



IMPACT ANALYSIS DOCUMENT

PocketMuseum

Riferimento	IAD_1.0
Versione	1.0
Data	22/06/2021
Destinatario	Andrea De Lucia
Presentato da	Simone Avolicino
Approvato da	



Revision History

Data	Versione	Cambiamenti	Autore
20/04/2021	0.1	Inizio stesura	Simone Avolicino
15/05/2021	0.2	Aggiunta SIS e CIS	Simone Avolicino
14/06/2021	0.3	Aggiunte AIS e conclusioni	Simone Avolicino
22/06/2021	1.0	Revisione	Simone Avolicino



SOMMARIO

1	Introduzione	2
1.1	Ambito del sistema	2
1.2	Panoramica sistema corrente	2
1.3	Attori	2
1.4	Requisiti funzionali	3
1.5	Database	3
1.6	Architettura	4
2	Impact Analysis CR_1	4
2.1	Descrizione	4
2.2	Requisiti funzionali	4
2.3	Class Diagram	4
2.4	Database	5
2.5	Grafo delle dipendenze	5
2.6	Starting Impact Set	6
2.7	Candidate Impact Set	6
2.7.1	Modifica del requisito 'Recensire un'opera'	6
2.7.2	CIS	6
2.8	Actual Impact Set	6
2.8.1	Modifica del requisito 'Recensire un'opera'	7
2.8.2	AIS	7
2.9	Conclusioni	7
2.9.1	Recall	7
2.9.2	Precision	7
2.9.3	Adeguacy	7
2.9.4	Effectiveness	7
3	Impact Analysis CR_2	8
3.1	Descrizione	8
3.2	Requisiti funzionali	8
3.3	Class Diagram	8
3.4	Database	8
3.5	Grafo delle dipendenze	9
3.6	Starting Impact Set	10
3.7	Candidate Impact Set	10
3.7.1	Aggiunta del requisito 'Visualizza Eventi'	10
3.7.2	Aggiunta del requisito 'Crea Evento'	10
3.7.3	Aggiunta del requisito 'Modifica Evento'	11
3.7.4	Aggiunta del requisito 'Elimina Evento'	11



3.7.5	CIS	11
3.8	Actual Impact Set	12
3.8.1	Aggiunta del requisito 'Visualizza Eventi'	12
3.8.2	Aggiunta del requisito 'Crea Evento'	12
3.8.3	Aggiunte del requisito 'Modifica Evento'	12
3.8.4	Aggiunta del requisito 'Elimina Evento'	13
3.8.5	AIS	13
3.9	False Positive Impact Set	13
3.10	Conclusioni	13
3.10.1	Recall	13
3.10.2	Precision	14
3.10.3	Adeguacy	14
3.10.4	Effectiveness	14
4	Impact Analysis CR_3	14
4.1	Descrizione	14
4.2	Starting Impact Set	14
4.3	Candidate Impact Set	15
4.3.1	Modifica UI/UX	15
4.3.2	CIS	15
4.4	Actual Impact Set	16
4.4.1	Modifica UI/UX	16
4.4.2	AIS	16
4.5	False Positive Impact Set	16
4.6	Conclusioni	17
4.6.1	Recall	17
4.6.2	Precision	17
4.6.3	Adeguacy	17
4.6.4	Effectiveness	17

1 Introduzione

1.1 Ambito del sistema

Il Museo intende fornire un sistema interconnesso, interattivo e multimediale, in cui la visita alle sale espositive non viene più effettuata da una guida fisica, ma attraverso una guida digitale. Grazie a questa innovazione intende accrescere il numero di visitatori cercando di attrarre e avvicinare anche le nuove generazioni, ovvero una tipologia di utenti più vicina alle nuove tecnologie fornendo loro la possibilità di conoscere la storia delle opere di loro interesse e rilasciare recensioni per aumentare la diffusione delle opere. Il Museo intende fornire la possibilità di acquistare biglietti online per poter controllare il flusso di visitatori rispettando le attuali normative anti-covid e inoltre permette la gestione delle opere e la gestione della contabilità.

Il sistema in particolare deve supportare:

- La registrazione del visitatore alla piattaforma;
- L'acquisto dei tickets d'ingresso al Museo;
- Gestione del flusso di ingresso dei visitatori;
- La visualizzazione e descrizione delle opere presenti nel museo in formato digitale;
- La possibilità di recensire le opere in modo da facilitare la visita turistica ad altri visitatori;
- Mantenere una contabilità degli acquisti;
- Aggiungere, modificare ed eliminare delle opere.

1.2 Panoramica sistema corrente

La piattaforma punta a rivoluzionare il mondo museale offrendo agli amministratori (Biglietteria e Guida) nuovi metodi di gestione e ai clienti la possibilità di fruire in modo digitale dei servizi messi a disposizione dal museo.

La piattaforma propone all'utente Guida nuovi strumenti per la gestione delle opere, permettendo inoltre all'utente Biglietteria l'accelerazione del processo di vendita e la semplificazione della gestione della contabilità.

Al cliente, è data la possibilità di:

- Compilare il modulo di registrazione per accedere al sistema;
- Acquistare biglietti in forma digitale o tradizionale;
- Visualizzare i biglietti acquistati;
- Visualizzare, per il tempo della visita, informazioni relative alle opere del museo;
- Recensire le opere del museo;
- Modificare la propria password, se registrato;

Alla guida, è data la possibilità di:

- Aggiungere, modificare o rimuovere le opere presenti all'interno del sistema;
- Modificare la propria password;

Alla biglietteria, è data la possibilità di:

- Vendere biglietti in formato fisico;
- Gestire la contabilità del museo;
- Modificare la propria password.

1.3 Attori

Gli attori principali del sistema sono 4:

- Guida
- Biglietteria

- Utente registrato
- Utente non registrato

1.4 Requisiti funzionali

Lato Utente Registrato

- RF1 – L'utente registrato deve poter accedere alla pagina personale.
- RF2 – L'utente registrato deve poter visualizzare i turni disponibili.
- RF3 – L'utente registrato deve poter acquistare il biglietto per entrare al museo.
- RF4 – L'utente registrato deve poter visualizzare la cronologia degli acquisti.
- RF5 – L'utente registrato deve poter visualizzare i biglietti acquistati e i relativi codici.
- RF6 – L'utente registrato deve poter visualizzare le opere durante la visita solo se ha inserito correttamente il codice del biglietto.
- RF7 – L'utente registrato deve poter ricercare le opere.
- RF8 – L'utente registrato deve poter recensire un'opera.
- RF10 – L'utente registrato deve poter modificare la propria password.
- RF11 – L'utente registrato deve poter recuperare la propria password.

Lato Utente Non Registrato

- RF12 – L'utente non registrato deve poter registrarsi al sistema.
- RF13 – L'utente non registrato deve poter accedere alla visita guidata tramite codice, se valido.
- RF14 – L'utente non registrato deve poter visualizzare le opere durante la visita solo se ha inserito correttamente il codice del biglietto

Lato Guida

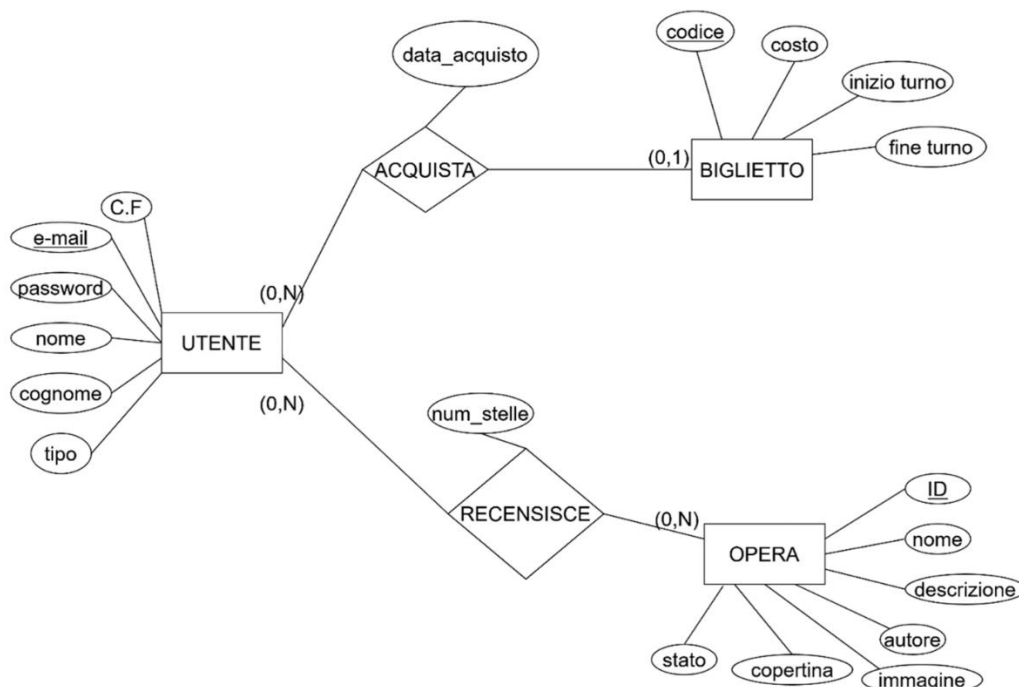
- RF15 – La guida deve poter accedere alla propria pagina personale.
- RF16 – La guida deve poter aggiungere un'opera.
- RF17 – La guida deve poter modificare un'opera.
- RF18 – La guida deve poter rimuovere un'opera.
- RF29 – La guida deve poter modificare la propria password.
- RF20 – La guida deve poter recuperare la propria password.

Lato Biglietteria

- RF21 – La biglietteria deve poter accedere alla propria pagina personale.
- RF22 – La biglietteria deve poter gestire la contabilità.
- RF23 – La biglietteria deve poter permettere l'acquisto del biglietto ad un utente non registrato.
- RF24 – La biglietteria deve poter modificare la propria password. Priorità Bassa
- RF25 – La biglietteria deve poter recuperare la propria password. Priorità Bassa

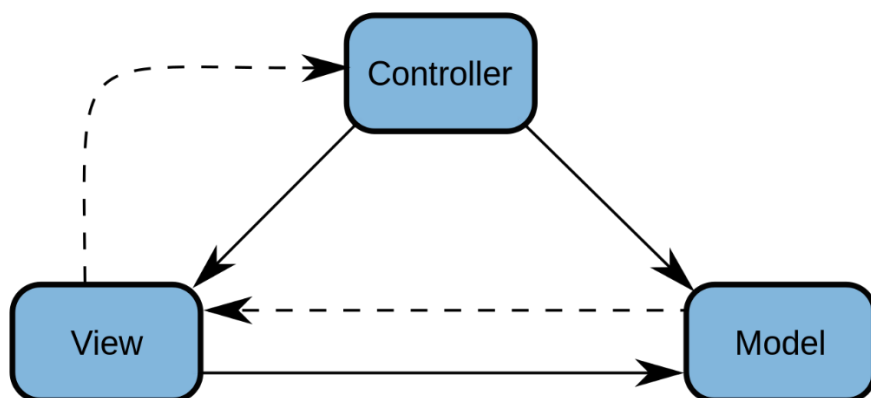
1.5 Database

Il sistema attuale utilizza un database relazionale Sql ed è il seguente:



1.6 Architettura

Il sistema attuale è basato su un'architettura MVC



2 Impact Analysis CR_1

2.1 Descrizione

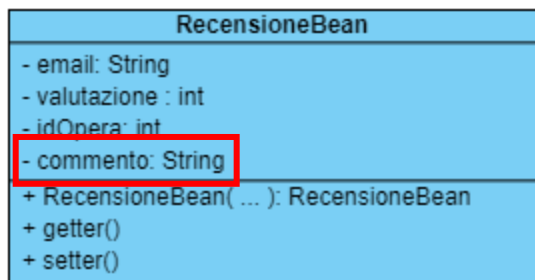
La change request CR_1 è relativa alla funzionalità già esistente 'Recensioni'. La modifica comporta l'inserimento di un nuovo attributo 'commento', il quale fornisce la possibilità all'attore Utente Registrato di inserire recensioni più dettagliate alle opere visualizzate.

2.2 Requisiti funzionali

A seguito delle modifiche, il requisito **RF8 – L'utente registrato deve poter recensire un'opera** subirà una modifica in termini di codice.

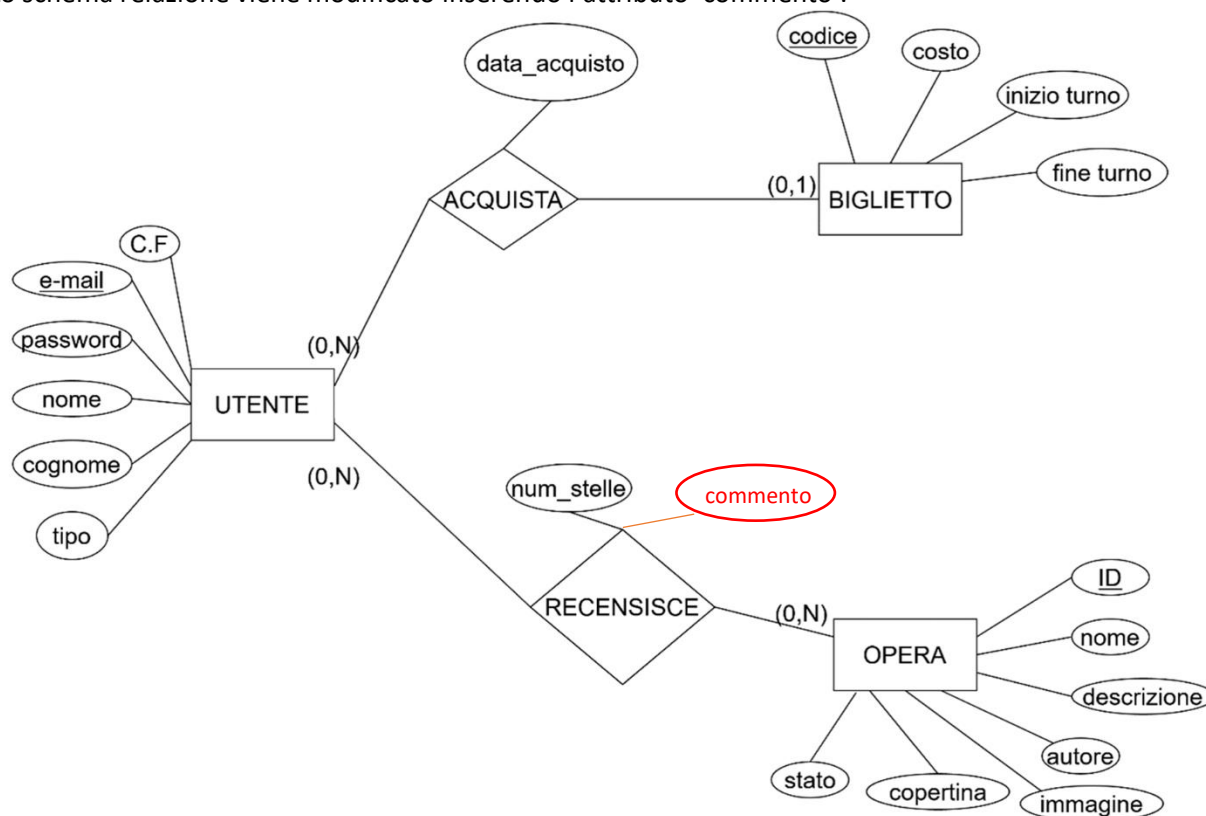
2.3 Class Diagram

La modifica di tale requisito comporta la modifica della classe RecensioneBean, diventando:



2.4 Database

Lo schema relazione viene modificato inserendo l'attributo 'commento'.



2.5 Grafo delle dipendenze

Generiamo il grafo delle dipendenze del sistema, in modo da avere una visione chiara e completa del sistema nel suo insieme, per poi andare ad analizzare la change request e vedere su quali parti del codice vanno ad impattare.

L'abbreviazione RF, si riferisce ai requisiti funzionali.

L'abbreviazione C si riferisce a Codice.

- C1 AggiungiRecensioneServlet.java
- C2 RecensioneBean.java
- C3 RecensioneModelDM.java
- C4 OpereTour.jsp

OUT(RF8) = {C1, C2, C3, C4}

OUT(C1) = {C2, C3, C4}

OUT(C2) = {}

OUT(C3) = {C2}
OUT(C4) = {}

2.6 Starting Impact Set

In questo paragrafo viene riportato lo Starting Impact Set, che rappresenta l'insieme delle componenti che verranno direttamente impattate da questa modifica. Nella tabella seguente verranno indicate le componenti presenti all'interno del nostro Starting Impact Set; l'impatto della modifica verrà valutato nel seguente modo:

- FORTE: nel caso in cui sia necessario riscrivere completamente l'artefatto;
- MEDIO: nel caso in cui sia necessario una rivisitazione e modifica di un artefatto;
- DEBOLE: nel caso in cui siano necessarie solo modifiche marginali;
- NULLO: nel caso in cui non siano presenti modifiche.

Artefatto	Impatto	Descrizione
Codice	Medio	Modificando il requisito RF_8, il codice relativo ad esso subirà delle modifiche per aggiungere un nuovo attributo.
DB	Medio	Modifica nell'entità 'Recensione' aggiungendo il nuovo attributo 'commento'.
Testing	Medio	La classe di test dovrà prendere come input anche il campo 'commento'

2.7 Candidate Impact Set

In questa sezione verranno indicati, per ogni requisito, eventuali impatti diretti e indiretti sulle altre componenti del sistema.

2.7.1 Modifica del requisito 'Recensire un'opera'

Tale modifica andrà ad impattare le classi 'AggiungiRecensioneServlet.java', 'RecensioneBean.java', 'RecensioneModelDM.java', 'AggiungiRecensioneTC' ed infine 'OpereTour.jsp'

CLASSE	IMPATTO
Control.opera.AggiungiRecensioneServlet.java	INDIRETTO
Model.opera.RecensioneBean.java	DIRETTO
Model.opera.RecensioneModelDM.java'	DIRETTO
webContent.OpereTour.jsp	DIRETTO
Test.AggiungiRecensioneTC	INDIRETTO

2.7.2 CIS

Di conseguenza, il nostro CIS ha cardinalità 5 ed è il seguente:

Control.opera.AggiungiRecensioneServlet.java
Model.opera.RecensioneBean.java
Model.opera.RecensioneModelDM.java'
webContent.OpereTour.jsp
Test.AggiungiRecensioneTC

2.8 Actual Impact Set

Di seguito sono elencate le componenti realmente modificate: tali componenti faranno parte del nostro Actual Impact Set (AIS).

2.8.1 *Modifica del requisito 'Recensire un'opera'*

2.8.1.1 **Control.opera.AggiungiRecensioneServlet.java**

La modifica ha previsto il controllo sull'ulteriore parametro 'commento' passato per l'inserimento della recensione.

2.8.1.2 **Model.opera.RecensioneBean.java**

La classe RecensioneBean è stata modificata aggiungendo al costruttore il parametro 'commento' oltre ai metodi getter e setter.

2.8.1.3 **Model.opera.RecensioneModelDM.java'**

Nella classe RecensioneModelDM sono stati modificati i metodi 'doSave' e 'doUpdate' nei quali si effettuano le query aggiungendo anche in questo caso l'attributo 'commento'.

2.8.1.4 **webContent.OpereTour.jsp**

Nella pagina OpereTour.jsp è stato modificato il form per l'inserimento del campo 'commento'.

2.8.1.5 **Test.AggiungiRecensioneTC**

La classe di test è stata modificata inserendo il parametro 'commento'.

2.8.2 *AIS*

Di conseguenza, il nostro AIS ha cardinalità 5 ed è il seguente:

Control.opera.AggiungiRecensioneServlet.java
Model.opera.RecensioneBean.java
Model.opera.RecensioneModelDM.java'
webContent.OpereTour.jsp
Test.AggiungiRecensioneTC

2.9 Conclusioni

Vengono mostrate le cardinalità dei nostri insiemi:

- Starting Impact Set |SIS| = 3
- Candidate Impact Set |CIS| = 5
- Actual Impact Set |AIS| = 5
- Discovered Impact Set |DIS| = 0
- False Positive Impact Set |FPIS| = 0

Per valutare l'accuratezza del processo di Impact Analysis sono state calcolate le seguenti metriche.

2.9.1 *Recall*

$$\text{Recall} = (CIS \cap AIS) / AIS = (5 \cap 5) / 5 = 100\%$$

2.9.2 *Precision*

$$\text{Precision} = (CIS \cap AIS) / CIS = (5 \cap 5) / 5 = 100\%$$

2.9.3 *Adeguacy*

2.9.3.1 *Inclusiveness*

AIS è incluso nel CIS dunque Inclusiveness = 1;

2.9.4 *Effectiveness*

2.9.4.1 *Ripple-sensitivity*

$$\text{Amplification} = |IISO| / |DISO| = 2/3 = 0,6$$

2.9.4.2 *Adherence*

$$\text{A-Ratio} = |AIS| / |CIS| = 5/5 = 1$$

3 Impact Analysis CR_2

3.1 Descrizione

La change request CR_2 prevede una nuova funzionalità, la quale coinvolge tutti gli attori del sistema. Gli attori Utente registrato, Utente non registrato e Biglietteria potranno visualizzare gli eventi che si terranno al museo; mentre, l'attore Guida sarà responsabile della creazione, modifica ed eliminazione degli eventi.

3.2 Requisiti funzionali

A seguito delle modifiche, verranno aggiunti i seguenti requisiti funzionali.

Lato Utente Registrato

RF26 – L'utente registrato deve poter visualizzare gli eventi del museo.

Lato Utente Non Registrato

RF27 – L'utente non registrato deve poter visualizzare gli eventi del museo.

Lato Guida

RF28 – La guida deve poter aggiungere un evento.

RF29 – La guida deve poter modificare un evento.

RF30 – La guida deve poter rimuovere un evento.

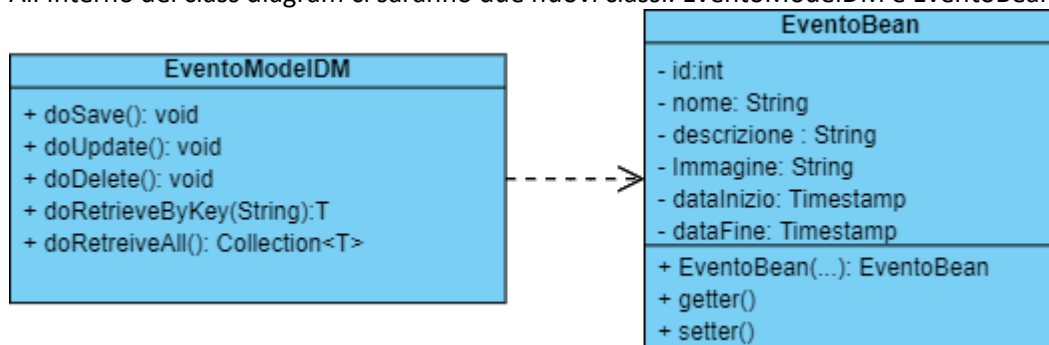
RF31 – La guida deve poter visualizzare gli eventi.

Lato Biglietteria

RF32 – La biglietteria deve poter visualizzare gli eventi del museo

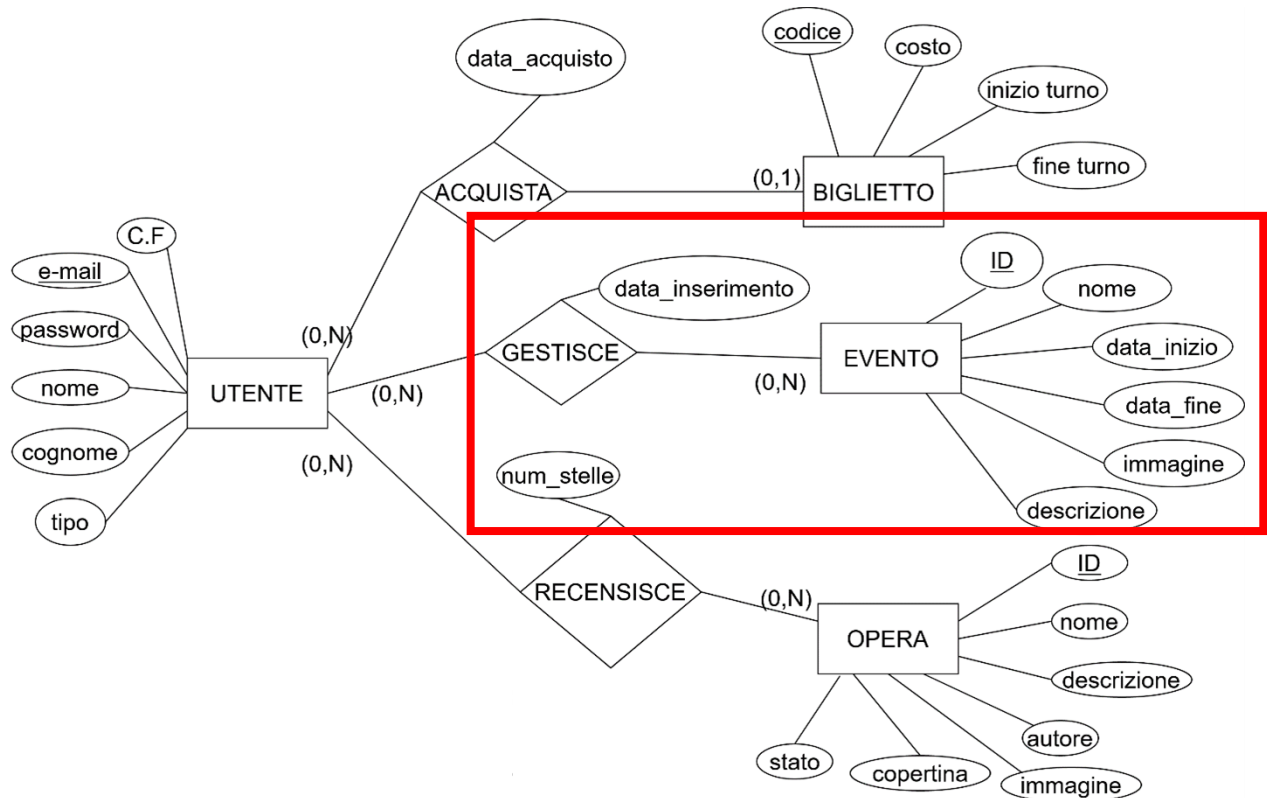
3.3 Class Diagram

All'interno del class diagram ci saranno due nuove classi: EventoModelDM e EventoBean.



3.4 Database

Lo schema relazionale viene modificato con l'inserimento dell'entità 'Evento' con i suoi attributi e la relazione che la unisce con l'entità 'Utente'



3.5 Grafo delle dipendenze

Generiamo il grado delle dipendenze del sistema, in modo da avere una visione chiara e completa del sistema nel suo insieme, per poi andare ad analizzare la change request e vedere su quali parti del codice va ad impattare.

L'abbreviazione RF, si riferisce ai requisiti funzionali.

L'abbreviazione C si riferisce a Codice.

- C5 MostraEventiServlet.java
- C6 EventoBean.java
- C7 EventoModelDM.java
- C8 Eventi.jsp
- C9 NavigationBar.jsp
- C10 InserisciEventoServlet.java
- C11 ModificaEventoServlet.java
- C12 ModificaEvento.jsp
- C13 EliminaEventoServlet.java
- C14 Eventi.js
- C15 MostraEventiTC
- C16 AggiungiEventoTC
- C17 ModificaEventoTC
- C18 EliminaEventoTC

OUT(RF26) = {C5, C6, C7, C8, C9, C14, C15}

OUT(RF27) = {C5, C6, C7, C8, C9, C14, C15}

OUT(RF28) = {C10, C7, C8, C14, C16}

OUT(RF29) = {C11, C7, C8, C12, C14, C17}

OUT(RF30) = {C13, C7, C8, C14, C18}

OUT(RF31) = {C5, C6, C7, C8, C9, C14, C15}
OUT(RF32) = {C5, C6, C7, C8, C9, C14, C15}

OUT(C5) = {C6, C7, C8}
OUT(C6) = {}
OUT(C7) = {C6}
OUT(C8) = {C6, C5, C10, C13, C14}
OUT(C9) = {C8}
OUT(C10) = {C6, C7, C8}
OUT(C11) = {C6, C7, C8}
OUT(C12) = {C6, C11, C14}
OUT(C13) = {C6, C7, C8}
OUT(C14) = {C5, C8, C12}

3.6 Starting Impact Set

In questo paragrafo viene riportato lo Starting Impact Set, che rappresenta l'insieme delle componenti che verranno direttamente impattate da queste aggiunte. Nella tabella seguente verranno indicate le componenti presenti all'interno del nostro Starting Impact Set; l'impatto della modifica verrà valutato nel seguente modo:

- FORTE: nel caso in cui sia necessario riscrivere completamente l'artefatto;
- MEDIO: nel caso in cui sia necessario una rivisitazione e modifica di un artefatto;
- DEBOLE: nel caso in cui siano necessarie solo modifiche marginali;
- NULLO: nel caso in cui non siano presenti modifiche.

Artefatto	Impatto	Descrizione
Requisiti Funzionali	Medio	Verranno aggiunti nuovi requisiti per tutti gli attori del sistema.
DB	Medio	Aggiunta della nuova entità 'Eventi'.
Codice	Medio	Verrà aggiunta la nuova funzionalità 'Eventi' con tutte le operazioni CRUD.
Testing	Medio	Nuove classi di test per le operazioni CRUD.

3.7 Candidate Impact Set

In questa sezione verranno indicati, per ogni requisito, eventuali impatti diretti e indiretti sulle altre componenti del sistema.

3.7.1 Aggiunta del requisito 'Visualizza Eventi'

Tale requisito provoca la creazione di nuove classi:

CLASSE	IMPATTO
Control.evento.MostraEventiServlet.java	INDIRETTO
Model.evento.EventoBean.java	DIRETTO
Model.evento.EventoModelDM.java	DIRETTO
Model.evento.EventoModel.java	DIRETTO
webContent.Eventi.jsp	DIRETTO
webContent.NavigationalBar.jsp	INDIRETTO
Eventi.js	DIRETTO
Test.MostraEventiTC	INDIRETTO

3.7.2 Aggiunta del requisito 'Crea Evento'

Tale requisito provoca la creazione di una nuova classe, 'InserisciEventoServlet.java' e la modifica di altre:

CLASSE	IMPATTO
--------	---------

Control.evento.InserisciEventoServlet.java	INDIRETTO
Model.evento.EventoModelDM.java	DIRETTO
Model.evento.EventoModel.java	DIRETTO
webContent.Eventi.jsp	DIRETTO
Eventi.js	DIRETTO
Test.AggiungiEventoTC	INDIRETTO

3.7.3 Aggiunta del requisito 'Modifica Evento'

Tale requisito crea nuove classi: 'ModificaEventiPageServlet.java', 'ModificaEventiServlet.java' e 'ModificaEventi.jsp'. Inoltre, andrà a modificare le classi 'EventiModelDM.java' e 'Eventi.jsp'

CLASSE	IMPATTO
Control.evento.ModificaEventoServlet.java	INDIRETTO
Model.evento.EventoModelDM.java	DIRETTO
Model.evento.EventoModel.java	DIRETTO
webContent.Eventi.jsp	DIRETTO
webContent.ModificaEvento.jsp	DIRETTO
Eventi.js	DIRETTO
Test.ModificaEventoTC	INDIRETTO

3.7.4 Aggiunta del requisito 'Elimina Evento'

Tale requisito provoca la creazione della nuova classe 'EliminaEventoServlet.java' e la modifica di altre: 'EventoModelDM.java', 'Eventi.jsp' e 'Eventi.js'

CLASSE	IMPATTO
Control.evento.EliminaEventoServlet.java	INDIRETTO
Model.evento.EventoModelDM.java	DIRETTO
Model.evento.EventoModel.java	DIRETTO
webContent.Eventi.jsp	DIRETTO
Eventi.js	DIRETTO
Test.EliminaEventoTC	INDIRETTO

3.7.5 CIS

Di conseguenza, il nostro CIS ha cardinalità 15 ed è il seguente:

webContent.NavigationalBar.jsp
Control.evento.MostraEventiServlet.java
Model.evento.EventoBean.java
Model.evento.EventoModelDM.java
Model.evento.EventoModel.java
webContent.Eventi.jsp
Control.evento.InserisciEventoServlet.java
Control.evento.ModificaEventoServlet.java
webContent.ModificaEvento.jsp
Control.evento.EliminaEventoServlet.java
Eventi.js
Test.MostraEventiTC
Test.AggiungiEventoTC
Test.ModificaEventoTC
Test.EliminaEventoTC

3.8 Actual Impact Set

Di seguito sono elencate le componenti realmente modificate: tali componenti faranno parte del nostro Actual Impact Set (AIS).

3.8.1 *Aggiunta del requisito 'Visualizza Eventi'*

3.8.1.1 **Control.evento.MostraEventiServlet.java**

È stata creata la servlet 'MostraEventiServlet' per restituire gli eventi disponibili.

3.8.1.2 **Model.evento.EventoBean.java**

È stata creata la classe 'EventoBean', la quale contiene il costruttore dell'evento e i metodi getter e setter.

3.8.1.3 **Model.evento.EventoModelDM.java**

È stata creata la classe 'EventoModelDM' con il metodo 'doRetrieveAll'.

3.8.1.4 **Model.evento.EventoModel.java**

Interfaccia per il metodo 'doRetrieve' che la classe EventoModelDM deve implementare.

3.8.1.5 **webContent.Eventi.jsp**

è stata creata la pagina 'Eventi' che mostra gli eventi disponibili.

3.8.1.6 **webContent.NavigationalBar.jsp**

La pagina 'NavigatiBar' è stata modificata inserendo nel menù il riferimento alla pagina 'Evento'.

3.8.1.7 **Test.MostraEventiTC**

Classe di test per la visualizzazione degli eventi.

3.8.2 *Aggiunta del requisito 'Crea Evento'*

3.8.2.1 **Control.evento.InserisciEventoServlet.java**

È stata creata la servlet 'InserisciEventoServlet' per effettuare l'inserimento di un nuovo evento.

3.8.2.2 **Model.evento.EventoModelDM.java**

È stata modificata la classe 'EventoModelDM' per inserire il metodo 'doSave'.

3.8.2.3 **Model.evento.EventoModel**

Interfaccia per il metodo 'doSave' che la classe EventoModelDM deve implementare.

3.8.2.4 **webContent.Eventi.jsp**

è stata modificata la pagina 'Eventi.jsp' inserendo il form per l'inserimento dei dati relativi all'evento che si vuole aggiungere.

3.8.2.5 **Test.AggiungiEventoTC**

Classe di test per l'aggiunta di un evento

3.8.3 *Aggiunte del requisito 'Modifica Evento'*

3.8.3.1 **Control.evento.ModificaEventoServlet.java**

È stata creata la servlet 'ModificaEventoServlet' per effettuare la modifica di un evento.

3.8.3.2 **Model.evento.EventoModelDM.java**

È stata modificata la classe 'EventoModelDM' per inserire il metodo 'doUpdate'.

3.8.3.3 **Model.evento.EventoModel**

Interfaccia per il metodo 'doUpdate' che la classe EventoModelDM deve implementare.

3.8.3.4 **webContent.Eventi.jsp**

è stata modificata la pagina 'Eventi.jsp' inserendo un button per reindirizzare l'attore guida alla pagina 'ModificaEvento'.

3.8.3.5 **webContent.ModificaEvento.jsp**

è stata creata la pagina 'ModificaEvento.jsp' che contiene il form per inserire i dati dell'evento che si vuole modificare.

3.8.3.6 **Test.ModificaEventoTC**

Classe di test per la modifica di un evento.

3.8.3.7 **Control.ModificaEventoPageServlet**

Classe che riceve in input l'id dell'evento da modificare.

3.8.4 Aggiunta del requisito 'Elimina Evento'

3.8.4.1 Control.evento.EliminaEventoServlet.java

È stata creata la servlet 'EliminaEventoServlet' per effettuare l'eliminazione di un evento.

3.8.4.2 Model.evento.EventoModelDM.java

È stata modificata la classe 'EventoModelDM' per inserire il metodo 'doDelete'.

3.8.4.3 Model.evento.EventoModel

Interfaccia per il metodo 'doDelete' che la classe EventoModelDM deve implementare.

3.8.4.4 webContent.Eventi.jsp

è stata modificata la pagina 'Eventi.jsp' inserendo un button per eliminare l'evento.

3.8.4.5 Test.EliminaEventoTC

Classe di test per la cancellazione di un evento.

3.8.5 AIS

Di conseguenza, il nostro AIS ha cardinalità 15 ed è il seguente:

webContent.NavigationalBar.jsp
Control.evento.MostraEventiServlet.java
Model.evento.EventoBean.java
Model.evento.EventoModelDM.java
Model.evento.EventoModel
webContent.Eventi.jsp
Control.evento.InserisciEventoServlet.java
Control.evento.ModificaEventoServlet.java
Control.evento.ModificaEventoPageServlet.java
webContent.ModificaEvento.jsp
Control.evento.EliminaEventoServlet.java
Test.MostraEventiTC
Test.AggiungiEventoTC
Test.ModificaEventoTC
Test.EliminaEventoTC

3.9 Discovered Impact Set

Il Discovered Impact set ha cardinalità 1 ed è il seguente:

control.ModificaEventoPageServlet.java
--

3.10 False Positive Impact Set

Il False Positive Impact set ha cardinalità 1 ed è il seguente:

Eventi.js

3.11 Conclusioni

Vengono mostrate le cardinalità dei nostri insiemi:

- Starting Impact Set $|SIS| = 4$
- Candidate Impact Set $|CIS| = 15$
- Actual Impact Set $|AIS| = 15$
- Discovered Impact Set $|DIS| = 1$
- False Positive Impact Set $|FPIS| = 1$

Per valutare l'accuratezza del processo di Impact Analysis sono state calcolate le seguenti metriche.

3.11.1 Recall

$Recall = (CIS \cap AIS) / AIS = (15 \cap 15) / 15 = 100\%$

3.11.2 Precision

$Precision = (CIS \cap AIS) / CIS = (15 \cap 15) / 15 = 100\%$

3.11.3 Adequacy

3.11.3.1 Inclusiveness

AIS è incluso nel CIS dunque Inclusiveness=1;

3.11.4 Effectiveness

3.11.4.1 Ripple-sensitivity

$Amplification = |IISO| / |DISO| = 9/6 = 1,5$

3.11.4.2 Adherence

$A-Ratio = |AIS| / |CIS| = 15/15 = 1$

4 Impact Analysis CR_3

4.1 Descrizione

La change request CR_3 comporta modifiche di design, infatti l'obiettivo è quello di aumentare l'usabilità del sistema attraverso la ridefinizione delle pagine jsp in termini di UI/UX

Di seguito sono riportate le pagine che subiranno delle modifiche grafiche:

1. webContent.Index.jsp
2. webContent.Tour.jsp
3. webContent.UserData.jsp
4. webContent.History.jsp
5. webContent.Opere.jsp
6. webContent.Login.jsp
7. webContent.Registrazione.jsp
8. webContent.OpereTour.jsp
9. webContent.NavigationalBar.jsp
10. AddChangeRemoveOpere.js
11. Buytjt.js
12. codeTour.js
13. historyBiglietteria.js
14. Login.js
15. opereTour.js
16. Registrazione.js
17. HomePage.js
18. Validation.js
19. ValidationBiglietto.js
20. PageBiglietto.js

4.2 Starting Impact Set

In questo paragrafo viene riportato lo Starting Impact Set, che rappresenta l'insieme delle componenti che verranno direttamente impattate da queste modifiche. Nella tabella seguente verranno indicate le componenti presenti all'interno del nostro Starting Impact Set; l'impatto della modifica verrà valutato nel seguente modo:

- FORTE: nel caso in cui sia necessario riscrivere completamente l'artefatto;
- MEDIO: nel caso in cui sia necessario una rivisitazione e modifica di un artefatto;
- DEBOLE: nel caso in cui siano necessarie solo modifiche marginali;
- NULLO: nel caso in cui non siano presenti modifiche.

Artefatto	Impatto	Descrizione
Implementazione	Medio	Verranno modificare tutte le interfacce del sistema

4.3 Candidate Impact Set

In questa sezione verranno indicati, per ogni requisito, eventuali impatti diretti e indiretti sulle altre componenti del sistema.

4.3.1 Modifica UI/UX

Per garantire una maggiore usabilità del sistema verranno modificate le pagine riportate nella tabella seguente.

CLASSE	IMPATTO
webContent.Index.jsp	DIRETTO
webContent.Tour.jsp	DIRETTO
webContent.UserData.jsp	DIRETTO
webContent.History.jsp	DIRETTO
webContent.Opere.jsp	DIRETTO
webContent.Login.jsp	DIRETTO
webContent.Registrazione.jsp	DIRETTO
webContent.OpereTour.jsp	DIRETTO
webContent.NavigationalBar.jsp	DIRETTO
AddChangeRemoveOpere.js	INDIRETTO
Buytjt.js	INDIRETTO
codeTour.js	INDIRETTO
historyBiglietteria.js	INDIRETTO
Login.js	INDIRETTO
opereTour.js	INDIRETTO
Registrazione.js	INDIRETTO
HomePage.js	INDIRETTO
Validation.js	INDIRETTO
ValidationBiglietto.js	INDIRETTO
PageBiglietto.js	INDIRETTO

4.3.2 CIS

Di conseguenza, il nostro CIS ha cardinalità 20 ed è il seguente:

webContent.Index.jsp
webContent.Tour.jsp
webContent.UserData.jsp
webContent.History.jsp
webContent.Opere.jsp
webContent.Login.jsp
webContent.Registrazione.jsp
webContent.OpereTour.jsp
webContent.NavigationalBar.jsp
AddChangeRemoveOpere.js
Buytjt.js
codeTour.js
historyBiglietteria.js
Login.js
opereTour.js
Registrazione.js
HomePage.js
Validation.js
ValidationBiglietto.js

PageBiglietto.js

4.4 Actual Impact Set

Di seguito sono elencate le componenti realmente modificate: tali componenti faranno parte del nostro Actual Impact Set (AIS).

4.4.1 Modifica UI/UX

Per garantire una maggiore usabilità del sistema verranno modificate le pagine riportate nella tabella seguente.

CLASSE
webContent.Index.jsp
webContent.Tour.jsp
webContent.UserData.jsp
webContent.History.jsp
webContent.Opere.jsp
webContent.Login.jsp
webContent.OpereTour.jsp
webContent.NavigationalBar.jsp
AddChangeRemoveOpere.js
Buytjt.js
codeTour.js
historyBiglietteria.js
Login.js
opereTour.js

4.4.2 AIS

Di conseguenza, il nostro AIS ha cardinalità 14 ed è il seguente:

webContent.Index.jsp
webContent.Tour.jsp
webContent.UserData.jsp
webContent.History.jsp
webContent.Opere.jsp
webContent.Login.jsp
webContent.OpereTour.jsp
webContent.NavigationalBar.jsp
AddChangeRemoveOpere.js
Buytjt.js
codeTour.js
historyBiglietteria.js
Login.js
opereTour.js

4.5 False Positive Impact Set

Il False Positive Impact Set ha cardinalità 6 ed è il seguente:

Registrazione.js
HomePage.js
Validation.js
ValidationBiglietto.js
PageBiglietto.js
WebContent.Registrazione.jsp

4.6 Conclusioni

Vengono mostrate le cardinalità dei nostri insiemi:

- Starting Impact Set $|SIS| = 1$
- Candidate Impact Set $|CIS| = 20$
- Actual Impact Set $|AIS| = 14$
- Discovered Impact Set $|DIS| = 0$
- False Positive Impact Set $|FPIS| = 6$

Per valutare l'accuratezza del processo di Impact Analysis sono state calcolate due metriche:

- Recall : percentuale degli impatti effettivi inclusi nel CIS
- Precision : percentuale di candidati che si stima possano essere impattati.

Affinché risulti efficiente tale processo, recall e precision dovrebbero avere valore maggiore del 50%.

4.6.1 Recall

$$\text{Recall} = (CIS \cap AIS) / AIS = (20 \cap 14) / 14 = 100\%$$

4.6.2 Precision

$$\text{Precision} = (CIS \cap AIS) / CIS = (20 \cap 14) / 20 = 70\%$$

4.6.3 Adequacy

4.6.3.1 Inclusiveness

AIS è incluso nel CIS dunque Inclusiveness=1

4.6.4 Effectiveness

4.6.4.1 Ripple-sensitivity

$$\text{Amplification} = |IISO| / |DISO| = 11/9 = 1,2$$

4.6.4.2 Adherence

$$\text{A-Ratio} = |AIS| / |CIS| = 14/20 = 0.7$$