



Progetto 10

FlySMM: *Applicativo di gestione dei biglietti di compagnia aerea*

Ingegneria del software A.A 2016/17

Andrea Biaggi 794873
Davide Bolzoni 797684
Gabriel Dos Santos 797785
Luca Lorusso 797805
Cesar Macias 793390

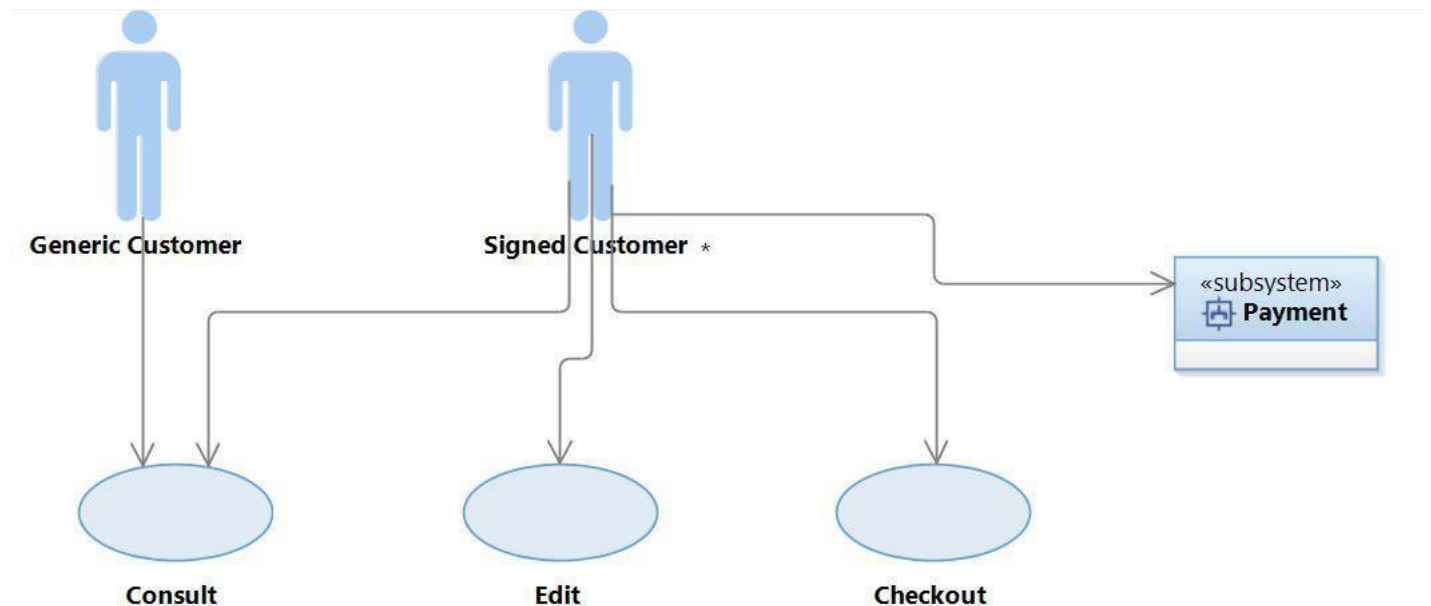
FlySMM è una web application che permette la prenotazione e l'acquisto di biglietti aerei. Ha quattro funzionalità principali: l'acquisto dei biglietti, la prenotazione, la gestione dei clienti e la gestione delle promozioni.

I casi d'uso sono stati progettati e verranno presentati divisi per funzionalità, allo scopo di una più chiara rappresentazione dei diagrammi.

Casi d'uso:

Prima funzionalità: sezione Sale

Il sottosistema di pagamento è stato progettato con uno stub che simula il comportamento classico di un gateway di pagamento. Come riferimento è stato realizzato il sistema di PayPal e l'implementazione è stata commentata.



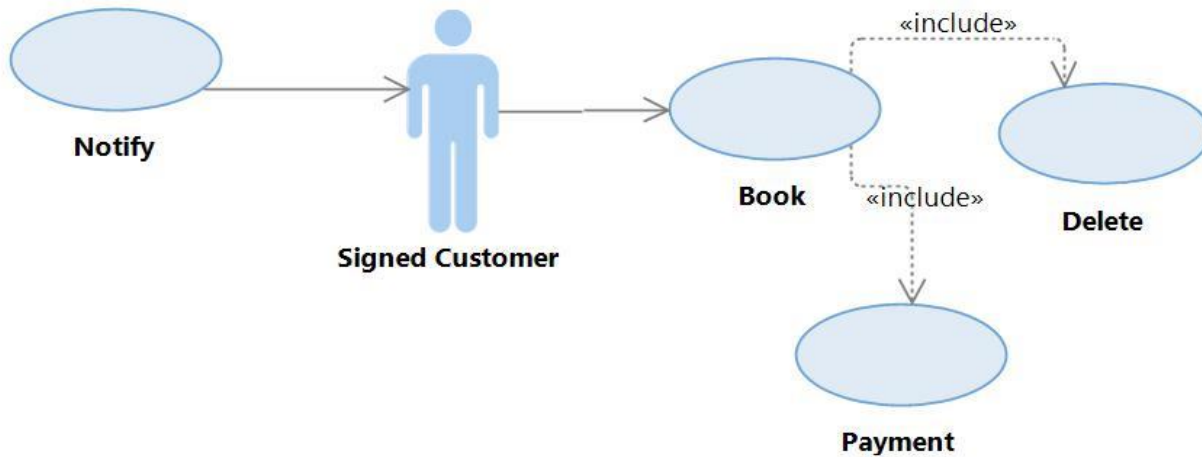
Sezione caso d'uso	Sale
Nome del caso d'uso	Consult
Portata	FlySMM
Livello	Obiettivo Utente
Attore primario	Customer/Signed Customer
Parti interessate e interessi	Prenotazioni ha l'interesse di ricevere le informazioni necessarie per procedere con la prenotazione
Precondizioni	-
Garanzie di successo	Bisogna avere a disposizione le informazioni dei voli scelti dall'utente come codice del volo, prezzo, numero di posti da prenotare
Scenario di successo primario	Il Customer/Signed Customer accede alla homepage del sistema FlySMM. Il sistema deve fornire all'utente la possibilità di inserire le località di partenza e di arrivo per il volo desiderato. Il sistema deve essere in grado di reperire i voli disponibile per la tratta scelta dall'utente e consentire all'utente di sceglierne una. Una volta confermata la scelta del volo desiderato il sistema procederà con la prenotazione del volo.
Estensioni	Non sono disponibili voli con i parametri inseriti dall'utente: Il Customer/Signed Customer accede alla homepage del sistema FlySMM il sistema deve fornire all'utente la possibilità di inserire le località di partenza e di arrivo per il volo desiderato, il sistema deve essere in grado di reperire i voli disponibile per la tratta scelta dall'utente non sono disponibili voli per i parametri inseriti il sistema: Il sistema cerca delle alternative cambiando data e ora avvisando l'utente delle modifiche apportate. Se non sono possibili alternative il sistema avvisa l'utente della indisponibilità del volo.
Requisiti speciali	Il sistema deve fornire i risultati della ricerca entro 1 minuto dalla richiesta.
Elenco delle variabili tecnologiche e dei dati	-
Frequenza di ripetizione	Elevata
Varie	-

Sezione caso d'uso	Sale
Nome del caso d'uso	Checkout
Portata	FlySMM
Livello	Obiettivo Utente
Attore primario	Signed Customer
Parti interessate e interessi	Customer perchè può fare anche lui checkout dopo essersi loggato
Precondizioni	L'utente deve essere registrato. L'utente deve essere loggato. Deve esistere una prenotazione a titolo dell'utente registrato.
Garanzie di successo	La prenotazione è stata registrata. Il pagamento è stato effettuato.
Scenario di successo primario	L'utente registrato arriva alla pagina per il checkout. Il sistema richiede le informazioni relative al pagamento. Il sistema le registra e le invia al sottosistema dei pagamenti per essere processate. Il sottosistema dei pagamenti restituisce l'ID della transazione e il codice di avvenuto pagamento. Il sistema comunica il successo della transazione.
Estensioni	Il sottosistema dei pagamenti rifiuta la transazione: Viene notificato all'utente il rifiuto della transazione insieme al codice dell'errore.
Requisiti speciali	-
Elenco delle variabili tecnologiche e dei dati	-
Frequenza di ripetizione	Media
Varie	-

Caso d'uso: Edit

L'utente deve autenticarsi con i suoi dati personali e il codice della prenotazione da modificare. Il sistema verifica i dati e presenta il dettaglio della prenotazione e il menù con le azioni che è possibile effettuare. L'utente sceglie il tipo di modifica da effettuare. Il sistema visualizza il form adeguato per la modifica della prenotazione. L'utente invia i nuovi dati. Il sistema stampa un report della modifica e l'eventuale importo dovuto per la modifica.

Seconda funzionalità: sezione Booking

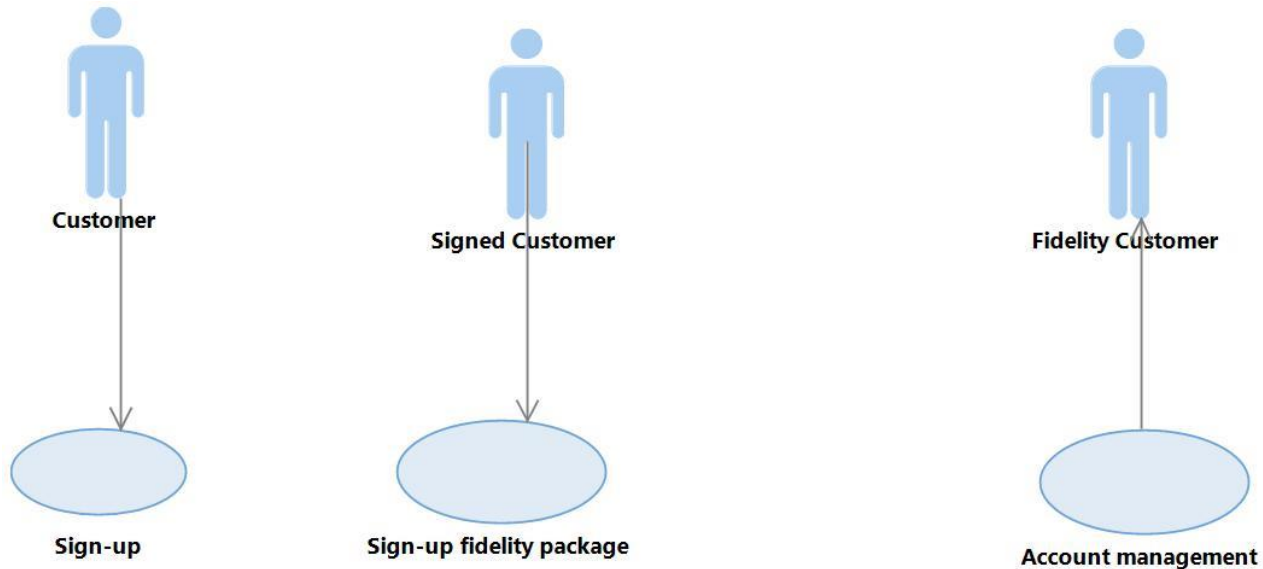


Caso d'uso Notify:

Scenario principale di successo: Il sistema avvisa il cliente, 24 ore prima, che la sua prenotazione sta per scadere e lo invita a pagare.

Sezione caso d'uso	Book
Nome del caso d'uso	Book
Portata	FlySMM
Livello	Obiettivo Utente
Attore primario	Signed Customer
Parti interessate e interessi	Signed Customer perchè si può fare una prenotazione solo se si è registrati/loggati
Precondizioni	L'utente deve essere registrato. L'utente deve essere loggato.
Garanzie di successo	La prenotazione è stata registrata.
Scenario di successo primario	L'utente registrato seleziona il/i volo/i. L'utente registrato inserisce i dati dei passeggeri e eventualmente sceglie il bagaglio. L'utente conferma i dati. Viene effettuata prenotazione.
Estensioni	Se la prenotazione viene effettuata meno di tre giorni prima della data di partenza del volo, il sistema deve obbligare l'utente a pagare subito.
Requisiti speciali	-
Elenco delle variabili tecnologiche e dei dati	-
Frequenza di ripetizione	Media
Varie	-

Terza funzionalità: sezione Customer



Caso d'uso Sign-up:

Sign up l'utente può procedere alla registrazione in qualsiasi momento purché avvenga prima della prenotazione del volo. l'utente inserisce i dati necessari per la registrazione (nome, cognome, indirizzo di residenza, data di nascita, e-mail e password) il sistema elabora i dati e controlla che tutti i campi obbligatori siano stati rispettati, se l'esito è positivo l'utente viene registrato nel DB e può procedere con la prenotazione/acquisto biglietto.

Caso d'uso Sign-up fidelity package:

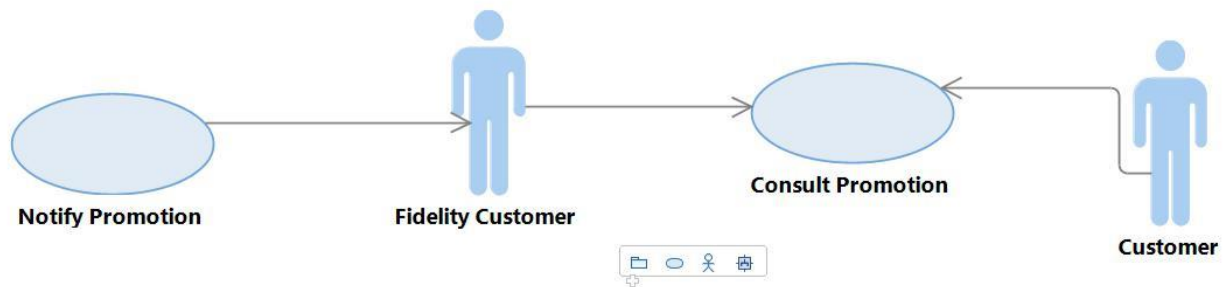
Sign up fidelity package il cliente già registrato può sottoscrivere il pacchetto fidelity. La seguente posizione prevede alcuni vantaggi, possibilità di accumulare punti in base alle miglia percorse e essere aggiornato tramite email sulle promozioni. Se un utente fidelity non acquista un biglietto aereo per più di due anni viene inserito in uno stato di "Infidelity". Viene cancellato dal programma se dopo un anno che è entrato nello stato "infidelity" non acquista nessun biglietto.

Caso d'uso Account Management:

Account Management la registrazione al pacchetto fidelity prevede dei vantaggi come la notifica di promozioni e l'accumulo di punti personali. Il sistema invia all'utente delle email per consultare le varie promozioni in riserbo per lui. I punti sono accumulabili tramite i viaggi stessi e vanno in base alle miglia percorse 1 punto = 1 miglio, il sistema incrementa automaticamente il punteggio.

L'utente in qualsiasi momento può ottenere/consultare l'estratto conto con i suoi punti e il dettaglio dei voli con le miglia.

Quarta funzionalità: sezione Promotion



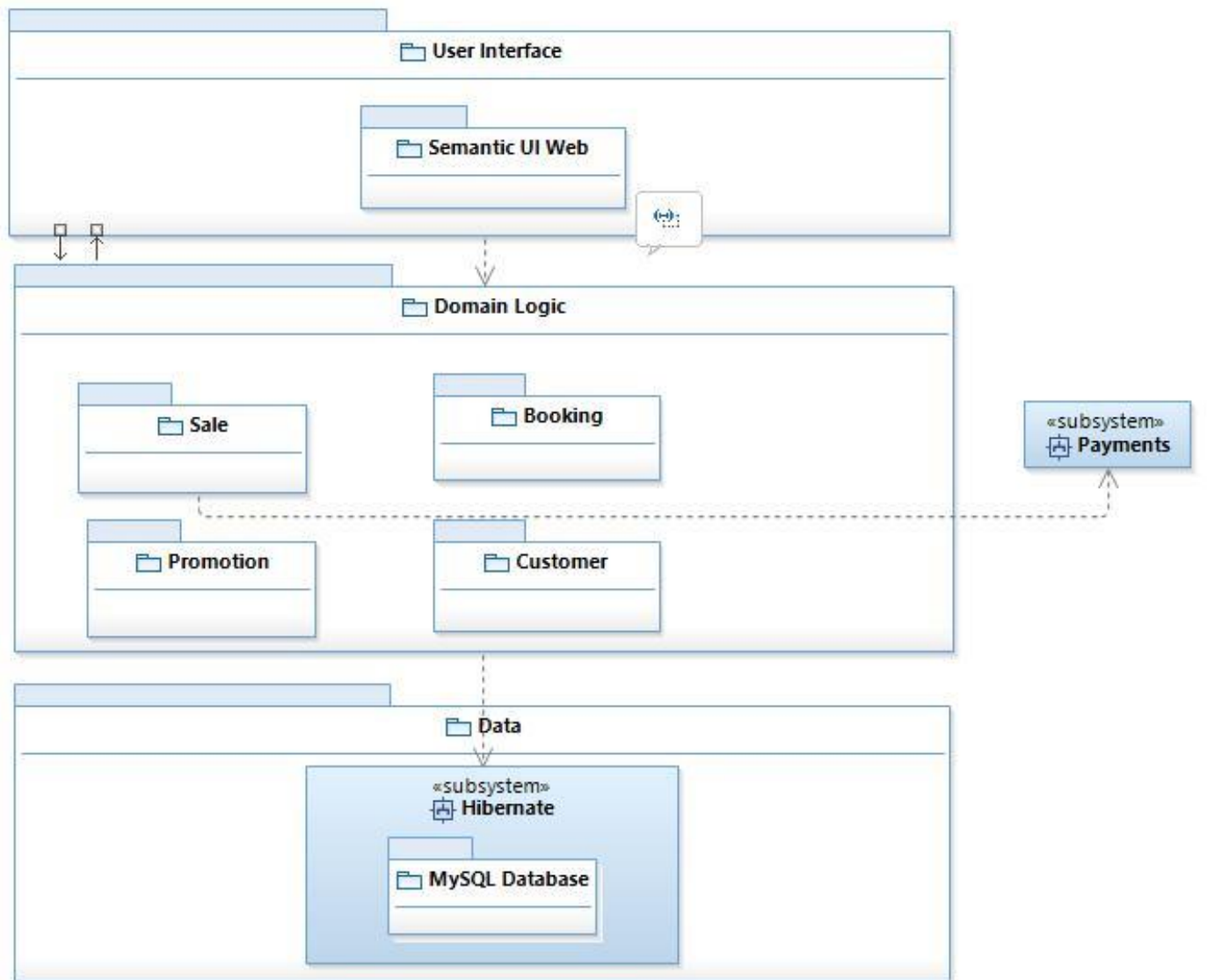
Caso d'uso Notify promotion:

Quando una nuova promo viene creata gli utenti fidelity ricevono una mail di avviso.

Caso d'uso Consult promotion:

All'interno della homepage gli utenti possono visualizzare le loro promozioni, in base al tipo di utente alcune promo verranno visualizzate o meno.

Diagramma dell'architettura logica:



La scelta dell'architettura logica è ricaduta su un modello a tre strati, perché consente di modulare interfacce ben definite e garantisce un'applicazione flessibile e facilmente manutenibile, sia per l'aggiunta che per la sostituzione di uno strato.

Diagramma Classi dominio:

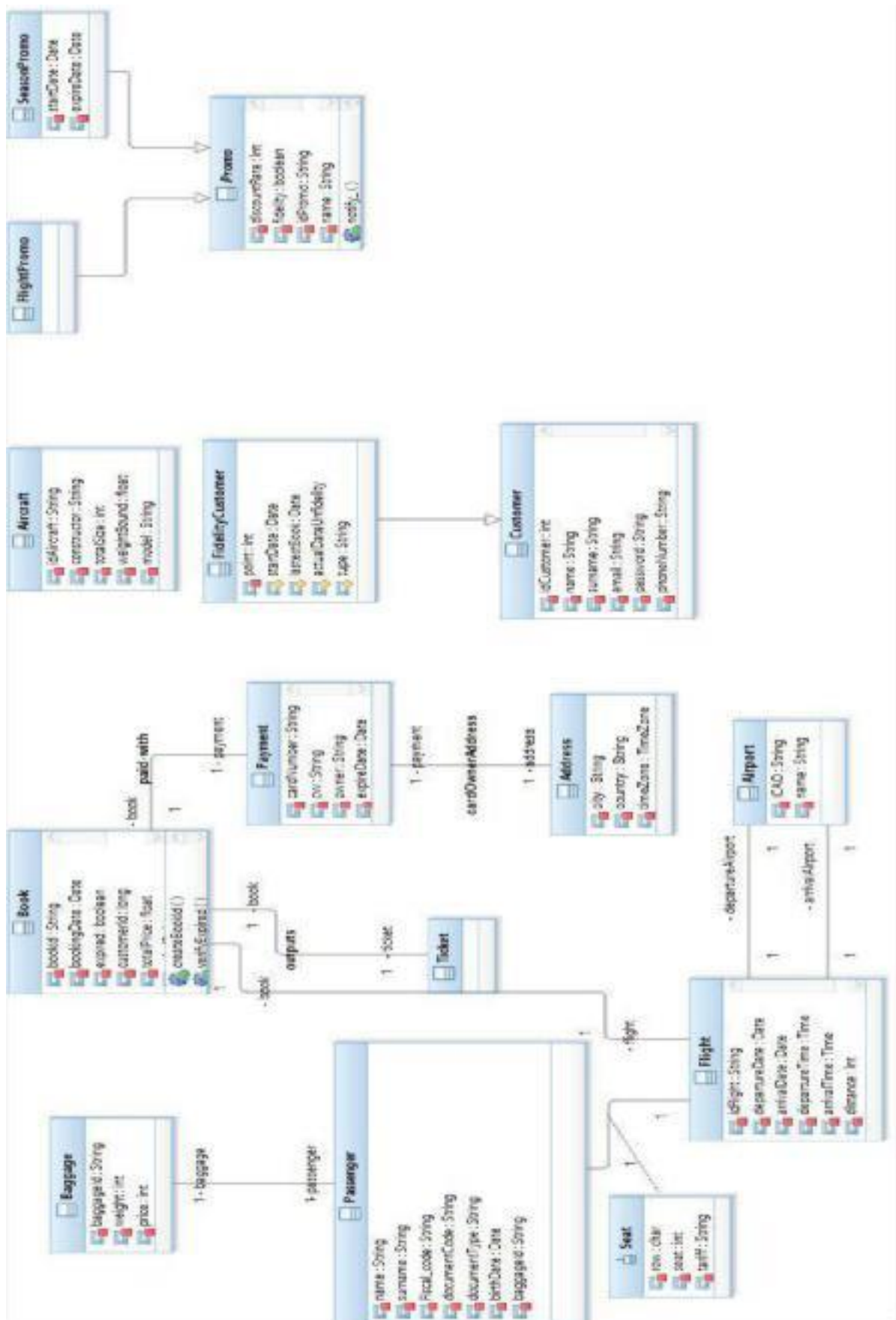
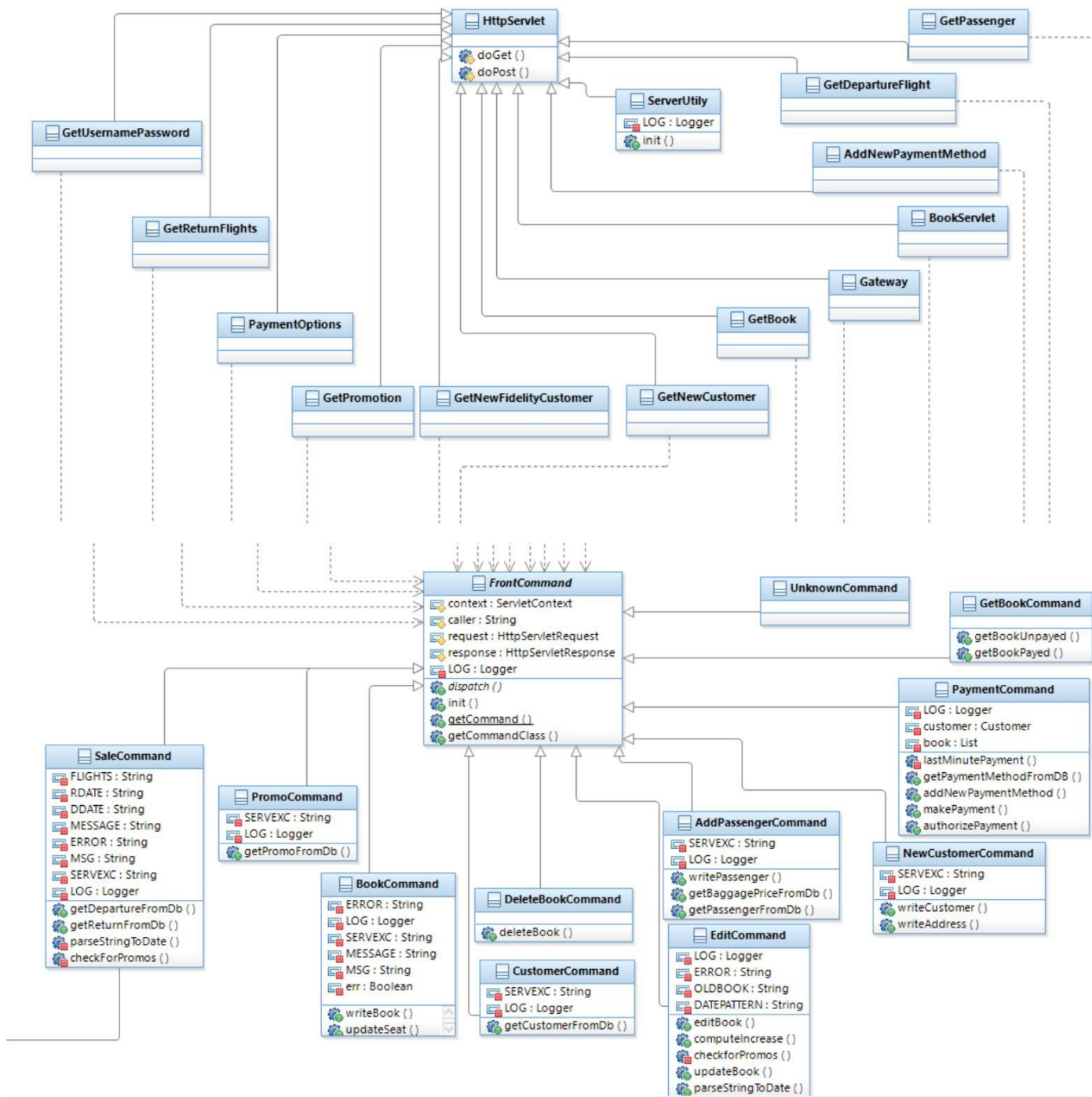


Diagramma delle classi di progetto:



La notifica periodica delle promozioni è effettuata all'interno dei costruttori, ogni qualvolta una nuova promozione viene istanziata, gli utenti fidelity ricevono una mail dal metodo `notify()`. Abbiamo deciso di usare il pattern architetturale **Front Controller**, anziché altri quali l'Application Controller poiché quest'ultimo sarebbe stato preferibile nel caso avessimo avuto diversi cambiamenti di stato per una risorsa, che non è il nostro caso, inoltre il Front Controller ha un'implementazione più semplice e lineare. Il design pattern **State** invece è stato scelto in quanto un Customer può entrare in due stati diversi: Fidelity State e Unfidelity State. Il cambiamento di stato è condizionato nelle specifiche della consegna. In ultimo, per ottimizzare le interazioni con il Database è stato scelto di utilizzare il pattern **Singleton** per l'apertura di una sessione, vista la costosità in termini di tempo, così da evitare che ad ogni modifica del database se ne apra una nuova.

Diagrammi di sequenza:

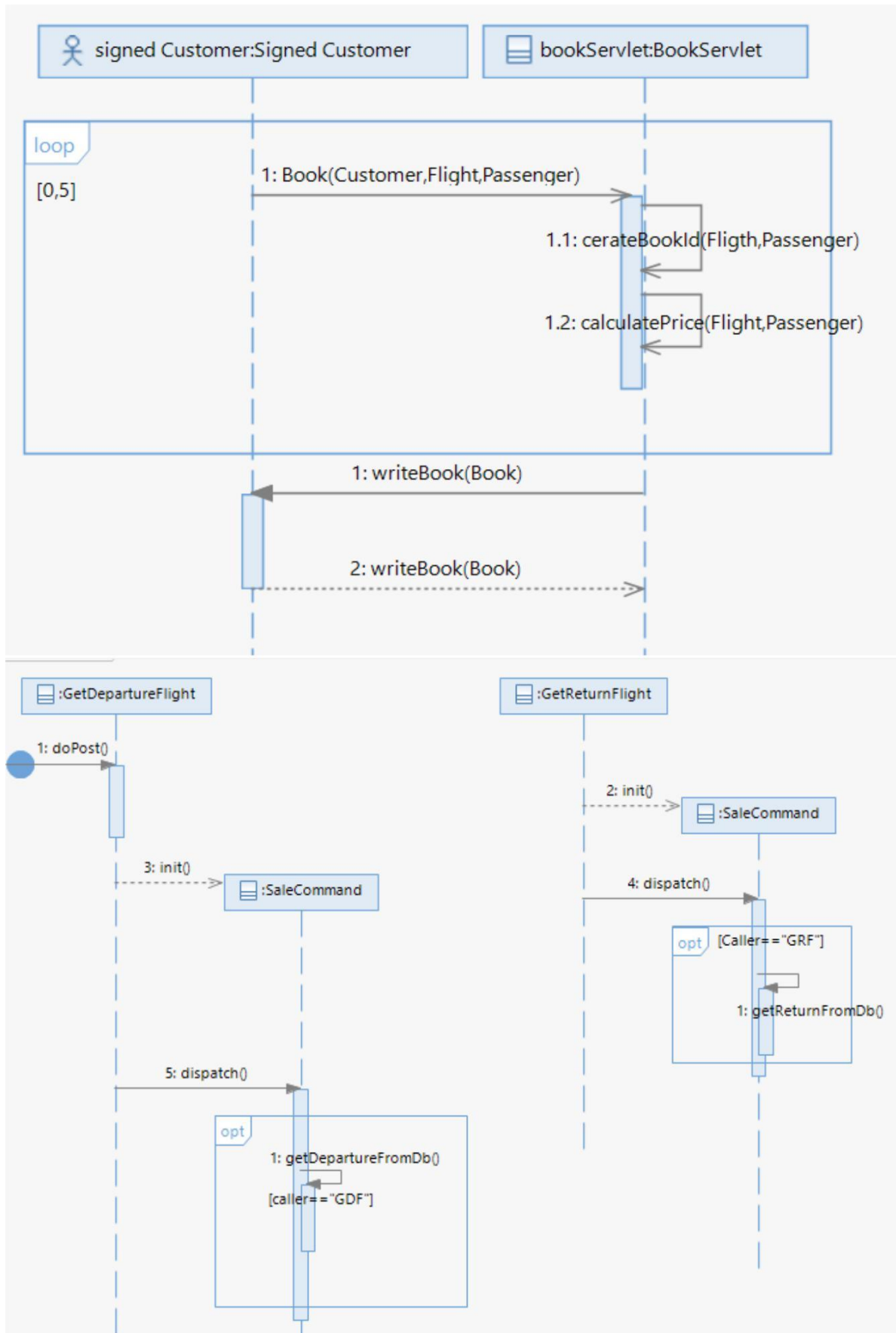
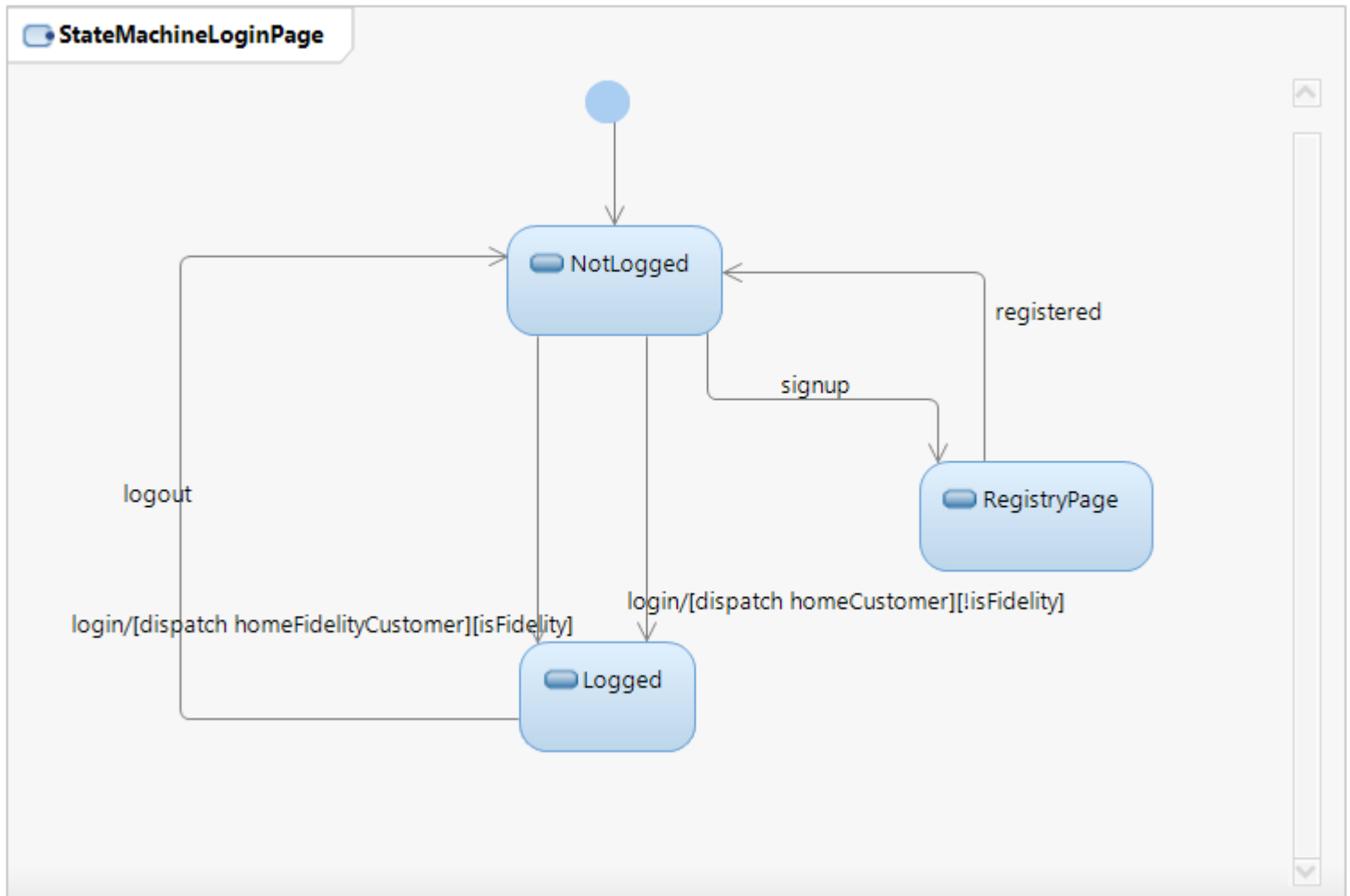
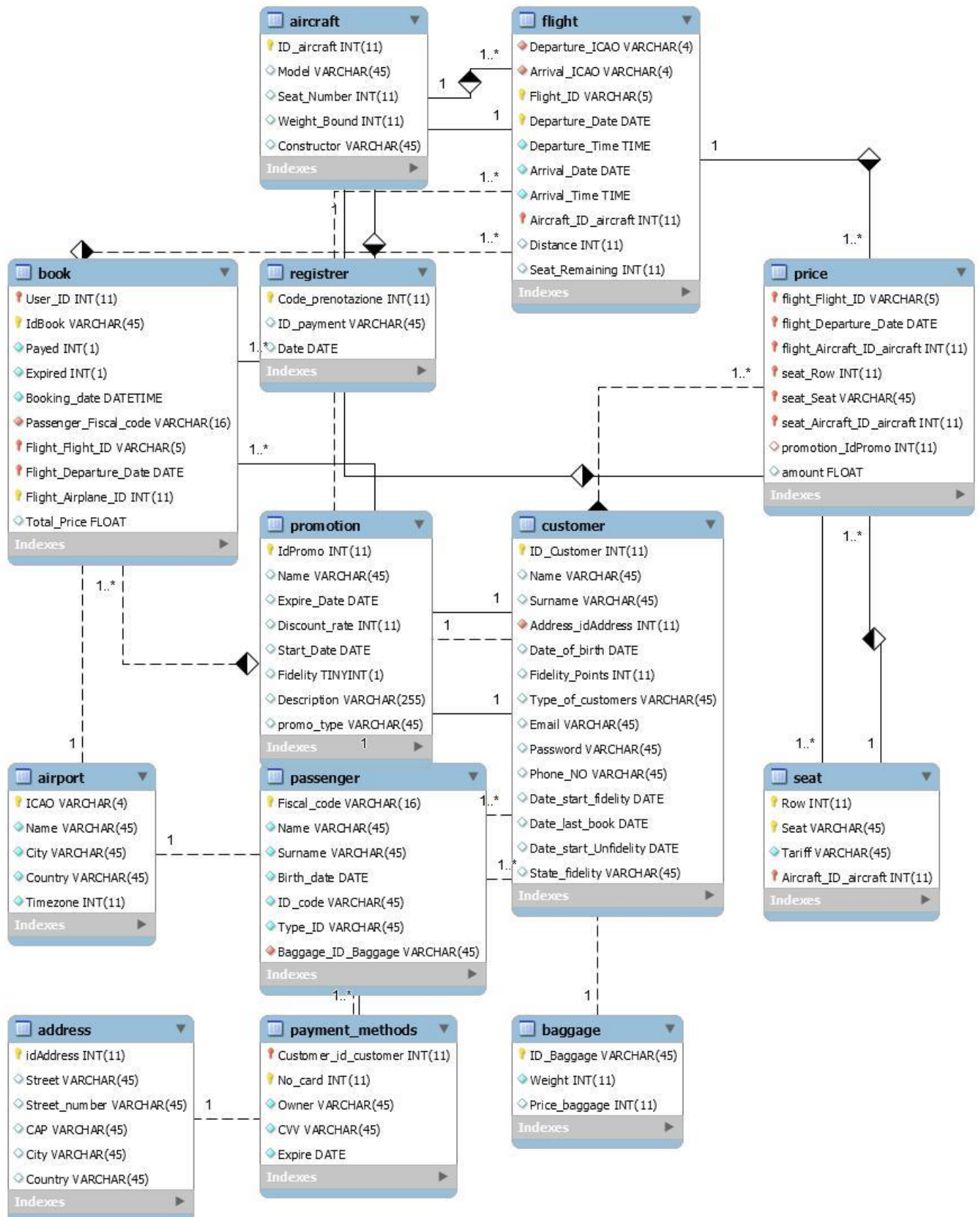


Diagramma di stato:



Modello del Database:



Risultato Sonarqube:

