

Fase di Elaborazione

Iterazione 2

Sommario

1. Workshop dei requisiti	4
1.1. UC3: Visualizza Prenotazioni Proiezione	4
1.2. UC4: Visualizza Programmazione	5
1.3. UC7: Gestisci Clienti	5
1.4. UC15: Login	7
2. Analisi	8
2.1. Introduzione	8
2.2. Caso d'uso UC3, Modello di dominio	8
2.3. Caso d'uso UC3, Diagramma di sequenza di sistema	8
2.4. Caso d'uso UC3, Contratti delle operazioni	8
2.5. Caso d'uso UC4, Modello di dominio	9
2.6. Caso d'uso UC4, Diagramma di sequenza di sistema	9
2.7. Caso d'uso UC4, Contratti delle operazioni	9
2.8. Caso d'uso UC7, Modello di dominio	9
2.9. Caso d'uso UC7, Diagramma di sequenza di sistema	9
2.10. Caso d'uso UC7, Contratti delle operazioni	10
2.11. Caso d'uso UC15, Modello di dominio	10
2.12. Caso d'uso UC15, Diagramma di sequenza di sistema	11
2.13. Caso d'uso UC15, Contratti delle operazioni	11
3. Progettazione	12
3.1. Caso d'uso UC3, Diagrammi di interazione	12
3.1.1. getPrenotazioniProiezione(codiceProiezione:String)	12
3.1.2. getStatoSalaProiezione(codiceProiezione:String)	12
3.2. Caso d'uso UC4, Diagrammi di interazione	13
3.2.1. getProiezioniPerData(data: Date)	13
3.3. Caso d'uso UC7, Diagrammi di interazione	13
3.3.1. nuovoCliente(codiceFiscale:String, nome:String, cognome:String, indirizzo:String)	13
3.4. Caso d'uso UC15, Diagrammi di interazione	14
3.4.1. autenticaUtente(username:String, password:String)	14
3.5. Modifiche ai diagrammi di sequenza della precedente iterazione	15
3.5.1. UC11 - nuovaPrenotazione(codiceProiezione:String)	15
3.5.2. UC11 - ottieniPostiDisponibili ()	15
3.5.3. UC11 - aggiungiBiglietto(numPosto: Integer)	17
3.5.4. UC11 - confermaPrenotazione()	17
3.6. Caso d'uso di avviamento, Diagramma di interazione	18

3.7.	Design Pattern GoF applicati.....	18
3.7.1.	Protection Proxy	18
3.7.2.	Singleton	19
3.8.	Diagramma delle classi complessivo.....	19
4.	Implementazione	20
5.	Testing	20
-	GestoreUtenti	20

1. Workshop dei requisiti

Per l'iterazione corrente sono stati presi in considerazione i casi d'uso UC3, UC4, UC7, UC15.

Per ciascuno di essi segue la descrizione in formato dettagliato.

1.1. UC3: Visualizza Prenotazioni Proiezione

Nome del caso d'uso	Visualizza Prenotazioni Proiezione
Portata	Applicazione EasyCinema
Livello	Obiettivo utente
Attore primario	Titolare
Parti interessate e interessi	<ul style="list-style-type: none">- Titolare: Data una proiezione, il Titolare intende conoscere le prenotazioni ad essa effettuate e lo stato complessivo della sala (in termini di posti).
Pre-condizioni	Il Titolare è identificato e autenticato (<u>Login</u>).
Post-condizioni	Tutte le prenotazioni relative alla proiezione e le informazioni sui posti liberi e occupati sono mostrate correttamente.
Scenario principale di successo	<ol style="list-style-type: none">1. Il Titolare inserisce il codice identificativo della proiezione di interesse.2. Il Sistema mostra le prenotazioni relative alla proiezione di interesse.3. Il Titolare intende conoscere anche lo stato della sala.4. Il Sistema mostra lo stato della sala in termini di posti disponibili e posti riservati.
Estensioni	<ol style="list-style-type: none">1a. Il codice inserito non esiste:<ol style="list-style-type: none">1. Il Sistema segnala l'errore invitando il Titolare a reinserire il codice.2. Il Titolare reinserisce il codice.2a. Il Titolare annulla l'operazione.3a. Il Titolare non è interessato allo stato della sala:<ol style="list-style-type: none">1. Il Sistema non prosegue oltre.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Svariate volte nell'arco della giornata.
Varie	Andrebbe proposta al Titolare la creazione di una nuova proiezione nel caso in cui la sala fosse completamente piena?

1.2. UC4: Visualizza Programmazione

Nome del caso d'uso	Visualizza Programmazione
Portata	Applicazione EasyCinema
Livello	Obiettivo utente
Attore primario	Cliente
Parti interessate e interessi	- Cliente: il Cliente intende conoscere le proiezioni che avranno luogo in una precisa data.
Pre-condizioni	Il Cliente è identificato e autenticato (<u>Login</u>).
Post-condizioni	Tutte le proiezioni che avranno luogo in data indicata dal cliente vengono visualizzate.
Scenario principale di successo	<ol style="list-style-type: none">1. Il Cliente intende conoscere le proiezioni in programmazione.2. Il Sistema richiede la data.3. Il Cliente immette la data.4. Il Sistema restituisce la totalità delle proiezioni in programmazione per la data immessa.
Estensioni	3a. La data inserita non è valida: <ol style="list-style-type: none">1. Il Sistema segnala l'errore invitando il Cliente ad inserire una data nel formato corretto.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Molto frequente.
Varie	Internazionalizzazione del formato di data e ora?

1.3. UC7: Gestisci Clienti

Nome del caso d'uso	Gestisci Clienti
Portata	Applicazione EasyCinema
Livello	Obiettivo utente
Attore primario	Titolare
Parti interessate e interessi	- Titolare: il Titolare intende aggiungere, visualizzare, modificare ed eliminare un cliente.
Pre-condizioni	Il Titolare è identificato e autenticato (<u>Login</u>).
Post-condizioni	Consistenza dei dati.
Scenario principale di successo	<ol style="list-style-type: none">1. Il Titolare richiede di aggiungere un nuovo cliente.2. Il Sistema richiede le informazioni relative al cliente (codice fiscale, nome, cognome, indirizzo).

	<ol style="list-style-type: none"> 3. Il Titolare immette le informazioni relative al cliente. 4. Il Sistema crea il nuovo cliente e notifica al Titolare l'esito positivo dell'operazione.
Estensioni	<ol style="list-style-type: none"> 1b. Il Titolare richiede la visualizzazione di tutti i clienti esistenti filtrati opportunamente: <ol style="list-style-type: none"> 1. Il Sistema richiede l'inserimento delle informazioni sulla base delle quali effettuare il filtraggio dei risultati. 2. Il Titolare inserisce le informazioni che desidera. 3. Il Sistema mostra l'elenco dei clienti che rispettano i requisiti. 1c. Il Titolare richiede la modifica di un cliente: <ol style="list-style-type: none"> 1. Il Titolare inserisce il codice fiscale del cliente da modificare. 2. Il Sistema presenta al titolare i dati attuali. 3. Il Titolare apporta i cambiamenti necessari. 4. Il Sistema aggiorna le informazioni del cliente. 1d. Il Titolare richiede l'eliminazione di un cliente. <ol style="list-style-type: none"> 1. Il Titolare inserisce il codice fiscale del cliente da eliminare. 2. Il Sistema chiede conferma dell'operazione. 3. Il Titolare conferma. 4. Il Sistema rimuove il cliente. 4a. Esiste già un altro cliente con lo stesso codice fiscale: <ol style="list-style-type: none"> 1. Il Sistema comunica l'errore al Titolare. 2. Il Titolare tenta una nuova creazione del cliente.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Occasionale.
Varie	Ogni nuovo cliente ha un credito iniziale non nullo? Cosa succede alle prenotazioni se un cliente viene rimosso?

1.4. UC15: Login

Nome del caso d'uso	Login
Portata	Applicazione EasyCinema
Livello	Sottofunzione
Attore primario	Utente
Parti interessate e interessi	Il Sistema deve poter essere utilizzato esclusivamente da persone identificate ed autenticate correttamente. Più precisamente un Utente del Sistema può appartenere ad una delle seguenti due categorie: Titolare o Cliente.
Pre-condizioni	Il Titolare è già presente nel sistema con credenziali di default (admin, admin). Il Cliente è stato registrato nel Sistema dal Titolare al momento della sua creazione (username e password coincidono con il codice fiscale).
Post-condizioni	L'Utente ha accesso al sistema.
Scenario principale di successo	<ol style="list-style-type: none"> 1. L'Utente intende utilizzare il Sistema. 2. Il Sistema richiede l'inserimento di username e password. 3. L'Utente inserisce username e password. 4. Il Sistema autentica l'Utente.
Estensioni	<p>4a Il Sistema non trova una corrispondenza tra gli utenti presenti nel Sistema per lo username e password inseriti:</p> <ol style="list-style-type: none"> 1. Il Sistema segnala il problema e richiede l'inserimento di nuove credenziali. 2. L'Utente inserisce nuovamente username e password.
Requisiti speciali	
Elenco delle varianti tecnologiche e dei dati	
Frequenza di ripetizione	Ad ogni nuovo accesso al sistema da parte dell'Utente.
Varie	Quali sono le credenziali di default per l'amministratore? Che caratteristiche devono avere username e password?

Al termine del workshop circa il 30% (33.3%) dei requisiti (a livello utente) sono stati descritti in dettaglio.

2. Analisi

2.1. Introduzione

Per l'iterazione 2, si è scelto di attenzionare lo scenario principale di successo dei seguenti requisiti:

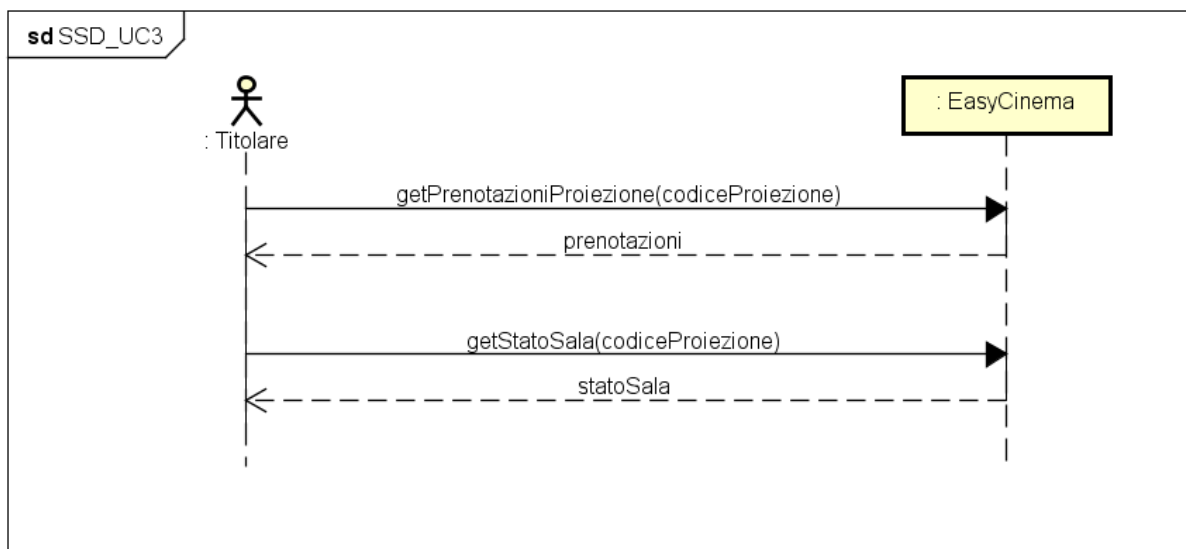
- Caso d'uso a livello utente UC3 (*Visualizza Prenotazioni Proiezione*).
- Caso d'uso a livello utente UC4 (*Visualizza Programmazione*).
- Caso d'uso a livello utente UC7 (*Gestisci Clienti*).
- Caso d'uso a livello di sottofunzione UC15 (*Login*): Viene attenzionato l'aspetto legato all'autenticazione.
- Caso d'uso di avviamento.
- Dati solo in memoria principale.

In questo capitolo viene descritta l'analisi svolta nell'iterazione 2, considerando separatamente i casi d'uso di interesse.

2.2. Caso d'uso UC3, Modello di dominio

Analizzando il caso d'uso in esame non sono state individuate modifiche da apportare al modello di dominio realizzato nella precedente iterazione; i concetti di Prenotazione e Proiezione sono infatti già presenti.

2.3. Caso d'uso UC3, Diagramma di sequenza di sistema



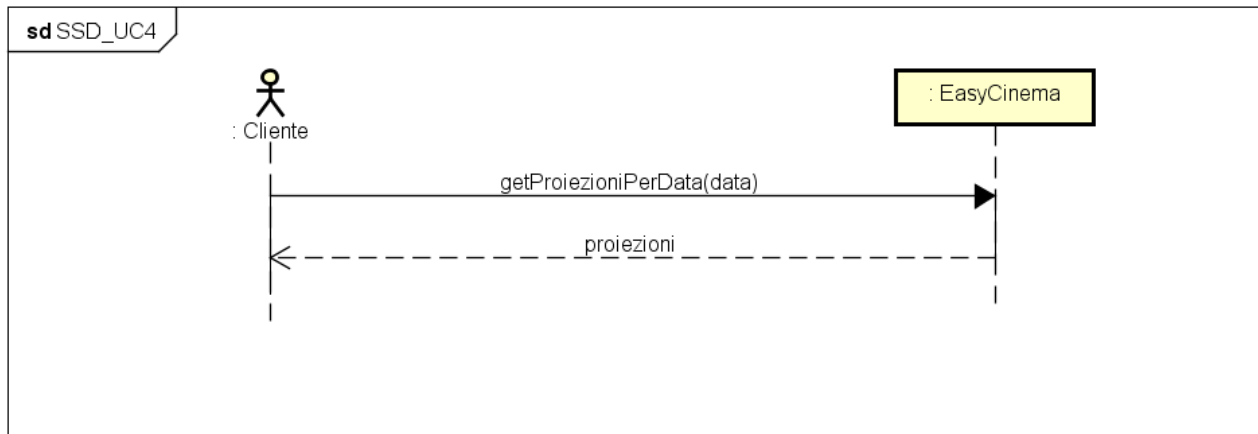
2.4. Caso d'uso UC3, Contratti delle operazioni

Le operazioni di sistema, riportate nel diagramma di sequenza di sistema di cui sopra, non determinano alterazioni dello stato degli oggetti del Modello di dominio in quanto sono delle semplici interrogazioni. Sulla base di queste considerazioni non si ritiene opportuno riportare alcun contratto.

2.5. Caso d'uso UC4, Modello di dominio

Anche per la realizzazione del caso d'uso UC4 non sono richieste modifiche al modello di dominio già presente.

2.6. Caso d'uso UC4, Diagramma di sequenza di sistema



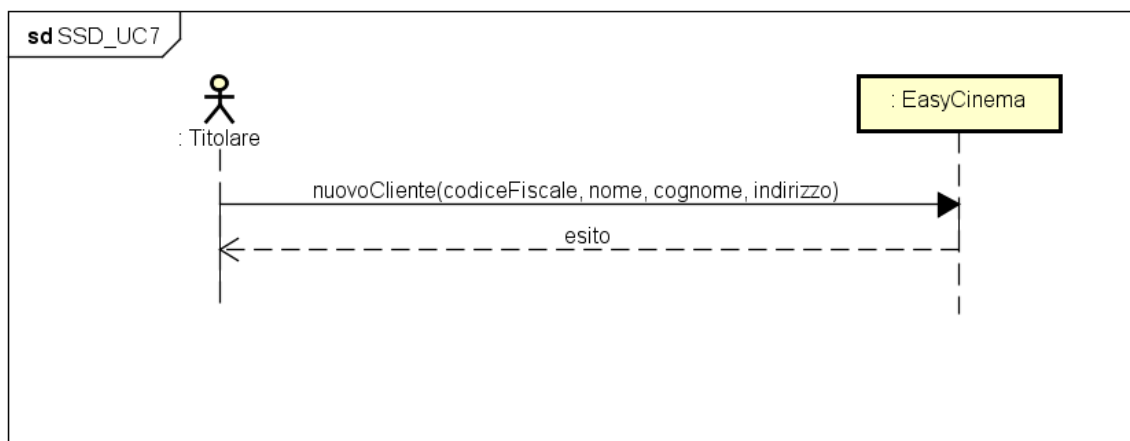
2.7. Caso d'uso UC4, Contratti delle operazioni

L'unica operazione di sistema individuata non richiede alterazioni di stato né a livello di istanza, né di attributi né tantomeno di associazioni. Dunque, non verrà presentato il contratto di tale operazione.

2.8. Caso d'uso UC7, Modello di dominio

Il focus del caso d'uso è sulla classe concettuale Cliente, la quale è già presente nel modello di dominio; esso dunque rimane invariato.

2.9. Caso d'uso UC7, Diagramma di sequenza di sistema

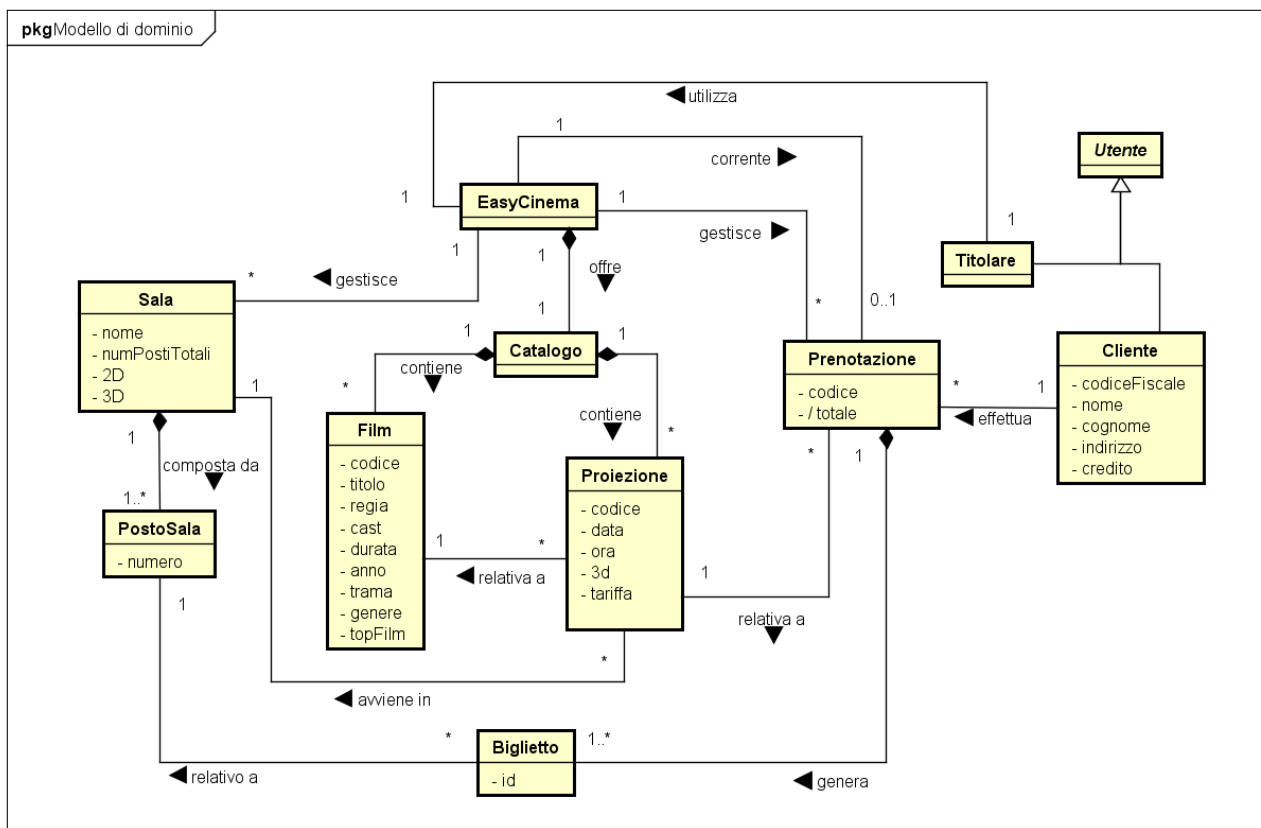


2.10. Caso d'uso UC7, Contratti delle operazioni

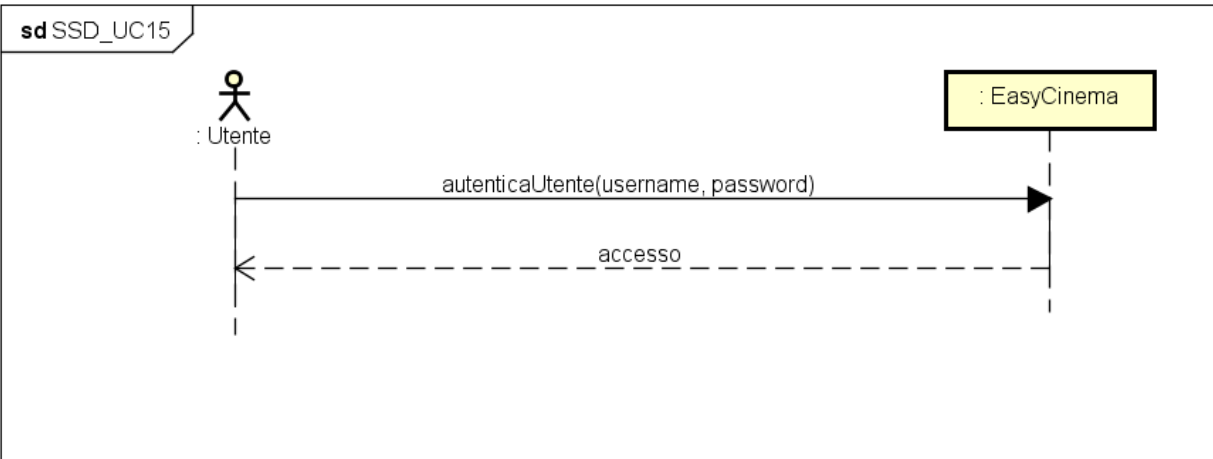
Operazione	nuovoCliente(codiceFiscale:String, nome:String, cognome:String, indirizzo:String)
Riferimenti	Caso d'uso: Gestisci Clienti
Pre-condizioni	- il Titolare t sta utilizzando il sistema
Post-condizioni	- è stata creata un'istanza c di Cliente; - gli attributi di c sono stati inizializzati.

2.11. Caso d'uso UC15, Modello di dominio

L'autenticazione degli utenti richiede la modifica del modello di dominio. Viene introdotta una nuova classe concettuale *Utente* superclasse di Titolare e Cliente. Essa, infatti, rappresenta un concetto più generale delle due e rispetta la regola *is-a* (ogni cliente/titolare è un utente) e la regola del *100%* (in merito agli attributi ed alle associazioni). *Utente* è inoltre una classe concettuale astratta in quanto un'istanza di *Utente* deve obbligatoriamente essere un Cliente o il Titolare.



2.12. Caso d'uso UC15, Diagramma di sequenza di sistema



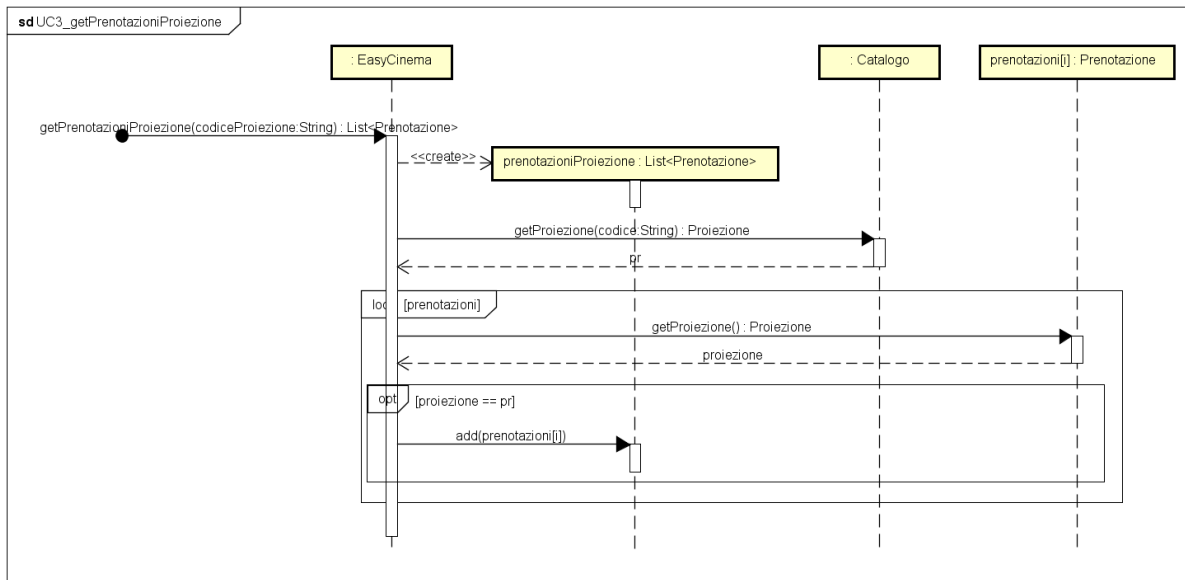
2.13. Caso d'uso UC15, Contratti delle operazioni

Operazione	autenticaUtente(username:String, password:String)
Riferimenti	Caso d'uso: Login
Pre-condizioni	
Post-condizioni	- Il particolare Utente u è stato associato con il GestoreUtenti tramite l'associazione "corrente";

3. Progettazione

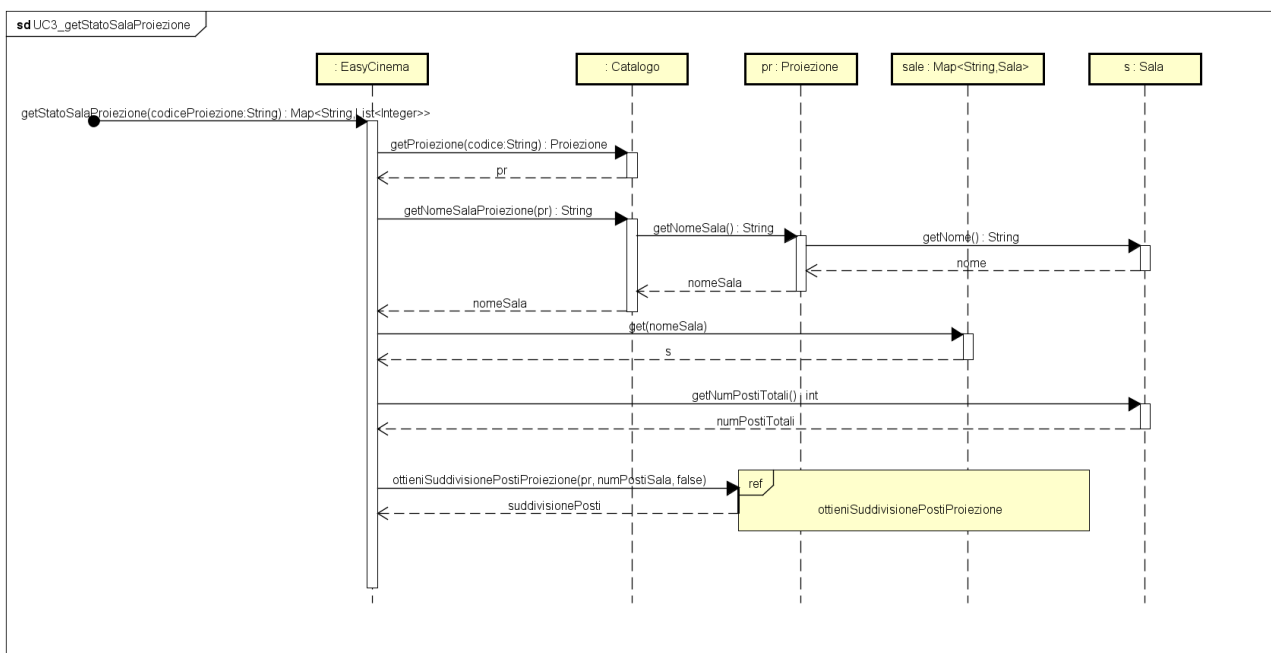
3.1. Caso d'uso UC3, Diagrammi di interazione

3.1.1. getPrenotazioniProiezione(codiceProiezione:String)



`EasyCinema` è responsabile della gestione delle prenotazioni, sarà dunque sua responsabilità creare e restituire la struttura contenente l'insieme delle prenotazioni di interesse. Per assolvere tale compito si avvale della classe software `Catalogo`, essa infatti gestisce le proiezioni. Infine, ciascuna prenotazione ha visibilità sulla `Proiezione` a cui fa riferimento. Tali considerazioni hanno guidato l'assegnazione delle responsabilità nel diagramma di sequenza sopra riportato.

3.1.2. getStatoSalaProiezione(codiceProiezione:String)

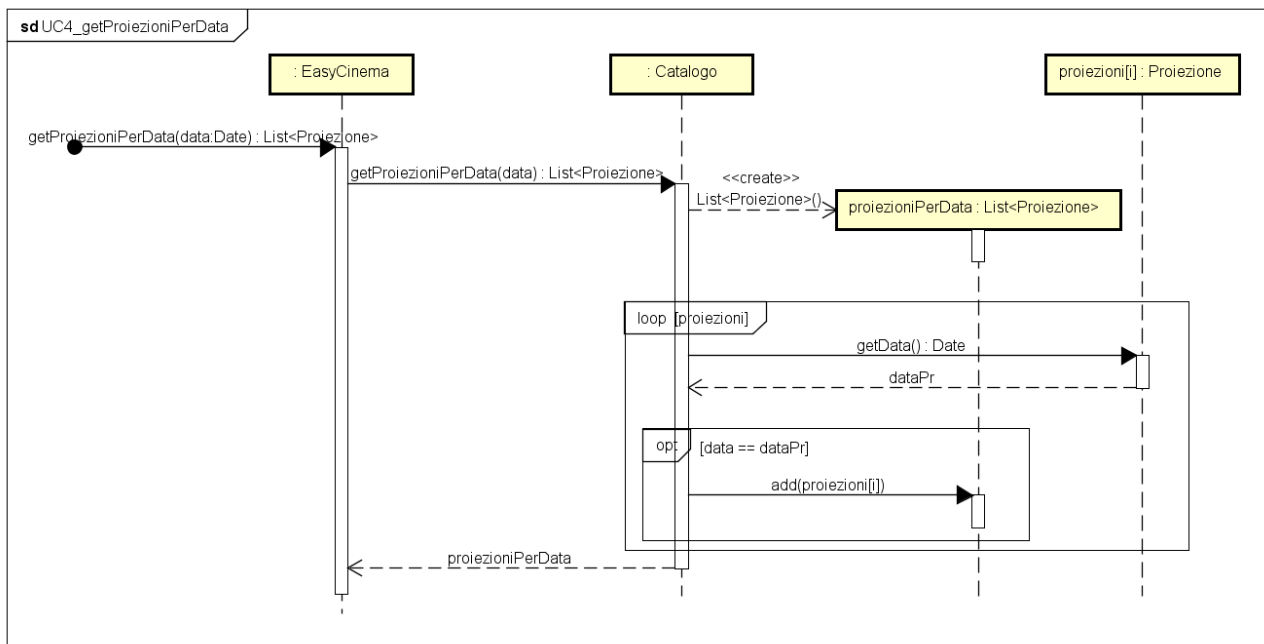


È stata assegnata una nuova responsabilità a Catalogo ovvero quella di recuperare il nome della Sala in cui ha luogo una Proiezione. Ciò è suggerito dal pattern Information Expert poiché Catalogo contiene le proiezioni.

Si fa uso della correlazione con il diagramma di interazione [ottieniSuddivisionePostiProiezione](#). Tale operazione, a partire dalle informazioni sulla Proiezione di interesse, il numero dei posti della Sala in cui avverrà la proiezione e se si tratta di una prenotazione in corso o meno (infatti vanno considerati come riservati anche quei posti selezionati nell'ambito della prenotazione corrente ma che non sono ancora stati confermati) ritorna una mappa avente per chiavi: "Disponibili" e "Occupati" e in corrispondenza di ciascuna chiave, come valore, una lista di interi (i numeri associati ai posti).

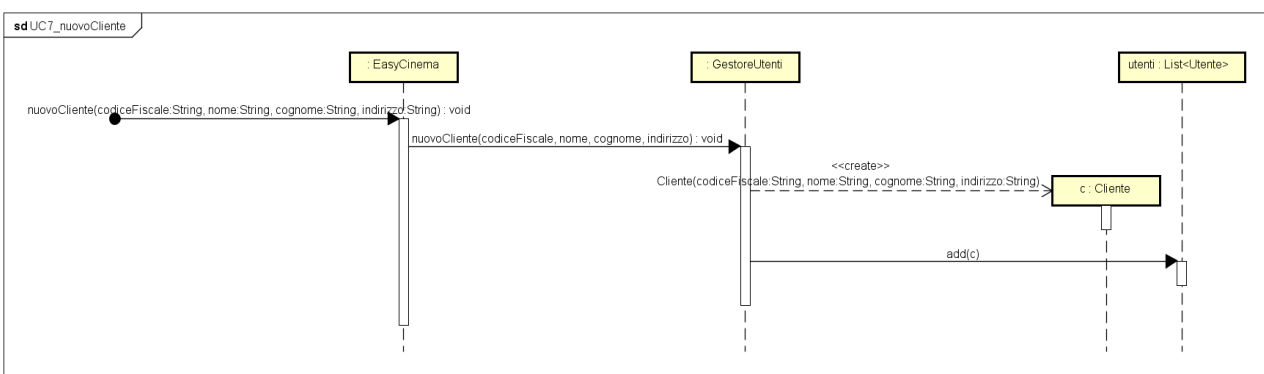
3.2. Caso d'uso UC4, Diagrammi di interazione

3.2.1. getProiezioniPerData(data: Date)



3.3. Caso d'uso UC7, Diagrammi di interazione

3.3.1. nuovoCliente(codiceFiscale:String, nome:String, cognome:String, indirizzo:String)

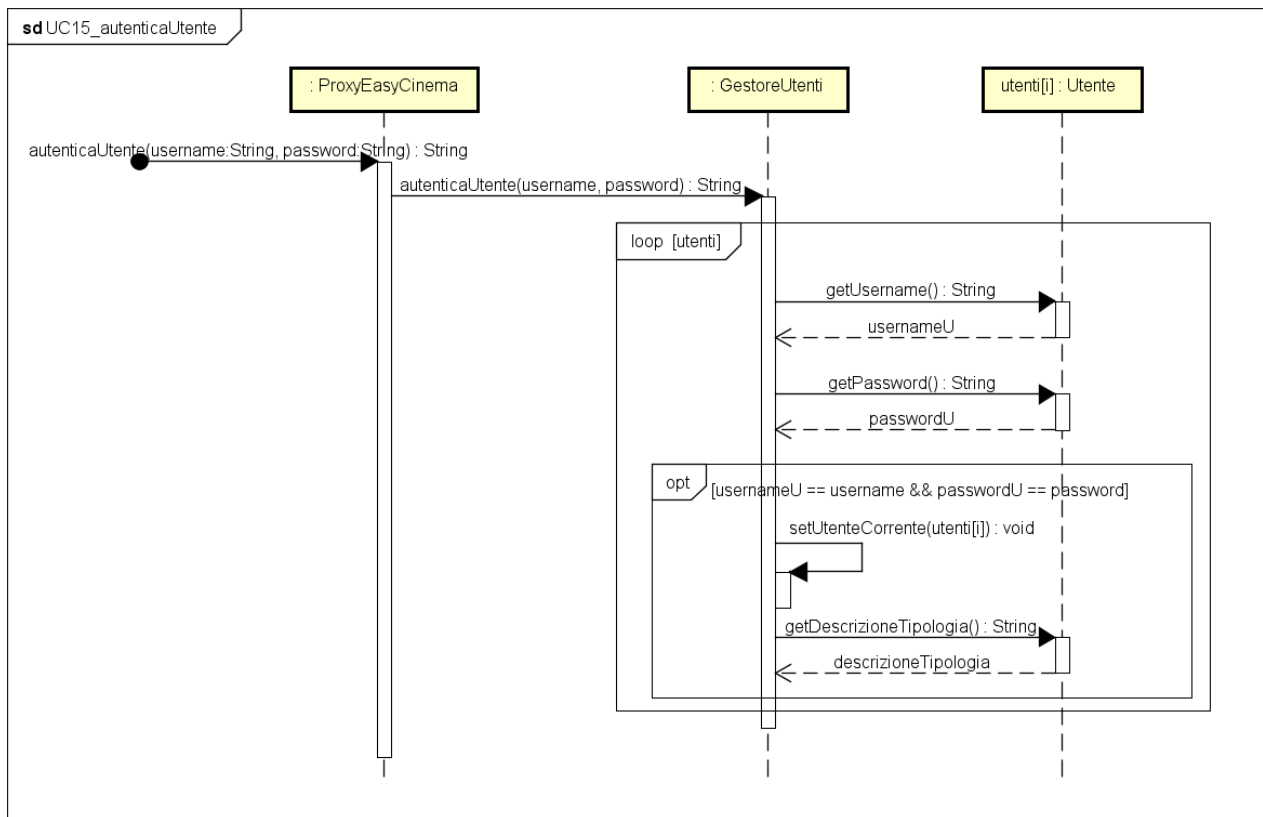


Inizialmente era stata individuata EasyCinema come responsabile della gestione degli utenti e dunque delle operazioni su di essi, in particolar modo per quelle riguardanti i clienti (creazione, modifica del credito ecc.). Volendo però ottenere un più basso accoppiamento e una più alta coesione (pattern GRASP *Low Coupling* e *High Cohesion*) si è scelto di ricorrere al Pattern GRASP *Pure Fabrication* introducendo la classe software GestoreUtenti, essa infatti non trova un riscontro a livello di modello di dominio.

La creazione dell'istanza di Cliente per il pattern GRASP *Creator* spetta dunque a GestoreUtenti in quanto esso aggrega oggetti di tipo Utente superclasse di Cliente.

3.4. Caso d'uso UC15, Diagrammi di interazione

3.4.1. autenticaUtente(username:String, password:String)

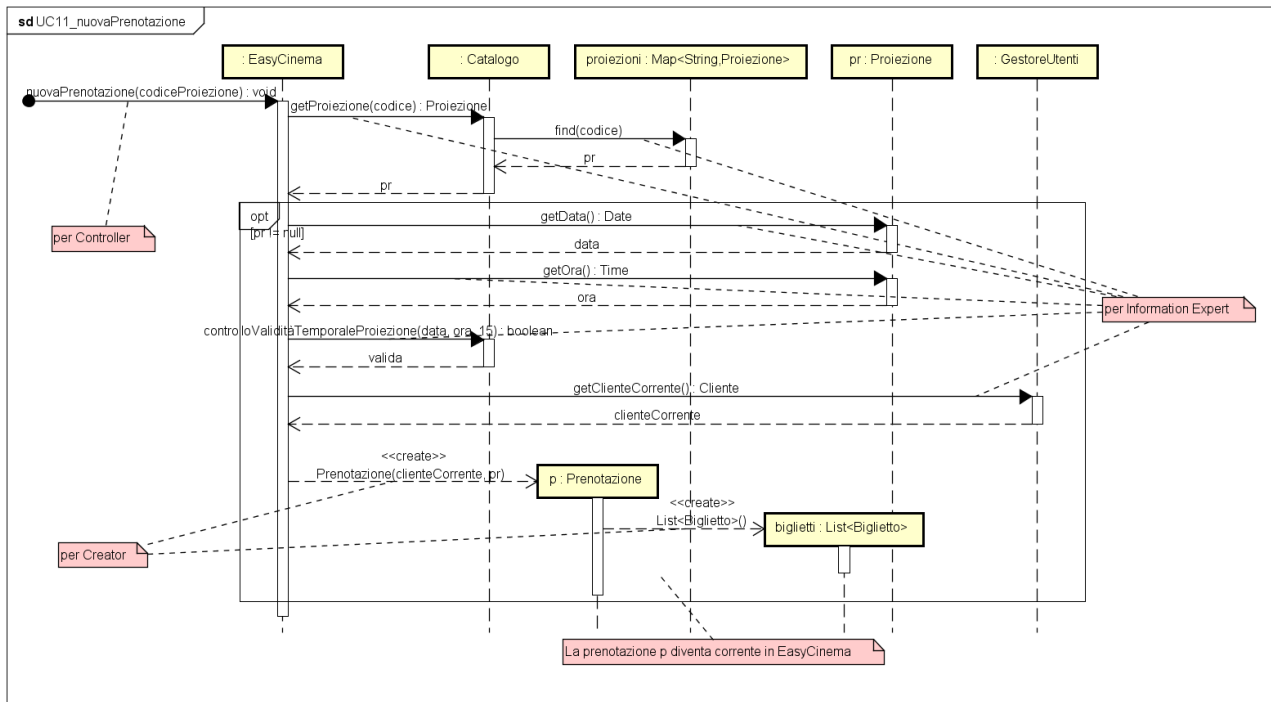


L'autenticazione dell'utente spetta, per *Information Expert*, a GestoreUtenti.

3.5. Modifiche ai diagrammi di sequenza della precedente iterazione

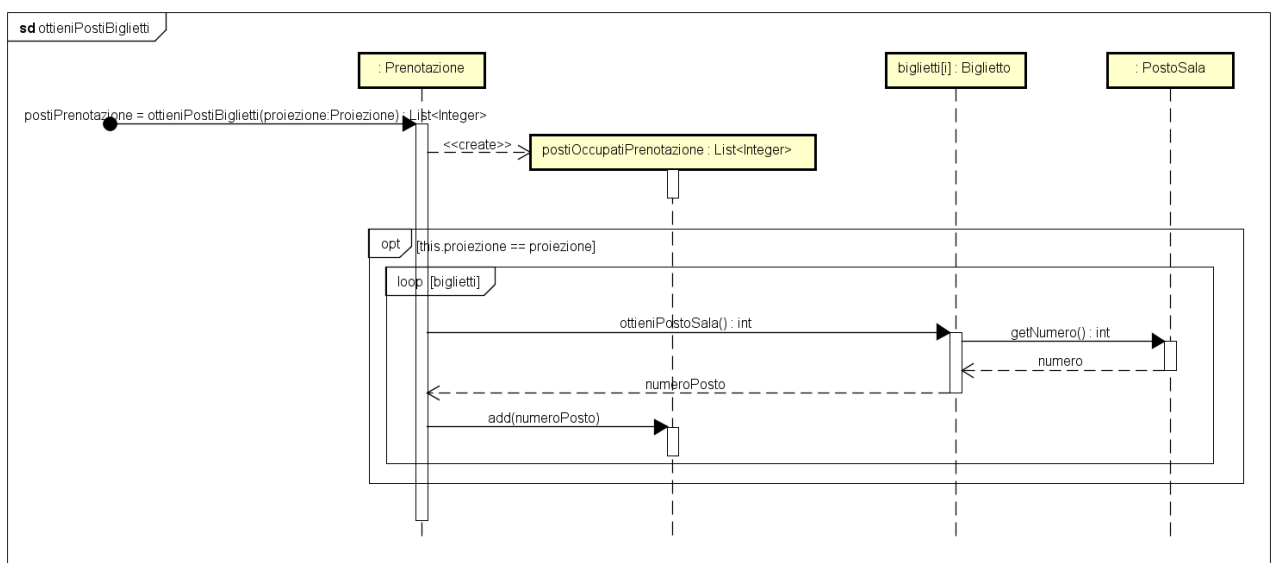
La progettazione dell'iterazione corrente ha determinato la modifica di alcuni dei diagrammi di sequenza sviluppata nella precedente iterazione. Tali diagrammi aggiornati vengono di seguito riportati.

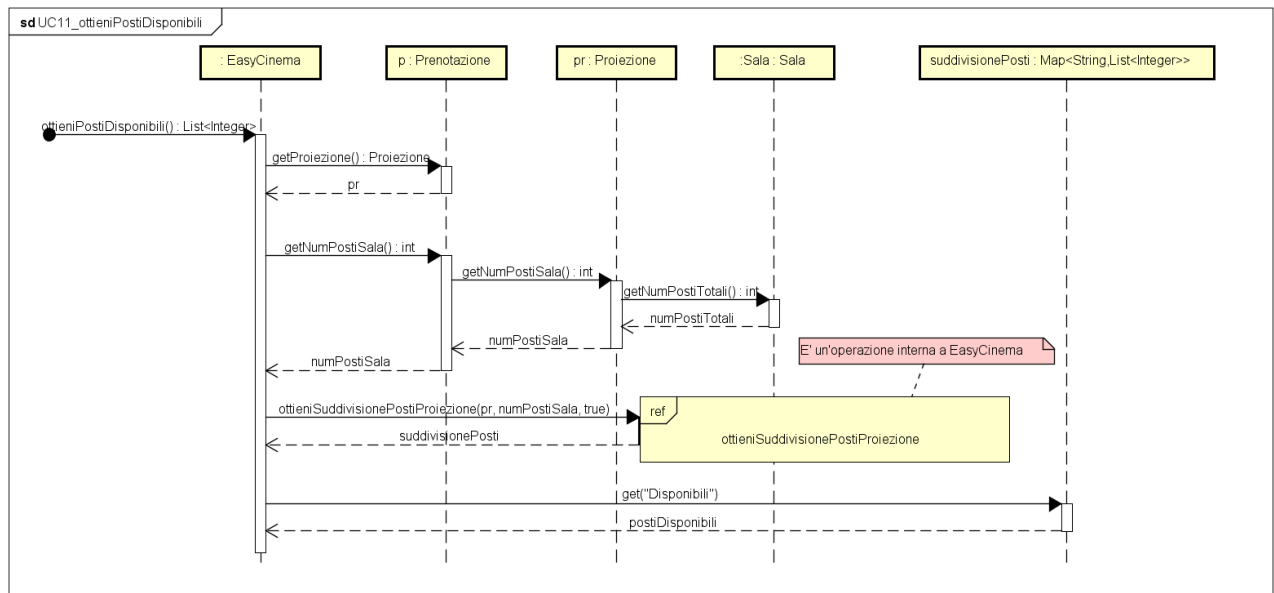
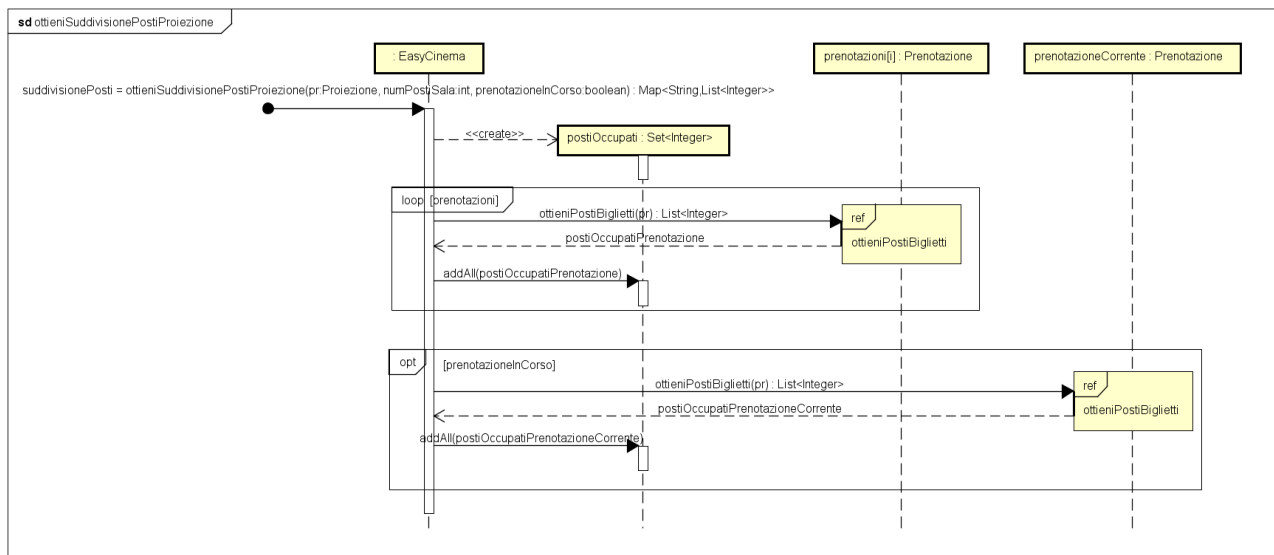
3.5.1. UC11 - nuovaPrenotazione(codiceProiezione:String)



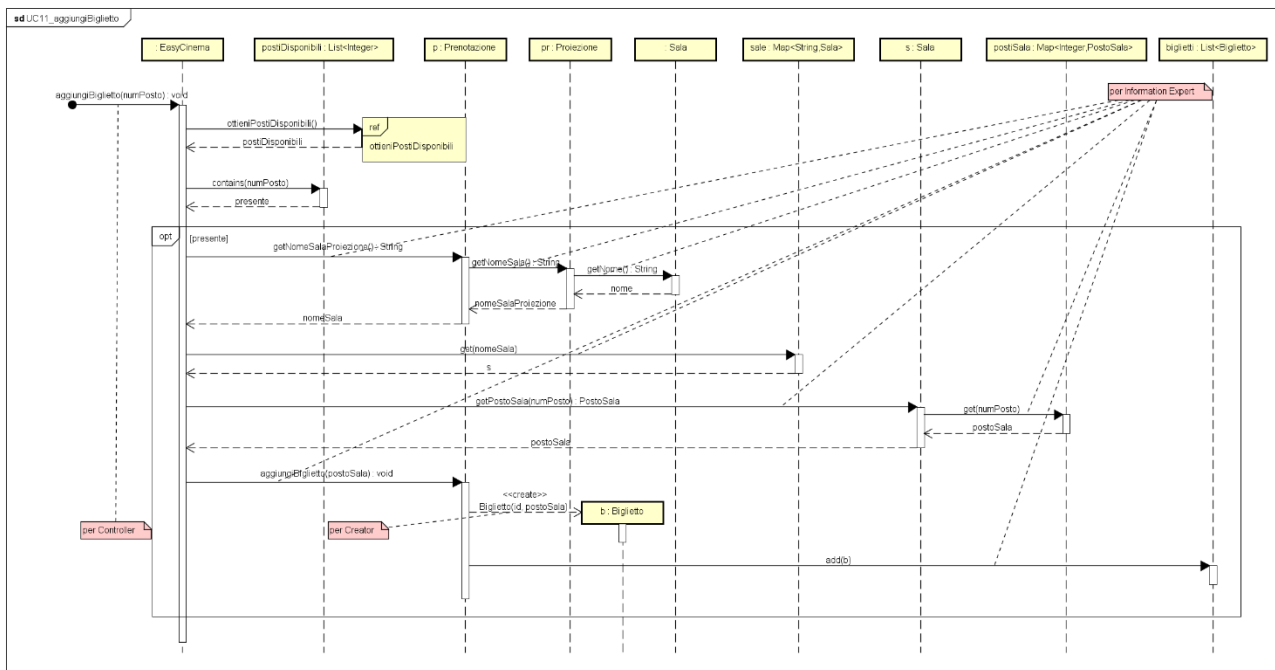
3.5.2. UC11 - ottieniPostiDisponibili ()

Essendo tale operazione piuttosto complessa è stata suddivisa in interazioni più semplici: `ottieniPostiBiglietti` e `ottieniSuddivisionePostiProiezione`. Esse vengono riportate e successivamente referenziate all'interno dei diagrammi di sequenza di seguito riportati.

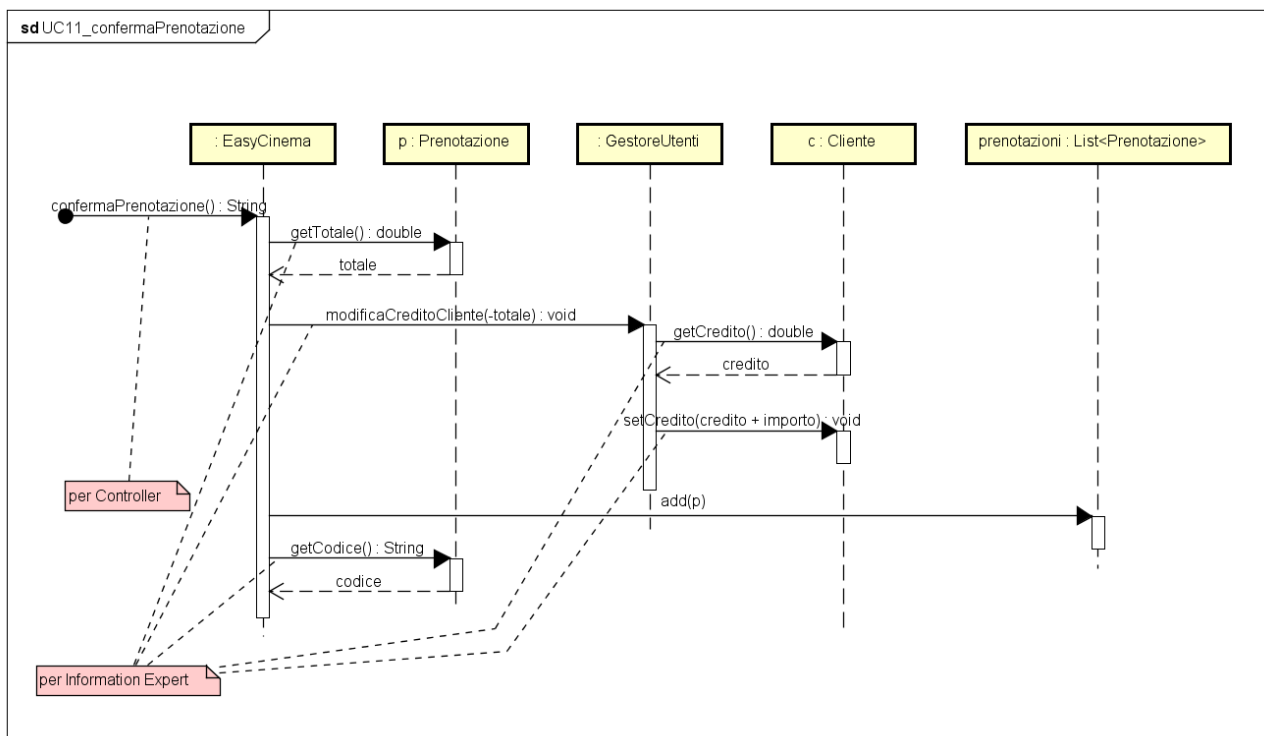




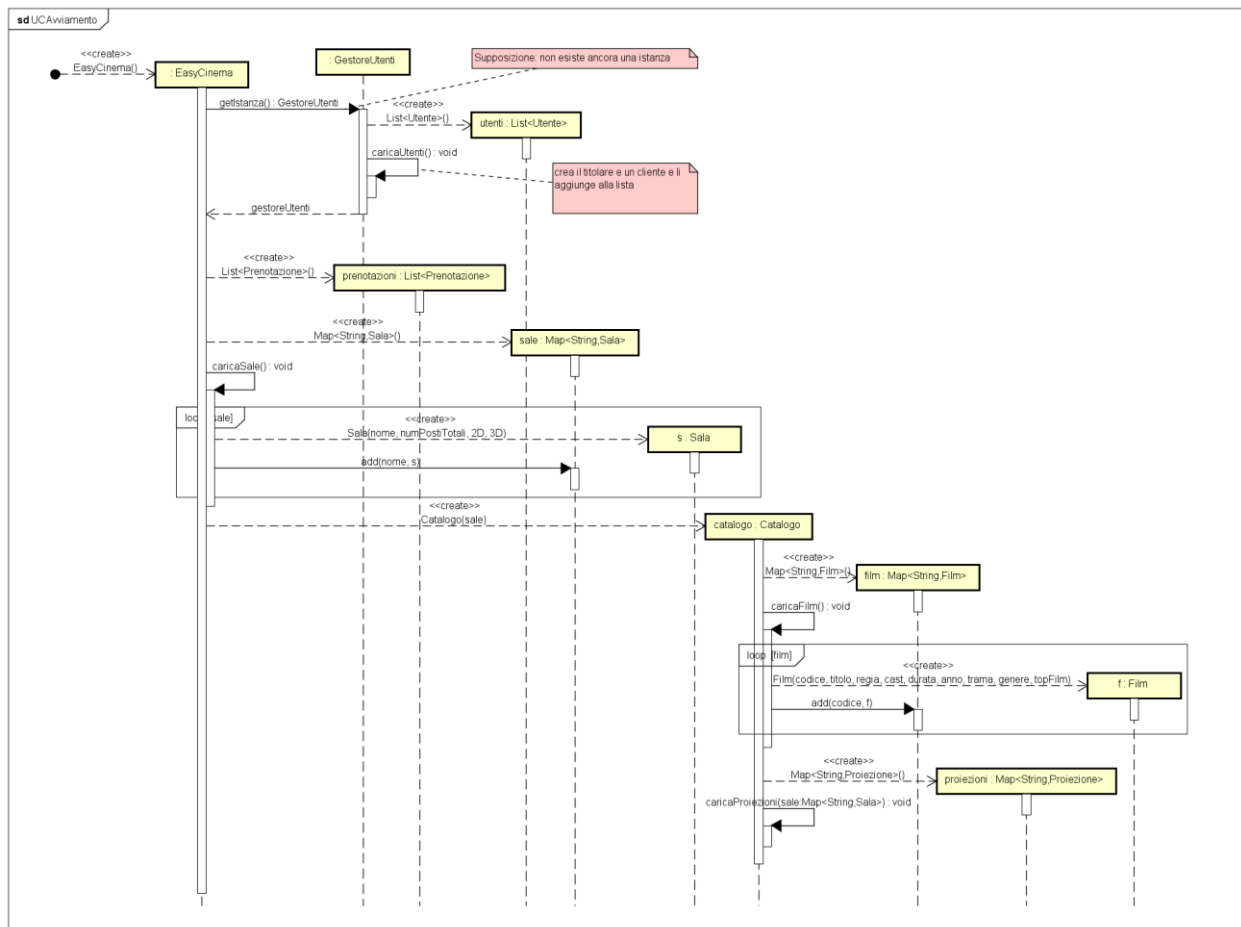
3.5.3. UC11 - aggiungiBiglietto(numPosto: Integer)



3.5.4. UC11 - confermaPrenotazione()



3.6. Caso d'uso di avviamento, Diagramma di interazione



3.7. Design Pattern GoF applicati

3.7.1. Protection Proxy

Il pattern Proxy, pattern strutturale basato su oggetti, permette di introdurre un oggetto da utilizzare come sostituto di un altro oggetto più complesso.

Più precisamente, si è fatto uso di un Protection Proxy il cui scopo è quello di controllare l'accesso all'oggetto di cui è il sostituto.

Nel caso del sistema software EasyCinema, solo gli utenti registrati e che hanno effettuato l'accesso possono interagire con il sistema. La classe software ProxyEasyCinema realizza il protection proxy passando le richieste provenienti dall'UI all'oggetto di tipo EasyCinema solo se l'utente è correttamente identificato e autenticato.

Il proxy espone gli stessi metodi dell'oggetto complesso che maschera; infatti, sia il proxy che l'oggetto da sostituire implementano un'interfaccia comune (IEasyCinema) in modo da risultare indistinguibili dall'esterno.

ProxyEasyCinema realizza un livello di indirection tra UI e il Sistema (pattern GRASP *Indirection*) ed è ottenuta per (pattern GRASP) *Pure Fabrication*.

Singleton è un design pattern creazionale. Permette di avere una sola istanza della classe a cui viene applicato fornendo un punto di accesso globale a tale istanza.

- EasyCinema: in qualità di Controller,
- ProxyEasyCinema: cosicché a livello di interfaccia utente non sia possibile crearne più istanze,
- GestoreUtenti: al fine di fornire un accesso globale e dunque condiviso a ProxyEasyCinema per l'autenticazione, mentre a EasyCinema per le operazioni sul Cliente.

[illegible]

4. Implementazione

Sono stati implementati gli scenari principali di successo e gli scenari alternativi per i casi d'uso presi in considerazione durante questa seconda iterazione. Solo per UC7, essendo questi un caso d'uso CRUD, ci si è limitati allo scenario principale di successo (considerando inoltre l'estensione 4a).

È stata aggiunta una schermata iniziale per l'autenticazione con successiva visualizzazione del menu principale, diverso a seconda della categoria di utente (titolare o cliente).

5. Testing

Sono state apportate le opportune modifiche nel setup dei vari unit tests sviluppati nell'iterazione precedente per adattarli alla nuova struttura del codice.

Poiché le operazioni relative ai casi d'uso UC3 e UC4 sono delle semplici interrogazioni si preferisce concentrarsi sulle operazioni dei casi d'uso UC7 e UC15; esse sono oggetto dei test.

Progettazione dei casi di test:

- GestoreUtenti

- **nuovoCliente**(String codiceFiscale, String nome, String cognome, String indirizzo): un nuovo cliente può essere aggiunto al sistema solo se non vi è già un altro cliente con lo stesso codice fiscale. Si vuole testare il numero di utenti presenti nel sistema.

Le condizioni di interesse individuate sono:

1. *testInserimentoValido*

- Input: generalità nuovo utente, nessun altro utente nel sistema.
- Output previsto: incremento unitario del numero di utenti.

2. *testDueInserimentiValidi*

- Input: generalità di due nuovi utenti, con codice fiscale distinto.
- Output previsto: incremento di due unità del numero di utenti.

3. *testCFDoppione*

- Input: generalità del nuovo utente con codice fiscale uguale ad un altro.
- Output previsto: generazione eccezione e numero utenti invariato.

- **autenticaUtente**(String username, String password): solo un utente (titolare/cliente) registrato al sistema può accedervi. Username e password sono di default uguali.

Le condizioni di interesse individuate sono:

1. *testCredenzialiErrate*

- Input: username e password errati.
- Output previsto: autenticazione fallita.

2. *testCredenzialiTitolare*

- Input: username e password associate al titolare.
- Output previsto: autenticazione titolare.

3. *testCredenzialiCliente*

- Input: username e password associate ad un cliente.
- Output previsto: autenticazione cliente.

- **controlloAutorizzazione**(Class categoriaUtenteAbilitata): un Cliente può accedere solo alle funzionalità ad esso indirizzate, lo stesso vale per il Titolare. Un utente che non è stato prima identificato e autenticato non può usufruire dei servizi del sistema.

Le condizioni di interesse individuate sono:

1. *testClienteRichiedePrivilegiTitolare*

- Input: cliente del sistema richiede servizi destinati al titolare.
- Output previsto: autorizzazione negata.

2. *testTitolareRichiedePrivilegiCliente*

- Input: il titolare del sistema richiede servizi destinati al cliente.
- Output previsto: autorizzazione negata.

3. *testClienteRichiedePrivilegiCliente*

- Input: cliente del sistema richiede servizi destinati al cliente.
- Output previsto: autorizzazione concessa.

4. *testTitolareRichiedePrivilegiTitolare*

- Input: il titolare del sistema richiede servizi destinati al titolare.
- Output previsto: autorizzazione concessa.

5. *testUtenteNonAutenticatoRichiedePrivilegiTitolare*

- Input: un utente non ancora autenticato richiede servizi destinati al titolare.
- Output previsto: autorizzazione negata.

6. *testUtenteNonAutenticatoRichiedePrivilegiCliente*

- Input: un utente non ancora autenticato richiede servizi destinati al cliente.
- Output previsto: autorizzazione negata.