHIESTA: 1123'. The interface is framed by a dark header and footer containing standard mobile navigation icons.

Autore

Titolo

Genere

Priorità

Vicinanza

Luogo

Inserisci Richiesta

ID RICHIESTA: 1123

Figure 15: wireframe InserisciRichiestaLibro

SEQUENCE DIAGRAM: InserisciRichiestaLibro

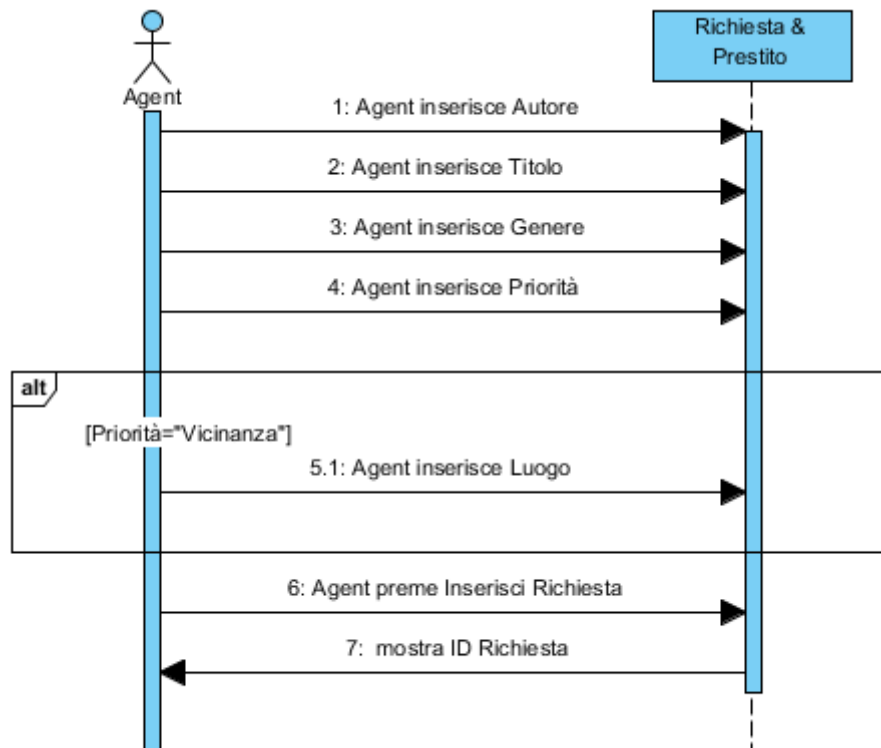


Figure 16: sequence diargam InserisciRichiestaLibro

USE CASE DETAILS: FiltraRicercaLibro

1. **Agent** inserisce **ID Richiesta**
2. **Agent** preme su **Visualizza**
3. **SYSTEM** mostra richiesta
4. **if** Priorità è Vicinanza
 - 4.1. **for each** Libro della lista
 - 4.1.1. **Agent** legge Proprietario
 - 4.1.2. **Agent** legge indirizzo del proprietario
 - 4.1.3. **Agent** calcola la distanza tra **Luogo** e l'indirizzo
 - end for each**
 - 4.2. **Agent** ordina i Libri in base alla vicinanza
 - end if**
5. **Agent** sceglie il primo libro della lista
6. **SYSTEM** visualizza il primo libro della lista al Lettore

Figure 17: use case dettagliato FiltraRicercaLibro

WIREFRAME: FiltraRicercaLibro

12:00

ID RICHIESTA

1123

Visualizza

Autore

J.R.R. Tolkien

Titolo

Il ritorno del re

Genere

Priorità

Vicinanza

Luogo

Pisa

J.R.R. Tolkien	Il ritorno del re	Pisa
J.R.R. Tolkien	Il ritorno del re	Lucca
J.R.R. Tolkien	Il ritorno del re	Pistoia
J.R.R. Tolkien	Il ritorno del re	Bologna

Figure 18: wireframe FiltraRicercaLibro

SEQUENCE DIAGRAM: FiltraRicercaLibro

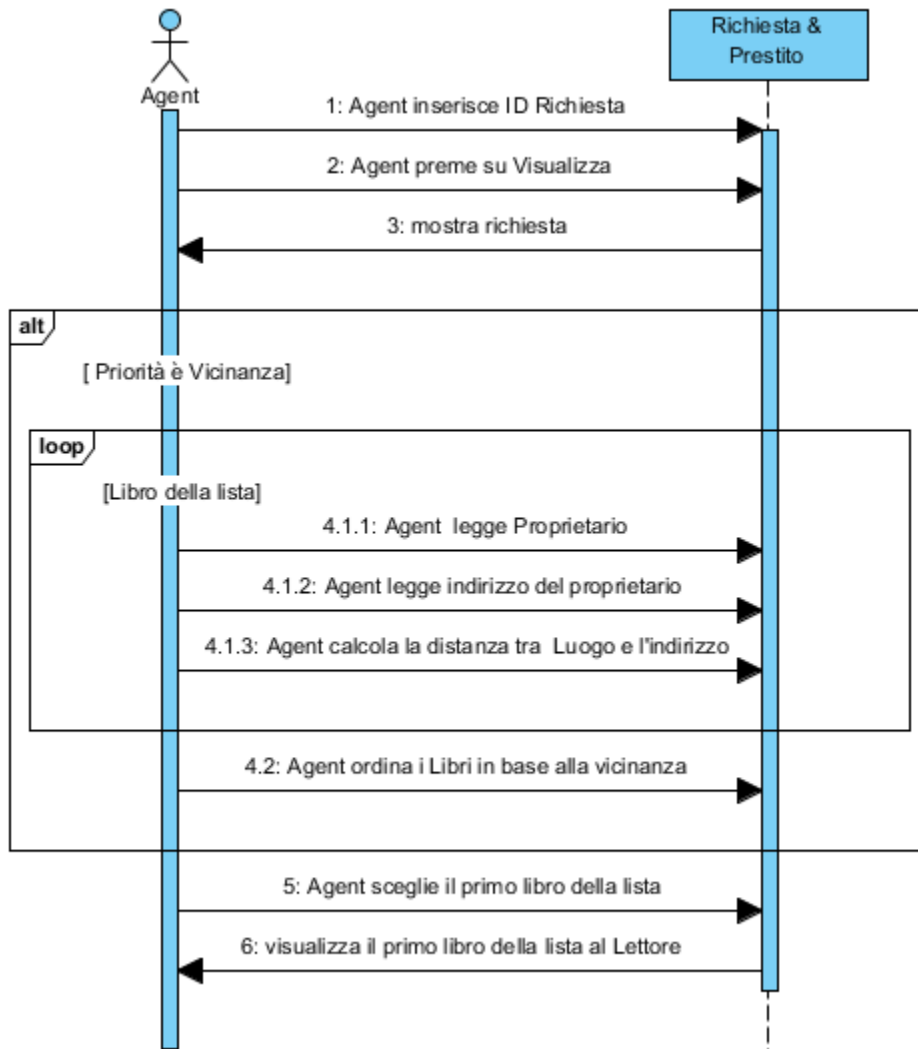


Figure 19: sequence diagram FiltraRicercaLibro

Class Diagram

Le funzionalità offerte da REP sono le seguenti:



Figure 20: class diagram Ricerca e Prestito

I metodi sono:

- **InserisciRichiestaLibro:** permette l'inserimento nel sistema di una richiesta effettuata dal lettore. Vengono specificati il titolo, l'autore o il genere e la priorità che può essere affidabilità, disponibilità o vicinanza. Nell'ultimo caso, è necessario inserire il luogo rispetto al quale calcolare la distanza. Ad inserimento effettuato alla richiesta viene associato un ID univoco
- **FiltraRicercaLibro:** permette di filtrare la lista di libri ottenuta in ingresso in modo tale da ottenere un unico risultato da fornire al lettore. Per fare ciò la lista di libri viene ordinata in base alla priorità specificata al momento dell'inserimento della richiesta, in particolare nel caso questa risulti essere affidabilità o disponibilità, l'ordinamento non è necessario in quanto la lista ricevuta è già ordinata in base a quel parametro. Viene quindi scelto il primo elemento della lista e proposto al lettore.

1.3.3 Appuntamento Per Consegna

Il servizio di Appuntamento Per Consegna(APC) viene utilizzato per la gestione della consegna del libro all'utente che lo ha richiesto in prestito, la consegna ha una doppia modalità di funzionamento: tramite utilizzo di corriere oppure direttamente di persona tra l'utente proprietario del libro e l'utente lettore. L'applicazione contiene le informazioni sulle agende di tutti gli utenti iscritti al sistema, esse vengono aggiornate con gli impegni che gli utenti hanno nel corso delle giornate. Il servizio mette a disposizione le seguenti funzionalità:

- Appuntamento tramite corriere
- Appuntamento di persona

Use case

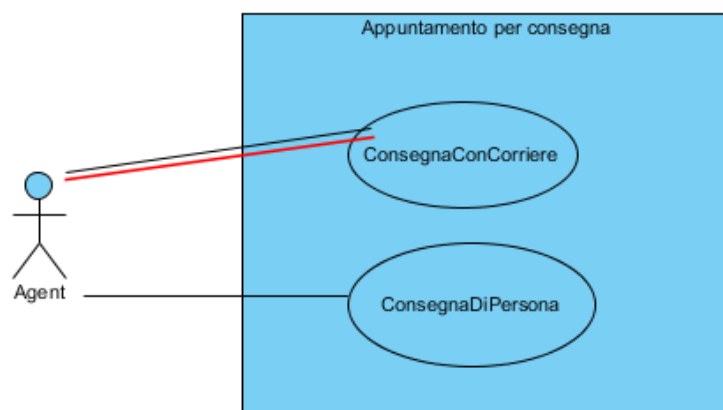


Figure 21: use case Appuntamento Per Consegna

USE CASE DETTAGLIATI

Nelle sezioni seguenti sono mostrati i dettagli dei casi d'uso, ovverosia quali sono le azioni dell'Agent quando usa i servizi di APC. Oltre al dettaglio dei casi d'uso sono aggiunti i wireframe dell'applicazione che emulano la user interface delle funzionalità usate dall'Agent.

Use Case Details: ConsegnaConCorriere






1.  Agent inserisce il nome del mittente
2.  Agent inserisce l'indirizzo del mittente
3.  Agent inserisce il nome del destinatario
4.  Agent inserisce l'indirizzo del destinatario
5.  Agent preme **Conferma Consegna**
6. **SYSTEM** mostra data e ora di ritiro libro
7. **SYSTEM** mostra data e ora di consegna libro
8. **SYSTEM** mostra il prezzo

Figure 22: use case dettagliato ConsegnaConCorriere

Wireframe: ConsegnaConCorriere



nome mittente:

indirizzo mittente:

nome destinatario:

indirizzo destinatario:

Conferma Consegna

data ritiro: ora ritiro:

data consegna: ora consegna:

prezzo:

Figure 23: wirefram Consegna Con Corriere

Sequence Diagram: ConsegnaConCorriere

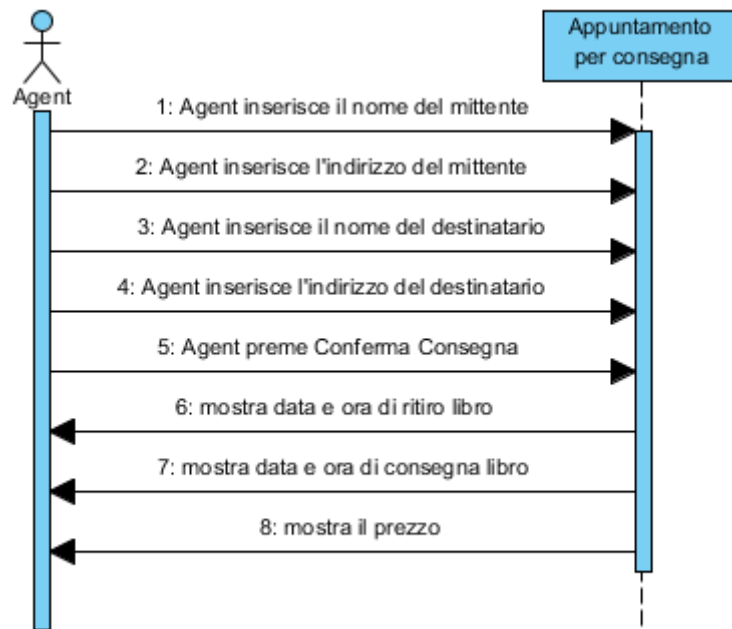


Figure 24: sequence diagram ConsegnaConCorriere

Use Case Details: ConsegnaDiPersona

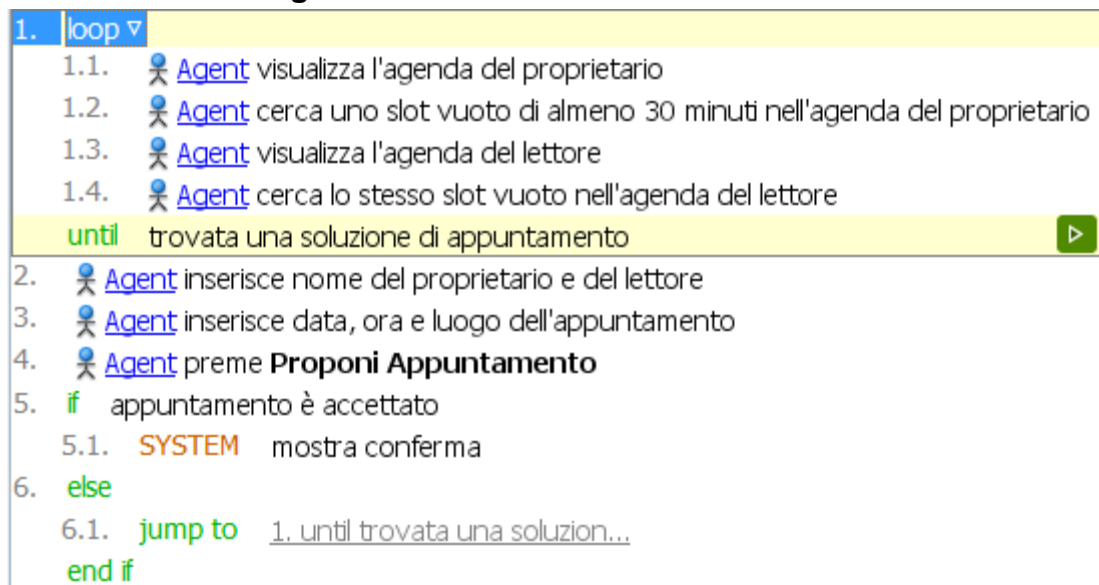


Figure 25: use case dettagliato ConsegnaDiPersona

Wireframe: ConsegnaDiPersona

Agenda di Mario Rossi

	Lunedì 29/05	Martedì 30/05	Mercoledì 31/05	Giovedì 1/06	Venerdì 2/06	Sabato 3/06	Domenica 4/06
9:00	9:00-11:00 Lezione	9:00 - 10:00 Riunione					
10:00					10:00-12:00 Lezione		
11:00							
12:00				12:00 - 14:00 Palestra			
13:00							
14:00							
15:00	15:00 - 16:30 Palestra		15:00-17:00 Lezione				
16:00							
17:00							

Figure 26: wireframe 1 ConsegnaDiPersona

12:00

mittente:

destinatario:

ora appuntamento:

data appuntamento:

luogo appuntamento:

Proponi Appuntamento

Figure 27: wireframe 2 ConsegnaDiPersona

Sequence Diagram: ConsegnaDiPersona

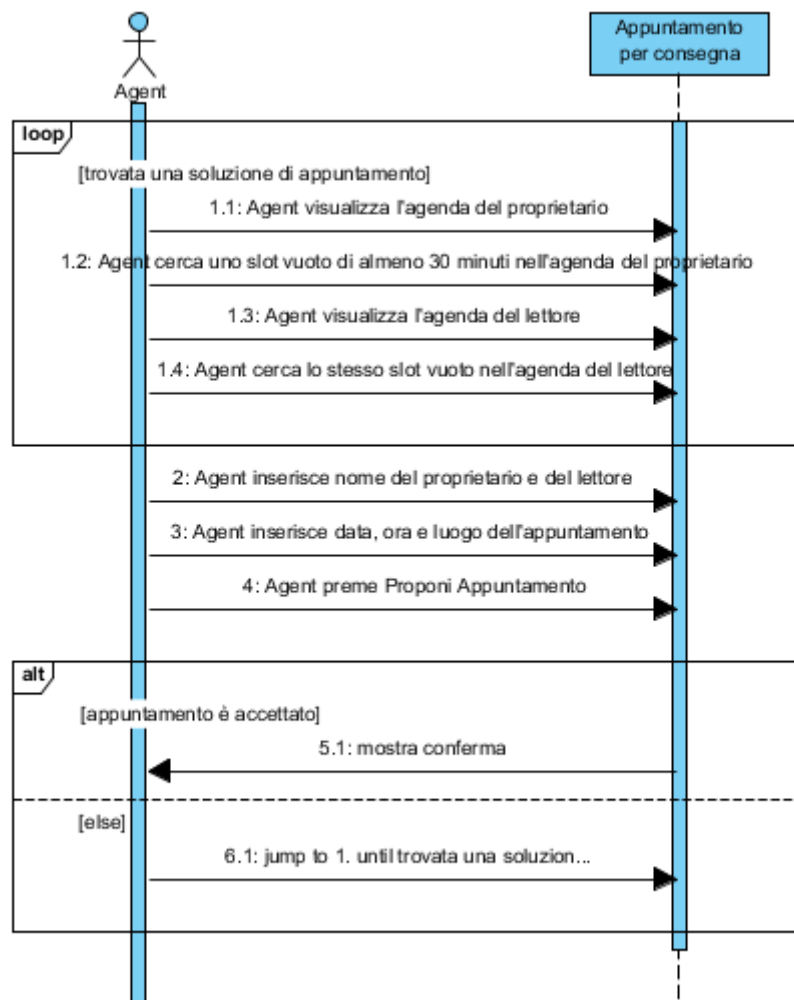


Figure 28: sequence diagram ConsegnaDiPersona

Class Diagram

Le funzionalità offerte da APC sono le seguenti:

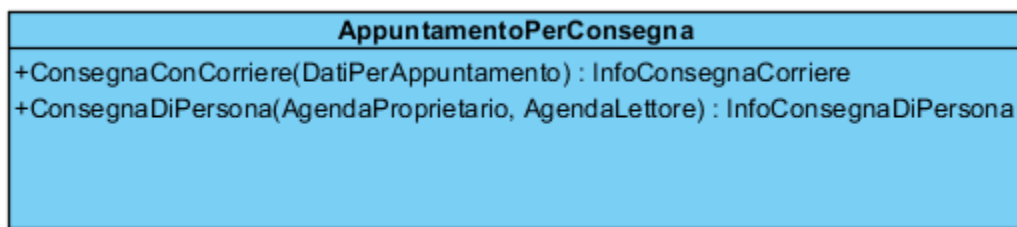


Figure 29: class diagram AppuntamentoPerConsegna

I metodi sono:

- **consegnaConCorriere:** consente di notificare al proprietario del libro ed all'utente che lo ha richiesto in prestito la data e l'orario in cui il corriere passerà dal proprietario a ritirare il libro e la data e l'orario in cui passerà dal lettore a consegnare il libro. Lo stesso processo, invertendo lettore e proprietario, avviene per la riconsegna del libro.
- **consegnaDiPersona:** propone al proprietario del libro ed all'utente che lo ha richiesto in prestito, data, orario e luogo per effettuare lo scambio basandosi sulle agende dei due utenti; in caso la proposta venga rifiutata il servizio propone un'altra data, orario e luogo fin quando gli utenti non abbiano accettato entrambi la medesima soluzione (si presume che una proposta venga accettata).

1.3.4 Recensione Libro/Utente

Il servizio di RecensioneLibroUtente gestisce il Sistema delle recensioni. Attraverso l'interfaccia fornita è possibile inserire nuove recensioni, sia per un libro che per un utente (sia esso lettore o il proprietario) composte da un testo ed un voto sintetico da 1 a 5, e di cercare recensioni riguardante libri o utenti. Il servizio offre le funzionalità di:

- inserimento nuove recensioni;
- lettura recensioni;

Use case

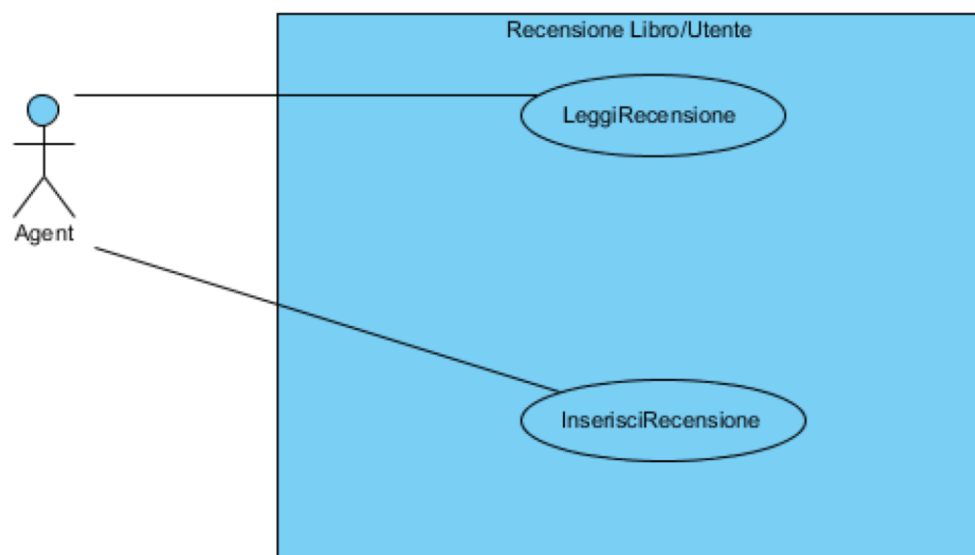


Figure 30: use case Recensione Libro Utente

USE CASE DETTAGLIATI

Nelle sezioni seguenti sono mostrati i dettagli dei casi d'uso, ovverosia quali sono le azioni dell'Agent quando usa i servizi di RLU. Oltre al dettaglio dei casi d'uso sono aggiunti i wireframe dell'applicazione che emulano la user interface delle funzionalità usate dall'Agent.

Use Case Details e Wireframe: LeggiRecensione

```
1. if Agent seleziona Libro
  1.1. Agent inserisce nome Libro
  1.2. if Agent seleziona Leggi Recensioni
    1.2.1. SYSTEM visualizza voto sintetico (stelle da 1 a 5)
    1.2.2. for each Recensione libro
      1.2.2.1. SYSTEM Visualizza recensione libro
    end for each
  end if
2. else if Agent seleziona Proprietario
  2.1. Agent inserisce nome Proprietario
  2.2. Agent seleziona Leggi Recensioni
  2.3. SYSTEM visualizza voto sintetico (stelle da 1 a 5)
  2.4. for each Recensione proprietario
    2.4.1. SYSTEM Visualizza recensioni proprietario
  end for each
  2.5. if Agent seleziona ordina per migliori
    2.5.1. SYSTEM ordina recensioni per punteggio sintetico migliore
  2.6. else
    2.6.1. SYSTEM ordina recensioni per punteggio sintetico peggiore
  end if
end if
```

Leggi recensione

☒ Libro

☐ Proprietario

Ordina per: ☒ migliori ☐ peggiori

Recensioni proprietario

☐ Recensioni libro

Figure 31: use case dettagliato e wireframe LeggiRecensione

Sequence Diagram: LeggiRecensione

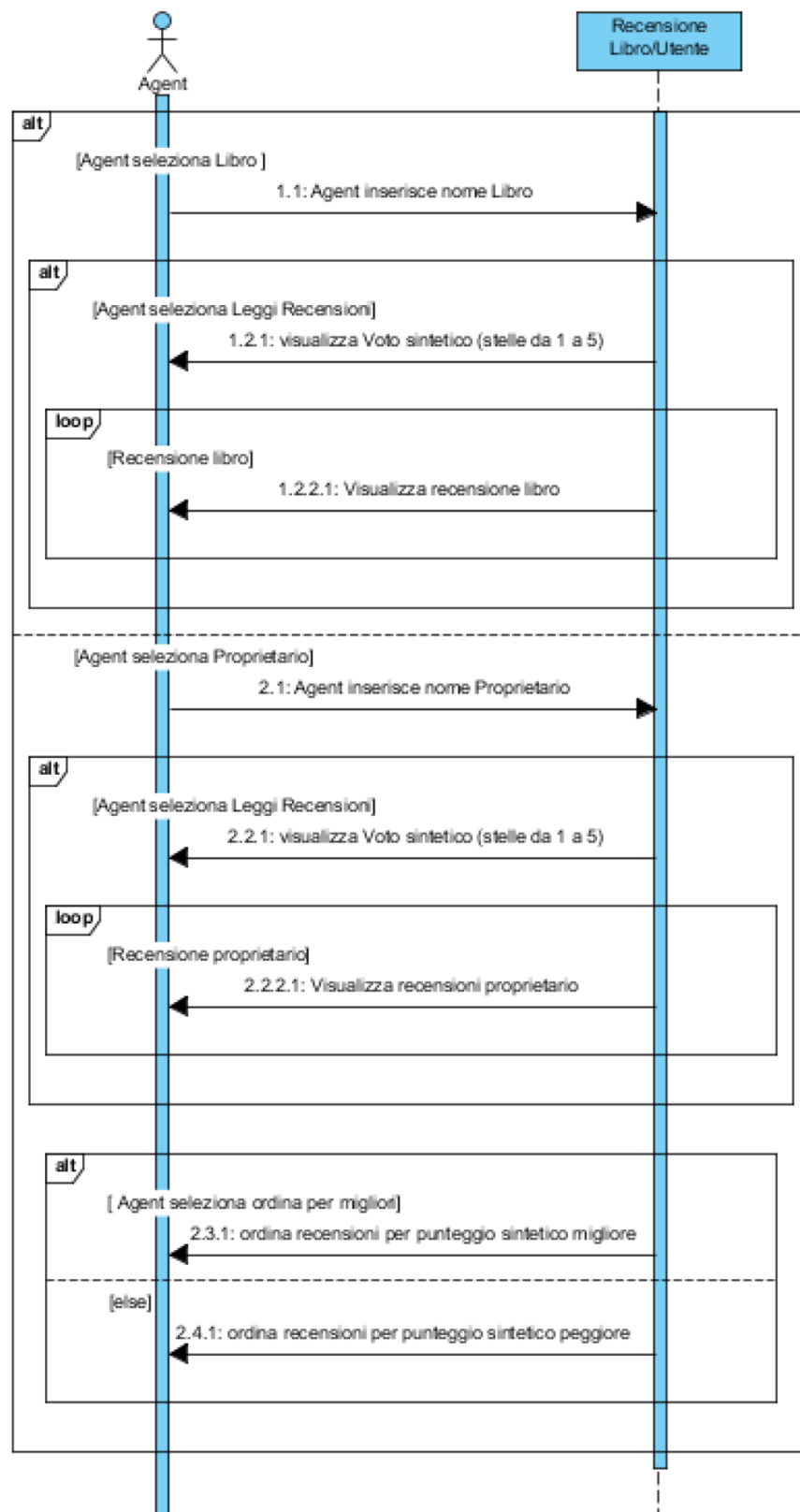


Figure 32: sequence diagram LeggiRecensione

Use Case Details e Wireframe: InserisciRecensione

1. **Agent** seleziona **Libro** o **Proprietario**
2. **Agent** inserisce valutazione sintetica
3. **Agent** scrive la recensione
4. **Agent** clicca su **Invia Recensione**
5. **SYSTEM** Visualizza Recensione inserita

Inserisci recensione

☒ Libro

☐ Proprietario

☆☆☆☆☆

Invia recensione

Figure 33: use case dettagliato e wireframe InserisciRecensione

Sequence Diagram: InserisciRecensione

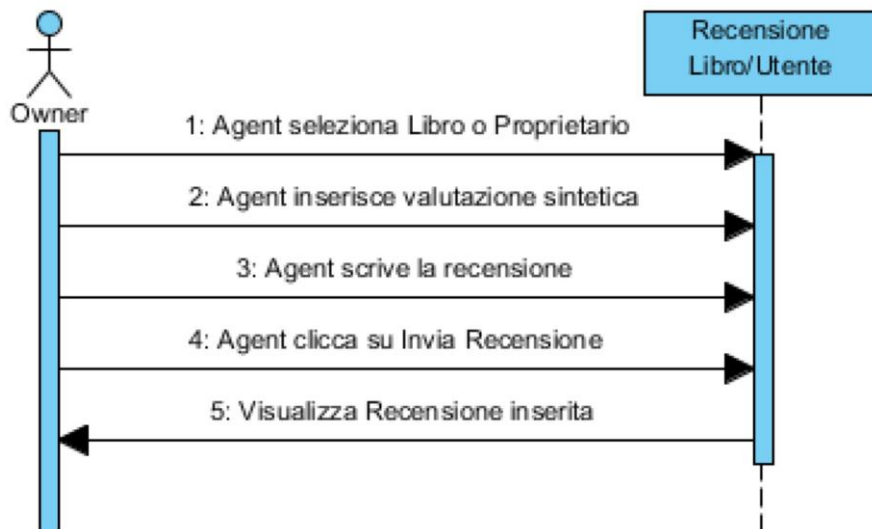


Figure 34: sequence diagram InserisciRecensione

Class Diagram

Le funzionalità offerte da RLU sono le seguenti:

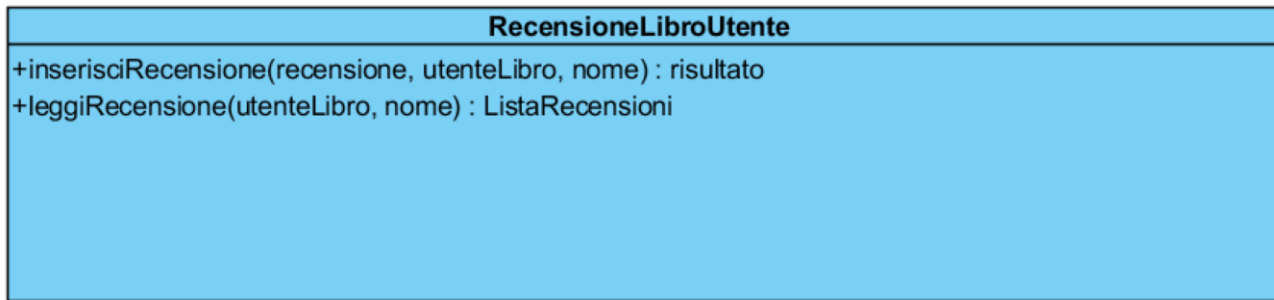


Figure 35: class diagram *RecensioneLibroUtente*

I metodi sono:

- **LeggiRecensione:** questo metodo prende come parametri un valore che discrimina la tipologia di recensione da cercare (utente o libro), la parola chiave (il nome del libro o dell'utente). L'output è una lista di recensioni, e un voto sintetico che rappresenta la media dei voti
- **InserisciRecensione:** questo metodo prende come parametri una Recensione, un valore che discrimina la tipologia di recensione da cercare (utente o libro) e la parola chiave (il nome del libro o dell'utente). L'output di questo metodo è una stringa di conferma di avvenuto inserimento della recensione, oppure un fault

1.4 Product Functions

A&R permette di richiedere un libro, la ricerca con determinati parametri, il filtraggio dei risultati in base ai parametri richiesti, la selezione del metodo di consegna e l'inserimento delle recensioni in maniera completamente automatizzata, in modo da permettere di eseguire efficientemente la condivisione dei libri.

L'applicativo, partendo da alcuni parametri scelti dal lettore, inserisce la richiesta tramite *Richiesta&Prestito*. Con quegli stessi parametri viene effettuata una ricerca all'interno di *Offerta&StatoLibro*, ottenendo in questo modo una lista di libri che soddisfino tali parametri. Questi risultati vengono quindi filtrati tenendo conto della priorità inserita dal lettore, in modo tale da estrapolare un singolo risultato che sarà proposto attraverso *Ricerca&Prestito*. Nel caso in cui il lettore accetti tale proposta e il libro sia disponibile, attraverso *Appuntamento per Consegna* vengono definite la modalità di consegna e le informazioni relative ad essa, per il libro scelto. In ogni caso lo stato del libro viene modificato contestualmente alla situazione, quindi nel caso in cui risulti già in prestito, lo stato cambierà in prenotato e non sarà più possibile ricavarlo. Nel caso invece in cui il libro sia disponibile, lo stato sarà modificato in "in prestito". Al momento della scadenza del prestito, sarà eseguita nuovamente la stessa procedura per la restituzione del libro. Infine, tramite *RecensioneLibroUtente* vengono recensiti il proprietario del libro e il libro stesso.

1.5 User characteristics

Il sistema è basato su un'interfaccia semplice ed intuitiva, non sono richieste particolari abilità dell'utente se non quelle di compilare i campi testuali ed eseguire delle scelte. Il sistema è automatizzato eccetto per l'interazione con l'utente, che potrebbe non accettare il libro proposto dal software o non dare la disponibilità per una determinata data di consegna.

1.6 Constraints

A&R usa gli applicativi OSL, RLU, APC e REP attraverso le interfacce fornite.

1.7 Assumptions and dependencies

- Non è previsto che attraverso APC non sia possibile determinare una data comune per la consegna di persona del libro
- Si suppone per semplicità che sia presente una recensione per ogni utente e per ogni libro in RLU
- Si suppone per semplicità che siano già presenti alcuni libri all'interno del catalogo di OSL
- Si suppone per semplicità che le agende di ogni utente siano già presenti all'interno di APC

1.8 Specifica dei requisiti

1.8.1 Interfaccia utente

- OSL
 - TrovaLibro: costituito da tre campi, ed un pulsante di ricerca
 - CambiaStatoLibro: costituito da due campi, Titolo, Proprietario ed un pulsante TrovaLibro
 - CambiaDataPrestito: costituito da due campi, Titolo, Proprietario ed un pulsante TrovaLibro
- REP
 - InserisciRichiestaLibro: costituita da cinque campi, Autore, Titolo, Genere, Priorità (Vicinanza, Affidabilità, Disponibilità), Luogo ed un pulsante Inserisci Richiesta
 - FiltraRicercaLibro: costituita da un campo ID Richiesta ed un pulsante Visualizza

- APC
 - ConsegnaConCorriere: costituita da quattro campi, nome mittente, indirizzo mittente, nome destinatario, indirizzo destinatario ed un -pulsante Conferma Consegna
 - ConsegnaDiPersona: costituita da cinque campi, mittente, destinatario, ora appuntamento, data appuntamento, luogo appuntamento ed un pulsante Proponi appuntamento

- RLU
 - LeggiRecensione: costituita da due radio button Libro, Proprietario e Ordina per migliori, peggiori, ed un pulsante Leggi Recensioni
 - InserisciRecensione: costituita da un radio button Libro, Proprietario, un ranking bar, un campo testo per inserire la recensione ed un pulsante Invia Recensione

1.8.2 Requisiti funzionali

ID	Descrizione
SR1	Il sistema deve ricevere una richiesta di inserimento libro
SR2	Il sistema deve ricevere il risultato dall'inserimento richiesta tramite il servizio REP
SR3	Il sistema deve ricevere una lista di libri che rispetti i criteri di ricerca dal processo Trova Libro tramite il servizio OSL
SR4	Il sistema deve ricevere il risultato del processo Filtra Ricerca Libro tramite il servizio REP
SR5	Il sistema deve ricevere il risultato del processo Leggi Recensione tramite il servizio RLU
SR6	Il sistema deve ricevere il risultato del processo Cambia Stato Libro tramite il servizio OSL
SR7	Il sistema deve ricevere il risultato del processo Consegna con Corriere tramite il servizio APC

SR8	Il sistema deve ricevere il risultato del processo Consegna di Persona tramite il servizio APC
SR9	Il sistema deve ricevere il risultato del processo Inserisci Recensione tramite il servizio RLU
SR10	Il sistema deve ricevere il risultato del processo Cambia Data Prestito tramite il servizio OSL

1.8.3 Interfaccia software

Il sistema A&R non deve interfacciarsi con nessun sistema software.

1.8.4 Requisiti di prestazioni

Sono richiesti tempi di risposta contenuti sia per le operazioni di prenotazione/riciesta sia per l'inserimento/riciesta delle recensioni.

1.8.5 Multiutenza

Le informazioni trattate dal sistema A&R devono poter essere accessibili da diversi dispositivi. Ciascun utente deve poter accedere alle funzionalità del sistema e gestire i libri, limitatamente alla propria Libreria.

1.8.6 Sicurezza

Il sistema A&R tratta informazioni sensibili che devono essere accessibili solamente ad un personale autorizzato. Un utente potrà accedere solo ed esclusivamente ai propri dati personali (indirizzi, agenda), per fare ciò sarà necessario fornire la password personale, comunicata all'atto della registrazione al servizio.

1.8.7 Attributi del sistema software

L'attività di manutenzione (software maintenance) comprende tutti gli interventi di modifica sul software, successivi alla consegna, effettuati allo scopo di:

- correzione degli errori presenti sin dalla fase di sviluppo o introdotti con manutenzioni successive (correttivi);
- aggiunta di nuove funzioni e miglioramento delle funzioni già presenti(perfettivi);
- adattamento ai cambiamenti software intervenuti nell'ambiente in cui è inserito (adattativi);
- apporto di miglioramenti per evitare problemi futuri (preventivi).

1.8.8 Altri requisiti

- **Database:** Il sistema A&R richiede uso di un RDBMS (Relational DataBase Management System), deve presentare i dati all'utente sotto forma di relazioni (tabelle).

2. Software Project Management Plan

2.1 Introduzione

2.1.1 Descrizione generale di progetto

Il progetto A&R Rental Book ha lo scopo di automatizzare e migliorare l'efficienza di un portale per la condivisione e scambio di libri tra utenti. Il software che deve essere sviluppato consiste in un modulo di coordinamento, il quale deve interagire con quattro servizi già presenti nel portale. Tramite il coordinamento di questi servizi, il sistema sarà in grado di prendere decisioni in maniera autonoma con lo scopo di ridurre costi, tempi di esecuzione e di attesa.

2.1.2 Project deliverables

Durante lo sviluppo del progetto verranno prodotti una serie di documenti. Tutti questi documenti saranno scritti in accordo agli standard ESA (software engineering & standardisation). L'intera documentazione sarà consegnata un giorno prima dell'esame.

2.1.3 Definizioni

Acronimo	Descrizione
PM	Project Manager
TS	Tool Specialist
L	Librarian
QE	Quality Engineer

2.2 Organizzazione di progetto

2.2.1 Gruppo di lavoro

Nome	Ruolo
Cascone Livio	Project Manager
Peraz Giuliano	Tool Specialist
Gallina Cristina	Librarian
Tanferna Valerio	Quality Engineer

2.2.2 Modello di processo

Data la pochissima esperienza del team nel campo di sviluppo ed integrazione software e la relativa semplicità del progetto da affrontare è stato scelto come modello di processo il modello waterfall con due cicli di iterazione al fine di poter correggere gli errori commessi durante il primo ciclo.

Il processo è diviso, quindi, in due cicli ripetuti sequenzialmente, ciascuno di essi costituito dalle stesse quattro fasi:

- Analisi dei requisiti
- Design
- Implementazione & Integrazione
- Testing

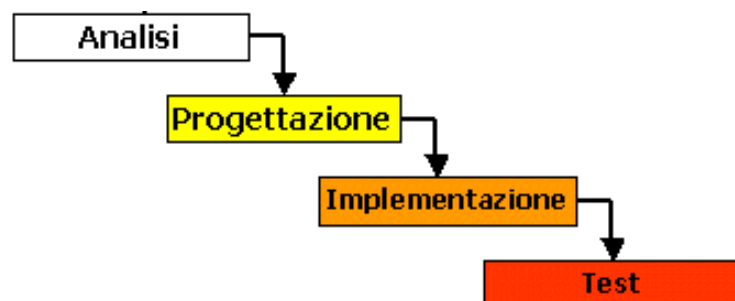


Figure 36: modello di processo

2.2.3 Responsabilità di progetto

Ruolo	Task
PM	Responsabile per lo sviluppo del processo, decide il modello di processo, coordina gli altri membri del team ed assegna loro i compiti
TS	Responsabile per l'utilizzo dei software e sostituisce il PM quando necessario
L	Responsabile per l'intera documentazione di progetto
QE	Responsabile per la qualità di progetto, collabora con il librarian per la gestione dei documenti o lo sostituisce quando necessario

2.2.4 Strumenti Utilizzati

- Visual Paradigm
- SoapUI
- Apache TomCat
- Hangouts
- Google docs
- Google Drive
- NetBeans
- SystemStar
- Notepad++

2.2.5 Funzionalità di supporto al progetto

Oltre al Project Management sono presenti due ulteriori funzionalità di management:

- Testing
 - L'attività di Software Testing controlla il software rispetto le sue specifiche. QE e TS collaborano alla stesura del Software Testing Plan (STP) nel quale saranno descritte tutte le attività di testing svolte
- Quality Assurance
 - Software Quality Assurance Plan (SQAP) viene redatto da QE e L e definisce tutte le azioni necessarie a fornire un prodotto conforme ai requisiti con un certo margine di sicurezza

2.2.6 Obiettivi e priorità

Gli obiettivi sono quelli di consegnare il prodotto entro i tempi prestabiliti ed ottenere una qualità di prodotto alta. Il PM ed il QE collaborano per raggiungere questi obiettivi rispettivamente controllando che il processo prosegua secondo i tempi previsti e monitorando la qualità del prodotto dopo ogni fase.

2.2.7 Assunzioni, Dipendenze e Vincoli

Le assunzioni di progetto sono le seguenti:

- A. Il gruppo è composto da 4 membri
- B. Tutti i software necessari allo sviluppo di A&R Rental Book sono disponibili
- C. A&R Rental Book ha 4 diverse applicazioni da gestire:
 - a. **Offerta & Stato Libro** application
 - b. **Richiesta & Prestito** application
 - c. **Appuntamento per consegna** application
 - d. **Recensione Libro/Utente** application

Le dipendenze di progetto sono le seguenti:

- A. A&R Rental Book interagisce attraverso le altre applicazioni attraverso le loro interfacce, se queste interfacce vengono modificate A&R Rental Book potrebbe avere un comportamento inatteso

2.3 Risk Management

Questa sezione individua e descrive una serie di possibili rischi per lo sviluppo del progetto. Inoltre sono descritte le azioni per l'individuazione, controllo ed eliminazione dei rischi individuati.

2.3.1 Identificazione dei rischi

Sono state identificate cinque categorie di rischi:

- 1. Tool risks
 - a. Il codice auto-generato potrebbe contenere inaccuratezze
 - b. Il codice auto-generato potrebbe contenere inefficienze
 - c. Alcuni programmi di sviluppo potrebbero non essere gratuiti
- 2. People risks:
 - a. I membri del gruppo potrebbero non avere l'esperienza necessaria per sviluppare il progetto
- 3. Organizational risks:

- a. Il gruppo potrebbe avere delle difficoltà a lavorare costantemente al progetto durante il periodo di esami
- 4. Requirements risks:
 - a. A causa della mancanza di esperienza dei membri del team, è possibile che le definizioni dei requisiti possano essere inaccurate
 - b. Il cliente potrebbe richiedere requisiti aggiuntivi in seguito
- 5. Estimation risks:
 - a. Il tempo necessario a completare il progetto potrebbe essere sottostimato
 - b. La dimensione del progetto potrebbe essere sottostimata

2.3.2 Analisi dei rischi

Per ognuno dei rischi precedentemente individuati specifichiamo probabilità (Bassa - Media - Alta) e l'effetto (Insignificante - Tollerabile - Grave - Molto Grave):

Rischio	Probabilità	Effetto
Il codice auto-generato potrebbe contenere inaccurately	Media	Tollerabile
Il codice auto-generato potrebbe contenere inefficienze	Media	Insignificante
Alcuni programmi di sviluppo potrebbero non essere gratuiti	Media	Insignificante
I membri del gruppo potrebbero non avere l'esperienza necessaria per sviluppare il progetto	Bassa	Molto Grave
Il gruppo potrebbe avere delle difficoltà a lavorare costantemente al progetto durante il periodo di esami	Alta	Molto Grave
A causa della mancanza di esperienza dei membri del team, è possibile che le definizioni dei requisiti possano essere inaccurate	Media	Grave
Il cliente potrebbe richiedere requisiti aggiuntivi in seguito	Bassa	Grave
Il tempo necessario a completare il progetto potrebbe essere sottostimato	Bassa	Grave
La dimensione del progetto potrebbe essere sottostimata	Bassa	Grave

2.3.3 Pianificazione dei rischi

Per ognuno dei rischi, che sono stati precedentemente individuati, si riporta una strategia per evitare o diminuire gli effetti:

Rischio	Strategia
Il codice auto-generato potrebbe contenere inaccuratezze	Utilizzare programmi professionali ed efficienti come visual paradigm
Il codice auto-generato potrebbe contenere inefficienze	Utilizzare programmi professionali ed efficienti come visual paradigm
Alcuni programmi di sviluppo potrebbero non essere gratuiti	Utilizzare la versione di prova
I membri del gruppo potrebbero non avere l'esperienza necessaria per sviluppare il progetto	Prima di iniziare ogni fase del progetto prevedere un breve periodo di studio
Il gruppo potrebbe avere delle difficoltà a lavorare costantemente al progetto durante il periodo di esami	Concordare, in anticipo, con ogni membro del gruppo ore e giorni di lavoro da dedicare al progetto
A causa della mancanza di esperienza dei membri del team, è possibile che le definizioni dei requisiti possano essere inaccurate	Evitare di procedere alla fase di analisi prima di essere sufficientemente sicuri dei requisiti scritti
Il cliente potrebbe richiedere requisiti aggiuntivi in seguito	Effettuare una o più interviste con il cliente per accertarsi di aver compreso tutti i requisiti richiesti
Il tempo necessario a completare il progetto potrebbe essere sottostimato	Seguire ogni fase e rivedere il tempo di lavoro programmato in accordo al modello di sviluppo waterfall
La dimensione del progetto potrebbe essere sottostimata	Seguire ogni fase e rivedere il tempo di lavoro programmato in accordo al modello di sviluppo waterfall

2.3.4 Monitoraggio dei rischi

Per ognuno dei rischi precedentemente individuati riportiamo alcuni possibili indicatori che potrebbero servire ad individuare e prevenire il rischio:

Rischio	Possibile Indicatore
Il codice auto-generato potrebbe contenere inaccuratezze	Parti mancanti del codice. Alcune funzionalità devo essere riscritte

Il codice auto-generato potrebbe contenere inefficienze	Alcune funzionalità devo essere riscritte
Alcuni programmi di sviluppo potrebbero non essere gratuiti	“Trial Version” nella Home Page del programma
I membri del gruppo potrebbero non avere l’esperienza necessaria per sviluppare il progetto	Il progetto prosegue per tentativi
Il gruppo potrebbe avere delle difficoltà a lavorare costantemente al progetto durante il periodo di esami	Il progetto prosegue lentamente
A causa della mancanza di esperienza dei membri del team, è possibile che le definizioni dei requisiti possano essere inaccurate	Requisiti incompleti, inconsistenti od in conflitto
Il cliente potrebbe richiedere requisiti aggiuntivi in seguito	Il cliente richiede di aggiungere dei requisiti
Il tempo necessario a completare il progetto potrebbe essere sottostimato	Il piano di sviluppo originale non viene rispettato
La dimensione del progetto potrebbe essere sottostimata	Il piano di sviluppo originale non viene rispettato

2.4 Analisi costi e tempi di sviluppo

In questa sezione verrà stimato il costo del software sia in termini di tempo che economici usando l’approccio function-point.

Questo metodo consente di misurare le funzioni del sistema durante la fase di analisi indipendentemente dal tipo di linguaggio di programmazione che verrà poi utilizzato in fase di programmazione. Il modello si basa sulla stima dei requisiti di sistema e sulle opzioni di design nonché sulle capacità del team di sviluppo sotto diversi aspetti.

Il costo del software verrà stimato attraverso l’aiuto del tool automatico SystemStar utilizzando il modello COCOMO II 2000.

2.4.1 Function Point technique

Secondo lo standard ISO/IEC 20926:2009 la tecnica dei function point è in grado di stimare il numero di linee di codice(LoC) del software necessarie.

La function point analysis consiste, inizialmente, nel determinare gli elementi fondamentali che costituiscono la visione esterna del sistema:

- External Input (EI)
- External Queries(EQ)
- External Output (EO)
- Logical Internal File (ILF)
- External Interface File (EIF)

Ogni elemento può essere classificato, secondo la sua complessità, in tre differenti modi: Simple, Average e High secondo le seguenti tabelle:

External Input (EI):

EI	1-4 DET	5-15 DET	16+ DET
0-1 FTR	Simple	Simple	Average
2 FTR	Simple	Average	High
3-4 FTR	Average	High	High

External Output(EO) / External Queries(EQ):

EO/EQ	1-5 DET	6-19 DET	20+ DET
0-1 FTR	Simple	Simple	Average
2 FTR	Simple	Average	High
3-4 FTR	Average	High	High

Internal logical file(ILF) / External Interface file(EIF):

ILF/EIF	1-19 DET	20-50 DET	51+ DET
1 RET	Simple	Simple	Average
2-5 RET	Simple	Average	High
6+ RET	Average	High	High

Dove DET, FTR e RET rappresentano rispettivamente:

- DET: Data Element Type

- FTR: File Type Referenced (IEF/EIF)
- RET: Record Element Type

Attraverso i pesi(w) assegnati ad ogni elemento, possiamo determinare la UFP (Unadjusted Function Point), la quale fornisce un'indicazione della complessità del sistema in termini di funzionalità secondo la seguente tabella:

Function	Complexity Weight (W)		
	Simple	Average	High
EI	3	4	6
EO	4	5	7
ILF	7	10	15
EIF	5	7	10
EO	3	4	6
Total	UFP Simple	UFP Medium	UFP High

I fattori di aggiustamento sono dei fattori che influenzano le caratteristiche generali del sistema e sono basati su 14 aspetti:

1. Data communication
2. Distributed data processing
3. Performance
4. Heavily used configuration
5. Transaction rate
6. Online data entry
7. End user efficiency
8. Online update
9. Complex processing
10. Reusability

- 11. Installation ease
- 12. Operational ease
- 13. Multiple sites
- 14. Facilitate change

Per ognuno di questi fattori si assegna un valore da 0 a 5 (Grado di Influenza, DI) in basa alla loro influenza nel sistema secondo la seguente tabella:

DI	Influenza
0.0	Nessuna
1.0	Insignificante
2.0	Moderata
2.5	Intermedia
3.0	Media
4.0	Significativa
5.0	Forte

2.4.2 Function Point analisi

Use Case	External Input (EI)		
	DET	Complexity	UFP
A&R Rental Book			
Prestito Libro	1	Simple	3
Totale	1	-	3

Use Case	External Output (EO)		
	DET	Complexity	UFP
A&R Rental Book			
Prestito Libro	1	Simple	4
Totale	1	-	4

Use Case	External Interface file (EIF)		
	DET	Complexity	UFP
A&R Rental Book			
Prestito Libro	4	Simple	5
Totale	4	-	20

Use Case	External Queries(EQ)		
	DET	Complexity	UFP
A&R Rental Book non utilizza un DB esterno e non invia query esterne			

Fattori di aggiustamento:

Fattore	Valore
Data communication	4
Distributed function	3
Performance	3
Heavily used configuration	0
Transaction rate	2
Online data entry	0

End user efficiency	2
Online update	0
Complex processing	0
Reusability	1
Installation ease	0
Operational ease	4
Multiple sites	0
Facilitate change	0
Total	19

2.4.3 COCOMO Analisi

L'analisi tramite il tool COCOMO consente di stimare il numero di persone-ora richieste per sviluppare il progetto; questa quantità è usata per determinare il tempo di sviluppo ed i costi di progetto.

L'analisi ha come input una serie di fattori:

Scale Driver:

1. **PREC (Precedentedness):** indica l'esperienza del team in progetti simili
2. **FLEX (Development Flexibility):** indica la flessibilità dei requisiti
3. **RESL (Architecture Risk resolution):** indica l'architettura ed il livello di rischio del software
4. **TEAM (Team cohesion):** indica il livello di coesione del team
5. **PMAT (Process Maturity):** indica il livello di conoscenza dei tool utilizzati

SCALE DRIVER	VALORE
PREC	Somewhat Unprecedented
FLEX	Some Relaxation
RESL	Little
TEAM	Basically Cooperative
PMAT	Lower Half

Cost Driver:

COST DRIVER	DESCRIZIONE	VALORE
ACAP	Analyst capability to develop the project	High
APEX	Team experience in this context	Low
PCAP	Programmer capability	Very High
PLEX	Team experience in the specific platform	Very High
LTEX	Programming languages and tool experience	Very High
PCON	Annual turnover of organization	Very High
TOOL	Use of software tools	Very High
SITE	Multi site development	Low
SCED	Development schedule	Nominal
TIME	Execution time constraint	Nominal
STOR	Main storage constraint	Nominal
PVOL	Platform volatility	Low
RELY	Required reliability	Nominal
DATA	Database size	Low
CPLX	Product complexity	Very Low
RUSE	Reusability requirement	Low
DOCU	Documentation match to lifecycle needs	Very Low

2.4.3 Risultati COSTAR**Report effort:**

Effort per Component (Person-Hours)					
Component Name	RQ	PD	DD	CT	IT
Component1	0	41	65	90	46
Effort Summary					
Component Totals	0	41	65	90	46
Grand Total	0	41	65	90	46

Report Cost:

Cost per Component (K\$)						Total
Component Name	RQ	PD	DD	CT	IT	RQ to IT
Component1	0.0	0.4	0.7	0.9	0.5	2.4
Cost Summary						
Component Totals	0.0	0.4	0.7	0.9	0.5	2.4
Grand Total	0.0	0.4	0.7	0.9	0.5	2.4

3. Quality Plan

3.1 Introduzione

Lo scopo di questo documento è quello di fornire metodi, standard e procedure usate nello sviluppo del software. In questa sezione si fa riferimento allo standard ISO/IEC 9126 costituito da una serie di attributi, ai quali è assegnato un peso in base all'influenza sulla qualità del progetto finale.

Il peso di un attributo viene assegnato in base alla seguente scala di valori:

- Basso
- Medio
- Alto

Questa metrica è stata usata per avere un sistema intuitivo per la definizione della priorità degli interventi che devono essere realizzati al fine di garantire e applicare tale modello di qualità.

Caratteristica	Sotto caratteristica	Peso
Funzionalità	Adeguatezza, interoperabilità	Alto
Affidabilità	Tolleranza agli errori	Medio
Usabilità	Apprendibilità	Alto
Efficienza	Comportamento temporale	Medio
Manutenibilità	Modificabilità	Medio

Portabilità	Adattabilità	Alto
-------------	--------------	------

3.2 Funzionalità

ID	Titolo	Descrizione	Metrica	Valore
QC1	Adeguatezza	A&R deve fornire tutti i set di funzioni specificati dall'utente	Numero di requisiti non rispettati	0
QC2	Interoperabilità	A&R deve essere in grado di comunicare con gli altri servizi minimizzando gli errori	(Numero totale di pacchetti correttamente inviati)/(Numero totale di pacchetti inviati)	<1%

3.3 Affidabilità

ID	Titolo	Descrizione	Metrica	Valore
QC3	Robustezza	A&R deve essere capace di evitare i fault	Numero di fault rilevati/numero di esecuzioni	<5%

3.4 Usabilità

ID	Titolo	Descrizione	Metrica	Valore
QC4	Apprendibilità	L'uso di A&R deve essere di facile comprensione	Numero di chiamate/numero mail all'assistenza	<1%

3.5 Efficienza

ID	Titolo	Descrizione	Metrica	Valore
QC5	Comportamento temporale	Capacità del software di soddisfare determinati tempi di risposta	Tempo di risposta	<2s

3.6 Manutenibilità

ID	Titolo	Descrizione	Metrica	Valore
QC6	Modificabilità	Capacità del software di permettere l'implementazione (progetto, codifica e documentazione) di una data modifica	Tempo medio di ogni modifica	<2 ore

3.7 Portabilità

ID	Titolo	Descrizione	Metrica	Valore
QC7	Adattabilità	Capacità del software di essere trasferito.	Numero di piattaforme supportate	N.A.

3.8 Process quality

L' ISO 12207 è lo standard che definisce tutti i task richiesti per lo sviluppo e la manutenzione di un software di qualità. Lo standard definisce un ciclo di vita del software, inclusi processi e attività applicate durante l'acquisizione e la configurazione del servizio del sistema.

L'obiettivo principale dello standard è definire un set comprensivo di processi che coprono l'intero ciclo di vita del software.

I processi adottati nello sviluppo sono classificati in tre tipi:

- Processo primario, attività relativa allo sviluppo del software
- Processi di supporto, attività relativa alla gestione della documentazione ed al processo di controllo di qualità
- Processi organizzativi, attività relativa alla gestione delle risorse

Processo primario:

- Acquisizione (gestione dei subfornitori)
- Fornitura (rapporti con il cliente)
- Sviluppo di sistemi software
- Gestione operativa (install. E servizi)
- Manutenzione

Processi di supporto:

- Documentazione del prodotto
- Gestione di versioni e configurazioni
- Assicurazione della qualità
- Validazione

Processi organizzativi:

- Gestione dei progetti
- Gestione delle infrastrutture
- Miglioramento del processo
- Formazione del personale

Documentazione del prodotto: attività di registrazione delle informazioni prodotte durante un processo

Gestione di versioni e configurazioni: attività di identificazione e controllo delle modifiche del software in via di sviluppo

Assicurazione della qualità: attività per garantire in modo obiettivo che i processi ed i prodotti siano conformi ai requisiti

Validazione: controllo del raggiungimento degli obiettivi

Gestione dei processi: attività basilari del management

Gestione delle infrastrutture: attività atte a fornire l'ambiente (software, hardware e logistica)

Miglioramento del processo: attività di un'organizzazione per stabilire e migliorare i propri processi

Formazione del personale: attività atte a fornire le competenze al personale.

4.Design

Questa sezione descrive la fase di design. Per ogni servizio sono riportati class diagram, object diagram e SoaML service interface diagram. Dopodichè è riportato il SoaML service interface diagram dell'agent ed il suo corrispondente sequence diagram.

4.1 Offerta & Stato Libro

Il servizio di Offerta e Stato Libri contiene un catalogo di Libri inseriti dagli utenti che vogliono condividere la propria libreria con altre persone e permette agli utenti di gestire il prestito per ogni libro di loro proprietà. Lo stato del libro può essere disponibile, in prestito o prenotato. Il servizio OSL imposta automaticamente la data di fine del prestito a 30 giorni dalla data di inizio prestito.

4.1.1 Class Diagrams

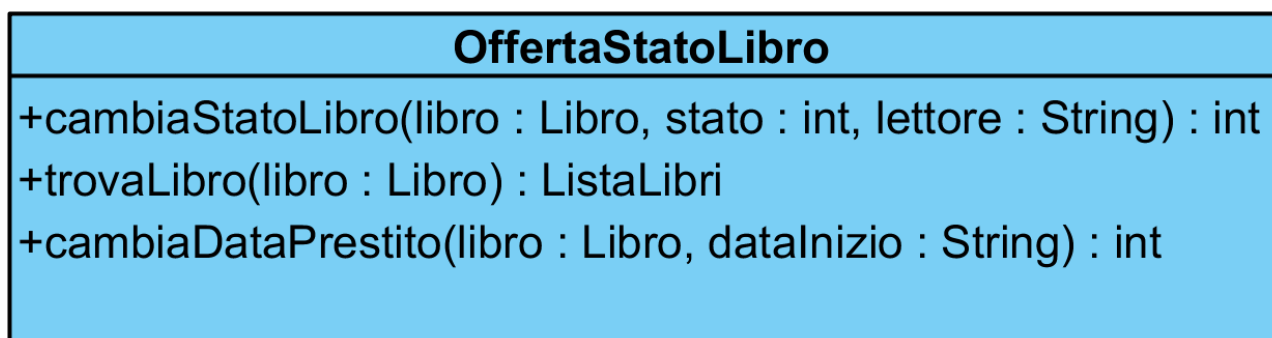


Figure 37: class diagram Offerta & Stato Libro

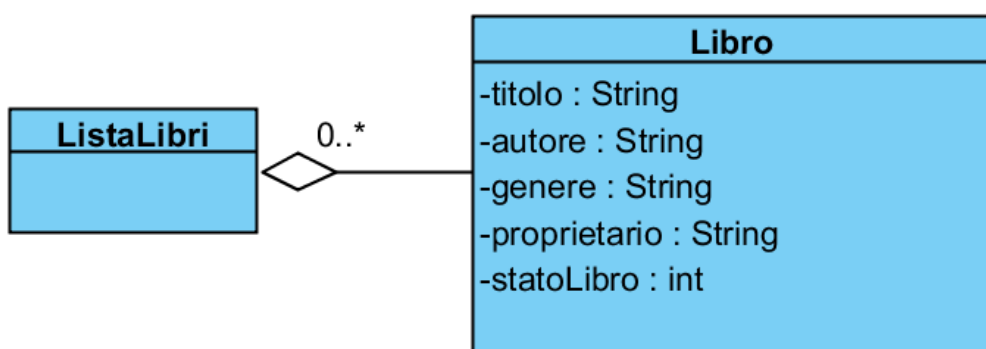


Figure 38: object class diagram Offerta & Stato Libro

- **ListaLibri**: lista container di Libri;
- **Libro**: rappresenta un libro ed è identificato univocamente dai due attributi Proprietario e Titolo.

4.1.2 SoaML Service Interface

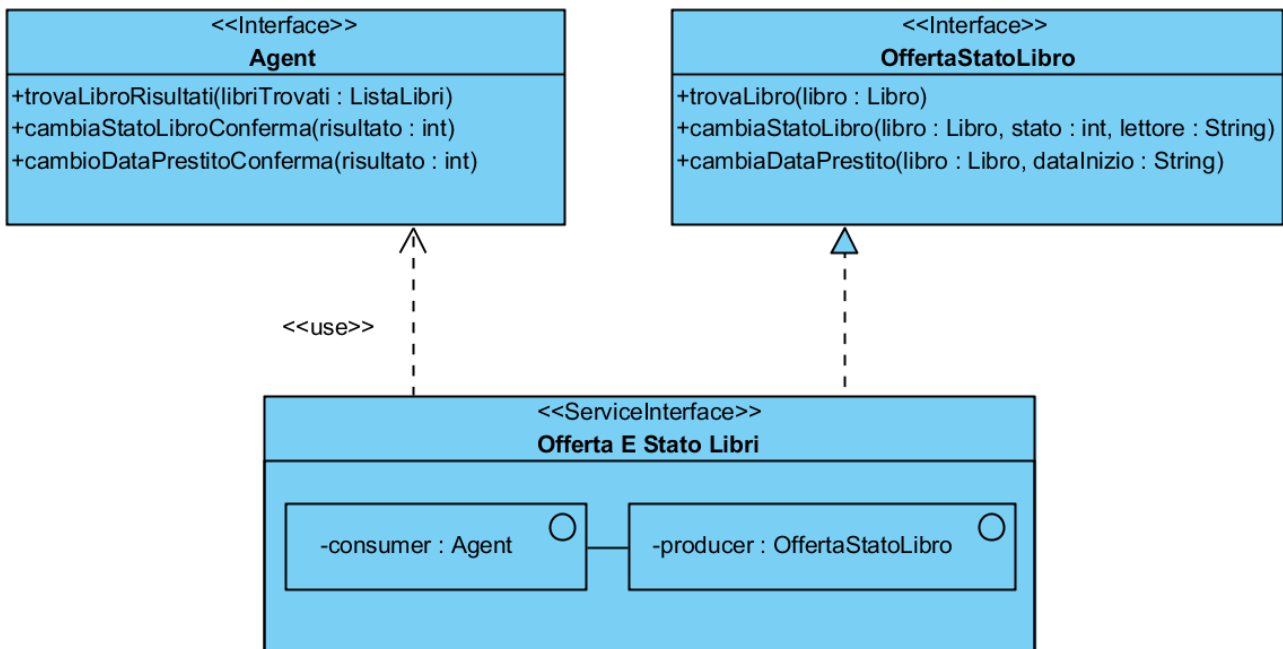


Figure 39: interfaccia SoaML Offerta e Stato Libro

4.2 Ricerca & Prestito

Il servizio Richiesta&Prestito si occupa di ricevere e salvare le richieste effettuate dagli utenti che possono richiedere un determinato titolo, un determinato autore o uno specifico genere letterario. Permette poi di filtrare i risultati ottenuti dalla ricerca tramite i parametri inseriti nella richiesta in modo tale da fornire un solo risultato al lettore. Il filtraggio è effettuato in base all'affidabilità del proprietario del libro, dalla vicinanza del libro al lettore o alla disponibilità del libro.

4.2.1 Class Diagrams

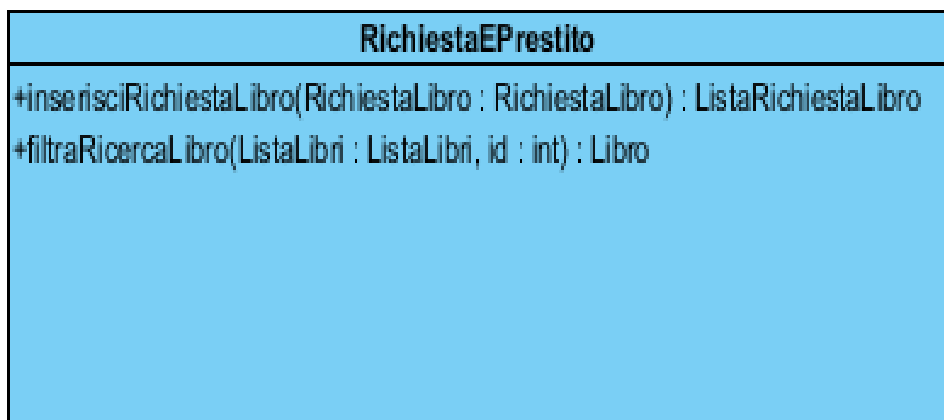


Figure 40: class diagram Ricerca e Prestito

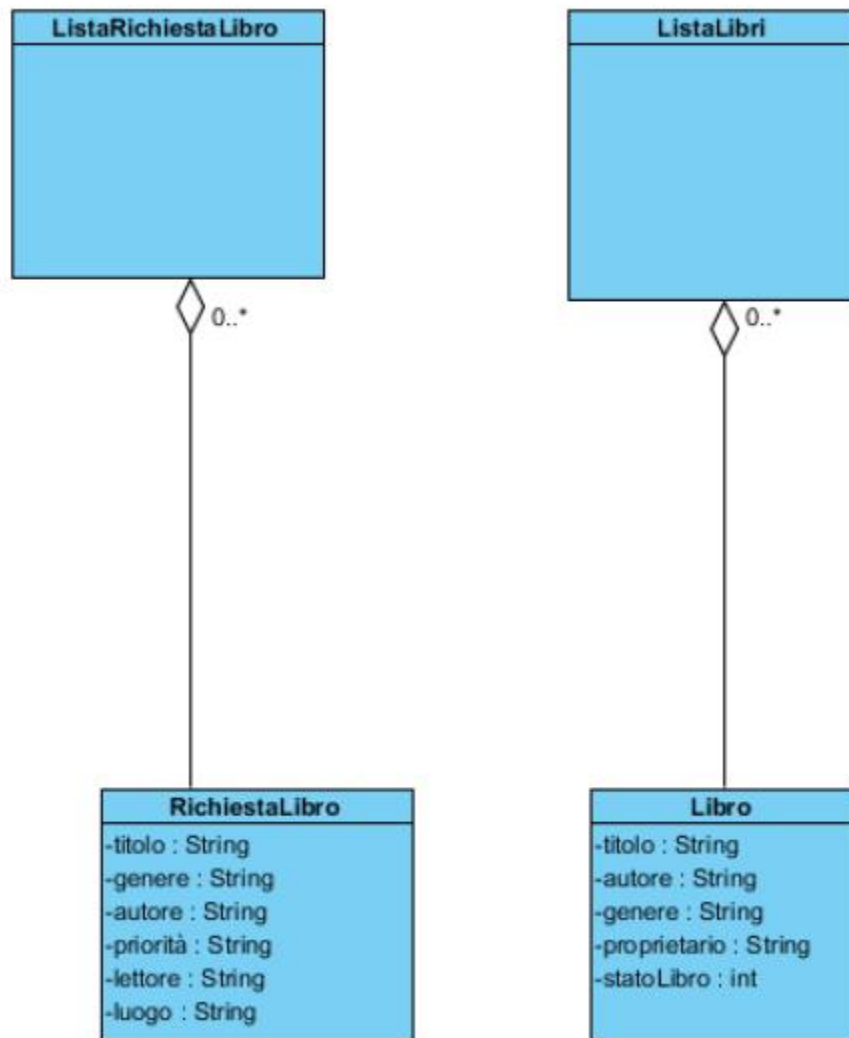


Figure 41: object class diagram Ricerca e Prestito

- **ListaRichiestaLibro**: rappresenta un'aggregazione di RichiestaLibro
- **RichiestaLibro**: rappresenta la richiesta effettuata dal lettore in cui sono specificati tutti i parametri inseriti
- **ListaLibri**: rappresenta un'aggregazione di Libri
- **Libro**: rappresenta un libro, identificato dalla coppia di valori titolo proprietario

4.2.2 SoaML Service Interface Diagram

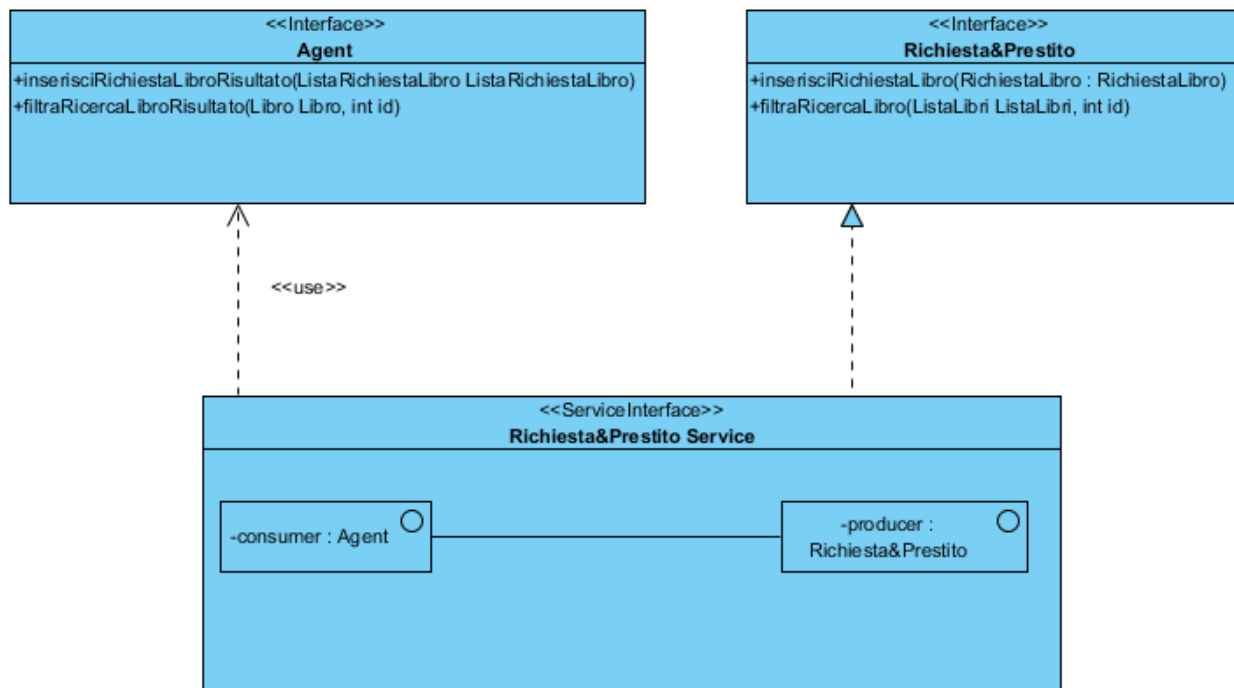


Figure 42: interfaccia SoaML Richiesta e Prestito

4.3 Appuntamento Per Consegna

Il servizio di Appuntamento Per Consegna(APC) viene utilizzato per la gestione della consegna del libro all'utente che lo ha richiesto in prestito, la consegna ha una doppia modalità di funzionamento: tramite utilizzo di corriere oppure direttamente di persona tra l'utente proprietario del libro e l'utente lettore. L'applicazione contiene le informazioni sulle agende di tutti gli utenti iscritti al sistema ed esse, vengono aggiornate con gli impegni che gli utenti hanno nel corso delle giornate.

4.3.1 Class Diagrams

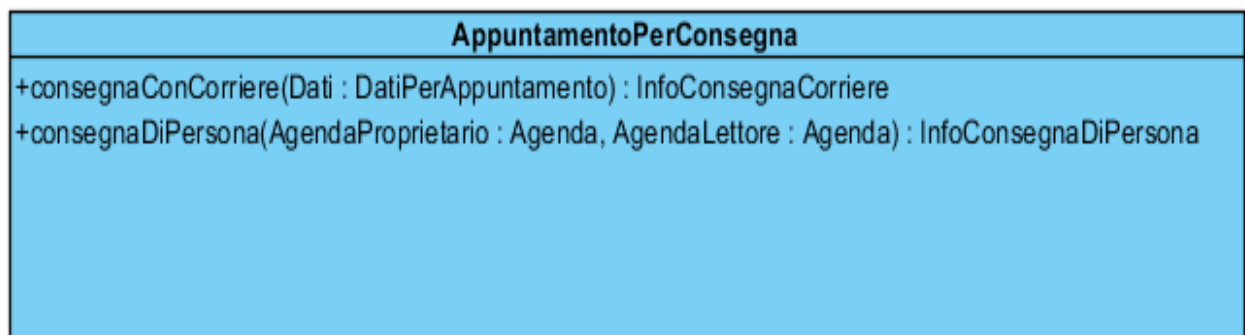


Figure 43: class diagram Appuntamento per Consegna

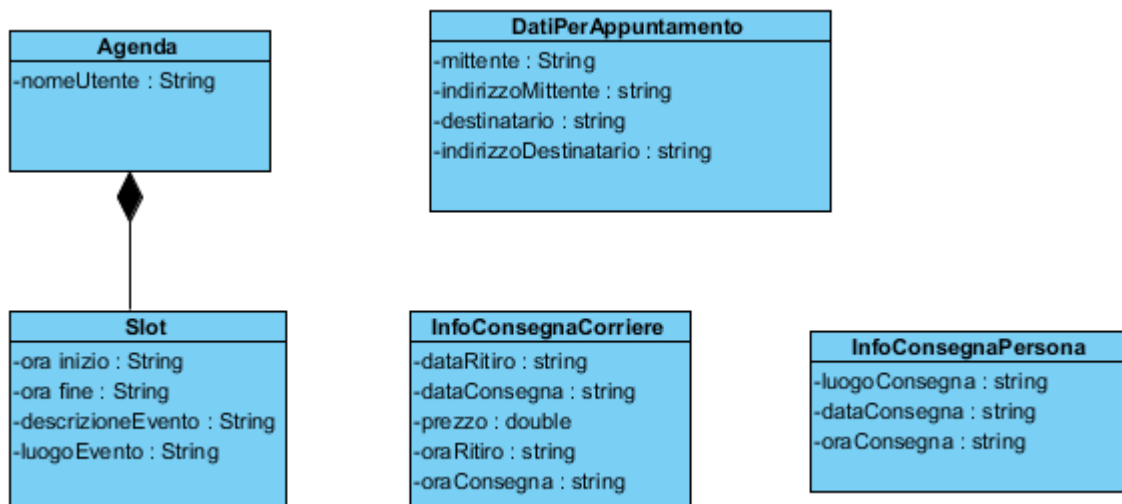


Figure 44: object class diagram Appuntamento per Consegna

- **Agenda:** costituito da una composizione di Slot ed identificata dal nome utente
- **Slot:** ogni slot rappresenta un appuntamento all'interno di una determinata Agenda
- **DatiPerAppuntamento:** rappresenta le informazioni necessarie al fine di poter individuare una soluzione per effettuare lo scambio del libro
- **InfoConsegnaCorriere:** rappresenta le informazioni necessarie per la consegna tramite corriere
- **InfoConsegnaPersona:** rappresenta le informazioni necessarie per la consegna di persona tra i due utenti

4.3.2 SoaML Service Interface Diagram

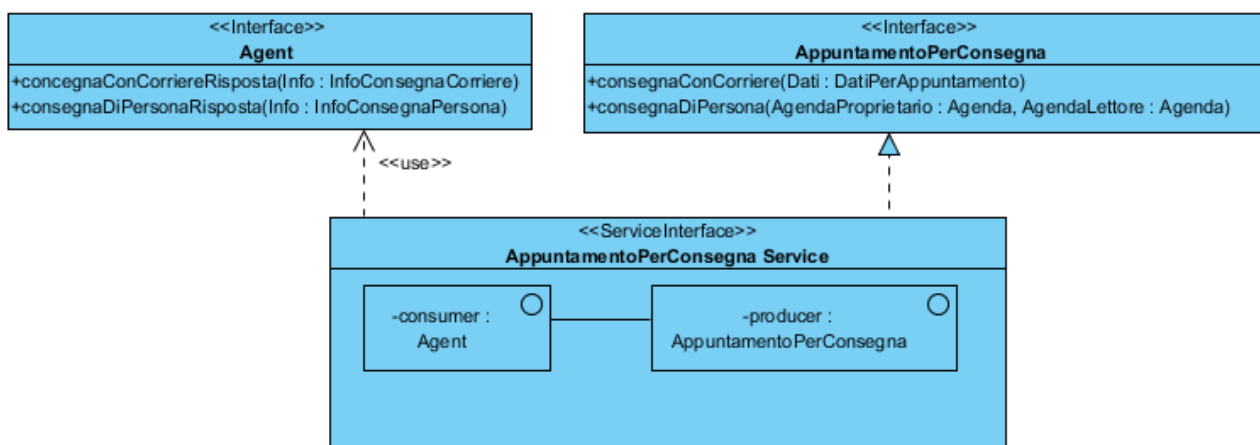


Figure 45: interfaccia SoaML Appuntamento per Consegna

4.4 Recensione Libro/Utente

Il servizio di Recensione Libro/Utente gestisce il Sistema delle recensioni. Attraverso l'interfaccia fornita è possibile inserire nuove recensioni, composte da un testo ed un voto sintetico da 1 a 5, e di cercare recensioni riguardante libri o utenti.

4.4.1 Class Diagrams

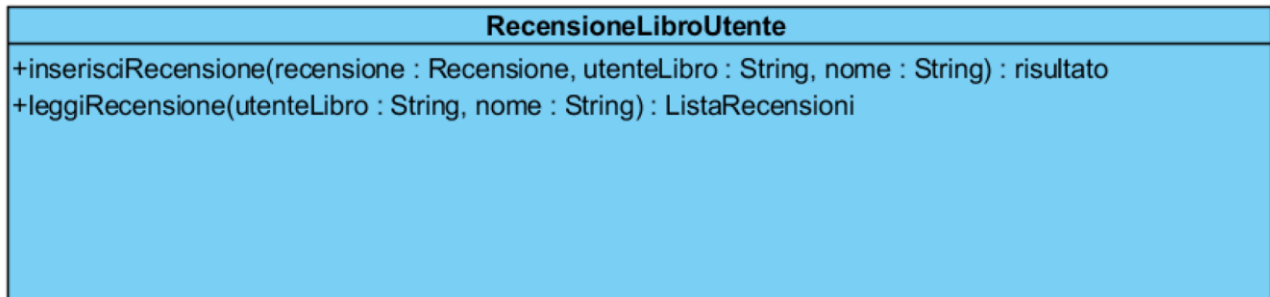


Figure 46: class diagram Recensione Libro Utente

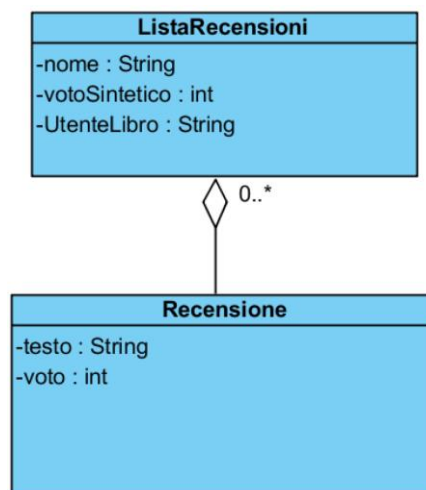


Figure 47: object class diagram Recensione Libro Utente

- **ListaRecensioni:** costituito da un'aggregazione di Recensioni ed identificata dal nome e dal tipo (utente o libro) ed un voto sintetico, composto dalla media dei voti delle recensioni
- **Recensione:** ogni recensione è composta da una stringa testuale ed un voto

4.1.2 SoaML Service Interface Diagram

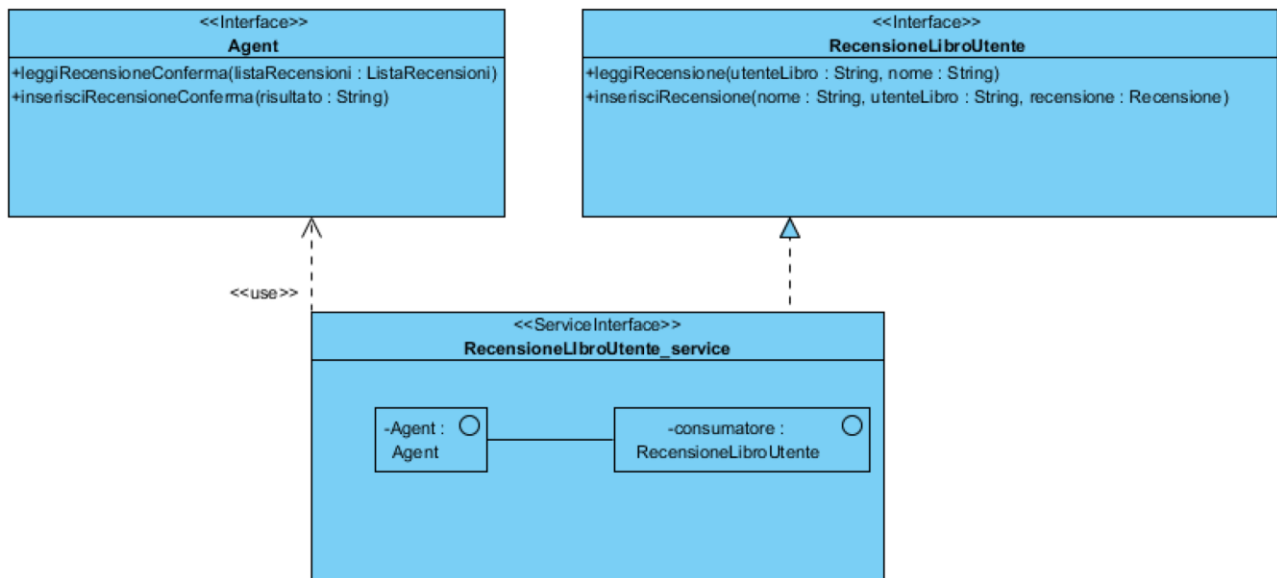


Figure 48: interfaccia SoaML RecensioneLibroUtente

4.5 Agent

4.5.1 SoaML Service Interface Diagram

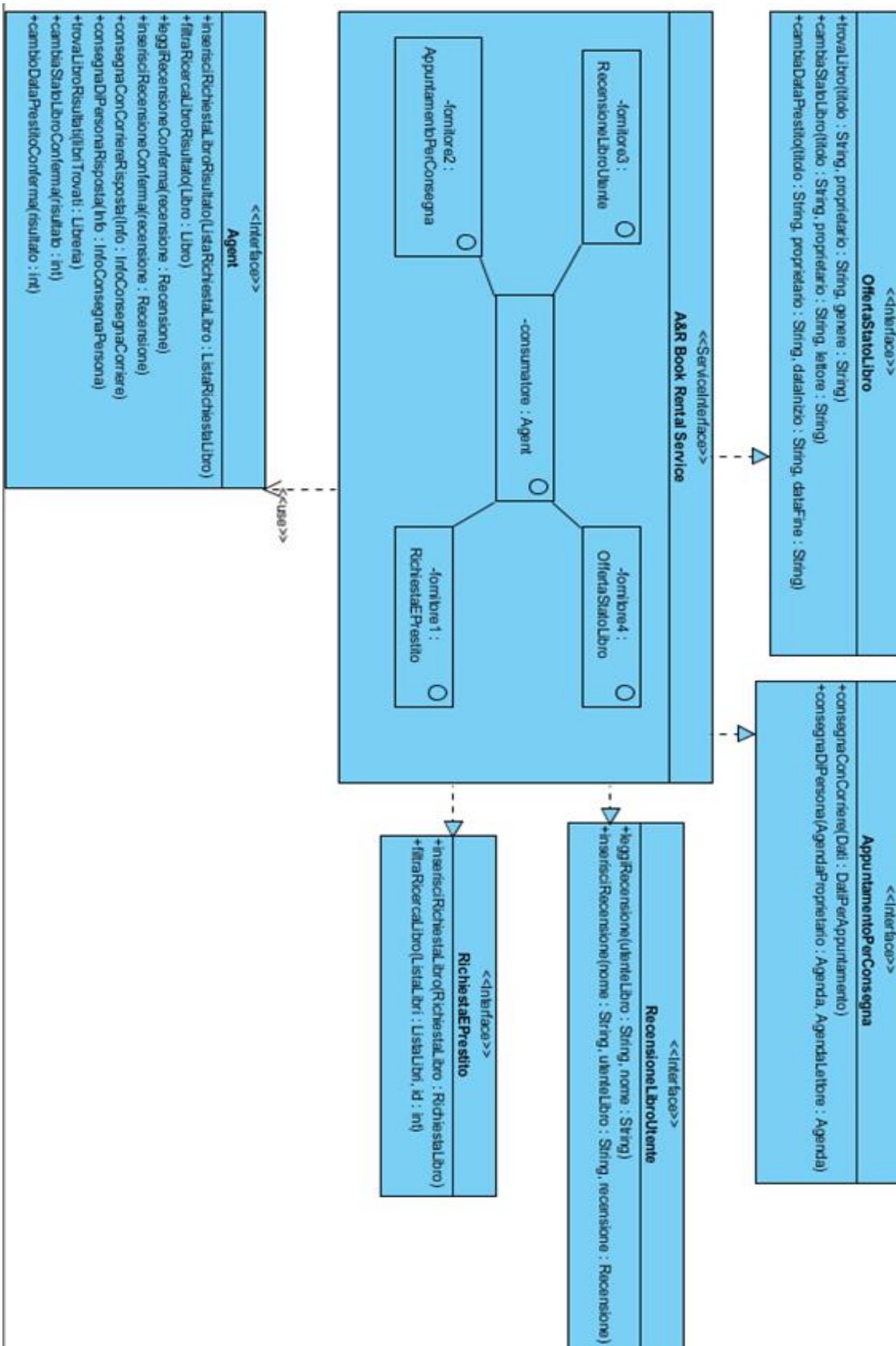


Figure 49: SoaML Service Interface Diagram

4.5.2 SoaML Sequence Diagram

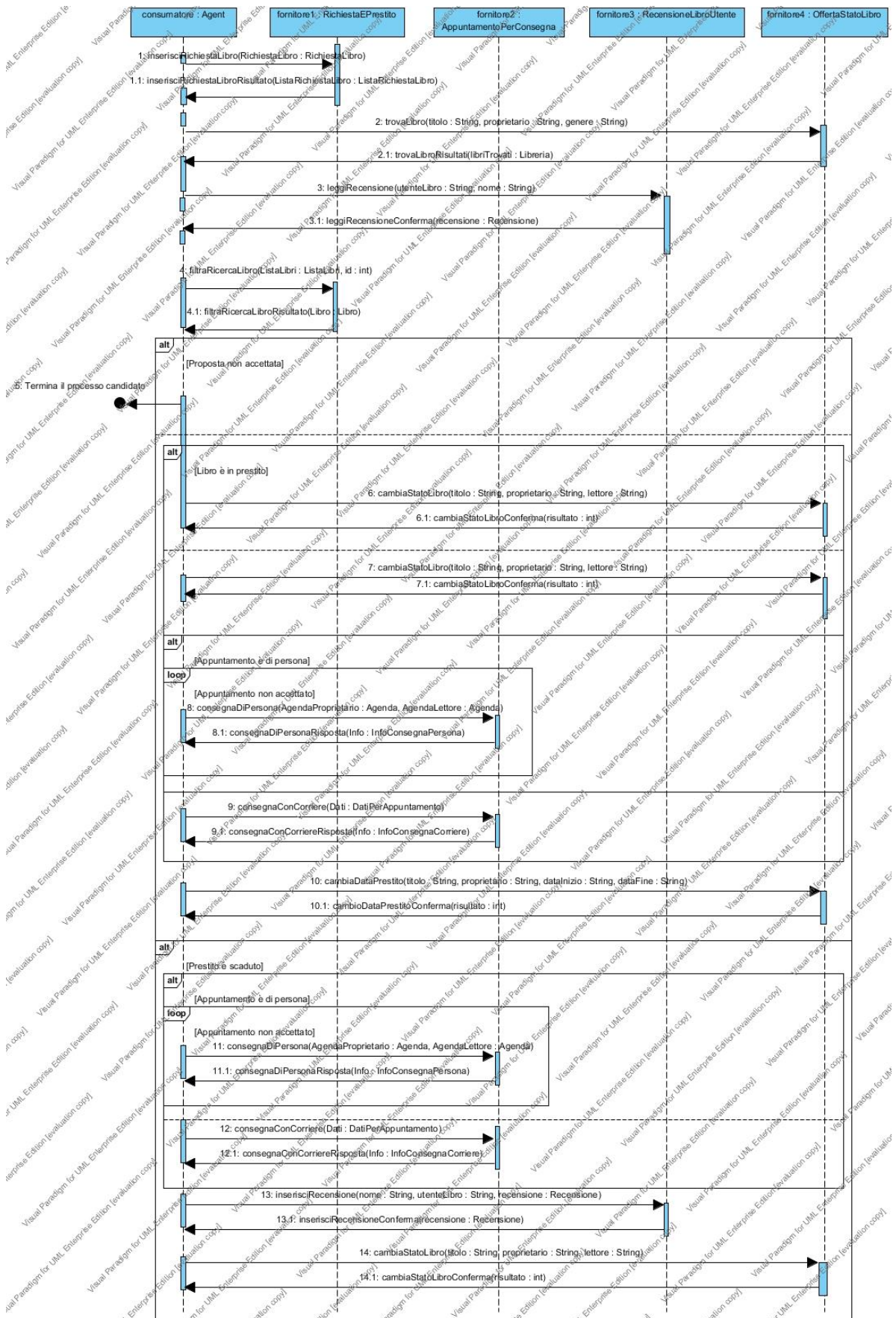


Figure 50: SoaML sequence diagram

5.Implementazione

Questo capitolo contiene il codice dei servizi e dell'agent ed anche le istanze XML ed i relativi schemi XSD delle strutture dati precedentemente definite.

5.1 Offerta & Stato Libri

5.1.1 Service Code

```
public class OffertaStatoLibro {
    private static double NONAVAILABILITY=0.01;

    public String cambiaStatoLibro(String Libro, int Stato, String Lettore) {
        if(Math.random()<=NONAVAILABILITY) return "fault";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;1"))&&(Stato==2))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;2";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;2"))&&(Stato==3))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;3";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;2"))&&(Stato==3))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;3";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;2"))&&(Stato==1))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;1";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;3"))&&(Stato==1))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;2";
        if((Libro.equals("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;3"))&&(Stato==1))
            Libro="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;2";
        return Libro;
    }

    public String trovaLibro(String Libro) {
        String ListaLibri=null;
        double ricerca=Math.random();
        if(ricerca<=NONAVAILABILITY) {
            ListaLibri = "fault";
        }
        if (ricerca<0.05){
            ListaLibri="nessun risultato";
        }
        else if (ricerca>=0.05 && ricerca<0.66){
```

```

        ListaLibri="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;1";
        ListaLibri+="|Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Marco Franco@Pisa;1";
        ListaLibri+="|Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Roma;2";
    }
    else if (ricerca>=0.66 && ricerca<1){
        ListaLibri="Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;2";
        ListaLibri+="|Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Marco Franco@Pisa;2";
        ListaLibri+="|Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Jenny Smith@Roma;2";
    }
    return ListaLibri;
}

public String cambiaDataPrestito(String Libro, String DataInizio) {
    String DataFine="01/01/2017";
    if(Math.random()<=NONAVAILABILITY){
        DataFine="fault";
    }
    if (DataInizio.equals("14/06/2017")==true){
        DataFine="14/07/2017";
    }
    return DataFine;
}
}

```

5.1.2 XML instance e XSD schema

• ListaLibri XSD Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ListaLibri">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Libro" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Libro">
    <xs:complexType>
      <xs:all>
        <xs:element name="titolo" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="autore" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="genere" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="proprietario" type="xs:string" minOccurs="1"
            maxOccurs="1" />
        <xs:element name="statoLibro" minOccurs="1" maxOccurs="1">
          <xs:simpleType>

```

```

    <xs:restriction base="xs:integer">
      <xs:minInclusive value="1" />
      <xs:maxInclusive value="3" />
    </xs:restriction>
  </xs:simpleType>
</xs:element>
</xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```

• ListaLibri XML Instance

```

<?xml version="1.0" encoding="UTF-8"?>
<ListaLibri xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='ListaLibri.xsd'>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>Franco Mario@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>Marco Franco@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>John Smith@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
</ListaLibri>

```

5.2 Ricerca & Prestito

5.2.1 Service Code

```

public class RichiestaEPrestito {
  public int inserisciRichiestaLibro(String RichiestaLibro) {
    if (Math.random() <= 0.01)
      return -1; // -1 indica la condizione di fault
    if(RichiestaLibro.equals("Harry Potter e la camera dei segreti,
fantasy,
J.K.Rowling, vicinanza, Mario Rossi, Pisa"))
      return 1;
  }
}

```

```

        if(RichiestaLibro.equals("Harry Potter e la camera dei segreti,
fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa"))
            return 2;
        if(RichiestaLibro.equals("Harry Potter e la camera dei segreti,
fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa"))
            return 3;
        return 0;
    }

    public String filtraRicercaLibro(String ListaLibri, int id) {
        if (Math.random() <= 0.01)
            return "fault";
        if (id == 1)
            return "Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;2";
        if (id == 2)
            return "Harry Potter e il calice di fuoco;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;2";
        if (id == 3)
            return "Harry Potter e il calice di fuoco;J.K.
Rowling;Fantasy;Franco Mario@Roma;1";
        return "";
    }
}

```

5.3.2 XML instance e XSD schema

• ListaLibri XSD Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ListaLibri">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="Libro" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="Libro">
    <xs:complexType>
      <xs:all>
        <xs:element name="titolo" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="autore" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="genere" type="xs:string" minOccurs="1" maxOccurs="1" />
        <xs:element name="proprietario" type="xs:string" minOccurs="1"
            maxOccurs="1" />
        <xs:element name="statoLibro" minOccurs="1" maxOccurs="1">
          <xs:simpleType>
            <xs:restriction base="xs:integer">
              <xs:minInclusive value="1" />
              <xs:maxInclusive value="3" />
            </xs:restriction>
          </xs:simpleType>
        </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>

```



```

    </xs:element>
  </xs:all>
</xs:complexType>
</xs:element>
</xs:schema>

```

• ListaLibri XML Instance

```

<?xml version="1.0" encoding="UTF-8"?>
<ListaLibri xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='ListaLibri.xsd'>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>Franco Mario@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>Marco Franco@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
  <Libro>
    <titolo>Harry Potter e la camera dei segreti</titolo>
    <autore>J.K. Rowling</autore>
    <genere>Fantasy</genere>
    <proprietario>John Smith@Pisa</proprietario>
    <statoLibro>1</statoLibro>
  </Libro>
</ListaLibri>

```

• ListaRichiestaLibro XSD schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ListaRichiesteLibro">
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="RichiestaLibro" maxOccurs="unbounded" />
      </xs:sequence>
    </xs:complexType>
  </xs:element>
  <xs:element name="RichiestaLibro">
    <xs:complexType>
      <xs:sequence>
        <xs:choice minOccurs="1" maxOccurs="unbounded">
          <xs:element name="titolo" type="xs:string" maxOccurs="1" />
          <xs:element name="autore" type="xs:string" maxOccurs="1" />
          <xs:element name="genere" type="xs:string" maxOccurs="1" />
        </xs:choice>
      </xs:sequence>
    </xs:complexType>
  </xs:element>

```

```

<xs:element name="lettore" type="xs:string" maxOccurs="1" maxOccurs="1" />
<xs:element name="priorità" type="xs:string" minOccurs="1" maxOccurs="1"/>
<xs:element name="luogo" type="xs:string" maxOccurs="1" />
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

- **ListaRichiestaLibro XML instance**

```

<?xml version="1.0" encoding="UTF-8"?>
<ListaRichiesteLibro xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:noNamespaceSchemaLocation='ListaRichiesteLibro.xsd'>
    <RichiestaLibro>
        <titolo>Harry Potter e il calice di fuoco</titolo>
        <autore>J.K. Rowling</autore>
        <genere>Fantasy</genere>
        <lettore>Franco Mario@Roma</lettore>
        <priorità>Affidabilità</priorità>
        <luogo>Roma</luogo>
    </RichiestaLibro>
    <RichiestaLibro>
        <titolo>Harry Potter e la camera dei segreti</titolo>
        <autore>J.K. Rowling</autore>
        <genere>Fantasy</genere>
        <lettore>Franco Mario@Roma</lettore>
        <priorità>vicinanza</priorità>
        <luogo>Roma</luogo>
    </RichiestaLibro>
</ListaRichiesteLibro>

```

5.3 Appuntamento Per Consegna

5.3.1 Service Code

```

public class AppuntamentoPerConsegna {
    public String consegnaConCorriere(String Dati) {
        if(Math.random()>0.99)
            return "fault";
        if(Dati.equals("mittente: Mario Rossi, Via Dio ti salvi-Pisa
destinatario:Giuseppe Verdi, Corso Italia-Milano"))
            return "14/06/2017,08:30-12:30,16/06/2017,14:30-18:30,3
Euro";
        return "";
    }

    public String consegnaDiPersona(String AgendaProprietario, String AgendaLettore)
    {
        if(Math.random()>0.99)
            return "fault";
    }
}

```

```

        if((AgendaProprietario.equals("Mario Rossi@10:30,12:00,palestra,CUS
Pisa|19:00,20:00,aperitivo con Giulia,Bazeel")) && AgendaLettore.equals("Marco
Bianchi@9:00,11:00,riunione,Facolta' di economia-Pisa|13:00,14:30,Pranzo con
Luca,Ristorante La Scaletta"))
            return "14/06/2017,15:00,Piazza dei Miracoli";
        else
            return "errore Agende";
    }
}

```

5.3.2 XML instance e XSD schema

• Agenda XSD Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="agenda">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="nomeUtente" type="xs:string" minOccurs="1" maxOccurs="1">
        </xs:element>
        <xs:element name="slot" maxOccurs="unbounded">
          <xs:complexType>
            <xs:all>
              <xs:element name="oraInizio" type="xs:string" minOccurs="1" maxOccurs="1">
              </xs:element>
              <xs:element name="oraFine" type="xs:string" minOccurs="1" maxOccurs="1">
              </xs:element>
              <xs:element name="descrizione" type="xs:string" minOccurs="1"
                maxOccurs="1">
              </xs:element>
              <xs:element name="luogo" type="xs:string" minOccurs="1" maxOccurs="1">
              </xs:element>
            </xs:all>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

• Agenda XML instance1

```

<?xml version="1.0" encoding="UTF-8"?>
<agenda xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='agenda.xsd'>
  <nomeUtente>Mario Rossi</nomeUtente>
  <slot>
    <oraInizio>10:30</oraInizio>
    <oraFine>12:00</oraFine>
    <descrizione>palestra</descrizione>

```

```

        <luogo>CUS Pisa</luogo>
    </slot>
    <slot>
        <oraInizio>19:00</oraInizio>
        <oraFine>20:00</oraFine>
        <descrizione>aperitivo con Giulia</descrizione>
        <luogo>Bazeel</luogo>
    </slot>
</agenda>

```

- **Agenda XML istance2**

```

<?xml version="1.0" encoding="UTF-8"?>
<agenda xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='agenda.xsd'>
  <nomeUtente>Marco Bianchi</nomeUtente>
  <slot>
    <oraInizio>09:00</oraInizio>
    <oraFine>11:00</oraFine>
    <descrizione>riunione</descrizione>
    <luogo>Facolta' di economia-Pisa</luogo>
  </slot>
  <slot>
    <oraInizio>13:00</oraInizio>
    <oraFine>14:30</oraFine>
    <descrizione>Pranzo con Luca</descrizione>
    <luogo>Ristorante La Scaletta</luogo>
  </slot>
</agenda>

```

- **DatiPerAppuntamento XSD Schema**

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="datiPerAppuntamento">
    <xs:complexType>
      <xs:all>
        <xs:element name="mittente" type="xs:string" minOccurs="1" maxOccurs="1">
          </xs:element>
        <xs:element name="indirizzoMittente" type="xs:string" minOccurs="1"
maxOccurs="1">
          </xs:element>
        <xs:element name="destinatario" type="xs:string" minOccurs="1" maxOccurs="1">
          </xs:element>
        <xs:element name="indirizzoDestinatario" type="xs:string" minOccurs="1"
maxOccurs="1">
          </xs:element>
      </xs:all>
    </xs:complexType>
  </xs:element>
</xs:schema>

```

- **DatiPerAppuntamento XML Instance**

```
<?xml version="1.0" encoding="UTF-8"?>
<datiPerConsegna xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='datiPerAppuntamento.xsd'>
  <mittente>Mario Rossi</mittente>
  <indirizzoMittente>Via Dio ti salvi-Pisa</indirizzoMittente>
  <destinatario>Giuseppe Verdi</destinatario>
  <indirizzoDestinatario>Corso Italia-Milano</indirizzoDestinatario>
</datiPerConsegna>
```

- **InfoConsegnaCorriere XSD Schema**

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="InfoConsegnaCorriere">
  <xs:complexType>
    <xs:all>
      <xs:element name="dataRitiro" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="dataConsegna" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="oraRitiro" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="oraConsegna" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="prezzo" type="xs:double" minOccurs="1" maxOccurs="1">
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:schema>
```

- **InfoConsegnaCorriere XML Instance**

```
<?xml version="1.0" encoding="UTF-8"?>
<infoConsegnaCorriere xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='infoConsegnaCorriere.xsd'>
  <dataRitiro>14/06/2017</dataRitiro>
  <dataConsegna>16/06/2017</dataConsegna>
  <oraRitiro>08:30-12:30</oraRitiro>
  <oraConsegna>14:30-18:30</oraConsegna>
  <prezzo>3 Euro</prezzo>
</infoConsegnaCorriere>
```

- **InfoConsegnaPersona XSD Schema**

```
<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema">
<xs:element name="infoConsegnaPersona">
  <xs:complexType>
    <xs:all>
      <xs:element name="luogo" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="data" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
      <xs:element name="ora" type="xs:string" minOccurs="1" maxOccurs="1">
      </xs:element>
    </xs:all>
  </xs:complexType>
</xs:element>
</xs:schema>
```

- **InfoConsegnaPersona XML Instance**

```
<?xml version="1.0" encoding="UTF-8"?>
<infoConsegnaPersona xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='infoConsegnaPersona.xsd' cos
  <luogo>Piazza dei Miracoli</luogo>
  <data>14/06/2017</data>
  <ora>15:00</ora>
</infoConsegnaPersona>
```

5.4 Recensione Libro/Utente

5.4.1 Service Code

```
public class RecensioneLibroUtente {

    public String inserisciRecensione(String recensione, String UtenteLibro,
String nome) {
        // TODO - implement RecensioneLibroUtente.inserisciRecensione
        if (Math.random() <= 0.01)
            return "fault";
        if(UtenteLibro.equals("Libro") && nome.equals("Il Signore Degli
Anelli") && recensione.equals("Bel libro, consigliato; 4"))
            return "true";
        if(UtenteLibro.equals("Utente") && nome.equals("Luca Bianchi") &&
recensione.equals("tutto ok; 4"))
            return "true";
        else
            return "";
    }

    public String leggiRecensione(String UtenteLibro, String nome) {
```

```

String listarecensioni = null;
if (Math.random() <= 0.01)
    return "fault";
// TODO - implement RecensioneLibroUtente.leggiRecensione
if(UtenteLibro.equals("Libro") && nome.equals("Il Signore Degli
Anelli")){
    listarecensioni = "Il signore degli anelli: 5";
    listarecensioni += " | 5; Ottimo libro | 5; Consigliato";
}
if(UtenteLibro.equals("Libro") && nome.equals("Harry Potter")){
    listarecensioni = "Harry Potter : 4";
    listarecensioni += " | 3; Sopravvalutato | 5; Bellissimo";
}
else if(UtenteLibro.equals("Utente") && nome.equals("Franco
Mario")){
    listarecensioni = "Franco Mario: 3";
    listarecensioni += " | 5; Tutto perfetto | 3; Bene ma non
benissimo";
}
else if(UtenteLibro.equals("Utente") && nome.equals("John Smith")){
    listarecensioni = "John Smith: 2";
    listarecensioni += " | 3; Gentile ma poco disponibile | 1;
Sconsigliato";
}
else if(UtenteLibro.equals("Utente") && nome.equals("Luca
Bianchi")){
    listarecensioni = "Luca Bianchi: 5";
    listarecensioni += " | 5; Gentile e disponibile | 5; Ottima
esperienza";
}
else if(UtenteLibro.equals("Utente") && nome.equals("Mario Rossi")){
    listarecensioni = "Mario Rossi: 1";
    listarecensioni += " | 1; Pessima esperienza, non consigliato
| 1; Non consigliato";
}
return listarecensioni;
}
}

```

5.4.2 XML instance e XSD schema

• ListaRecensioni XSD Schema

```

<?xml version="1.0" encoding="UTF-8" ?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="ListaRecensioni">
    <xs:complexType>
      <xs:sequence>
        <xs:element type="xs:string" name="nome"/>
        <xs:element type="xs:byte" name="votoSintetico"/>
        <xs:element type="xs:string" name="UtenteLibro"/>
        <xs:element name="Recensione" maxOccurs="unbounded" minOccurs="0">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element type="xs:byte" name="voto"/>
    <xs:element type="xs:string" name="testo"/>
  </xs:sequence>
</xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

● recensione utente XML Instance

```

<?xml version="1.0" encoding="UTF-8"?>
<ListaRecensioni xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='ListaRecensioni.xsd'>
  <nome>Luca Bianchi</nome>
  <votoSintetico>5</votoSintetico>
  <UtenteLibro>Utente</UtenteLibro>
  <Recensione>
    <voto>5</voto>
    <testo>Gentile e disponibile</testo>
  </Recensione>
  <Recensione>
    <voto>5</voto>
    <testo>Ottima esperienza</testo>
  </Recensione>
  <Recensione>
    <voto>4</voto>
    <testo>tutto ok</testo>
  </Recensione>
</ListaRecensioni>

```

● recensione libro XML Instance

```

<?xml version="1.0" encoding="UTF-8"?>
<ListaRecensioni xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:noNamespaceSchemaLocation='ListaRecensioni.xsd'>
  <nome>Il Signore Degli Anelli</nome>
  <votoSintetico>5</votoSintetico>
  <UtenteLibro>Libro</UtenteLibro>
  <Recensione>
    <voto>5</voto>
    <testo>Ottimo libro</testo>
  </Recensione>
  <Recensione>
    <voto>5</voto>
    <testo>Consigliato</testo>
  </Recensione>
</ListaRecensioni>

```


5.5 Agent

5.5.1 Agent.java Class

```
import java.util.logging.Level;
import java.util.logging.Logger;
import java.util.regex.*;

public class Agent {
    public static String OSL_url =
"http://localhost:8080/axis2/services/OffertaStatoLibro";
    public static String REP_url =
"http://localhost:8080/axis2/services/RichiestaEPrestito";
    public static String RLU_url =
"http://localhost:8080/axis2/services/RecensioneLibroUtente";
    public static String APC_url =
"http://localhost:8080/axis2/services/AppuntamentoPerConsegna";

    public static Messages Mex = new Messages();
    public static class OSLException extends Exception{};
    public static class REPException extends Exception{};
    public static class RLUException extends Exception{};
    public static class APCException extends Exception{};
    private static void print(String msg) {System.out.println(msg);}

    private static void printFashion(String Msg, String typeMsg){
        String regex="";
        if(typeMsg.equals("response"))
            regex = ".*<ns:return>(.*?)</ns:return>.*";
        else
            regex = ".*<axis:args0>(.*?)</axis:args0>.*";
        Pattern pattern = Pattern.compile(regex);
        Matcher matcher = pattern.matcher(Msg);

        if(matcher.matches()) {
            print(matcher.group(1));
        }
        else{
            try {
                throw new Exception("Errore in printFashion");
            } catch (Exception ex) {
                Logger.getLogger(Agent.class.getName()).log(Level.SEVERE, null,
ex);
            }
        }
    }

    public static void main(String[] args){
        int tests = -1;
        int gestioni_corrette = 0;
        double OSLFaultNum = 0, REPFaultNum = 0,
        RLUFaultNum = 0, APCFaultNum = 0;
        double OSLCalls = 0, REPCalls = 0, RLUCalls = 0, APCCalls = 0;
```

```

    double TotalFault = 0;
    int Reject = 0;

    while(++tests < 150000){
        try{
            boolean consegnaDiPersona = false;
            String resp = null;
            String richiestaLibro="";
            String recensioni="";
            String Libro="";
            String req = "";

            int i=1;
            String DatiConsegna="mittente: Mario Rossi, Via Dio ti salvi-
Pisa destinatario:Giuseppe Verdi, Corso Italia-Milano";

            double richiesta = Math.random();
            if(richiesta <= 0.33)
                richiestaLibro = Mex.requestIR_1;
            else if(richiesta> 0.33 && richiesta<=0.66)
                richiestaLibro = Mex.requestIR_2;
            else
                richiestaLibro = Mex.requestIR_3;

            print("##### NUOVA RICHIESTA LIBRO#####\n");
            printFashion(richiestaLibro,"request");
            print("call RichiestaEPrestito: inserisciRichiestaLibro");
            int IDrichiesta=-1;
            resp = SOAPClient.sendMess(REP_url, richiestaLibro);
            REPCalls++;
            if(resp.contains("<ns:return>1</ns:return>"))
                IDrichiesta = 1;
            if (resp.contains("<ns:return>2</ns:return>"))
                IDrichiesta = 2;
            if(resp.contains("<ns:return>3</ns:return>"))
                IDrichiesta = 3;
            //System.out.println(resp);
            System.out.println("id richiesta "+ IDrichiesta);
            if (IDrichiesta == -1) throw new REPEXception();

            print("##### INSERITA RICHIESTA LIBRO#####\n");
            print("call OffertaStatoLibro: trovaLibro");
            String listaLibri = SOAPClient.sendMess(OSL_url,
Mex.requestTL_1);
            OSLCalls++;
            if(listaLibri.contains("<ns:return>fault</ns:return>")) throw
new OSLEXception();
            //if(listaLibri.equals(Mex.responseTL_fault)) throw new
OSLEXception();
            if(listaLibri.contains("nessun risultato")){
                print("##### NESSUN LIBRO TROVATO#####\n");
                print("##### FINE #####\n");
                continue;
            }
        }
    }

```

```

    }

    print("##### LIBRI TROVATI #####\n");

    if(IDrichiesta==2){
        print("Priorità = affidabilità");
        print("call RecensioneLibroUtente: leggiRecensione");
        String tempResp = SOAPClient.sendMess(RLU_url,
Mex.requestLR_3);
        RLUCalls++;

        if(tempResp.contains("<ns:return>fault</ns:return>")) throw
new RLUException();
        //if(tempResp.equals(Mex.responseLR_fault)) throw new
        RLUException();

        else recensioni = tempResp;

        tempResp = SOAPClient.sendMess(RLU_url, Mex.requestLR_4);
        RLUCalls++;

        if(tempResp.contains("<ns:return>fault</ns:return>")) throw
new RLUException();
        //if(tempResp.equals(Mex.responseLR_fault)) throw new
        RLUException();

        else recensioni += tempResp;

        tempResp = SOAPClient.sendMess(RLU_url, Mex.requestLR_5);
        RLUCalls++;
        if(tempResp.contains("<ns:return>fault</ns:return>")) throw
new RLUException();
        else recensioni += tempResp;
    }

    print("call RichiestaEPrestito: filtraRicercaLibro");

    switch(IDrichiesta){
        case 1:
            req = Mex.requestFRL_1;
            break;
        case 2:
            req = Mex.requestFRL_2;
            break;
        case 3:
            req = Mex.requestFRL_3;
            break;
        default:
            break;
    }
    Libro = SOAPClient.sendMess(REP_url, req);
    REPCalls++;
    if(Libro.contains("<ns:return>fault</ns:return>")) throw new
    REPEException();

    print("##### PROPOSTO UN LIBRO #####\n");
    if(Math.random() < 0.1 ){

```

```

        print("##### PROPOSTA LIBRO NON ACCETTATA #####\n");
        print("##### FINE #####\n");
        Reject++;
        continue;
    }
    printFashion(Libro,"response");

    print("call OffertaStatoLibro: cambiaStatoLibro");
    if(Libro.contains("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Franco Mario@Roma;1")){
        print("LIBRO DISPONIBILE!!!!!!");
        Libro=SOAPClient.sendMess(OSL_url, Mex.requestCSL_1);
OSLCalls++;
        if(Libro.contains("<ns:return>fault</ns:return>")) throw new
OSLException();
        //if(Libro.equals(Mex.responseCSL_fault)) throw new
OSLException();
        print("##### STATO DEL LIBRO MODIFICATO #####\n");
        print("LIBRO PRESO IN PRESTITO!!!!!!");
        printFashion(Libro,"response");
    }

    if(Libro.contains("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;2")){
        req = Mex.requestCSL_3;
        print("LIBRO IN PRESTITO E PRENOTABILE!!!!!!");
        Libro=SOAPClient.sendMess(OSL_url, req);
        OSLCalls++;
        if(Libro.contains("<ns:return>fault</ns:return>")) throw new
OSLException();
        print("##### STATO DEL LIBRO MODIFICATO #####\n");
        print("LIBRO PRENOTATO!!!!!!");
        printFashion(Libro,"response");
    }

    if(Libro.contains("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;2")){
        req = Mex.requestCSL_2;
        print("LIBRO IN PRESTITO E PRENOTABILE!!!!!!");
        Libro=SOAPClient.sendMess(OSL_url, req);
        OSLCalls++;
        if(Libro.contains("<ns:return>fault</ns:return>")) throw new
OSLException();
        print("##### STATO DEL LIBRO MODIFICATO #####\n");
        print("LIBRO PRENOTATO!!!!!!");
        //print(Libro);
        printFashion(Libro,"response");
    }

    print("##### CONSEGNA DEL LIBRO #####\n");
    if(Math.random() < 0.5){
        print("CONSEGNA DI PERSONA!!!!!!");
        consegnaDiPersona = true;
    }

```

```

        print("call AppuntamentoPerConsegna: consegnaDiPersona");
        do{
            print("TENTATIVO ACCORDO DI CONSEGNA NUMERO "+ i);
            resp = SOAPClient.sendMess(APC_url, Mex.requestCDPers);
            APCCalls++;
            if(resp.contains("<ns:return>fault</ns:return>")) throw
new APCEException();
            i++;
        }while(Math.random() < 0.5);
        print("##### PROPOSTA CONSEGNA ACCETTATA #####\n");
    }
    else{
        print("CONSEGNA CON CORRIERE!!!!!!");
        print("call AppuntamentoPerConsegna: consegnaConCorriere");
        resp=SOAPClient.sendMess(APC_url, Mex.requestCCC);
        APCCalls++;
        if(resp.contains("<ns:return>fault</ns:return>")) throw new
APCEException();
        print("##### NOTIFICA CONSEGNA CORRIERE #####\n");
    }
    print("call OffertaStatoLibro: cambiaDataPrestito");
    resp=SOAPClient.sendMess(OSL_url, Mex.requestCDP_ok);
    OSLCalls++;
    if(resp.contains("<ns:return>fault</ns:return>")) throw new
OSLException();
    print("##### DATE PRESTITO AGGIORNATE #####\n");

    print("\n\n##### RESTITUZIONE LIBRO #####\n");

    if(Math.random()<0.5){
        print("PRESTITO SCADUTO!!!!!!");
        if(consegnaDiPersona){
            print("CONSEGNA DI PERSONA!!!!!!");
            print("call AppuntamentoPerConsegna:
consegnaDiPersona");
            i = 1;
            do{
                print("TENTATIVO ACCORDO DI CONSEGNA NUMERO "+ i);
                resp=SOAPClient.sendMess(APC_url, Mex.requestCDPers);
                APCCalls++;
                if(resp.contains("<ns:return>fault</ns:return>"))
throw new APCEException();
                i++;
            }while(Math.random() < 0.5);
            print("##### PROPOSTA CONSEGNA ACCETTATA #####\n");
        }
        else{
            print("CONSEGNA CON CORRIERE!!!!!!");
            print("call AppuntamentoPerConsegna:
consegnaConCorriere");
            resp=SOAPClient.sendMess(APC_url, Mex.requestCCC);
            APCCalls++;

```

```

        if(resp.contains("<ns:return>fault</ns:return>")) throw
new APCEException();
        print("##### NOTIFICA CONSEGNA CORRIERE #####\n");
    }

    print("call RecensioneLibroUtente: inserisciRecensione");
    resp=SOAPClient.sendMess(RLU_url, Mex.requestIRec_1);
RLUCalls++;
        if(resp.contains("<ns:return>fault</ns:return>")) throw new
RLUEException();
    resp=SOAPClient.sendMess(RLU_url, Mex.requestIRec_2);
RLUCalls++;
        if(resp.contains("<ns:return>fault</ns:return>")) throw new
RLUEException();
    print("##### RECENSIONE INSERITA #####\n");
    print("call OffertaStatoLibro: cambiaStatoLibro");
    if(Libro.contains("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;Luca Bianchi@Pisa;3")) req = Mex.requestCSL_6;
    if(Libro.contains("Harry Potter e la camera dei segreti;J.K.
Rowling;Fantasy;John Smith@Pisa;3")) req = Mex.requestCSL_5;
    else req = Mex.requestCSL_4;
    Libro=SOAPClient.sendMess(OSL_url, req);
    OSLCalls++;
    if(Libro.contains("<ns:return>fault</ns:return>")) throw new
OSLEException();
    //if(Libro.equals(Mex.responseCSL_fault)) throw new
OSLEException();
    print("##### STATO DEL LIBRO MODIFICATO #####\n");
    }
    else{
        print("PRESTITO NON SCADUTO!!!!!!");
    }
    print("##### FINE #####\n");
}
catch(OSLEException e){
    OSLFaultNum ++;
    TotalFault ++;
    System.out.println("fault OSL");
    print("##### FINE #####\n");
}
catch(REPEException e){
    REPFaultNum ++;
    TotalFault ++;
    System.out.println("fault REP");
    print("##### FINE #####\n");
}
catch(RLUEException e){
    RLUFaultNum ++;
    TotalFault ++;
    System.out.println("fault RLU");
    print("##### FINE #####\n");
}
catch(APCEException e){

```

```

        APCFaultNum ++;
        TotalFault ++;
        System.out.println("fault APC");
        print("##### FINE #####\n");
    }
    catch(Exception e){}
    //tests++;
}

// STATISTICHE
System.out.printf("STATISTICHE \n");
System.out.printf("-----\n");
System.out.print("Test eseguiti: " + tests + "\n");
System.out.printf("-----\n");
System.out.print("Gestioni corrette: " + (tests-TotalFault) + "\n");
System.out.printf("-----\n");
System.out.print("Fault calls: " + TotalFault + "\n");
System.out.printf("-----\n");
System.out.printf("Agent Availability: %.4f%\n", ((1-(TotalFault/tests))
* 100));
System.out.printf("OSL fault rate: %.8f\n", (OSLFaultNum/OSLCalls));
System.out.printf("REP fault rate: %.8f\n", (REPFaultNum/REPCalls));
System.out.printf("APC fault rate: %.8f\n", (APCFaultNum/APCCalls));
System.out.printf("RLU fault rate: %.8f\n", (RLUFaultNum/RLUCalls));
System.out.println("OSL fault : " + OSLFaultNum);
System.out.println("REP fault : " + REPFaultNum);
System.out.println("APC fault : " + APCFaultNum);
System.out.println("RLU fault : " + RLUFaultNum);
System.out.println("OSL call : " + OSLCalls);
System.out.println("REP call : " + REPCalls);
System.out.println("APC call : " + APCCalls);
System.out.println("RLU call : " + RLUCalls);
}
}

```

5.5.2 Messages.java Class

*Vedere **FinalReport\srcMessages.java***

5.5.3 SOAPClient.java Class

```

import javax.xml.soap.*;
import javax.xml.parsers.*;
import org.w3c.dom.*;
import javax.xml.transform.dom.*;
import java.io.*;

public class SOAPClient {

    public static String sendMess(String url, String mess) throws Exception {
        SOAPConnection soapConnection =
        SOAPConnectionFactory.newInstance().createConnection();
    }
}

```

```

        SOAPMessage soapResponse = soapConnection.call(createSOAPRequest(mess),
url);
        ByteArrayOutputStream out = new ByteArrayOutputStream();
        soapResponse.writeTo(out);
        soapConnection.close();
        String resp = new String(out.toByteArray());
        return resp;
    }

    private static SOAPMessage createSOAPRequest(String m) throws Exception {
        SOAPMessage soapMessage = MessageFactory.newInstance().createMessage();
        Document document =
DocumentBuilderFactory.newInstance().newDocumentBuilder().parse(new
ByteArrayInputStream(m.getBytes()));
        soapMessage.getSOAPPart().setContent(new DOMSource(document));
        soapMessage.saveChanges();
        return soapMessage;
    }
}

```

6.Test Plan

6.1 Introduzione

Lo scopo di questi test è verificare che il prodotto A&R Book Rental Service sia conforme alle specifiche scritte nel documento dei requisiti (SRS) e che il comportamento sia quello atteso dagli utenti. Questo test plan sarà composto da 2 livelli di test:

1. testing funzionale (functional testing);
2. testing strutturale (structural testing);

Nella sezione dei risultati dei test saranno riportate anche le statistiche di errore e la disponibilità del servizio attesa.

6.2 Functional testing

6.2.1 Funzionalità da testare

Saranno testate tutte le funzionalità descritte dal documento dei requisiti SRS.

6.2.2 Approccio

Si deve creare un test case per ogni requisito specificato nel SRS. Per ogni test case si deve verificare la risposta del sistema. In questo livello di testing si userà la tecnica del black boxing, costruendo degli input di test per i servizi usati e verificandone la risposta.

6.2.3 Pass Criteria

I test si considerano superati quando l'output di tutti i test è lo stesso dell'output atteso scritto nel documento dei requisiti (SRS).

6.2.4 Fail Criteria

I test si considerano falliti nel caso in cui un solo test non dia i risultati descritti nel documento dei requisiti.

6.2.5 Test Deliverables

I documenti prodotti da questo livello di testing saranno i seguenti:

1. Tabella dei test case
2. Output del sistema
3. Statistiche

TABELLA DEI TEST CASE

Requisito	Caso di test	Output desiderato	ID Output reale
SR1	<ul style="list-style-type: none">• Nuova richiesta	Il sistema mostra "NUOVA RICHIESTA LIBRO"	O1
SR2/SR3	<ul style="list-style-type: none">• Nuova richiesta• Inserimento richiesta• Lista Libri	Il sistema utilizza l'applicazione REP per il servizio di inserisciRichiestaLibro e riceve la conferma di inserimento, utilizza il servizio di OSL per il servizio trova libro e riceve la lista di libri trovati	O2
SR4	<ul style="list-style-type: none">• Nuova richiesta• Inserimento richiesta• Lista Libri• Libro Proposto	Il sistema utilizza l'applicazione REP per il servizio filtraRicercaLibro e riceve il libro proposto	O3

SR5	<ul style="list-style-type: none"> • Nuova richiesta • Inserimento richiesta • Lista Libri • Recensioni 	Il sistema utilizza l'applicazione RLU per il servizio leggiRecensione e riceve la lista di recensioni	O4
SR6	<ul style="list-style-type: none"> • Nuova richiesta • Inserimento richiesta • Lista Libri • Libro Proposto • Libro preso in prestito/prenotato 	Il sistema utilizza l'applicazione OSL per il servizio cambiaStatoLibro e riceve la conferma del cambio di stato	O5
SR7/SR10	<ul style="list-style-type: none"> • Nuova richiesta • Inserimento richiesta • Lista Libri • Libro Proposto • Libro preso in prestito/prenotato • Consegna con corriere • Aggiornamento date prestito 	Il sistema utilizza l'applicazione APC per il servizio consegnaConCorriere e riceve la notifica della data di prelievo/consegna, utilizza l'applicazione OSL per il servizio cambiaDataPrestito e riceve la conferma dell'aggiornamento delle date di prestito	O6
SR8	<ul style="list-style-type: none"> • Nuova richiesta • Inserimento richiesta • Lista Libri • Libro Proposto • Libro preso in prestito/prenotato • Consegna di persona 	Il sistema utilizza l'applicazione APC per il servizio consegnaDiPersona e riceve la notifica della data,luogo,ora di scambio	O7

SR9	<ul style="list-style-type: none"> • Nuova richiesta • Inserimento richiesta • Lista Libri • Libro Proposto • Libro preso in prestito/prenotato • Consegna con corriere/Consegna di persona • Scrittura recensione 	Il sistema utilizza l'applicazione RLU per il servizio inserisciRecensione e riceve la conferma di inserimento della recensione	O8
-----	---	---	----

Output del sistema

ID	Output del sistema
O1	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa</p>
O2	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa</p> <p>call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 2</p> <p>##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro</p> <p>##### LIBRI TROVATI #####</p>

O3	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, vicinanza, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 1 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e la camera dei segreti;J.K. Rowling;Fantasy;John Smith@Pisa;2</p>
O4	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 2 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>Priorità = affidabilità call RecensioneLibroUtente: leggiRecensione</p>
O5	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, vicinanza, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 1 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e la camera dei segreti;J.K. Rowling;Fantasy;John Smith@Pisa;2 call OffertaStatoLibro: cambiaStatoLibro LIBRO IN PRESTITO E PRENOTABILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>LIBRO PRENOTATO!!!!!! Harry Potter e la camera dei segreti;J.K. Rowling;Fantasy;John Smith@Pisa;3</p>

O6

NUOVA RICHIESTA LIBRO

Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa
call RichiestaEPrestito: inserisciRichiestaLibro

id richiesta 3

INSERITA RICHIESTA LIBRO

call OffertaStatoLibro: trovaLibro

LIBRI TROVATI

call RichiestaEPrestito: filtraRicercaLibro

PROPOSTO UN LIBRO

Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1

call OffertaStatoLibro: cambiaStatoLibro

LIBRO DISPONIBILE!!!!!!

STATO DEL LIBRO MODIFICATO

CONSEGNA CON CORRIERE!!!!!!

call AppuntamentoPerConsegna: consegnaConCorriere

NOTIFICA CONSEGNA CORRIERE

call OffertaStatoLibro: cambiaDataPrestito

DATE PRESTITO AGGIORNATE

O7	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 3</p> <p>##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro</p> <p>##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro</p> <p>##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1 call OffertaStatoLibro: cambiaStatoLibro LIBRO DISPONIBILE!!!!!!</p> <p>##### STATO DEL LIBRO MODIFICATO #####</p> <p>CONSEGNA DI PERSONA!!!!!!</p> <p>call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1</p> <p>##### PROPOSTA CONSEGNA ACCETTATA #####</p> <p>call OffertaStatoLibro: cambiaDataPrestito</p> <p>##### DATE PRESTITO AGGIORNATE #####</p>
----	---

O8

NUOVA RICHIESTA LIBRO

Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa

call RichiestaEPrestito: inserisciRichiestaLibro

id richiesta 3

INSERITA RICHIESTA LIBRO

call OffertaStatoLibro: trovaLibro

LIBRI TROVATI

call RichiestaEPrestito: filtraRicercaLibro

PROPOSTO UN LIBRO

Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1

call OffertaStatoLibro: cambiaStatoLibro

LIBRO DISPONIBILE!!!!!!

STATO DEL LIBRO MODIFICATO

CONSEGNA CON CORRIERE!!!!!!

call AppuntamentoPerConsegna: consegnaConCorriere

NOTIFICA CONSEGNA CORRIERE

call OffertaStatoLibro: cambiaDataPrestito

DATE PRESTITO AGGIORNATE

RESTITUZIONE LIBRO

PRESTITO SCADUTO!!!!!!

CONSEGNA CON CORRIERE!!!!!!

call AppuntamentoPerConsegna: consegnaConCorriere

NOTIFICA CONSEGNA CORRIERE

call RecensioneLibroUtente: inserisciRecensione

RECENSIONE INSERITA

Statistiche

Test eseguiti	150000
OSL call	453568
REP call	289831
APC call	274404
RLU call	257450
OSL fault	1634
REP fault	2838
APC fault	2733
RLU fault	2615
Gestioni corrette	140180
Fault calls	9820
Agent Availability	93,4533%

- **OSL/REP/APC/RLU call:** il numero di chiamate ai servizi di OSL, REP, APC, RLU rispettivamente
- **OSL/REP/APC/RLU fault:** il numero di volte in cui i servizi OSL, REP, APC, RLU non sono disponibili
- **Agent Availability:** calcolata come $(1 - (Fault\ calls \div Test\ eseguiti)) \times 100$

Applicazione	Risposta	Probabilità
OSL	risposta attesa	99,64%
	fault	0,360%
REP	risposta attesa	99,031%
	fault	0,979%

APC	risposta attesa	99,004%
	fault	0,996%
RLU	risposta attesa	98,085%
	fault	1,015%

6.3 Structural testing

6.3.1 Funzionalità da testare

Dovranno essere testate tutte le righe di codice del modulo dell'Agent e, quindi, tutte le funzionalità del software.

6.3.2 Approach

Si dovrà creare un test case per ogni requisito specificato nel SRS. Per ogni test case si dovrà verificare che la risposta del sistema sia quella attesa. In questo livello di testing si userà la tecnica del white boxing usando i criteri di Statement Coverage e di Decision Coverage, costruendo degli input di test per le funzionalità usate in modo che ogni istruzione del programma sarà eseguita almeno una volta da almeno un test dell'insieme di dati di testing. I due criteri di copertura del codice, con i documenti prodotti usando tali criteri, sono compresi entrambi in questa sezione.

6.3.3 Pass Criteria

I test si considereranno superati quando l'output di tutti i test risulterà essere l'output atteso per il requisito testato.

6.3.4 Fail Criteria

I test si considereranno falliti nel caso in cui uno solo dei test non darà i risultati attesi.

6.3.5 Test Deliverables

I documenti prodotti da questo livello di testing saranno i seguenti:

1. Control Flow Graph (CFG) dell'Agent
2. Control Flow Graph (CFG) di ogni funzionalità dell'Agent
3. Insieme dei dati di test
4. Output del sistema dato dall'esecuzione dei test usando il criterio di Statement Coverage

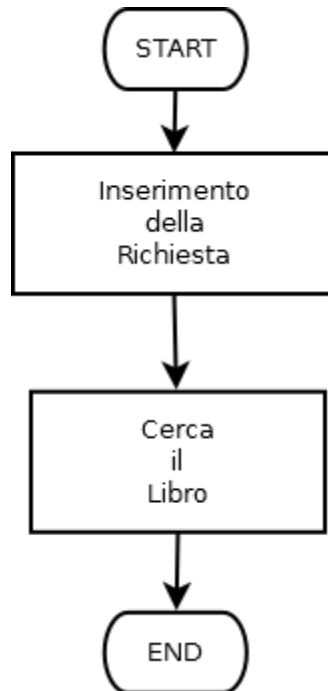
5. Output del sistema dato dall'esecuzione dei test usando il criterio di Decision Coverage

CFG dell'Agent



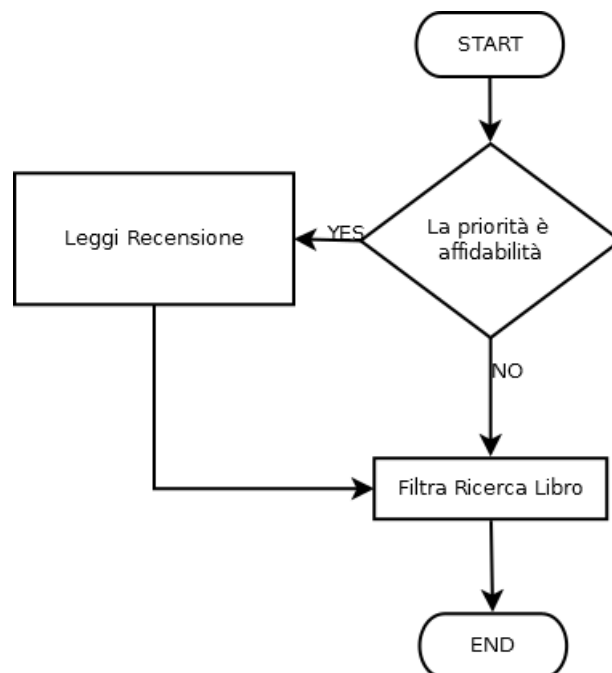
CFG RICERCA DEL LIBRO

Questa funzionalità si compone dei blocchi logici individuati a partire dal Control Flow Graph dell'Agent.



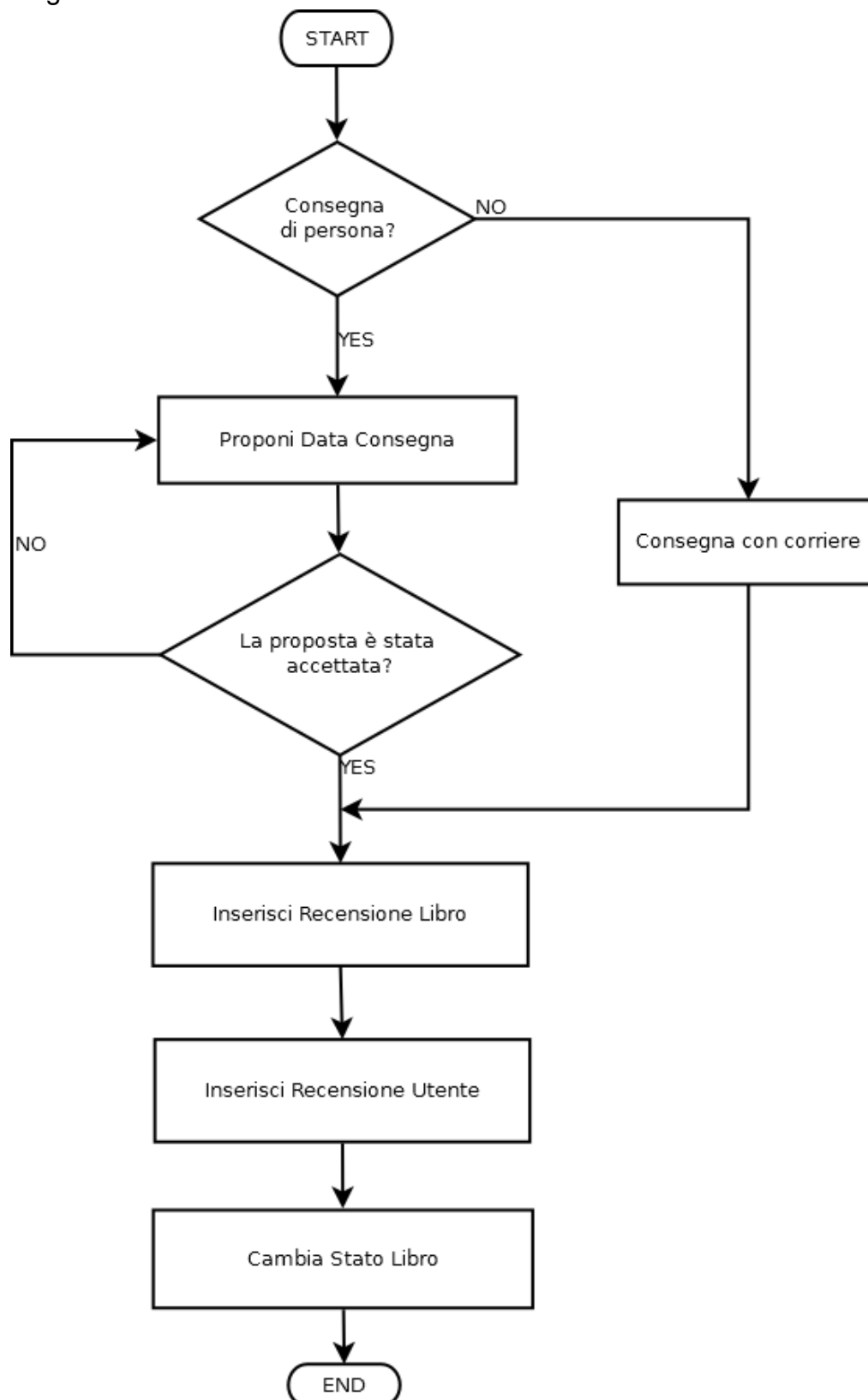
CFG FILTRAGGIO RISULTATI RICERCA

Questa funzionalità si compone dei blocchi logici individuati a partire dal Control Flow Graph dell'Agent.



CFG: SPEDIZIONE DEL LIBRO

Questa funzionalità si compone dei blocchi logici individuati a partire dal Control Flow Graph dell'Agent.



STATEMENT COVERAGE

Il criterio chiamato Statement Coverage prevede che ogni riga di codice del programma sia eseguita almeno una volta da almeno un test del set di dati di testing. Il set di dati è riportato nella seguente tabella:

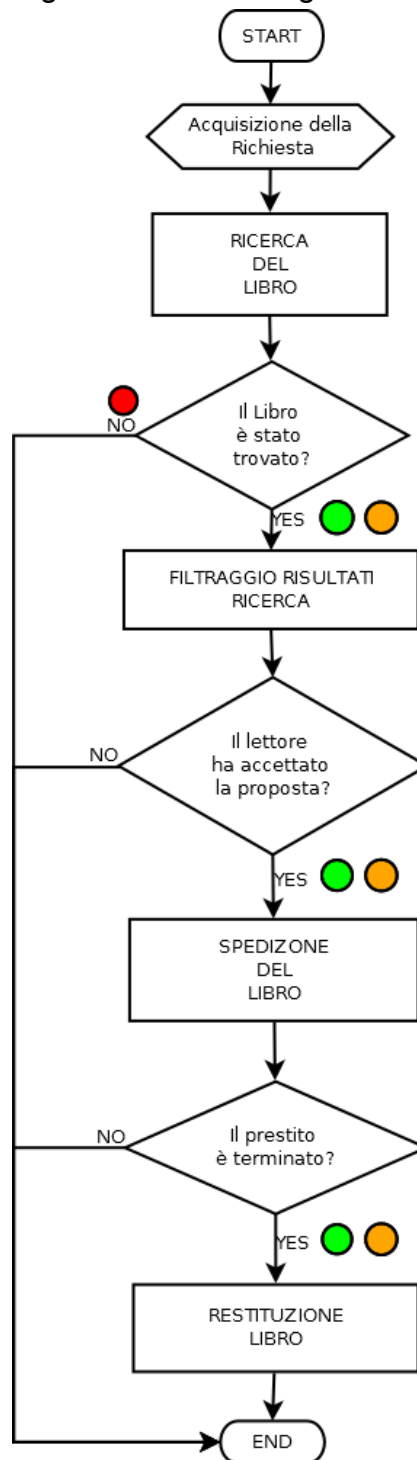
Input	Colore	ID
<ul style="list-style-type: none">• Ricerca il libro.• Il libro non è stato trovato.	ROSSO	SET1
<ul style="list-style-type: none">• Ricerca il libro.• Il libro è stato trovato.• La priorità è affidabilità.• Il lettore ha accettato la proposta.• La consegna è di persona.• La proposta di consegna è stata accettata.• Il prestito è terminato.• La consegna è di persona.• La proposta di consegna è stata accettata.	VERDE	SET2
<ul style="list-style-type: none">• Ricerca il libro.• Il libro è stato trovato.• La priorità non è affidabilità.• Il lettore ha accettato la proposta.• La consegna non è di persona.• Il prestito è terminato.• La consegna non è di persona.	ARANCIONE	SET3

I colori usati indicano i rami del diagramma di flusso che il test seguirà per ogni modulo definito nella parte dei Control Flow Graph. È stato omesso il grafico del modulo di ricerca del libro perché non contiene al suo interno dei branch condizionali.

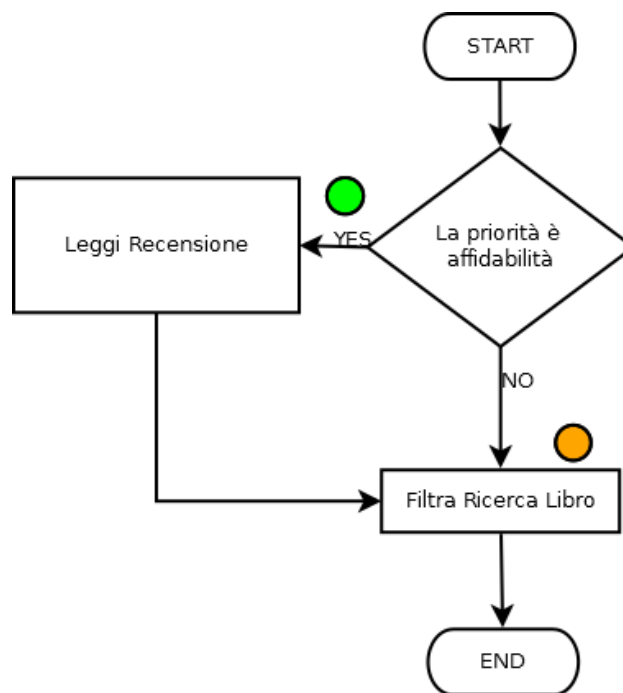
Nelle pagine seguenti sono visualizzati graficamente quali sono i path che gli insiemi di test seguono durante l'esecuzione del programma.

CFG Statement Coverage: Agent

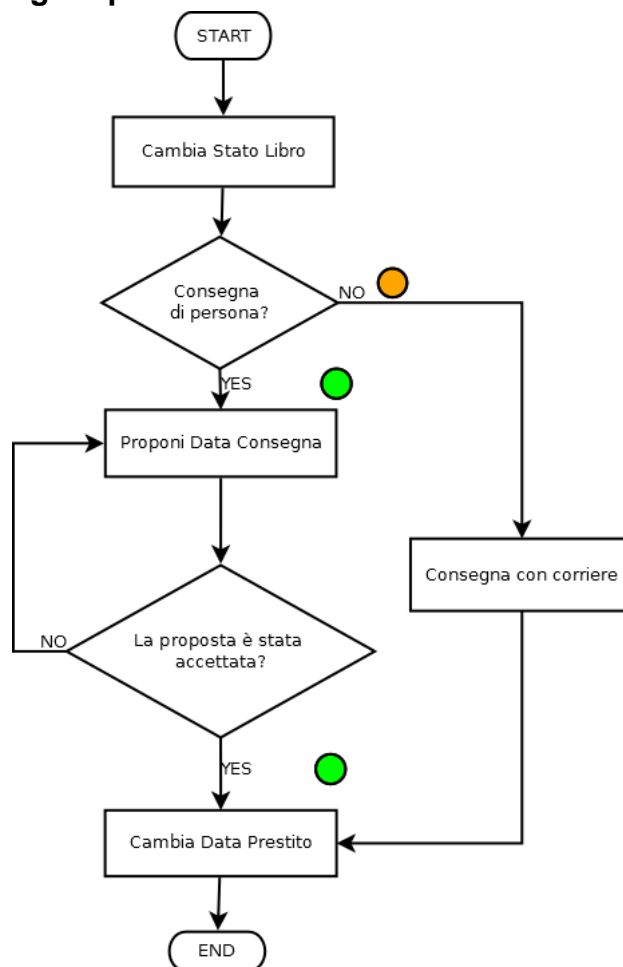
Il test set SET1 deve coprire le righe di codice che gestiscono l'uscita dal programma.



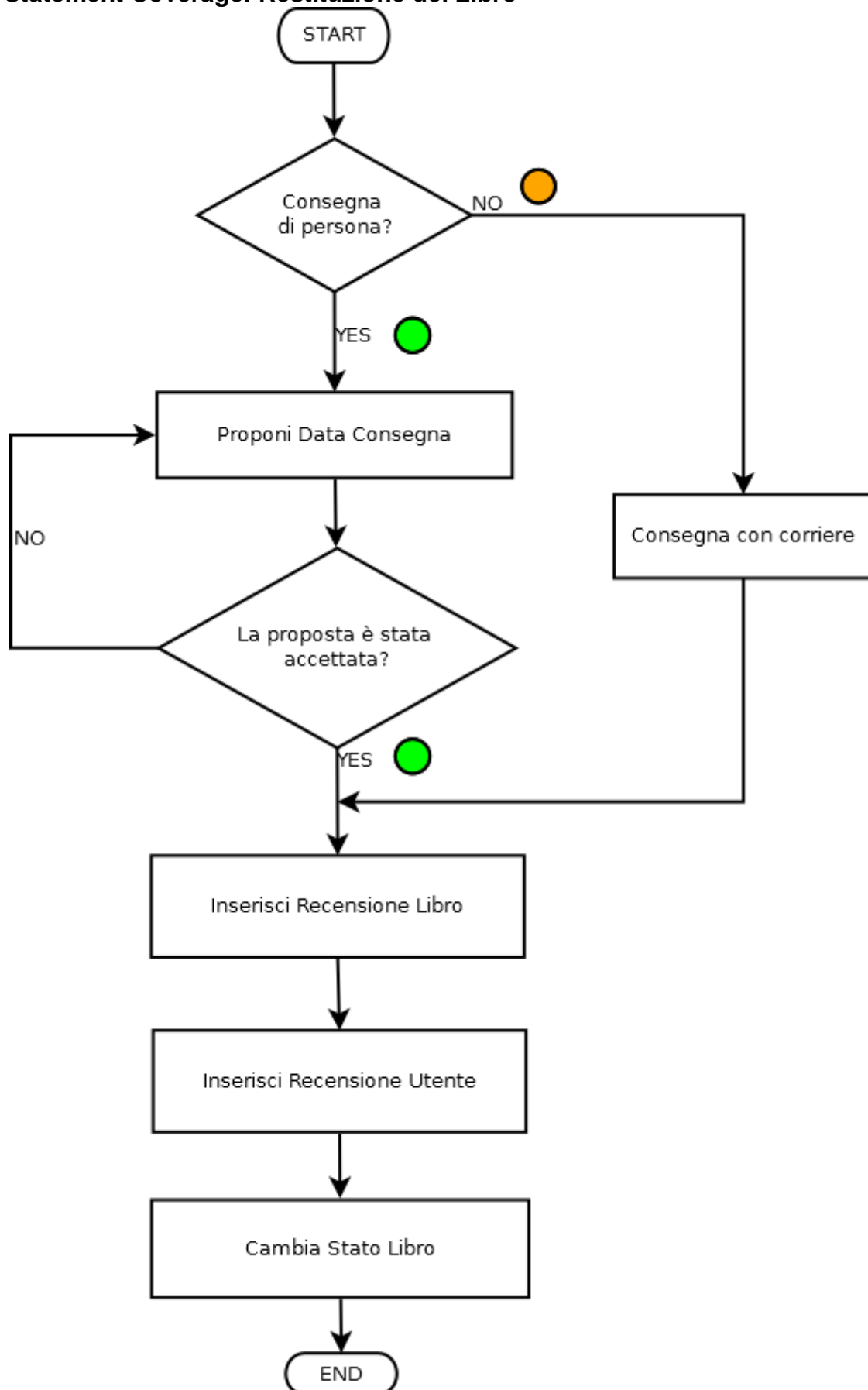
CFG Statement Coverage: Filtraggio Risultati Ricerca



CFG Statement Coverage: Spedizione del Libro



CFG Statement Coverage: Restituzione del Libro



OUTPUT DEL CRITERIO STATEMENT COVERAGE

Test ID	Output del sistema
SET1	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, vicinanza, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 1 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### NESSUN LIBRO TROVATO #####</p> <p>##### FINE #####</p>
SET2	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 2 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>Priorità = affidabilità call RecensioneLibroUtente: leggiRecensione call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Luca Bianchi@Pisa;2 call OffertaStatoLibro: cambiaStatoLibro LIBRO IN PRESTITO E PRENOTABILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>LIBRO PRENOTATO!!!!!! Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Luca Bianchi@Pisa;3 ##### CONSEGNA DEL LIBRO #####</p> <p>CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1 ##### PROPOSTA CONSEGNA ACCETTATA#####</p> <p>call OffertaStatoLibro: cambiaDataPrestito ##### DATE PRESTITO AGGIORNATE#####</p> <p>##### RESTITUZIONE LIBRO #####</p>

	<p>PRESTITO SCADUTO!!!!!! CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1 ##### PROPOSTA CONSEGNA ACCETTATA #####</p> <p>call RecensioneLibroUtente: inserisciRecensione ##### RECENSIONE INSERITA #####</p> <p>call OffertaStatoLibro: cambiaStatoLibro ##### STATO DEL LIBRO MODIFICATO#####</p> <p>##### FINE #####</p>
SET3	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 3 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1 call OffertaStatoLibro: cambiaStatoLibro LIBRO DISPONIBILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>CONSEGNA CON CORRIERE!!!!!! call AppuntamentoPerConsegna: consegnaConCorriere ##### NOTIFICA CONSEGNA CORRIERE #####</p> <p>call OffertaStatoLibro: cambiaDataPrestito ##### DATE PRESTITO AGGIORNATE #####</p> <p>##### RESTITUZIONE LIBRO #####</p> <p>PRESTITO SCADUTO!!!!!! CONSEGNA CON CORRIERE!!!!!! call AppuntamentoPerConsegna: consegnaConCorriere ##### NOTIFICA CONSEGNA CORRIERE #####</p> <p>call RecensioneLibroUtente: inserisciRecensione ##### RECENSIONE INSERITA #####</p> <p>call OffertaStatoLibro: cambiaStatoLibro ##### STATO DEL LIBRO MODIFICATO ##### ##### FINE #####</p>

DECISION COVERAGE

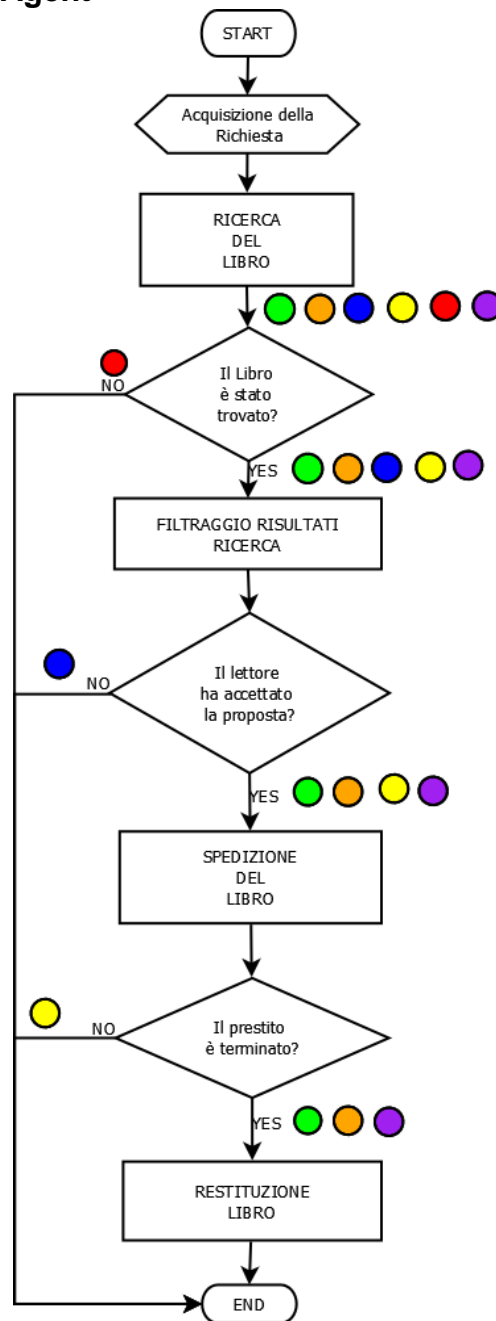
Il criterio chiamato Decision Coverage prevede che siano coperti tutti i possibili percorsi decisionali presenti nel codice dell'Agent. Il set di dati è riportato nella seguente tabella:

Input	Colore	ID
<ul style="list-style-type: none">● Ricerca il libro● Il libro non è stato trovato	ROSSO	T1
<ul style="list-style-type: none">● Ricerca il libro● Il libro è stato trovato● La priorità non è affidabilità● Il lettore ha accettato la proposta● La consegna è di persona● La proposta di consegna è stata accettata● Il prestito è terminato● La consegna è di persona● La proposta di consegna è stata accettata	VERDE	T2
<ul style="list-style-type: none">● Ricerca il libro● Il libro è stato trovato● La priorità non è affidabilità● Il lettore ha accettato la proposta● La consegna non è di persona● Il prestito è terminato● La consegna non è di persona	ARANCIONE	T3
<ul style="list-style-type: none">● Ricerca il libro● Il libro è stato trovato● La priorità è affidabilità● Il lettore non ha accettato la proposta	BLU	T4
<ul style="list-style-type: none">● Ricerca del libro● Il libro è stato trovato● La priorità è affidabilità● Il lettore ha accettato la proposta● La consegna è di persona● La proposta non è stata accettata● La proposta è stata accettata● Il prestito non è terminato	GIALLO	T5
<ul style="list-style-type: none">● Ricerca del libro● Il libro è stato trovato● La priorità non è affidabilità● Il lettore ha accettato la proposta● La consegna è di persona● Il prestito è terminato● La consegna è di persona	VIOLA	T6

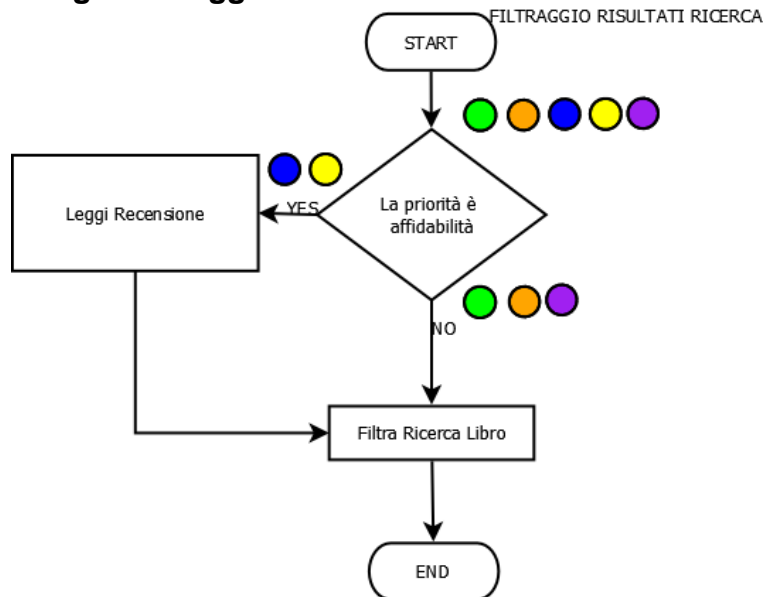
I colori usati indicano i rami del diagramma di flusso che il test seguirà per ogni modulo definito nella parte dei Control Flow Graph. È stato omesso il grafico del modulo di ricerca del libro perché non contiene al suo interno dei branch condizionali.

Nelle pagine seguenti sono visualizzati graficamente quali sono i path che gli insiemi di test seguono durante l'esecuzione del programma.

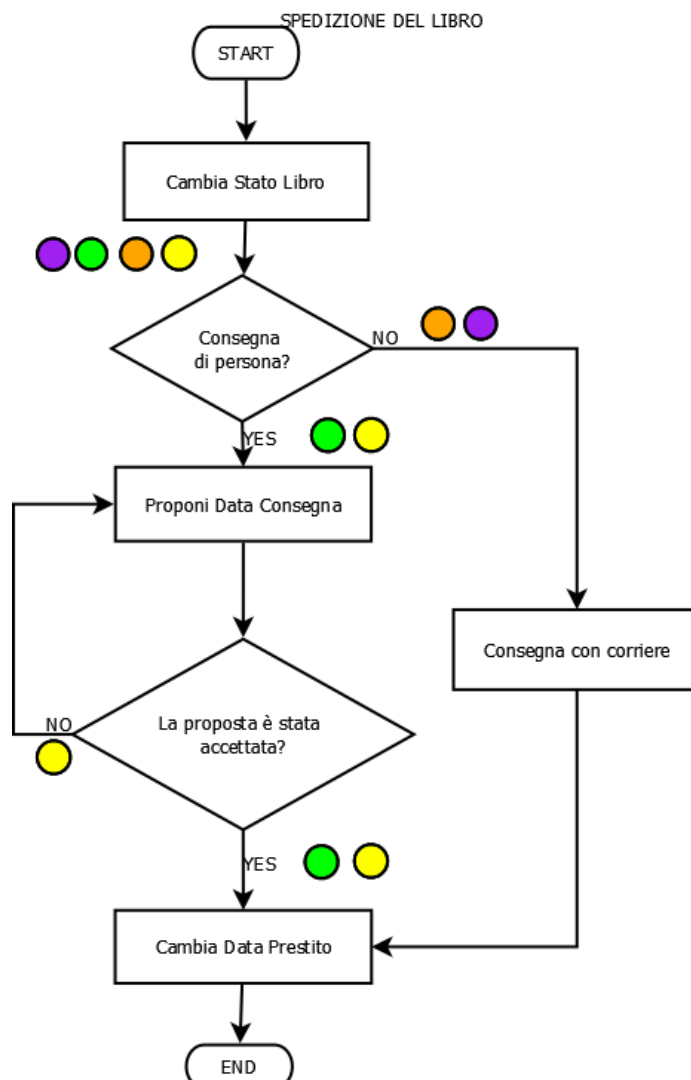
CFG Decision Coverage: Agent



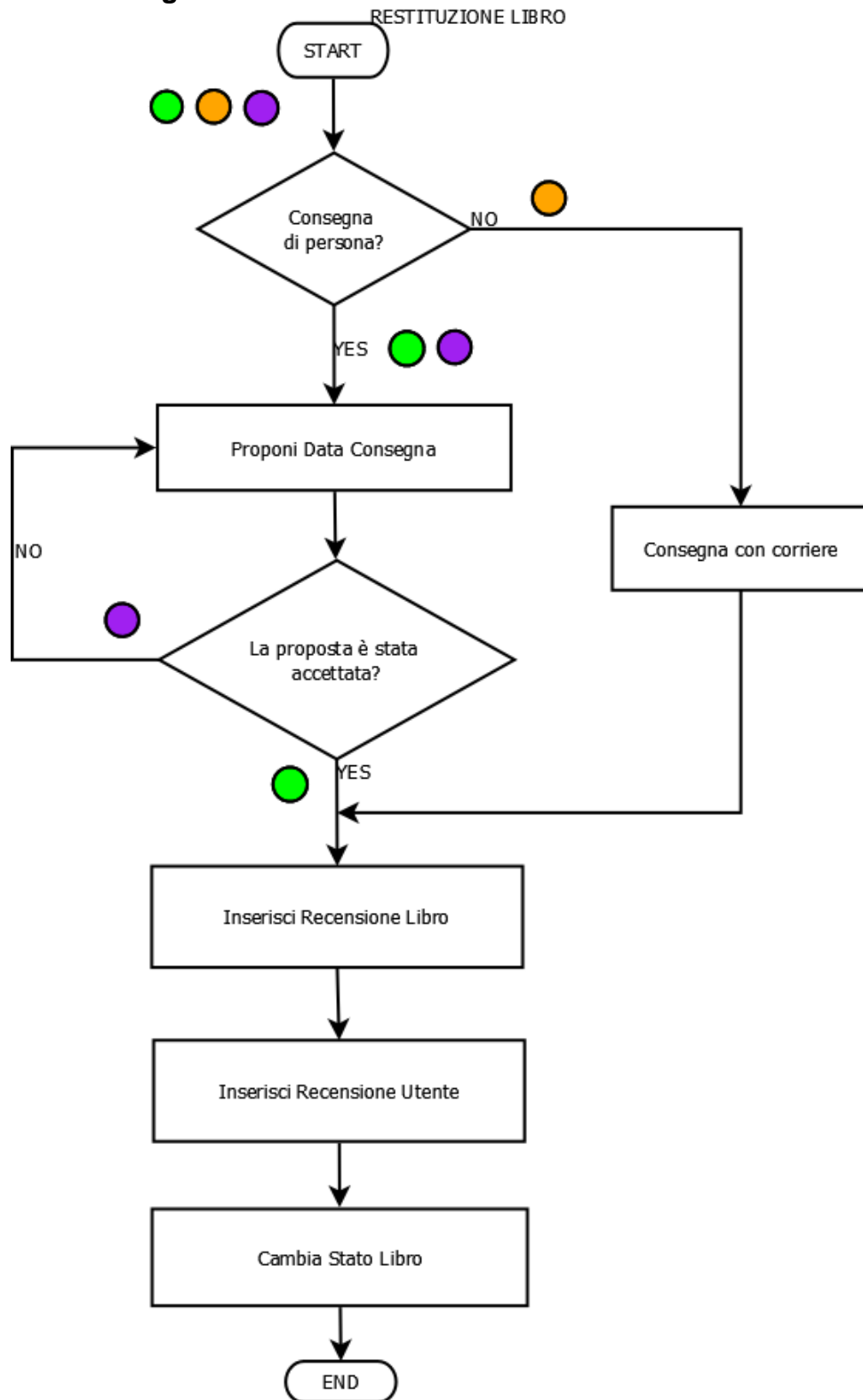
CFG Decision Coverage: Filtraggio Risultati Ricerca



CFG Decision Coverage: Spedizione del Libro



CFG Decision Coverage: Restituzione Libro



Test ID	Output di sistema
T1	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, vicinanza, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 1 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### NESSUN LIBRO TROVATO #####</p> <p>##### FINE #####</p>
T2	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 3 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1 call OffertaStatoLibro: cambiaStatoLibro LIBRO DISPONIBILE!!!!!! ##### STATO DEL LIBRO MODIFICATO#####</p> <p>CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1 ##### PROPOSTA CONSEGNA ACCETTATA#####</p> <p>call OffertaStatoLibro: cambiaDataPrestito ##### DATE PRESTITO AGGIORNATE #####</p> <p>##### RESTITUZIONE LIBRO #####</p> <p>PRESTITO SCADUTO!!!!!! CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1 ##### PROPOSTA CONSEGNA ACCETTATA#####</p> <p>call RecensioneLibroUtente: inserisciRecensione ##### RECENSIONE INSERITA #####</p> <p>call OffertaStatoLibro: cambiaStatoLibro ##### STATO DEL LIBRO MODIFICATO #####</p> <p>##### FINE #####</p>
T3	<p>##### NUOVA RICHIESTA LIBRO #####</p>

	<p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 3 ##### INSERTA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1 call OffertaStatoLibro: cambiaStatoLibro LIBRO DISPONIBILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>CONSEGNA CON CORRIERE!!!!!! call AppuntamentoPerConsegna: consegnaConCorriere ##### NOTIFICA CONSEGNA CORRIERE#####</p> <p>call OffertaStatoLibro: cambiaDataPrestito ##### DATE PRESTITO AGGIORNATE #####</p> <p>##### RESTITUZIONE LIBRO #####</p> <p>PRESTITO SCADUTO!!!!!! CONSEGNA CON CORRIERE!!!!!! call AppuntamentoPerConsegna: consegnaConCorriere ##### NOTIFICA CONSEGNA CORRIERE#####</p> <p>call RecensioneLibroUtente: inserisciRecensione ##### RECENSIONE INSERITA #####</p> <p>call OffertaStatoLibro: cambiaStatoLibro ##### STATO DEL LIBRO MODIFICATO #####</p> <p>##### FINE #####</p>
T4	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 2 ##### INSERTA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>Priorità = affidabilità call RecensioneLibroUtente: leggiRecensione call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>##### PROPOSTA LIBRO NON ACCETTATA#####</p> <p>##### FINE #####</p>

T5	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, affidabilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 2 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>Priorità = affidabilità call RecensioneLibroUtente: leggiRecensione call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO ##### Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Luca Bianchi@Pisa;2 call OffertaStatoLibro: cambiaStatoLibro</p> <p>LIBRO DISPONIBILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona TENTATIVO ACCORDO DI CONSEGNA NUMERO 1 TENTATIVO ACCORDO DI CONSEGNA NUMERO 2 TENTATIVO ACCORDO DI CONSEGNA NUMERO 3 TENTATIVO ACCORDO DI CONSEGNA NUMERO 4 TENTATIVO ACCORDO DI CONSEGNA NUMERO 5 ##### PROPOSTA CONSEGNA ACCETTATA#####</p> <p>call OffertaStatoLibro: cambiaDataPrestito ##### DATE PRESTITO AGGIORNATE #####</p> <p>##### RESTITUZIONE LIBRO #####</p> <p>PRESTITO NON SCADUTO!!!!!! ##### FINE #####</p>
T6	<p>##### NUOVA RICHIESTA LIBRO #####</p> <p>Harry Potter e la camera dei segreti, fantasy, J.K.Rowling, disponibilita, Mario Rossi, Pisa call RichiestaEPrestito: inserisciRichiestaLibro id richiesta 3 ##### INSERITA RICHIESTA LIBRO #####</p> <p>call OffertaStatoLibro: trovaLibro ##### LIBRI TROVATI #####</p> <p>call RichiestaEPrestito: filtraRicercaLibro ##### PROPOSTO UN LIBRO #####</p> <p>Harry Potter e il calice di fuoco;J.K. Rowling;Fantasy;Franco Mario@Roma;1 call OffertaStatoLibro: cambiaStatoLibro LIBRO DISPONIBILE!!!!!! ##### STATO DEL LIBRO MODIFICATO #####</p> <p>CONSEGNA DI PERSONA!!!!!! call AppuntamentoPerConsegna: consegnaDiPersona</p>

<p>TENTATIVO ACCORDO DI CONSEGNA NUMERO 1</p> <p>##### PROPOSTA CONSEGNA ACCETTATA#####</p> <p>call OffertaStatoLibro: cambiaDataPrestito</p> <p>##### DATE PRESTITO AGGIORNATE #####</p> <p>##### RESTITUZIONE LIBRO #####</p> <p>PRESTITO SCADUTO!!!!!!</p> <p>CONSEGNA CON CORRIERE!!!!!!</p> <p>call AppuntamentoPerConsegna: consegnaConCorriere</p> <p>##### NOTIFICA CONSEGNA CORRIERE#####</p> <p>call RecensioneLibroUtente: inserisciRecensione</p> <p>##### RECENSIONE INSERITA #####</p> <p>call OffertaStatoLibro: cambiaStatoLibro</p> <p>##### STATO DEL LIBRO MODIFICATO #####</p> <p>##### FINE #####</p>
--