UNIVERSITÀ
DEGLI STUDI
DI BRESCIA

DIPARTIMENTO DI INGEGNERIA

Elaborato di Ingegneria del Software

Parte A. Release 2.

SOFTWARE PER LA GESTIONE DI FILE MULTIMEDIALI

Alberto Rovetta

Lorenzo Tomasetti

Denis Barbascumpa

Anno Accademico 2018/2019

REQUISITI FUNZIONALI

- Il modello di processo da adottare è incrementale/iterativo.
- Il linguaggio di programmazione da utilizzare è Java.
- L'architettura esterna da realizzare per l'applicazione è stand alone.
- Non è richiesta la creazione di una interfaccia utente grafica.
- Non è richiesto l'impiego di alcun DBMS (Data Base Management System).
- I file che si occupano dei salvataggi dei Prestiti, degli Utenti, degli Operatori e dei Fruitori si creano automaticamente al primo utilizzo del programma, ma la cartella 'src/Local_database' deve essere creata prima della prima esecuzione. Eventuali spostamenti, modifiche dei file destinati al salvataggio comporteranno un malfunzionamento del programma.
- La gestione delle scadenze dei Prestiti e dei Fruitori viene effettuata ad ogni accensione del sistema. Tale funzionamento è stato pensato consapevolmente che sarebbe più opportuno effettuare una verifica ogni giorno alle 00:00.

SCELTE PROGETTUALI VERSIONE 2

- Per implementare le risorse multimediali abbiamo creato una Classe astratta 'Risorsa' che contiene tutti i metodi utili per la gestione delle risorse. Ogni 'Risorsa' è definita univocamente da un ID ed al numero di copie presenti. Abbiamo quindi creato la classe 'Libro', che estende 'Risorsa', che contiene un arrayList dei dati relativi alla risorsa libro(titolo, autore/i, numero di pagine, ecc...),E' stato scelto questo percorso in modo che sia poi possibile inserire altri tipi di risorse creando la relativa classe estendendo la classe padre 'Risorsa'.
- Per implementare le categorie delle risorse abbiamo creato una Classe 'Categoria' che contiene un arrayList di Risorsa ed un arrayList di Categoria. Abbiamo scelto questo approccio in quanto è possibile in futuro avere più sottocategorie e non soltanto una come specificato per questa versione dal testo.

CASI D'USO (Modalità testuale)

Nome	**Operatore Loggato**
Attori	Operatore
Precondizioni	L'utente sia loggato come operatore
Scenario Principale	1. Il Sistema mostra all'Operatore le operazioni che può compiere. 2. L'operatore richiede al Sistema di visualizzare tutti i Fruitori. 3. Il Sistema visualizza la lista di tutti i Fruitori. Torna a 1.
Scenario Alternativo (a)	2.a L'Operatore sceglie di tornare alla condizione di Utente. Fine..
Postcondizioni (a)	L'Operatore torna ad essere un Utente.
Scenario Alternativo (b)	2.b L'Operatore sceglie di visualizzare a video tutti i Libri. 3.b Il Sistema visualizza tutti i Libri.Torna al punto 1.
Scenario Alternativo (c)	2.c L'Operatore sceglie di aggiungere la descrizione ad un Libro. 3.c Include<<Aggiungi Descrizione>>. Fine..
Scenario Alternativo (d)	2.d L'Operatore sceglie di rimuovere la descrizione ad un Libro. 3.d Include<<Rimuovi Descrizione>>. Fine..

Nome	**Aggiungi Descrizione**
Attore	Operatore
Precondizioni	Il Libro è presente
Scenario Principale	1. L'Operatore inserisce l'ID del Libro. 2. L'Operatore inserisce la nuova descrizione del Libro. 3. La risorsa viene aggiornata con la nuova descrizione. Fine..
Precondizioni (a)	Il Libro non è presente
Scenario Alternativo (a)	1.a L'Operatore inserisce l'ID del Libro. 2.a Il Sistema visualizza un messaggio di errore. Fine..

Nome	**Rimozione Descrizione**
Attori	Operatore
Precondizioni	Il Libro è presente
Scenario Principale	1. L'Operatore inserisce l'ID del Libro. 2. Il Sistema rimuove la descrizione al Libro. 3. La risorsa viene aggiornata con la nuova descrizione. Fine.
Precondizioni (a)	Il Libro non è presente
Scenario Alternativo (a)	1.a L'Operatore inserisce l'ID del Libro. 2.a Il Sistema visualizza un messaggio di errore. Fine.

DIAGRAMMA CASI D'USO

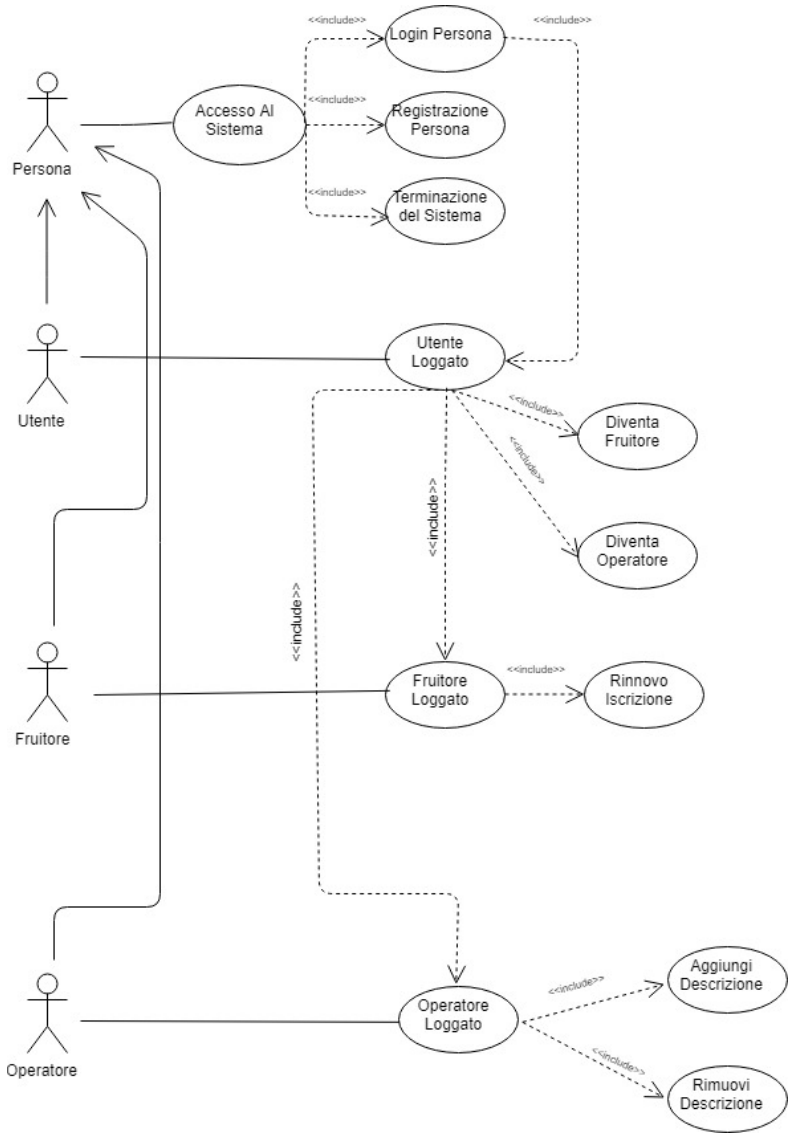

DIAGRAMMA UML