

DOCUMENTAZIONE PROGETTO

INGEGNERIA DEL

SOFTWARE



Matteo Mainenti
VR397625

Negozio di CD

Indice

1. Introduzione	3
2. Use Case e Schede Di Specifica	4
2.1 Login & Registrazione	5
2.2 Visualizzazione Catalogo & Dettagli Prodotto	7
2.3 Modificare il Carrello (Aggiungi e Rimuovi	9
2.4 Visualizza Carrello	11
2.5 Ricerca	12
2.6 Acquista	13
3. Implementazione	15
3.1 Pattern MVC	15
3.2 File XML	16
3.3 Classi Utilizzate	16
3.4 Nota: Observer	21
4. Activity & Sequence Diagram	21
4.1 Avvio del programma	21
4.2 Login & Registrazione	23
4.3 Search	24
4.4 Attività Utente Registrato	25
5. Fase di Testing	27
6. Scelte Progettuali	28

1. Introduzione

Per la seguente documentazione è stato richiesto di progettare un sistema informativo capace di gestire la compravendita di CD.

Per fare ciò è stato scelto di rappresentare il sistema del punto di vista dell'utente, il cui scopo sarà interagire con le schermate del programma per svolgere le attività richieste , ovvero la possibilità di poter effettuare un login sulla piattaforma e avere un carrello dedicato a cui successivamente potrà aggiungere e/o rimuovere gli elementi che preferisce.

In alternativa l'utilizzatore , nell'eventualità in cui non voglia acquistare nessun prodotto, può anche scegliere di visualizzare le informazioni relative ai vari CD, oppure semplicemente effettuare una ricerca secondo diverse modalità, in base al proprio gradimento.

In questa documentazione sono presenti Class Diagram delle classi utilizzate, Use Case Diagram delle azioni utilizzabili dall'utente, Sequence Diagram e Activity Diagram per le interazioni tra le varie componenti e tra utente e progetto.

Nei diagrammi richiesti vengono illustrate le interazioni tra le varie componenti del sistema, il cui input iniziale sarà sempre dato dall'utente, per poter osservare e comprendere al meglio la struttura del progetto nella sua totalità. Per completezza sono stati aggiunti diagrammi più generali, come i Sequence Diagram, in cui gli attori sono semplicemente l'utente, la vista e il sistema stesso.

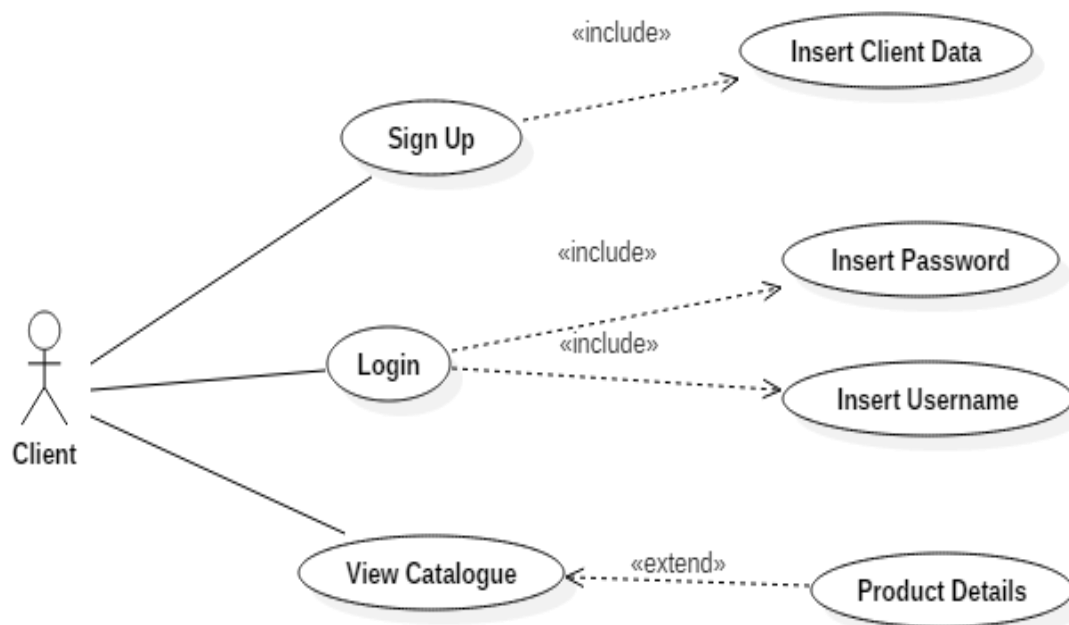
Per il progetto è stato scelto di utilizzare un pattern di tipo MVC, in cui le parti vengono suddivise in Model, Control e View per poi interagire tra di loro (Più approfondito nella sezione dedicata).

2. Use Case e Schede di Specifica.

Gli Use Case del progetto dal punto di vista dell'utente sono illustrate di seguito. Ogni Use Case è accompagnato dalla relativa scheda di specifica e dal relativo Sequence Diagram.

Nello Use Case Diagram viene visualizzato l'insieme delle azioni che un utente può compiere all'interno del programma. Ogni utente, persino quelli che non hanno effettuato il Login, possono visualizzare il Catalogo e vedere le informazioni relative al CD di preferenza.

Per accedere alle altre funzionalità, ovvero visualizzare il Carrello e aggiungere/rimuovere elementi da esso è permesso solo ed unicamente dopo il Login.



2.1 Login & Registrazione

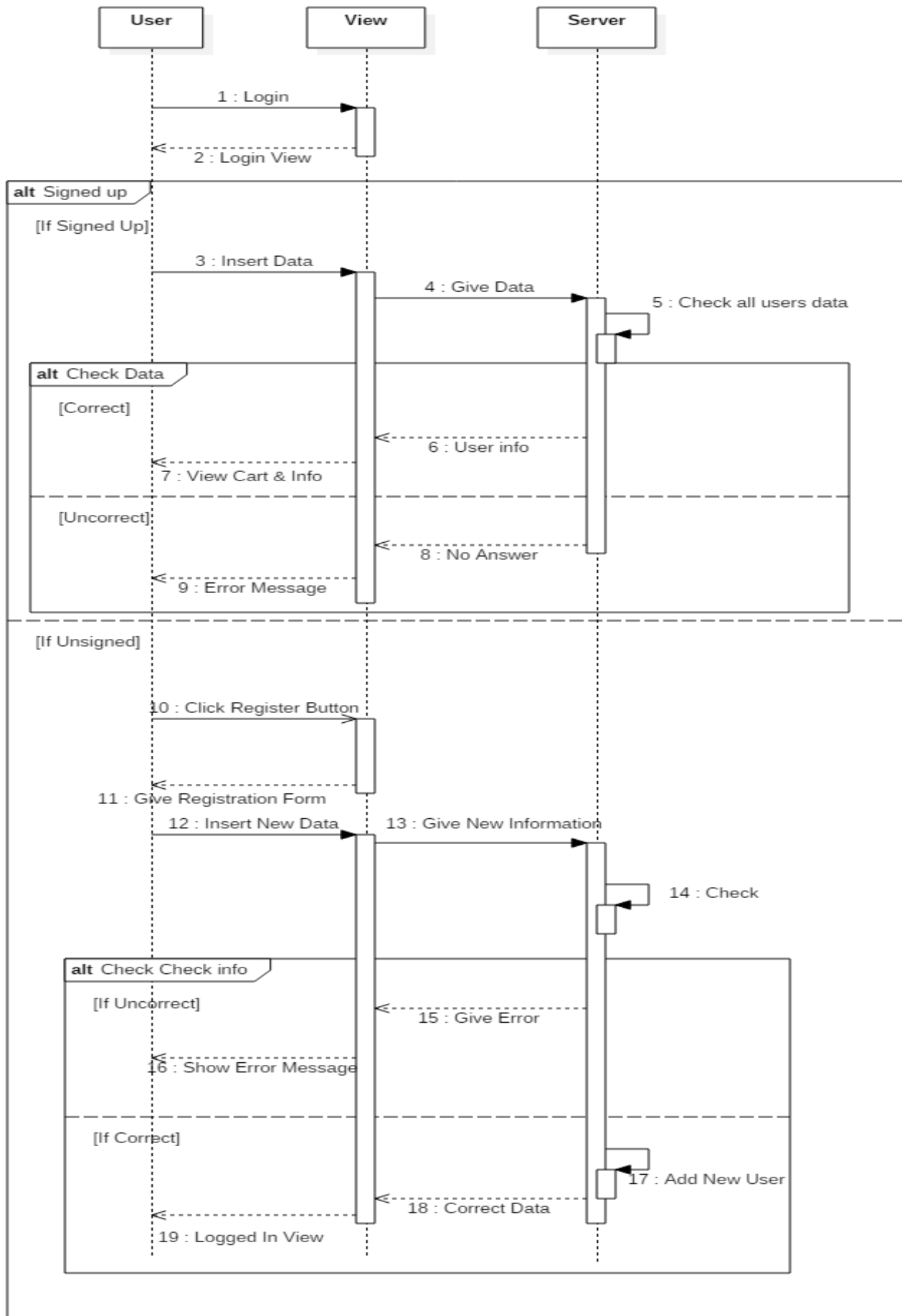
Per accedere al sistema come utente registrato ci sono due metodi, essere già registrati ed effettuare il Login, oppure Registrarsi. In entrambi i casi, successivamente al completamento di una di queste due azioni, l'utente visualizzerà una nuova finestra con azioni aggiuntive che verranno elencate nelle prossime pagine.

Il cliente durante il login deve solo inserire i dati che ha inserito precedentemente durante la registrazione, il sistema controllerà la lista di tutti gli utenti registrati confrontando la Password e Username inseriti con quelle che ha nel proprio database. Se verrà trovato un match tra i dati inseriti e quelli presenti nel database, allora l'utente avrà effettuato il Login, in caso contrario gli verrà richiesto di inserire nuovamente i dati.

Nel caso della registrazione invece, all'utente verrà richiesto di inserire tutte le sue credenziali, ovvero username, password, nome, cognome, codice fiscale, città di provenienza e numero di telefono. Successivamente all'inserimento di tutti i dati il sistema controllerà la correttezza di questi, se i dati rispettano le politiche di inserimento allora un nuovo utente verrà registrato nel database e verrà visualizzata la nuova schermata utente, in caso contrario verrà visualizzato un messaggio di errore per richiedere di inserire nuovamente i dati considerati errati.

Caso d'uso: Login
ID: UC1
Precondizioni: Il cliente ha già effettuato l'operazione di registrazione in passato.
Sequenza degli eventi: <ol style="list-style-type: none"> 1. Il caso d'uso inizia quando l'user clicca il tasto Login nella schermata principale 2. L'user inserisce il proprio username e password 3. Avviene un controllo sulla correttezza dei dati inseriti <ol style="list-style-type: none"> 3.1 I dati sono corretti, avviene il login 3.2 I dati sono errati, chiede di inserire di nuovo i dati
Post Condizioni: Avviene l'accesso al proprio profilo utente, è possibile visualizzare il carrello e aggiungervi relativi articoli

Caso d'Uso: Sign up
ID: UC2
Precondizioni: L'utente ha aperto il programma e clicca sul tasto di login.
Sequenza degli eventi: <ol style="list-style-type: none"> 1. L'utente clicca sul tasto registra. 2. L'utente inserisce tutti i dati richiesti 3. Controllo dei dati inseriti <ol style="list-style-type: none"> 3.1 I dati sono errati, chiede di inserirli di nuovo 3.2 I dati sono corretti, l'utente è registrato 4. L'utente vede la schermata UserView
PostCondizioni: Avviene l'accesso al proprio profilo utente, è possibile accedere a tutte le funzionalità ad esso collegato.
Sequenza alternativa: L'utente decide di uscire dalla finestra di registrazione



2.2 Visualizzazione Catalogo & Dettagli Prodotto

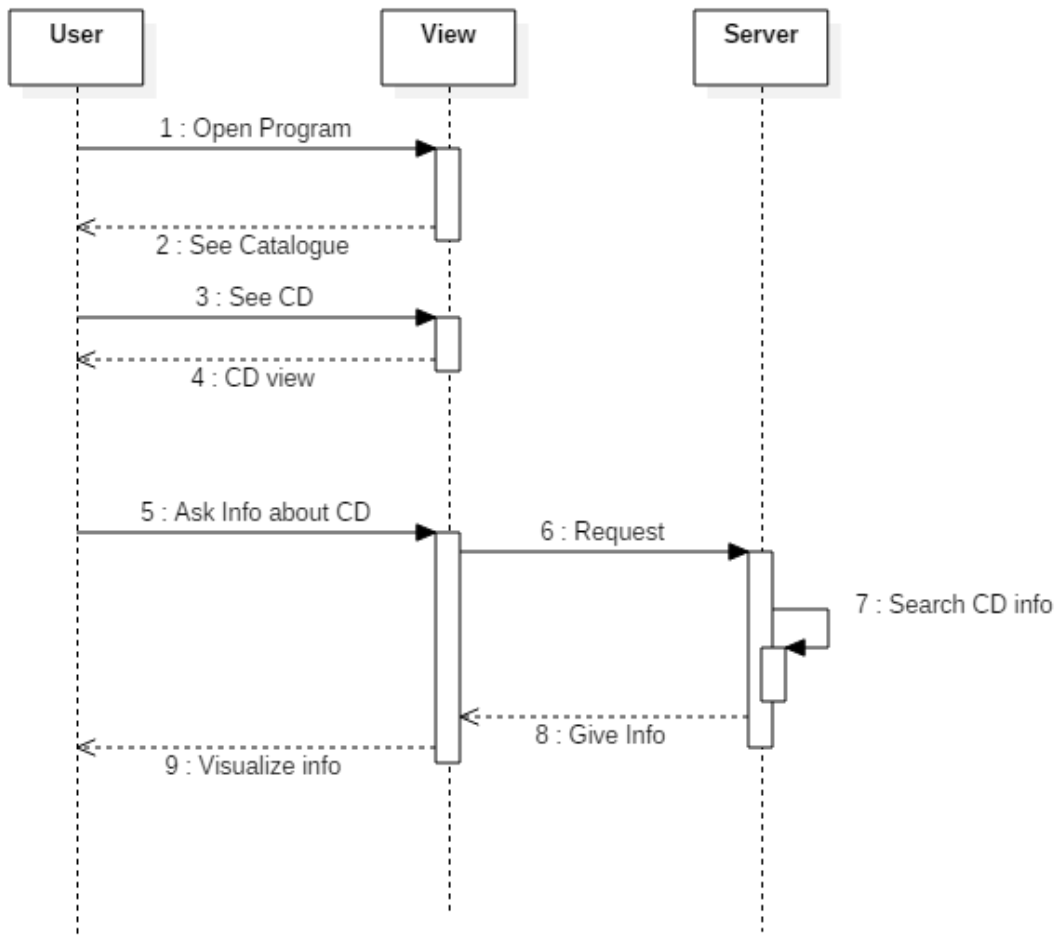
Visualizzare il catalogo permette all'utente di essere a conoscenza di quali elementi lo compongono. Grazie alla visualizzazione del catalogo è possibile navigare tra i vari elementi. L'unica azione che l'utente che non ha effettuato il login nel sistema può effettuare è solo visualizzare le informazioni relative a un singolo CD. Solo ed unicamente dopo il login l'utente potrà decidere se aggiungere l'elemento visualizzato al proprio carrello.

L'utente quando richiede le informazioni di un CD può visualizzare nome del CD, autore, data di rilascio, codice di riferimento, tracklist, genere e prezzo.

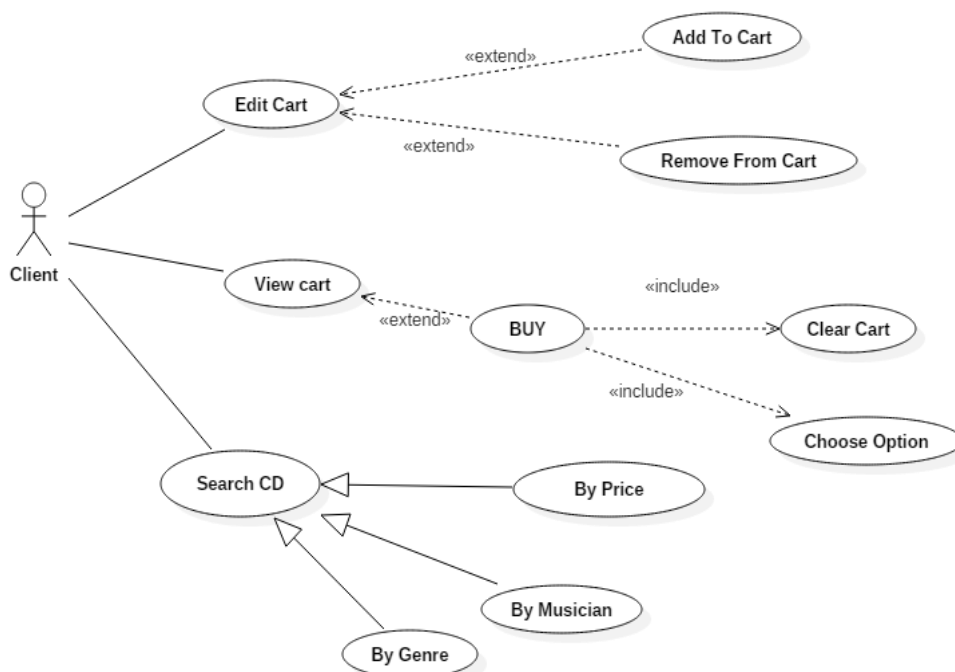
Il sistema ovviamente tiene conto dell'elemento visualizzato da parte dell'utente, in modo che quando l'utente voglia visualizzare delle informazioni il sistema possa prenderle e comunicare le info da visualizzare.

Caso d'uso: Product Details
ID:UC3
Precondizioni: L'utente sta visualizzando il catalogo, selezionando un particolare CD
Sequenza eventi: <ol style="list-style-type: none">1. L'utente preme il tasto info.2. L'utente visualizza le informazioni relative al CD
Postcondizioni: l'utente ha ottenuto le info relative ad uno specifico CD.

Caso d'uso: View Catalogue
ID:UC4
Precondizioni: il programma è stato aperto
Sequenza eventi: <ol style="list-style-type: none">1. L'utente all'apertura visualizza il contenuto del catalogo2. È possibile visualizzare un cd alla volta3. Per visualizzare un altro cd basterà premere i tasti NEXT o PREV
Postcondizioni: l'utente visualizza il cd attuale



Il seguente Use Case Diagram indica le rimanenti azioni che può effettuare l'utente, ovvero la ricerca di alcuni CD in base al prezzo, al musicista o al genere. Mentre le azioni per visualizzare il carrello e di modificarlo sono possibili solo dopo il login.



2.3 Modificare il Carrello (Aggiungi e Rimuovi)

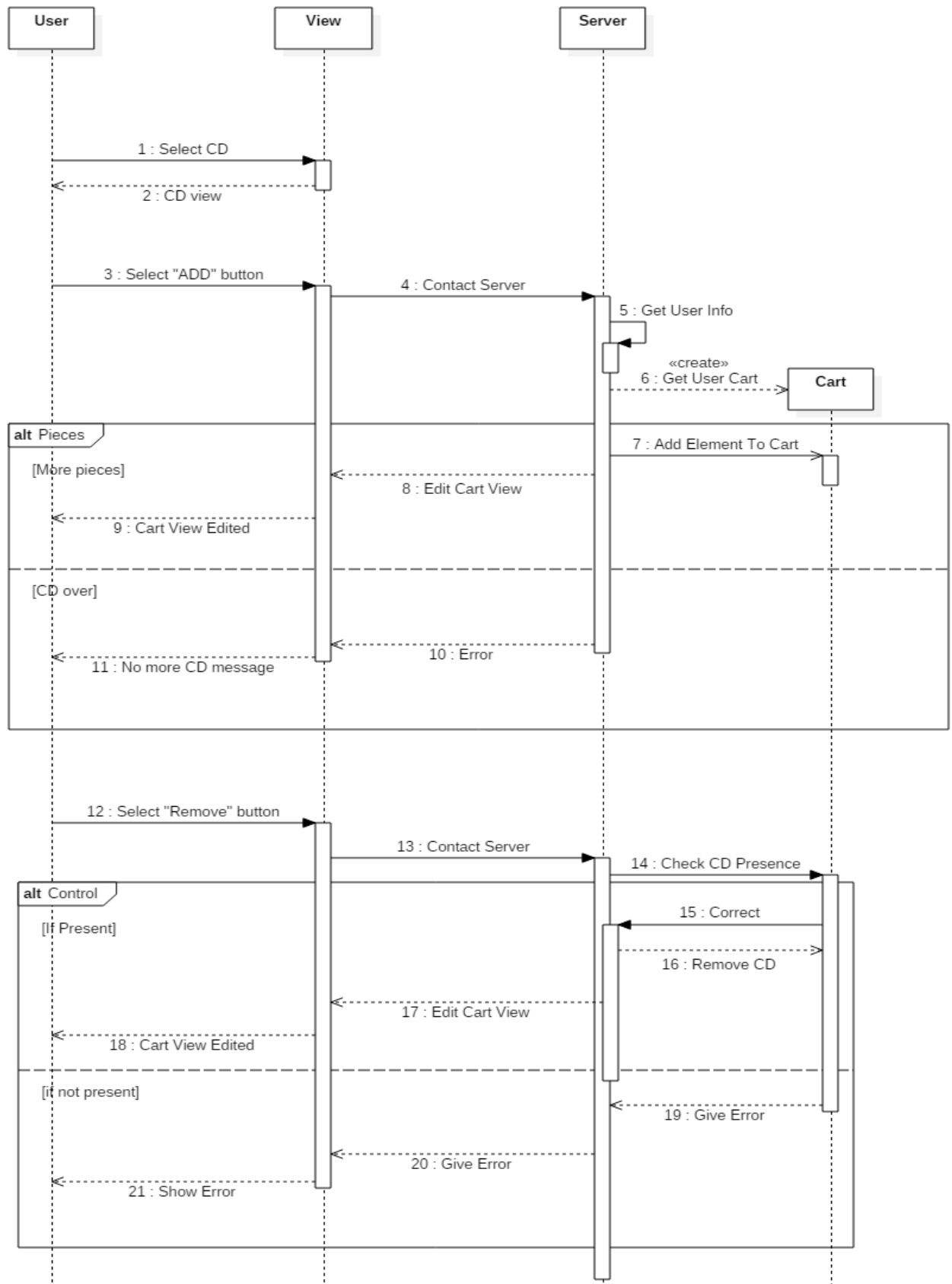
Aggiungere e rimuovere elementi da un carrello è una parte fondamentale nel programma, in quanto permette di spostare pezzi dal catalogo per metterli a disposizione dell'utente che vuole effettuare l'acquisto. L'utente sceglie un CD, dopodiché premendo il tasto di aggiunta viene chiamato il sistema che cerca le informazioni inerenti all'utente per poi andare a selezionare il relativo carrello a cui aggiungere l'elemento selezionato. L'aggiunta viene effettuata solo se nel catalogo sono ancora disponibili dei pezzi del CD selezionato, in caso contrario viene segnalato un errore.

Per quanto riguarda la rimozione, il sistema effettua un controllo sul carrello dell'utente per capire se il CD selezionato dall'utente è effettivamente presente nel suo carrello. Nel caso in cui il CD sia presente, ne rimuove un'unità, in caso contrario dà un segnale di errore in quanto un utente non può rimuovere un oggetto non presente nel proprio carrello.

Alla fine di entrambi i processi il sistema aggiorna il carrello e il suo prezzo totale e l'utente potrà vedere la finestra del proprio carrello modificata.

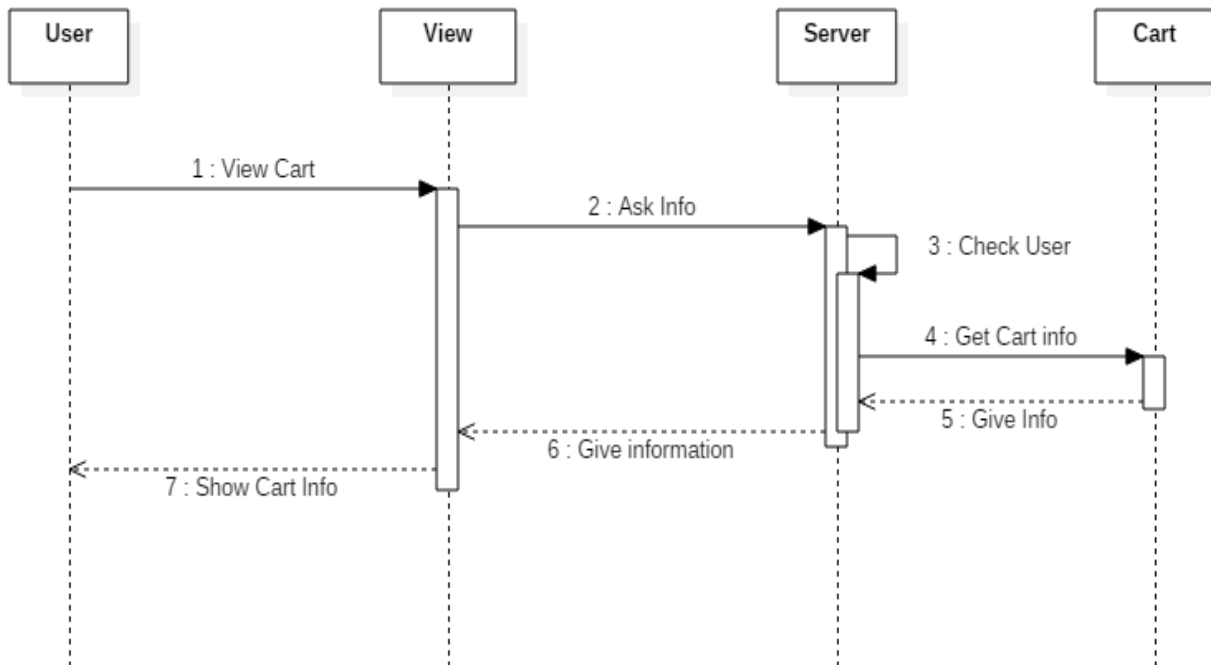
Caso d'uso: Remove from cart
ID:UC5
Precondizioni: L'utente ha effettuato il login e ha precedentemente aggiunto degli elementi al proprio carrello
Sequenza eventi: <ol style="list-style-type: none"> 1. L'utente preme sul tasto Remove su un Cd 2. Viene effettuato un controllo <ol style="list-style-type: none"> 2.1 Il cd è contenuto nel carrello 2.2 Il cd non è contenuto nel carrello
Postcondizioni: il carrello è stato aggiornato e un elemento del CD scelto è stato rimosso.

Caso d'uso: Add to Cart
ID: UC6
Precondizioni: L'utente ha effettuato il login e sta visualizzando il catalogo
Sequenza eventi: <ol style="list-style-type: none"> 1. L'utente preme su ADD 2. Controllo sulla quantità di pezzi <ol style="list-style-type: none"> 2.1 Sono ancora disponibili dei pezzi 2.2 Non sono più disponibili dei pezzi 3.
PostCondizioni Il CD viene aggiunto al carrello e il prezzo del carrello aumenta



2.4 Visualizzare il Carrello

Ogni utente può visualizzare su richiesta il contenuto del proprio carrello, in modo da vedere i CD precedentemente aggiunti. Ogni volta che viene effettuata la richiesta il sistema va a prelevare le informazioni dell'utente per selezionare il carrello del richiedente, successivamente invia alla view le informazioni necessarie.



Casi d'Uso: View Cart

ID:UC7

Precondizioni: L'utente ha effettuato l'accesso.

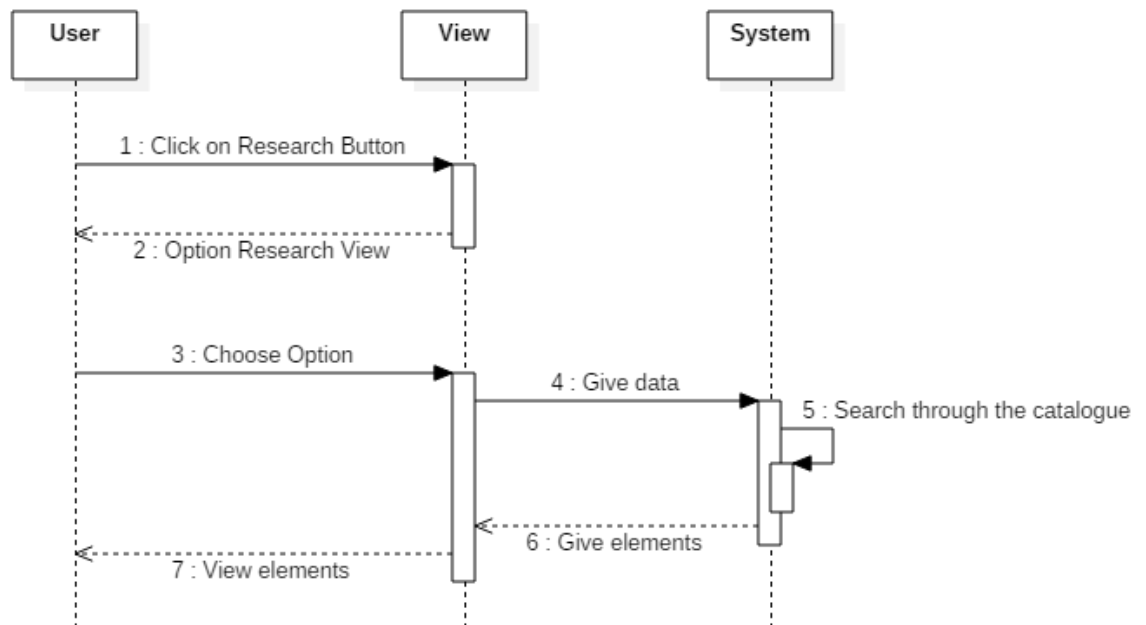
Sequenza eventi:

1. Dalla schermata l'utente clicca su "cart"
2. L'utente visualizza il contenuto del carrello e il prezzo totale dei suoi prodotti.

PostCondizioni: dopo aver visualizzato il carrello l'utente è libero di continuare ad aggiungere o rimuovere elementi al proprio carrello.

2.5 Ricerca

Un utente fin dalla schermata iniziale può cercare dei CD secondo determinate caratteristiche grazie al comando di ricerca. Dopo aver premuto il tasto di ricerca all'utente viene richiesto secondo quale criterio vuole vedere i CD, grazie a questa scelta l'utente sarà in grado di visualizzare CD di un determinato autore, di un determinato genere o con un determinato prezzo. Il sistema secondo la richiesta ricevuta sfoglierà tutto il catalogo in cerca dei CD le cui caratteristiche corrispondono a quelle richieste, dopodiché li raccoglierà e li mostrerà all'utente come risultato.



Casi d'Uso: Search CD

ID:UC8

PreCondizioni: L'utente ha aperto il programma

Sequenza eventi:

1. L'utente clicca sul tasto search
2. Viene aperta una finestra in cui l'utente può scegliere con quali parametri cercare i prodotti
 - 2.1 Cerca per prezzo
 - 2.2 Cerca per musicista
 - 2.3 Cerca per genere
3. Avvia ricerca

Postcondizioni: Successivamente l'utente visualizza i risultati della propria ricerca

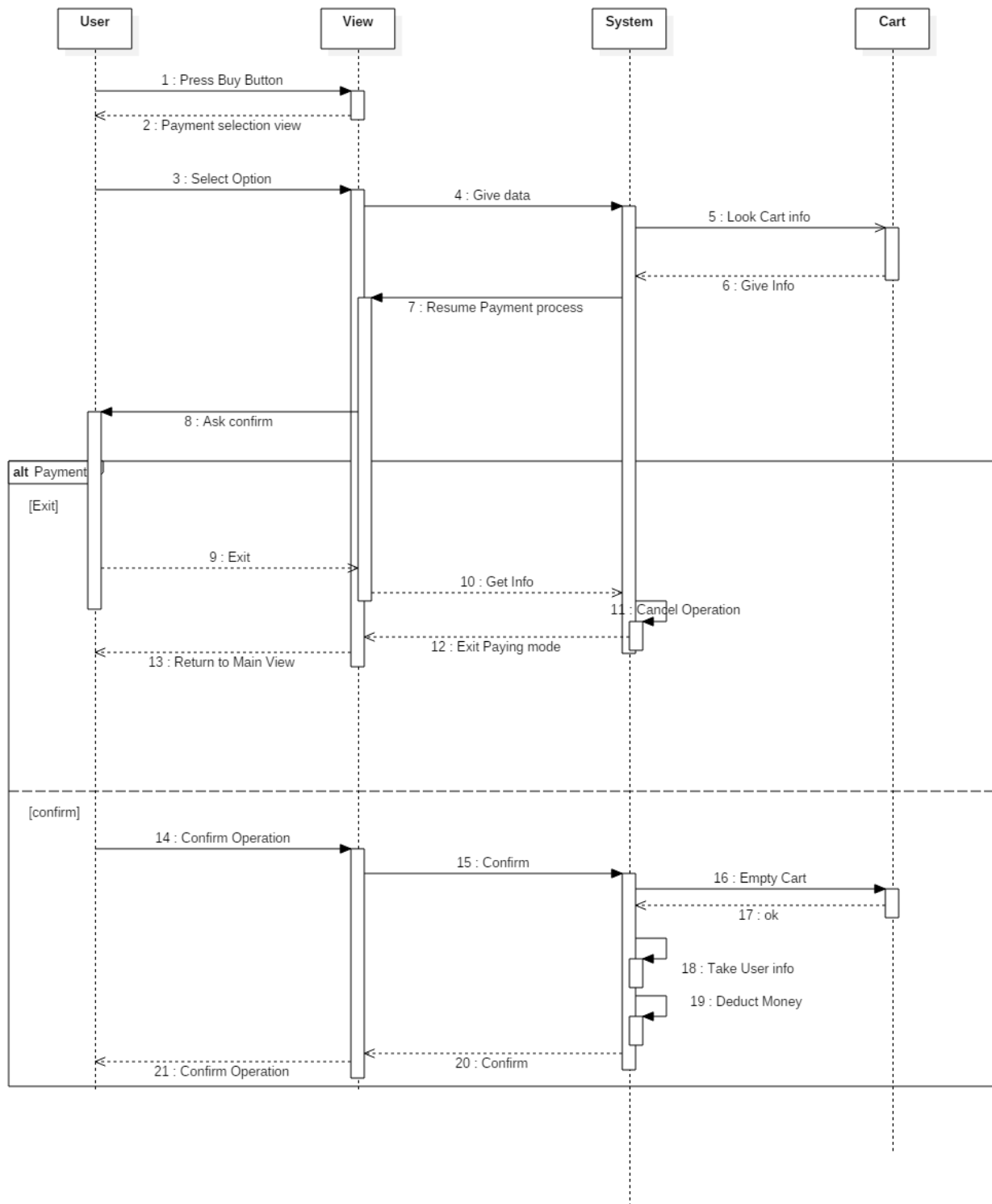
2.6 Acquista

Dopo aver selezionati i CD che si vogliono acquistare, averli aggiunti al carrello ed eventualmente averne visualizzato il contenuto, è possibile effettuare una transizione grazie al comando di acquisto. Non appena il cliente clicca sul comando gli vengono proposte due opzioni possibili, ovvero pagare il contanti alla consegna o il pagamento tramite carta prepagata.

Dopo aver selezionato un metodo di pagamento, il sistema offre all'utente di visualizzare un riassunto della transizione che presenta importo totale, CD acquistati e metodo di pagamento e rimane in attesa di una conferma.

In questo caso l'utente può scegliere di uscire dalla modalità di pagamento o di confermare. Nel primo caso il sistema cancellerà la transizione, nel secondo caso svuoterà il carrello dell'utente e completerà l'acquisto.

Casi d'uso: Buy
ID:UC9
Precondizioni: l'utente ha effettuato il login e aggiunto elementi al proprio carrello
Sequenza d'eventi: <ol style="list-style-type: none">1. L'utente vede il conto totale e visualizza le modalità di pagamento2. L'utente sceglie una modalità di pagamento<ol style="list-style-type: none">2.1 L'utente conferma2.2 L'utente esce
Postcondizioni: L'utente completa l'acquisto e <u>il</u> carrello dell'utente viene svuotato



3. Implementazione

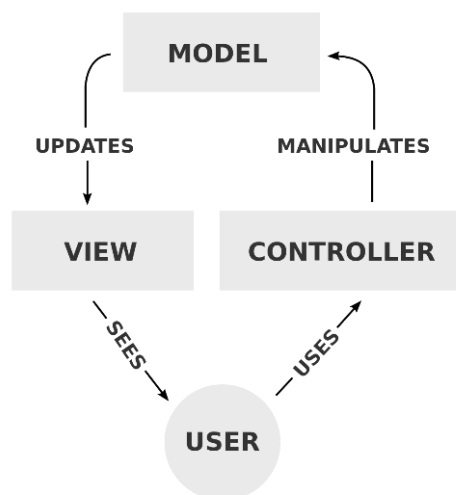
3.1 Pattern MVC

Per l'implementazione del progetto è stato utilizzato il Pattern MVC, che prevede l'utilizzo di tre tipi diversi di classi con funzionalità ben precise. MVC prevede l'interazione tra tre tipi di componenti.

La parte di Model, con il quale l'utente non avrà interazione, contiene tutte le informazioni relative ai dati degli utenti, CD e Catalogo. L'utente, in base alle sue azioni, potrà interagire con le parti relative a View e Controller, in quanto la View racchiude tutta la parte grafica del progetto (schermata iniziale, di ricerca, di visualizzazione del carrello etc..) e, come prevede il pattern, la sua schermata dipende dal contenuto di Model e viene aggiornata in base ad esso. Controller contiene tutte le componenti con cui l'utente può interagire, infatti qualsiasi evento causato dall'utente verrà intercettato dal Controller che grazie ai comandi Getters and Setters forniti da View e può modificare i dati del programma in base alle azioni eseguite.

Nell'architettura MVC la separazione tra Model, View e Controller non è solo concettuale, infatti prevede che le componenti siano codificate su file distinti.

L'utilizzo di MVC semplifica molto la parte di progettazione in quanto sin da subito, in base ai requisiti richiesti, si può decidere quali classi svolgeranno quali azioni e quale sarà la relazione tra queste. Per esempio sapendo di dover implementare un servizio di acquisti, subito sapremo che dovremo implementare una schermata per visualizzare il prodotto, un controller per aggiungerlo al carrello e un Model per inserire i dati di un determinato prodotto all'interno del carrello stesso, di conseguenza sapremo che per ogni requisito ci saranno almeno 3 componenti.



3.2 File XML

Vista la necessità di dover salvare le informazioni relative ai CD e ai Clienti, all'apertura del programma vengono analizzati due file di tipo XML. I tag dei vari file sono le variabili che poi verranno passate al costruttore delle classi Customer e CD, che successivamente verranno aggiunte alle classi contenenti il catalogo e la totalità degli utenti precedentemente registrati. In breve, i file XML inizializzano tutta la parte iniziale del programma relativa alle componenti Model del pattern spiegato precedentemente.

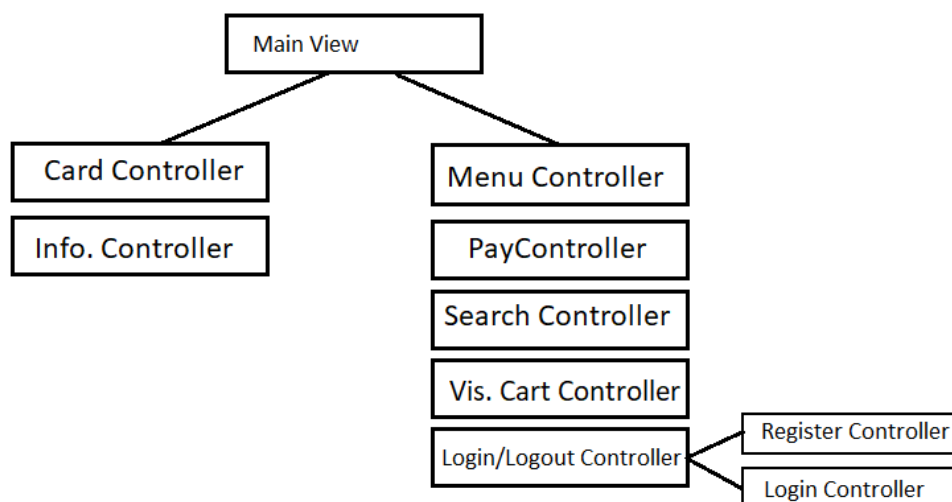
E' stato scelto XML rispetto alle altre possibili opzioni in quanto ci sembrava più semplice estrarre ed aggiungere informazione a questa tipologia di file.

3.3 Classi Utilizzate

Di seguito andremo a elencare le classi utilizzate, i diagrammi di classe e l'appartenenza al pattern MVC. Iniziamo elencando il rapporto tra le classi di visualizzazione e di controllo.

All'apertura del programma l'utente visualizza il contenuto della MainView, in cui viene visualizzato il contenuto del catalogo e le possibili interazioni col programma.

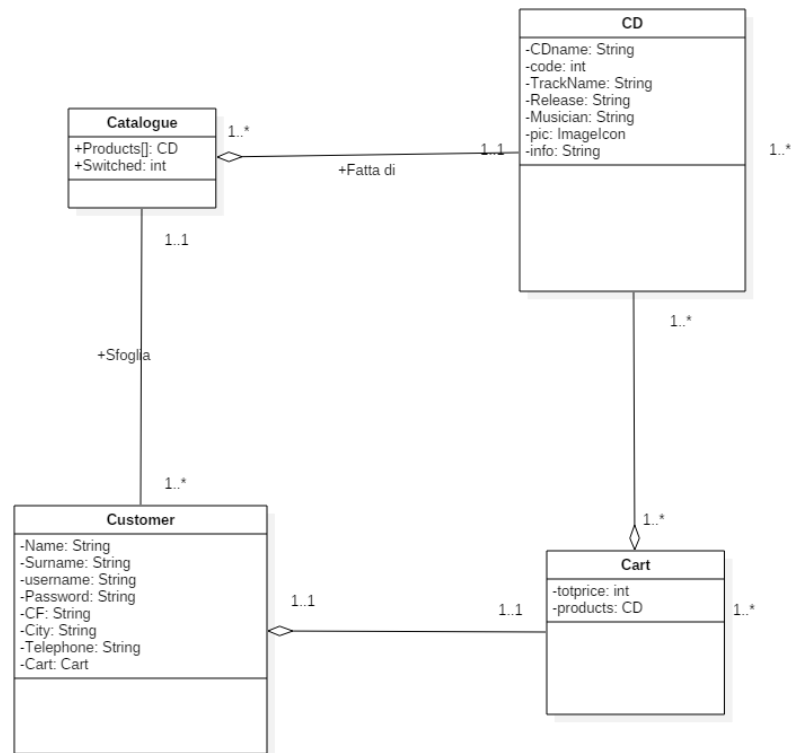
Dalla classe MainView è possibile richiamare tutti i controller per l'interazione, ovvero quelli per le funzioni richieste di Login, Visualizzazione Carrello, Ricerca e Acquisti.



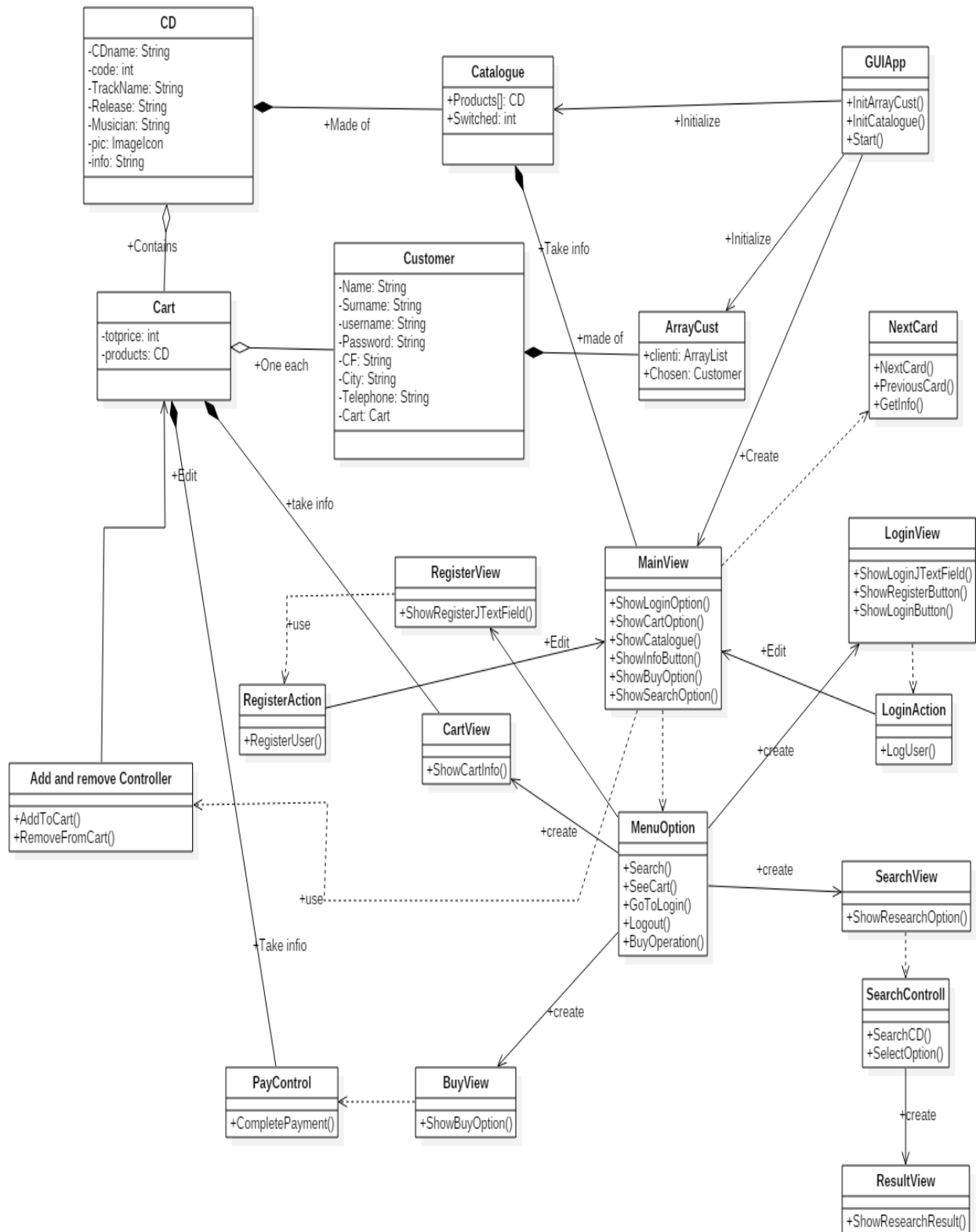
Come dal disegno, notiamo che nella parte relativa al CardController abbiamo che quest'ultimo ha la possibilità di richiamare il controller che gestisce la visualizzazione delle informazioni relative a un elemento del catalogo, mentre il Menu Controller contiene tutti i controller delle varie opzioni richieste dai requisiti del progetto.

Da precisare nel complesso di questo progetto, è che ogni controller dopo aver verificato o modificato degli elementi all'interno delle componenti Model crea una specifica finestra di View. Per esempio nel Search Controller, successivamente all'attivazione dell'evento verrà visualizzata una nuova finestra dove poter compilare i campi di ricerca. (Parte integrata meglio successivamente) .

Per quanto riguarda il package Model, esistono varie classi che lo compongono. Abbiamo 4 classi principali, ovvero Customer, CD, Catalogue e ArrayCustomer. All'avvio del programma infatti, la classe di partenza che legge il contenuto dei file XML crea delle istanze di classi CD che verranno poi inserite all'interno di Catalogue e delle istanze di classi Customer che verranno inserite all'interno di ArrayCustomer, che diventerà una lista contenente tutti i clienti precedentemente registrati nel sistema.

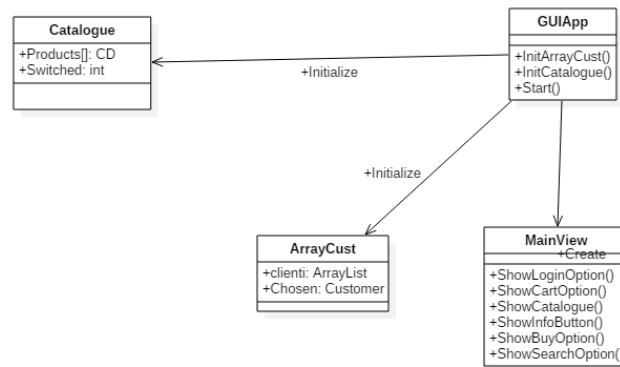


Come si può notare nel diagramma sopra elencato, per ogni cliente abbiamo un solo carrello, un Catalogo è formato da CD e un Cart può contenere dei CD (a relazione semplice). Per ultimo, c'è una relazione tra cliente e catalogo, in cui un cliente può vedere un solo catalogo, ma un catalogo può essere visto da più clienti.

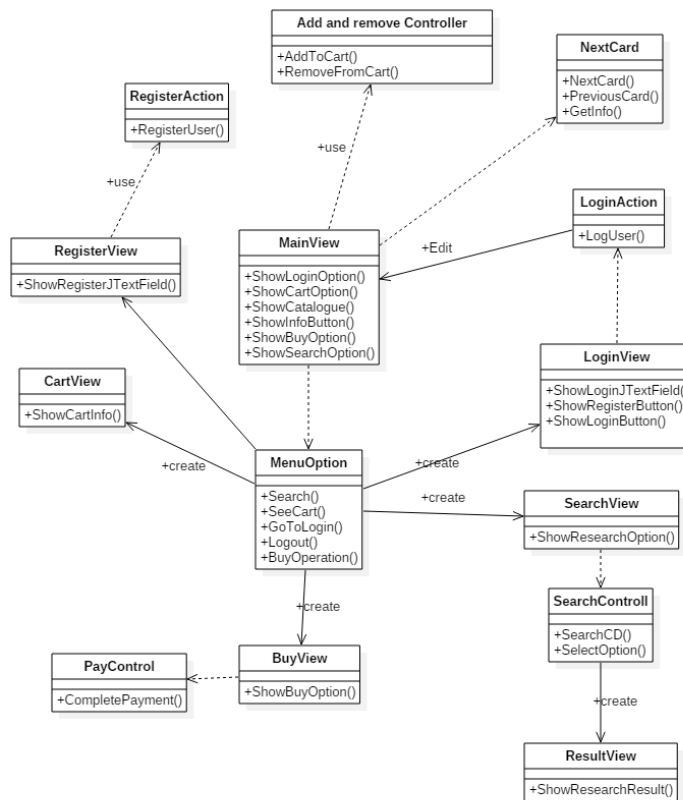


Class Diagram autogenerato alla fine del fascicolo.

Iniziamo un'analisi passo passo del Class Diagram precedentemente visualizzato.



La classe GUIApp è la classe iniziale, che inizializza l'array contenente gli utenti registrati, il catalogo (Leggendo le informazioni necessarie dai file XML, utilizzando i tag come costruttori delle nuove istanze delle classi) e crea la finestra iniziale.

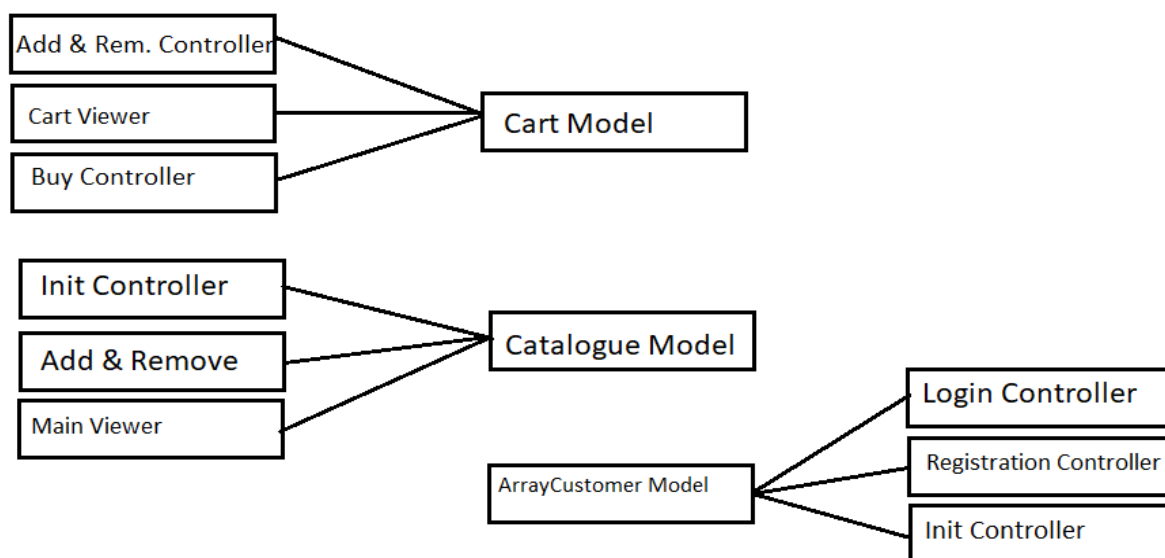


Dalla classe View MainView è possibile richiamare due tipi diversi di Controller, ovvero NextCard, che cambierà il CD visualizzato nel catalogo o visualizzerà le informazioni relative al CD (in base al tasto cliccato), o MenuOption, che in base al menù cliccato aprirà una delle possibili classi View in cui l'utente potrà inserire le informazioni dedicate al completamento di un'azione specifica. Ogni volta che vengono inserite delle informazioni, gli ActionListener vengono attivati effettuando controlli per produrre il risultato dell'operazione desiderata.

Le classi Model queste vengono modificate dai relativi Controller. Nel dettaglio, la classe Cart viene utilizzata da Cart Viewer per visualizzare il contenuto del carrello, mentre il controller di Add & Remove ne aggiunge o rimuove dei pezzi. Tutti gli elementi del carrello vengono eliminati successivamente all'utilizzo del BuyController, la cui esecuzione effettuerà un acquisto.

Sul Model del catalogo invece agiscono il controller iniziale, che lo inizializza, l'Add & Remove, che per ogni aggiunta al carrello rimuove un elemento al catalogo e viceversa e il Main Viewer che visualizzerà gli elementi in esso contenuto.

Per ultimo, l'ArrayCustomer, che contiene i profili utenti registrati, viene utilizzato come controllo dal Login Controller, che controlla se i dati inseriti da un User corrispondono a qualche utente precedentemente registrato. Questa classe Model viene inizializzata dalla stessa classe che inizializza il Catalogo e viene modificata ogni volta che Registration Controller aggiunge un nuovo utente al file XML.



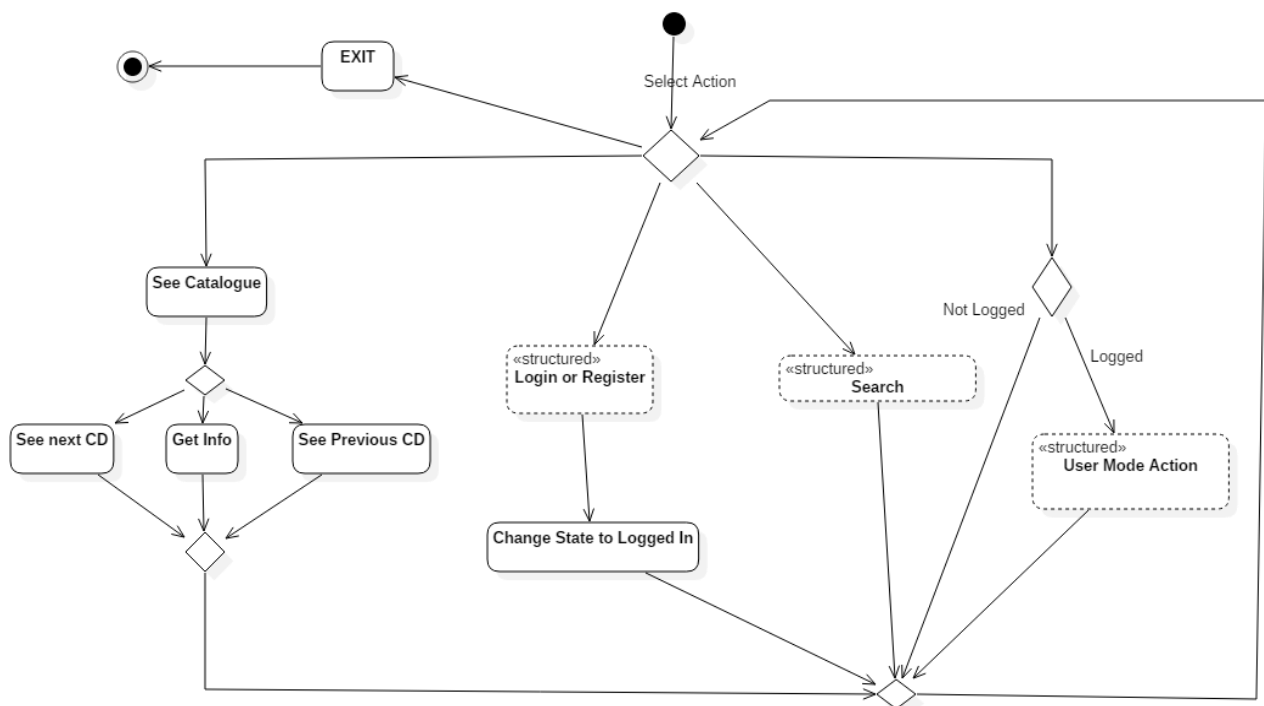
3.4 Nota: Observer

Un piccolo approfondimento va fatto anche sull'utilizzo degli ActionListener all'interno del programma. Gli ActionListener riprendono in parte l'utilizzo del Pattern Observer. Nell'Observer sono presenti due utenti principali, il Subject e l'Observer, in cui il Subject è l'oggetto osservato e l'Observer è l'osservatore. Il pattern si basa su uno o observer, che vengono registrati per gestire un evento che potrebbe essere generato dall'oggetto "osservato", infatti in questo Pattern è presente la molteplicità uno a molti. Nel progetto i Subject sono i componenti della View, mentre gli Observer le classi del package Controller, che sono in attesa di un evento scatenato dall'utente per attivarsi, ciò implica una relazione molti a molti. La relazione molti a molti è data dal fatto che una classe del Controller può essere collegata a più componenti e che in base al componente che esegue un evento il Controller agisca in maniera diversa, infatti ogni classe all'avvio del proprio metodo principale fa un controllo sulla sorgente dell'evento scatenato e grazie a questa analisi riesce a capire efficacemente quale azione intraprender per il conseguimento del programma.

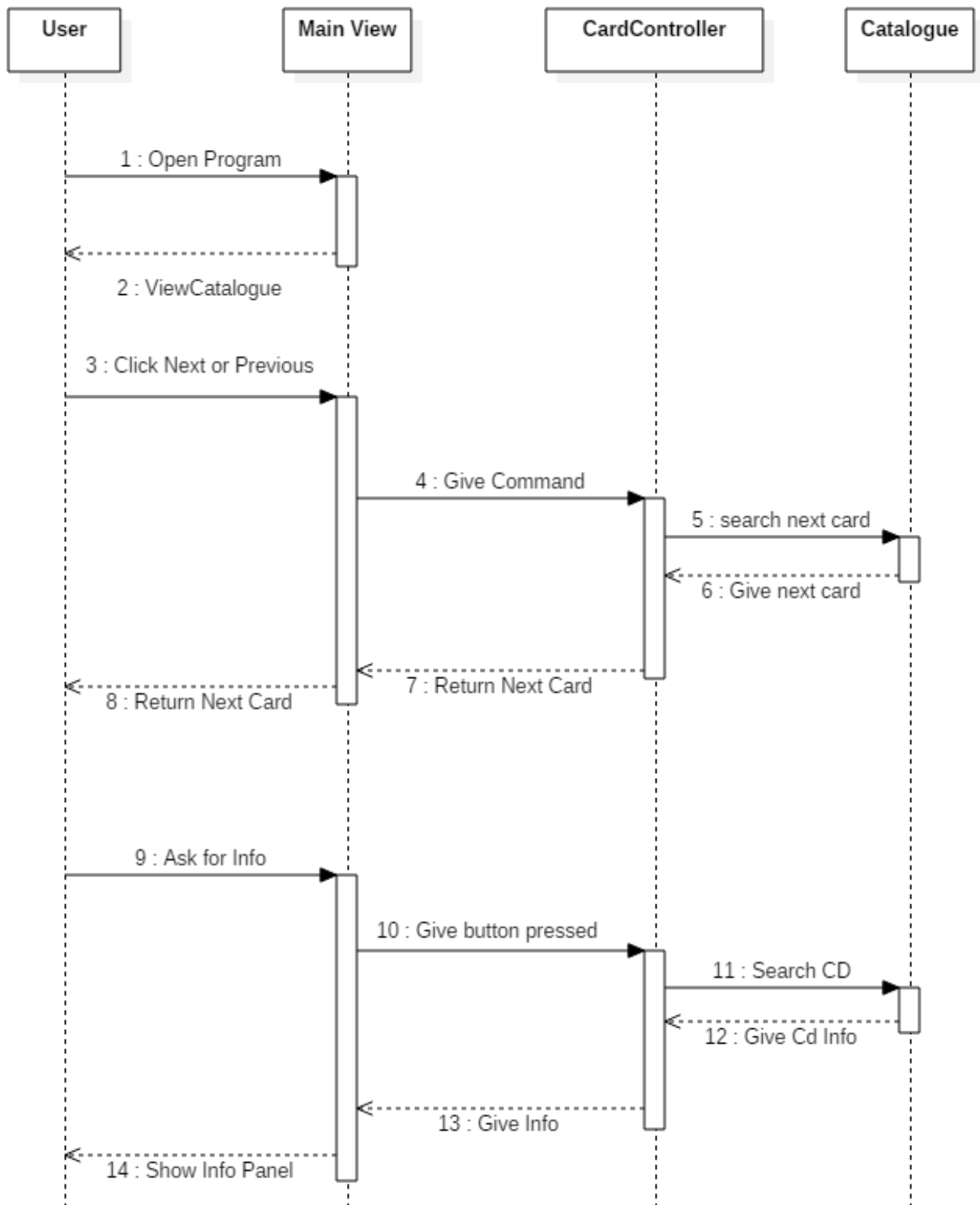
4. Activity & Sequence Diagrams

In seguito, grazie al supporto degli Activity e Sequence Diagrams andremo a vedere nel dettaglio il funzionamento del programma in tutte le sue sfaccettature.

4.1 Avvio del programma

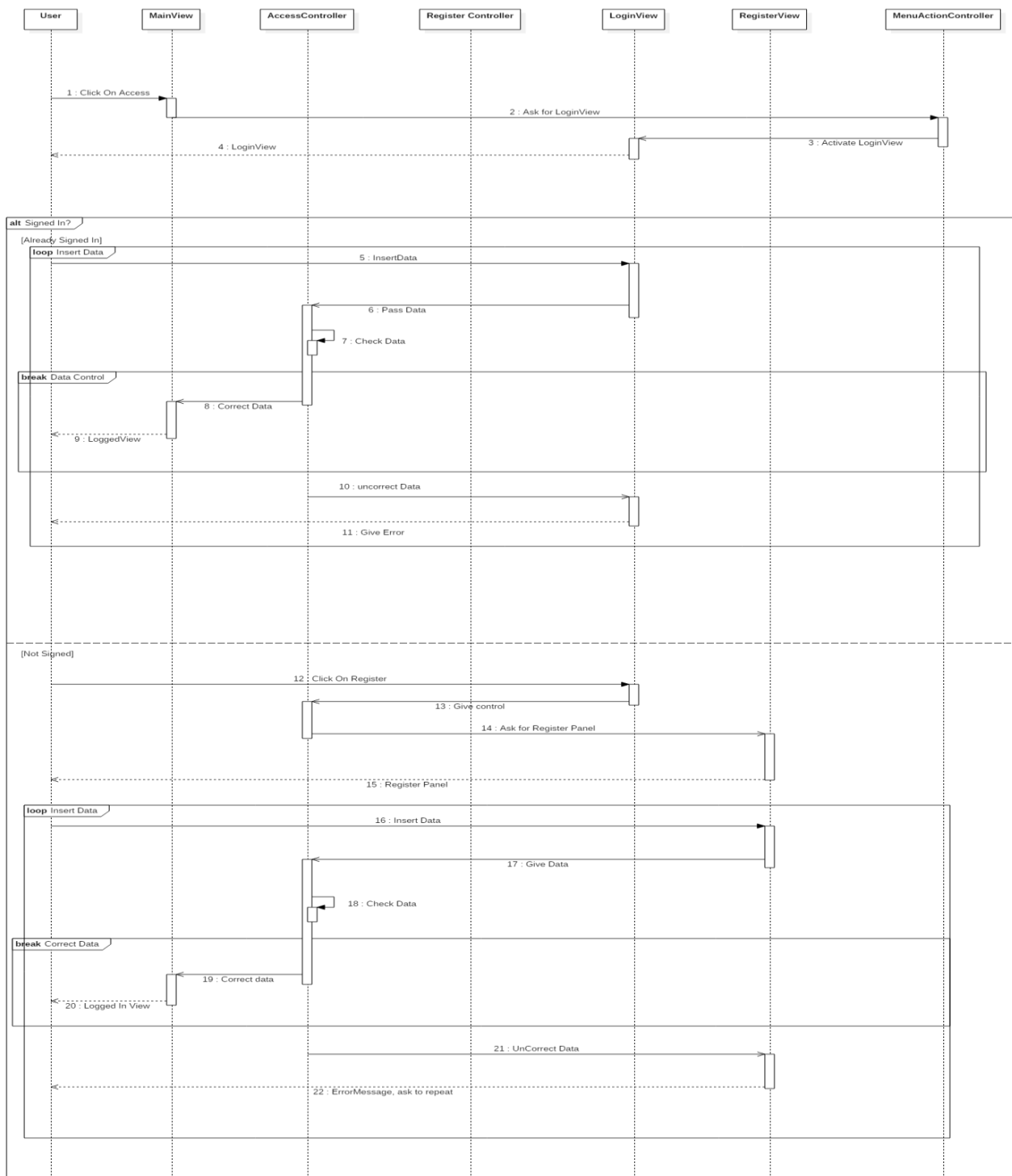


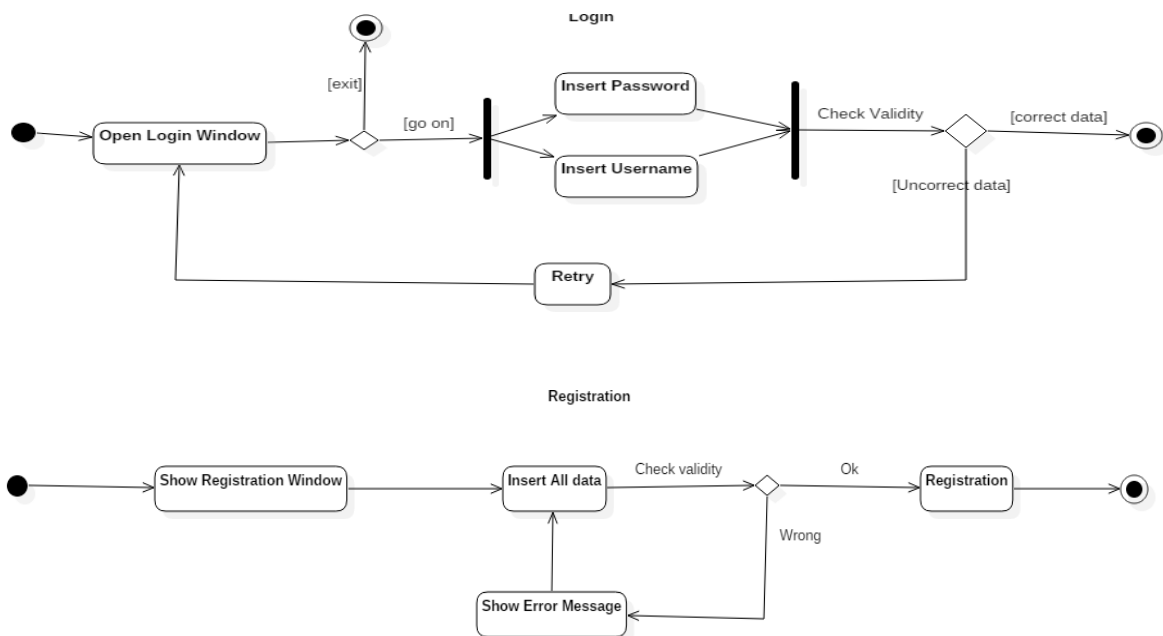
All'avvio del programma l'utente si ritrova nella schermata iniziale, in cui deve decidere che azione compiere, inizialmente può solamente effettuare un'operazione di ricerca, effettuare il login o vedere il catalogo. Vedere il catalogo comprende scegliere tra visualizzare le informazioni relative al CD visualizzato oppure visualizzarne un altro.



4.2 Login & Registrazione

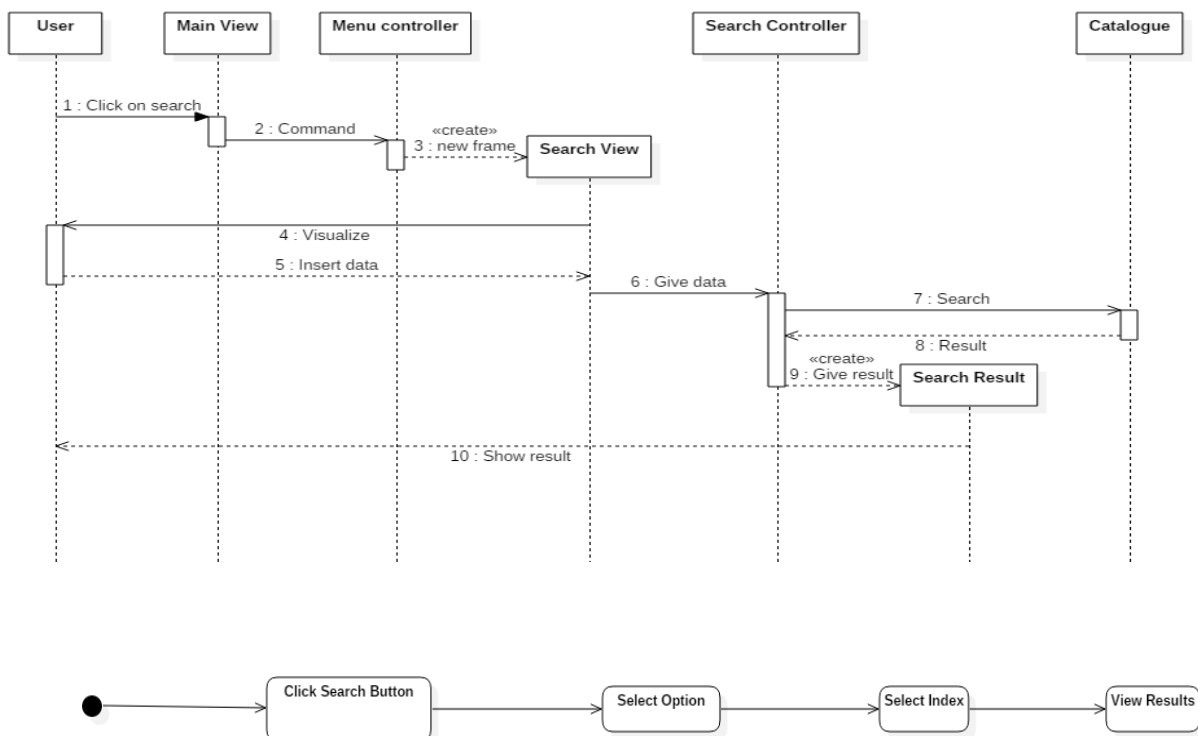
Per accedere ad opzioni aggiuntive ogni utente deve effettuare il Login, inserendo i propri username e password. Nel caso un utente non abbia un username e password personali è possibile effettuare la registrazione inserendo le proprie credenziali. Dopo l'inserimento dei dati viene effettuato un controllo (Nel caso della registrazione viene effettuata sulla correttezza dei dati inseriti, mentre nel login viene effettuato un paragone tra i dati inseriti e quelli presenti)





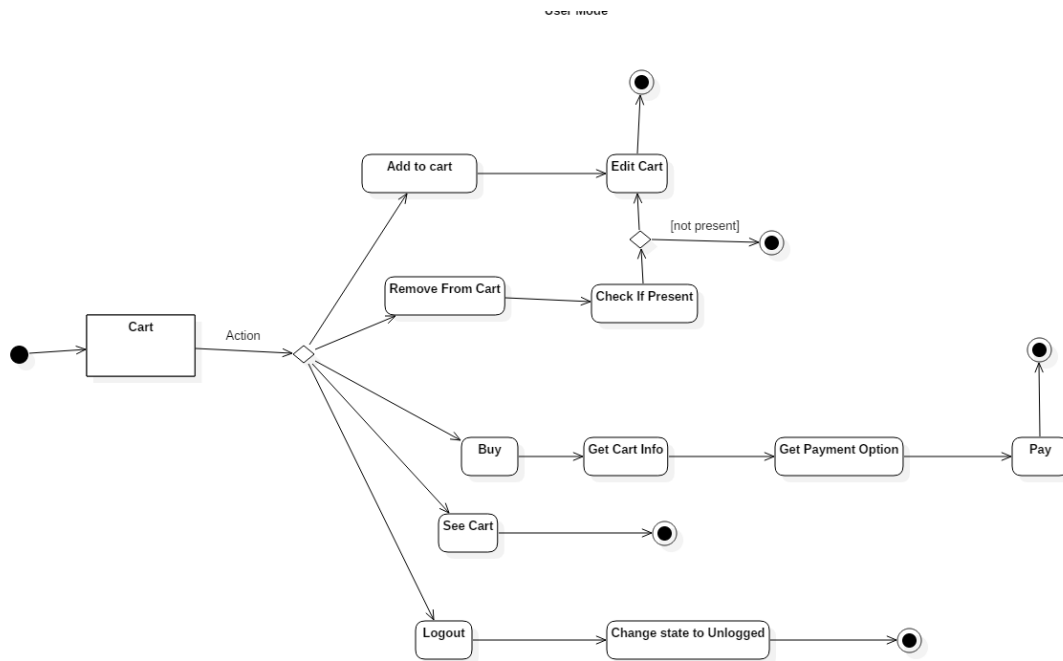
4.3 Search

Nell'opzione di ricerca, dopo aver selezionato il comando dedicato, il Menù controller richiama la finestra per inserire le opzioni di ricerca, una volta inserito secondo quali criteri si vuole ricercare un determinato CD il search controller cerca all'interno del catalogo tutti gli elementi che rispettano le caratteristiche inserite e le mostra all'utente in una nuova finestra.



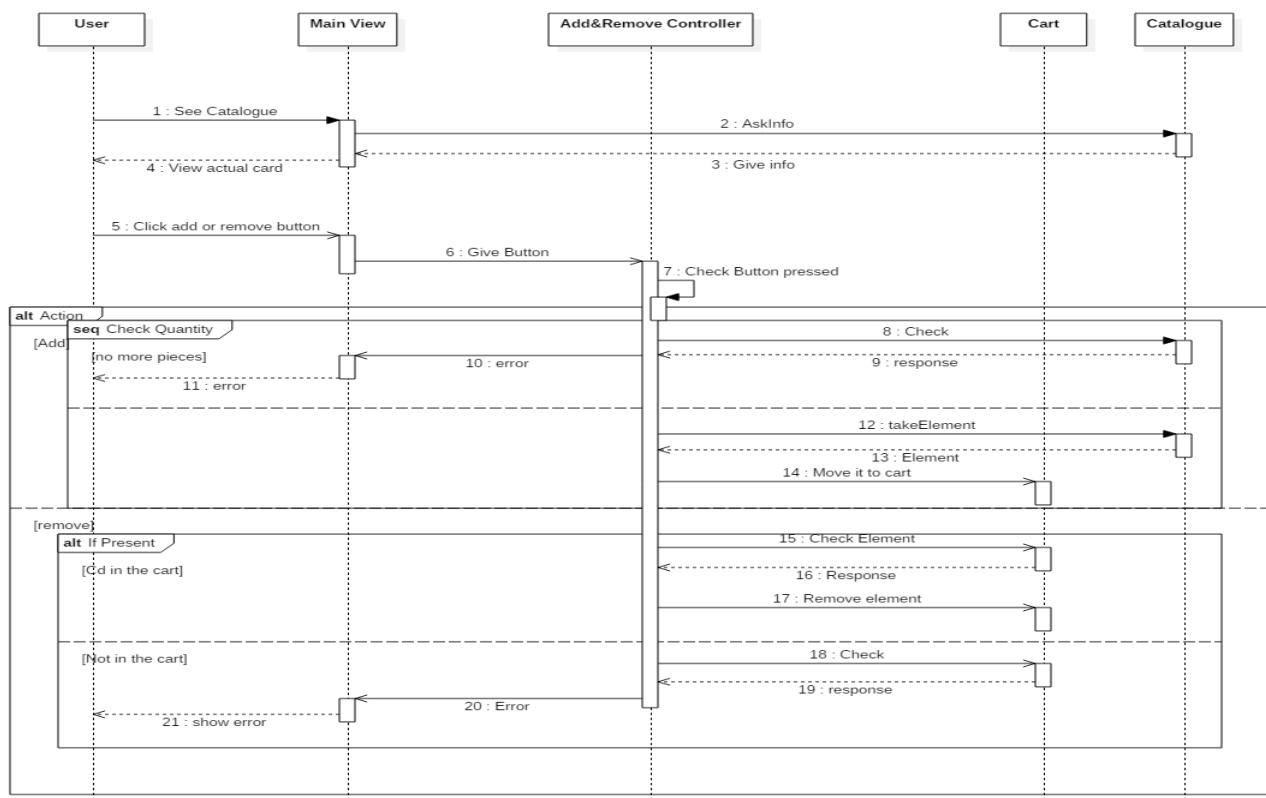
4.4 Attività Utente Registrato

Dopo aver effettuato il login, all'utente viene data la possibilità di effettuare nuove operazioni all'interno del programma, come detto in precedenza.

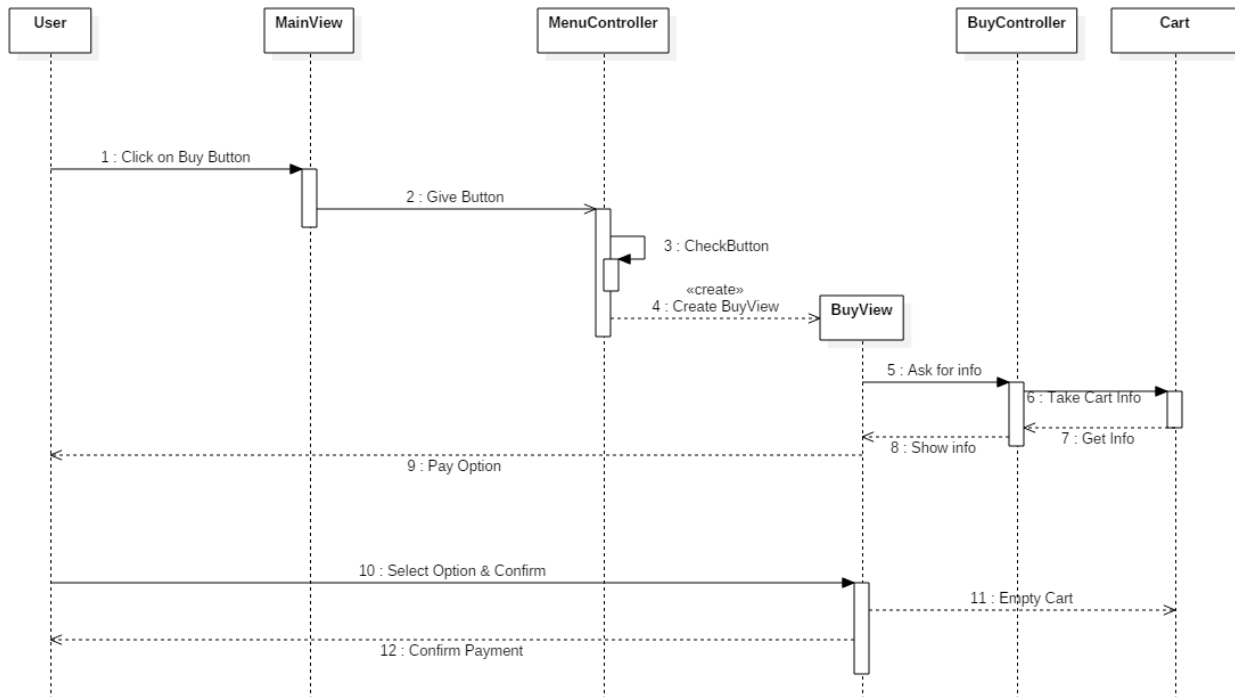


Di seguito, i sequence diagram dall'alto al basso:

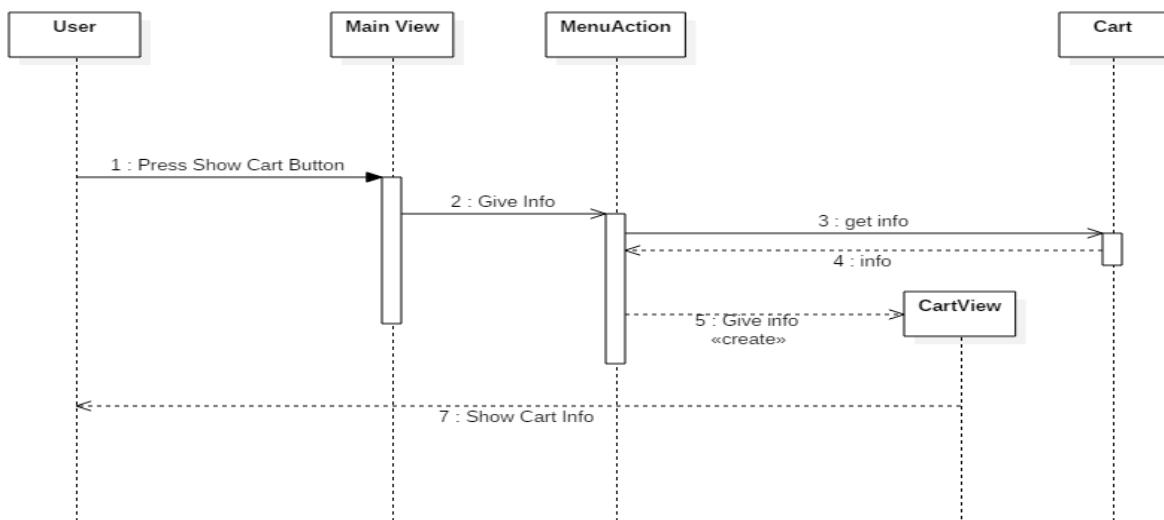
-Aggiungi & Rimuovi



- Acquista



- Visualizza Carrello



5. Fase di Testing

Per la fase di Testing è stato seguito un processo continuo che alternava l'implementazione delle componenti singole e il controllo sulla correttezza del funzionamento. Inizialmente, dopo aver letto e deciso quali parti del programma implementare, venivano scritte componenti del programma, partendo dalla View principale fino alle parti più piccole, per poi essere successivamente testate. Per ogni nuova componente venivano testate la corretta visualizzazione, funzionamento o accesso, in base al proprio package che poteva essere di tipo View, Controller o Model. Dopo aver creato la View principale sono stati controllati i collegamenti alle parti del Controller per assicurarsi che le chiamate alla funzione di ActionListener fossero eseguite nei modi desiderati.

Per ogni ActionListener veniva effettuato un controllo sul modo in cui accedeva alle variabili che doveva modificare, questo passaggio è stato effettuato in modo particolarmente accurato per la funzione iniziale, il cui compito era inizializzare le variabili di tipo Model che successivamente avrebbero influenzato la vista del programma.

Ogni test veniva fatto per assicurarsi che le componenti fossero visualizzate nel punto giusto e richiamassero la funzione desiderata per modificare le variabili nei modi concordati dopo la stesura delle specifiche.

Dopo essersi assicurati che ogni macro-funzione del programma (elencate nei paragrafi precedenti, ovvero ricerca, visualizzazione carrello, aggiungere al carrello etc..) funzionasse a dovere, è stato effettuato un test sul funzionamento del sistema in generale, ovvero un System Testing, per assicurarsi che le varie componenti non creassero conflitti tra di loro, per esempio modificando una parte del Model in due modi contrapposti, o che l'utilizzo di una funzione non compromettesse l'utilizzo di un'altra.

Una volta essersi assicurati del funzionamento complessivo del sistema, si è passato alla ricerca di eventuali bug di sistema provando a dare input volontariamente erronei per vedere come reagiva il sistema. Dopo aver fatto emergere dei nuovi bug e averli successivamente corretti si è giunti all'ultima fase, il test sugli utenti.

Come utenti di prova sono stati considerati due tipi di utenti:

- Utenti che erano già a conoscenza del progetto in generale, quindi a conoscenza di cosa doveva fare il programma in tutte le sue sfaccettature, con conoscenze informatiche e capaci di dare consigli su come migliorare l'implementazione da un punto di vista pratico (ovvero compagni di corso e/o università)
- Utenti totalmente esterni a cui è stata descritta la dinamica del programma e cosa doveva fare, in modo da capire come un nuovo utente si approcciava per la prima volta al prototipo realizzato. Questi tipi di utenti erano capaci di dare consigli più ad alto livello, in modo da poter migliorare l'approccio più esterno al programma.

Entrambi i punti di vista sono stati utili per migliorare il progetto nel suo complesso.

Dopo aver coperto tutte le possibili sfaccettature emerse dagli utenti di prova, il programma è stato considerato completo nell'implementazione delle specifiche indicate e pronto alla consegna.

6. Scelte Progettuali

Per l'implementazione del prototipo sono state scelte di effettuare le seguenti operazioni dal punto di vista dell'utente: visualizzazione catalogo, aggiunta e rimozione elementi dal carrello, visualizzazione carrello, ricerca elementi in base a specifiche, possibilità di login e logout, effettuare acquisti, visualizzare informazioni relative ad ogni elemento.

Per quanto riguarda la visualizzazione del catalogo, è stato utilizzato un Layout di tipo CardLayout in quanto grazie al suo lato estetico permetteva di scorrere le foto dei vari CD in modo sequenziale così che chiunque la utilizzasse non si ritrovasse in una schermata iniziale piena di titoli confusionari, ma si concentrasse su un elemento per volta. Per tenere conto degli spostamenti effettuati nel catalogo, è stato utilizzato un contatore che aumentava e diminuiva ogni volta che venivano premuti i tasti "NEXT" e "PREV".

Le informazioni relative a ogni elemento sono disponibili grazie al tasto "INFO" per cui non è stata implementata una finestra di alcun tipo, ma è stato usato una finestra di MatDialog con tutte le informazioni che era necessario mostrare.

Per quanto riguarda la ricerca, bisogna precisare che è stato deciso di mostrare i risultati in una finestra a scorrimento separata, in modo che l'utente che la utilizzi sappia solo quali sono i risultati in base ai parametri scelti, ma che non possa fare azioni sugli elementi visualizzati. Nel caso in cui l'utente voglia aggiungere elementi al proprio carrello basterà cercarli nella visualizzazione catalogo e aggiungerli.

L'alternativa per fare il modo che dopo la ricerca si potesse effettuare delle azioni sui risultati, sarebbe stata ricreare una finestra del tipo MainView visualizzando solo i risultati della ricerca invece della totalità del catalogo, azzerando il contatore che indicava gli spostamenti. Inizialmente, dopo aver implementato entrambe le alternative, è stato deciso di tenere la prima, ovvero quella con la finestra a scorrimento, in quanto sembrava l'ipotesi più elegante (L'altra opzione descritta in precedenza è stata tenuta come parte commentata all'interno del codice).

Per quanto riguarda la parte dell'acquisto è stata implementata in modo semplicistico, ovvero è stata data la possibilità di scegliere la modalità di consegna tramite dei checkbox mentre si visualizzava l'importo totale da pagare. E' stato inserito a solo scopo di completezza del programma, in quanto è stato ritenuto che un programma per acquisti dovesse implementare il tasto "BUY", anche se questa funzione non voleva essere tra le parti approfondite del programma. Il tasto BUY infatti, si limita solo a svuotare il contenuto del carrello, senza controllare i dati bancari del cliente o un possibile importo su una carta di credito.