



EvoCinema

“ Sistema per la gestione di un cinema ”

Versione 1.0

Documento di Analisi e Manutenzione

Coordinatori del progetto

Prof. Andrea De Lucia - Top Manager
Francesco Vicidomini - Project Manager
Ferdinando D'Avino - Project Manager

Partecipanti

Luca Strefezza	0512102474	strluca94@gmail.com
Angelo Stefano D'Auria	0512102630	angelodauria91@gmail.com
Gianluca Villani	0512102990	lucassalerno1995@gmail.com
Giuseppe D'Ambrosio	0512103472	giuseppe.dambrosio14@gmail.com
Giuseppe Apuzzo	0512103920	g.apuzzo94@gmail.com
Sara De Filippo	0512103430	s.defilippo93@gmail.com
Antonio Giulio	0512103098	antonio.giulio96@gmail.com
Michele Delli Paoli	0512103820	mikeledellipaoli@gmail.com
Giuseppe Del Gaudio	0512103690	ciaogiuseppe96@gmail.com
Pietro Dell'Isola	0512103866	dellisola.pietro@gmail.com
Emanuele Buono	0512102370	squareman93@gmail.com
Francesco De Feo	0512103274	francescodefeo94@gmail.com

Revision history

Data	Versione	Descrizione	Autori
10/01/2018	1.0	Stesura del documento	Ferdinando D'Avino, Francesco Vicidomini

1. Scopo del Documento	4
2. Panoramica del sistema attuale	4
2.1. Attori e funzionalità	4
2.2. Design implementazione e testing del sistema attuale	4
3. Analisi della modifica richiesta	4
4. Individuazione soluzione progettuale	4
4.1. Problematiche affrontate	4
4.2. Soluzione individuata	4
4.3. Soluzione alternativa	4
5. Identificazione dell' Impact Set	4
6. Studio di fattibilità	4
6.1. Identificazione, descrizione e valutazione dei costi	4
6.2. Identificazione, descrizione e valutazione dei benefici	4
6.3. Identificazione, descrizione e classificazione dei rischi	4
6.4. Risultati previsti	4

1.Scopo del Documento

In questo documento verranno descritti gli obiettivi del processo di reingegnerizzazione del gestionale proposto, allo scopo di realizzare una piattaforma più completa che prenderà il nome di **EvoCinema**.

In riferimento al documento “Change Request” verranno descritte tutte le nuove funzionalità implementate e verrà stilata una analisi dell’impatto dei cambiamenti sul sistema, evidenziando i rischi, i costi e i benefici.

Inoltre verrà presentato uno studio di fattibilità con obiettivo di analizzare le successive fasi di progettazione, implementazione e testing.

2.Panoramica del sistema attuale

Il sistema attuale è un gestionale per un cinema, esso si rivolge a due categorie di utenti quali i “**clienti**” del cinema e gli “**amministratori**” dello stesso.

Ogni cliente ha la possibilità di visualizzare la programmazione proposta dalla struttura, divisa in sala, orari ed eventualmente prenotare o acquistare biglietti.

L’amministratore può invece, oltre che visualizzare la programmazione, gestirla per intero aggiungendo o rimuovendo spettacoli, sale e sconti, visualizzare le informazioni relative al rendimento della struttura in base a diversi filtri.

Allo stato attuale il sistema è rivolto a una cerchia ristretta di utenti, in quanto per usufruire dei servizi esposti è necessario configurare la macchina dell’utente per permettere l’installazione.

2.1. Attori e funzionalità

Il sistema attuale prevede i seguenti attori con relative funzionalità:

- Amministratore
 - Login
 - Inserimento \ Modifica \ Cancellazione Spettacoli
 - Inserimento \ Modifica \ Cancellazione Politiche di Sconto
 - Inserimento \ Modifica \ Cancellazione Sale
 - Abilitare \ Disabilitare Sconti

- Visualizzazione degli Incassi
- Riservare o rendere indisponibili posti in sala
- Utente
 - Login
 - Registrazione
 - Acquisto / Prenotazione posti in sala
 - Visualizzazione programmazione

2.2. Design implementazione e testing del sistema attuale

Il sistema attuale è stato concepito come un' architettura single layer che sfrutta un sistema di storage locale, nel quale la logica di business è implementata sia in classi che gestiscono le interfacce grafiche che in altre riservate ad essa.

Il sistema è stato sviluppato interamente in JavaSE, più precisamente:

- Java Util per le strutture dati
- Java Swing per le interfacce grafiche
- Storage di oggetti serializzati su file

Implementazione

L'implementazione del sistema non segue alcun pattern, in particolare è presente una pessima divisione delle responsabilità tra le varie classi. Pertanto il lavoro di manutenzione punterà oltre che al cambiamento dell'architettura anche ad una riorganizzazione delle classi.

Testing

Vista la mancanza di documentazione e la mancanza di classi di test riteniamo che il sistema non sia stato testato. Si prevede quindi l'implementazione di componenti a questo scopo.

3. Analisi della modifica richiesta

La change request effettuata prevede di migliorare i requisiti funzionali del sistema attuale ed adattarli ad una piattaforma web. In più è prevista l'aggiunta di funzionalità del tutto nuove come ad esempio una libreria per film ed una sezione nella quale sono riassunte in grafici varie analytics utili al gestore.

4. Individuazione soluzione progettuale

4.1. Problematiche affrontate

Issue 1: Quali funzionalità conservare?

Proposal 1.1

Conservare tutte le vecchie funzionalità.

Proposal 1.2

Conservare solo le funzionalità di iscrizione al sistema, implementando dal principio le altre.

Criterion 1.1

Nonostante i difetti il sistema risulta funzionante, conservare almeno a livello logico le funzionalità ci garantisce che le componenti siano predisposte all'interazione.

Criterion 1.2

Riscrivere totalmente le funzionalità ci permetterebbe di analizzare il sistema dal principio già consapevoli delle debolezze e dei problemi individuati nel sistema proposto.

Argument 1.1

Riscrivere tutte le funzionalità ha costi troppo elevati e richiede troppo tempo.

Argument 1.2

Conservare le vecchie funzionalità potrebbe rendere complessa l'integrazione dei nuovi requisiti funzionali.

Resolution 1

Analizzando i punti Argument 1.1 e Argument 1.2 si è deciso di risolvere l'issue 1 applicando la Proposal 1.1.

Issue 2: Conservare e/o aggiungere nuovi attori?

Proposal 2.1

Mantenere tutti e soli gli attori attuali.

Proposal 2.2

Mantenere tutti gli attori attuali aggiungendo l'attore **Operatore**.

Criterion 2.1

Gli attori attuali sono sufficienti per la gestione del sistema.

Criterion 2.2

Il ruolo dell'attore **Operatore** è necessario per una nuova e migliorata divisione delle responsabilità.

Argument 2.1

Il ruolo dell'**Operatore** può essere svolto dal **Gestore** risulta quindi inutile e dispendioso riorganizzare la struttura per inserire adeguatamente questo attore. Inoltre tale approccio ci impedisce gran parte del riutilizzo delle componenti.

Argument 2.2

L'attuale struttura gerarchica del sistema limita il sistema alla sola gestione di piccole realtà imprenditoriali, in quanto il **Gestore** sarebbe il delegato anche per le operazioni di cassa.

Resolution 2

Analizzando i punti Argument 2.1 e Argument 2.2 si è deciso di risolvere l'issue 2 applicando la Proposal 2.2.

Issue 3: Gestire il pagamento con credito virtuale?

Proposal 3.1

Associare ad ogni account un credito, ricaricabile nella struttura del Cinema.

Proposal 3.2

Permettere un sistema di pagamento che sfrutta E-Commerce.

Criterion 3.1

Ottenere una gestione semplificata del pagamento, ampliando l'utenza anche ad una fascia più giovane che presumibilmente non dispone di metodi di pagamento classici.

Criterion 3.2

L'utenza avrà già confidenza con il mondo dei pagamenti online, facendo risultare il tutto più user-friendly.

Argument 3.1

L'integrazione di un sistema di E-Commerce nel sistema risulterà eccessivamente complessa.

Argument 3.2

Il metodo di pagamento tramite credito virtuale è potenzialmente meno sicuro.

Resolution 3

Analizzando i punti Argument 3.1 e Argument 3.2 si è deciso di risolvere l'issue 3 applicando la Proposal 3.1.

Issue 4: Integrare nel sistema il sottosistema Libreria?

Proposal 4.1

Inserire il sottosistema Libreria dando agli utenti la possibilità di valutare le diverse pellicole presenti

Proposal 4.2

Non inserire il sottosistema Libreria

Criterion 4.1

Aumentare la possibilità di interazione degli utenti con il sistema, agevolando contemporaneamente il lavoro del Gestore che potrà sfruttare i feedback rilasciati per fornire il giusto tipo di intrattenimento.

Criterion 4.2

Conservare l'attuale funzionamento del sistema.

Argument 4.1

L'utilità del sistema resterebbe limitata alla prenotazione/acquisto di posti in sala, limitando notevolmente l'affluenza alla piattaforma.

Argument 4.2

L'aggiunta di questa nuova funzionalità risulterebbe lunga e complessa, inoltre non è garantito il successo della manovra per aumentare l'affluenza.

Resolution 4

Analizzando i punti Argument 4.1 e Argument 4.2 si è deciso di risolvere l'issue 4 applicando la Proposal 4.1.

Issue 5: Quando e come aggiungere un film in libreria?

Proposal 5.1

La gestione della libreria è affidata completamente al Gestore che decide autonomamente quando e cosa aggiungere.

Proposal 5.2

Semi-Automatizzare il popolamento della libreria, facendo in modo che ogni film che esce dalla programmazione sarà reperibile immediatamente in essa. Fornire anche la possibilità di aggiungere pellicole autonomamente.

Criterion 5.1

Affidare le responsabilità del sottosistema ad un singolo ruolo garantisce che il risultato sia quello desiderato.

Criterion 5.2

Il carico di lavoro gravante sul Gestore risulta notevolmente inferiore senza perdere la ricchezza dei contenuti della libreria.

Argument 5.1

La propagazione degli errori è sempre da evitare, semi-automatizzare il processo di popolamento della Libreria sfruttando la Programmazione del Cinema, conserverebbe eventuali errori di quest'ultima sezione.

Argument 5.2

Il carico di lavoro risulterebbe notevolmente superiore, aumentando la possibilità di errore da parte del Gestore.

Resolution 5

Analizzando i punti Argument 5.1 e Argument 5.2 si è deciso di risolvere l'issue 5 applicando la Proposal 5.2.

Issue 6: Inserire un sottosistema per la visualizzazione degli Analytics?

Proposal 6.1
Aggiungere la gestione degli Analytics più completa rispetto a quella del sistema attuale.
Proposal 6.2
Conservare l'attuale gestione degli Analytics.

Criterion 6.1
Si mette a disposizione del Gestore uno strumento che permette di analizzare, oltre che agli incassi, diverse nuove informazioni utili riassunte in grafici.
Criterion 6.2
Attualmente il sistema fornisce semplicemente le informazioni fondamentali relative agli incassi.

Argument 6.1
La semplice gestione dei dati attuale non fornisce una panoramica completa relativa all'andamento della struttura nel tempo. Inoltre la sola gestione dell'incasso risulta limitativa in un sistema Web-Oriented che di norma si basa sullo studio di molteplici fattori.
Argument 6.2
L'integrazione di questo sottosistema risulterebbe complicata in quanto il team di sviluppo dovrà essere istruito riguardo libreria grafiche e di rendering.

Resolution
Analizzando i punti Argument 6.1 e Argument 6.2 si è deciso di risolvere l'issue 6 applicando la Proposal 6.1.

Issue 7: Aggiungere nuovi Gestori?

Proposal 7.1

Impedire che coesistano più Gestori.

Proposal 7.2

Consentire ad un Gestore di aggiungerne un altro.

Criterion 7.1

Lasciare il controllo dell'intero sistema nelle mani di un singolo Gestore impedisce problemi nella divisioni delle competenze di questo ruolo.

Criterion 7.2

Più Gestori che lavorano contemporaneamente possono godere di un carico di lavoro inferiore compiendo comunque tutti i doveri del ruolo.

Argument 7.1

Un controllo da parte di un singolo individuo risulterebbe ingestibile in caso di realtà più grandi.

Argument 7.2

Molteplici Gestori danno luogo alla necessità di una fase aggiuntiva di organizzazione del lavoro.

Resolution 7

Analizzando i punti Argument 7.1 e Argument 7.2 si è deciso di risolvere l'issue 7 applicando la Proposal 7.2.

Issue 8: Dare all'utente la possibilità di cancellare il proprio account?

Proposal 8.1

L'Utente può cancellare il proprio account richiedendo la cancellazione ad un Operatore.

Proposal 8.2

L'Utente può cancellare il proprio account autonomamente.

Criterion 8.1

Cancellare il proprio account tramite una richiesta al Gestore permette a quest'ultimo di monitorare gli utenti che lasciano il sistema, scoprendo magari il motivo della scelta.

Criterion 8.2

L'eliminazione autonoma velocizzerebbe il processo, diminuendo il lavoro dell'Operatore.

Argument 8.1

La gestione richiede troppo lavoro per gli Operatori.

Argument 8.2

L'eliminazione automatica non permettere di migliorarsi scoprendo il motivo della cancellazione.

Resolution

Analizzando i punti Argument .1 e Argument .2 si è deciso di risolvere l'issue applicando la Proposal .2.

Issue 9: Quali linguaggi utilizzare per l'evoluzione del sistema?

Proposal 9.1

Utilizzare HTML5, CSS3, JavaScript, JQuery per il lato Client e JavaEE per il Server.

Proposal 9.2

Utilizzare HTML5, CSS3, JavaScript, JQuery per il lato Client ed utilizzare solo JSP e Servlet per il Server.

Criterion 9.1

La fase di implementazione risulta notevolmente più rapida per la potenza dell'ambiente.

Criterion 9.2

Il team è già istruito riguardo l'utilizzo di questa tecnologia.

Argument 9.1

Far utilizzare al team di sviluppo solo JSP e Servlet si impedirebbe al team di utilizzare altri moduli di JavaEE che potrebbero semplificare e velocizzare lo sviluppo.

Argument 9.2

Il tempo che si recupererebbe utilizzando l'ambiente JavaEE andrebbe perso per la formazione del team di sviluppo.

Resolution

Analizzando i punti Argument 9.1 e Argument 9.2 si è deciso di risolvere l'issue 9 applicando la Proposal 9.2.

Issue :

Proposal .1
Proposal .2

Criterion .1
Criterion .2

Argument .1
Argument .2

Resolution
Analizzando i punti Argument .1 e Argument .2 si è deciso di risolvere l'issue applicando la Proposal .2.

4.2. Soluzione individuata

La soluzione individuata consiste nell'unione di tutte le "Resolution" individuate nel paragrafo 4.1. La soluzione avrà le seguenti caratteristiche:

1. Conservare tutte le vecchie funzionalità.
2. Mantenere tutti gli attori attuali aggiungendo l'attore **Operatore**.
3. Associare ad ogni account un credito, ricaricabile nella struttura del Cinema.
4. Inserire il sottosistema Libreria dando agli utenti la possibilità di valutare le diverse pellicole presenti
5. Semi-Automatizzare il popolamento della libreria, facendo in modo che ogni film che esce dalla programmazione sarà reperibile immediatamente in essa. Fornire anche la possibilità di aggiungere pellicole autonomamente.
6. Aggiungere la gestione degli Analytics più completa rispetto a quella del sistema attuale.
7. Più Gestori che lavorano contemporaneamente possono godere di un carico di lavoro inferiore compiendo comunque tutti i doveri del ruolo.
8. L'Utente può cancellare il proprio account autonomamente.
9. Utilizzare HTML5, CSS3, JavaScript, JQuery per il lato Client ed utilizzare solo JSP e Servlet per il Server.

4.3. Soluzione alternativa

Le soluzioni alternative individuate sono rappresentate da tutte le 'Proposal' e gli 'Argument' affrontati nel paragrafo 4.1 che non sono stati selezionati per la soluzione adottata.

5. Identificazione dell' Impact Set

La soluzione individuata comporta la modifica di diverse componenti(classi) del sistema attuale.

In seguito verrà descritto il Candidate Impact Set che conterrà l'insieme delle componenti modificate durante la fase di manutenzione, per ogni componente modificata verrà data una descrizione della modifica e il tipo di impatto che questa ha avuto sulla componente.

L'interfaccia grafica del sistema verrà completamente rivoluzionata ma questa dovrà supportare le funzionalità attuale e permettere di aggiungere di nuove. A tale scopo questi cambiamenti riguardanti i requisiti funzionali verranno specificati nel documento di analisi dei requisiti.

Per quanto riguarda il documento di System Design ci saranno cambiamenti che riguardano l'architettura e il mapping Hardware/Software.

Nella tabella seguente verranno indicate le componenti impattate, l'impatto della modifica verrà valutato su tre livelli:

- **FORTE:** nel caso in cui sia necessario riscrivere completamente l'artefatto.
- **MEDIO:** nel caso in cui risulta necessario apportare forti modifiche all'artefatto.
- **DEBOLE:** nel caso in cui siano necessarie solo modifiche marginali.

<i>Artefatto</i>	<i>Impatto</i>	<i>Descrizione</i>
Identificazione Attori	MEDIA	Come precedentemente descritto nel capitolo 4 sarà necessario l'aggiunta di un nuovo attore: l' Operatore .
Requisiti Funzionali	FORTE	Nonostante si sia scelto di conservare tutti i vecchi requisiti, ne sono stati aggiunti molti nuovi. Inoltre i requisiti conservati subiranno comunque importanti modifiche.
Casi d'uso	FORTE	Il livello d'impatto risulta simile rispetto al caso dei Requisiti Funzionali.
Requisiti non funzionali	FORTE	La reingegnerizzazione verso un sistema Web-Oriented ci mette di fronte a requisiti non funzionali del tutto nuovi.

Volendo scendere nel dettaglio, risulta impossibile trovare componenti candidate, in quanto, il sistema attuale non presenta un Object Design Document da analizzare.

Concentriamo lo studio quindi sulle diverse classi del sistema attuale che subiranno modifiche:

Package “grafica”: Contenente tutte le classi che gestiscono l’interfaccia grafica del nostro sistema Stand Alone. Considerando che si sta migrando l’applicazione verso il Web tutte le classi verranno rimpiazzate.

Artefatto	Impatto	Descrizione
MyLog.java	FORTE	La classe verrà sostituita
Home.java	FORTE	La classe verrà sostituita
HomeAdmin.java	FORTE	La classe verrà sostituita
MyAddSpettacolo.java	FORTE	La classe verrà sostituita
MyReplaceSpettacolo.java	FORTE	La classe verrà sostituita
MyPosti.java	FORTE	La classe verrà sostituita
MyPostiCliente.java	FORTE	La classe verrà sostituita
FrameError.java	FORTE	La classe verrà sostituita
MyRegistrazione.java	FORTE	La classe verrà sostituita
MySconti.java	FORTE	La classe verrà sostituita

Package “cinema”: Contenente tutte le classi che gestiscono la logica di business, e le astrazioni degli oggetti. L’impatto sarà forte sulle componenti che contengono logica di business, per via della migrazione verso il web, meno sulle classi model.

Artefatto	Impatto	Descrizione
Cinema.java	FORTE	La classe verrà sostituita
CinemaGestore.java	FORTE	La classe verrà sostituita
Cliente.java	MEDIO	La classe verrà decorata con altri attributi utili alla gestione dei nuovi

		requisiti funzionali.
Film.java	MEDIO	L'impatto sarà il medesimo avuto per Cliente.java
Gestore.java	FORTE	La classe verrà sostituita
Prenotazioni.java	FORTE	La classe verrà sostituita per risultare compatibile con la nuova gestione degli acquisti.
Scontabili.java	FORTE	L'interfaccia verrà eliminata.
Sconto.java	MEDIO	La classe verrà arricchita con nuovi attributi e funzionalità.
Spettacolo.java	FORTE	La classe verrà sostituita

6. Studio di fattibilità

6.1. Identificazione, descrizione e valutazione dei costi

Identificazione	Valutazione	Motivazione
Modifica dei requisiti funzionali	MEDIO	Il sistema attuale risulta abbastanza povero, più che di manutenzione si parla di cambiamento per adattare gli stessi al web.
Aggiunta di nuovi requisiti funzionali	FORTE	Le funzionalità aggiunte risultano molto più numerose rispetto a ciò che si riesce a conservare.
Aggiunta Libreria	MEDIO	Va completamente aggiunta ma la difficoltà del task non risulta eccessiva.
Aggiunta Analytics	FORTE	Oltre che dover essere completamente aggiunta, richiede skills tecniche che il team ancora non possiede.
Aggiunto attore Operatore	MEDIA	Nonostante l'attore verrà aggiunto in questa nuova versione, molte delle sue funzionalità vengono "ereditate" dal Gestore della piattaforma attuale.
Garantire la Struttura Three-Tier	FORTE	Il sistema attuale non presenta questo tipo di architettura, quindi il sistema

		andrà ricostruito rispettando i vincoli della stessa.
Implementazione con linguaggi HTML5, CSS3, JSP, Servlet, JQuery, JavaScript	DEBOLE	La quasi totalità del team ha familiarità con le tecnologie citate.
Compatibilità Cross-Platform	DEBOLE	Fornita automaticamente dal framework Bootstrap 3.
Testing funzionale	FORTE	Le componenti del sistema attuale non sono state testate, in più il cambio di piattaforma non ci avrebbe permesso di riciclare per intero i casi di test.

6.2. Identificazione, descrizione e valutazione dei benefici

Identificazione	Valutazione	Motivazioni
Accesso Multiplatforma	FORTE	L'accesso multiplatforma è garantito dal fatto che le piattaforme web sono accessibili comodamente tramite Browser.
Estensione del dominio applicativo	MEDIA	Il dominio applicativo è pressoché inalterato, il sistema resta un gestionale per un cinema che offre però molti più servizi.
Aggiunta Libreria	FORTE	Migliorato il livello di interazione del cliente con la piattaforma, fornendo informazioni utili ai Gestori del cinema.
Aggiunta Analytics	FORTE	Migliorata l'esperienza del Gestore che ora ha a disposizione svariate informazioni sull'andamento del cinema.
Scontabili.java	FORTE	L'interfaccia verrà eliminata.
Sconto.java	MEDIO	La classe verrà arricchita con nuovi attributi e funzionalità.
Spettacolo.java	FORTE	La classe verrà sostituita

6.3. Identificazione, descrizione e classificazione dei rischi

<i>Identificazione</i>	<i>Probabilità</i>	<i>Motivazioni</i>
Introduzione nuovi fault	FORTE	Come già spiegato le modifiche stravolgeranno il sistema aumentando drasticamente la possibilità di nuovi errori
Malfunzionamento del Server	DEBOLE	Un malfunzionamento del Server risulta molto improbabile.
Attacchi informatici	FORTE	Il porting della piattaforma sul web ovviamente ci espone al rischio di attacchi
Carico di lavoro eccessivo per il Server	DEBOLE	Le operazioni che il Server va a compiere non risultano per nulla difficili computazionalmente
Carenza di Skill all'intero del team	FORTE	Il team conosce la maggior parte delle tecnologie che andremo ad utilizzare, la mancanza di esperienze precedenti per quanto riguarda il team-work potrebbe portare molteplici problemi anche dal punto di vista tecnico.

6.4. Risultati previsti

La realizzazione del sistema software è definita secondo la soluzione individuata nel capitolo 4, tenendo in considerazione costi, benefici e rischi riportati nel capitolo 6

