

UNIVERSITÀ
degli STUDI
di CATANIA

DIPARTIMENTO DI INGEGNERIA
ELETTRICA ELETTRONICA E INFORMATICA

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA INFORMATICA

Roberto Ravale 1000044127

NAVIGATOR

Anno Accademico 2023/2024

INDICE

1. IDEAZIONE E ANALISI DEI REQUISITI	3
1.1 INTRODUZIONE	3
1.2 REQUISITI	4
1.3 OBIETTIVO DEI CASI D'USO	5
1.4 MODELLI DEI CASI D'USO.....	6
1.5 REGOLE DI DOMINIO	13
1.6 GLOSSARIO.....	14
1.7 SPECIFICHE SUPPLEMENTARI	15
2 ANALISI ORIENTATA AGLI OGGETTI	16
2.1 INTRODUZIONE	16
2.2 MODELLO DI DOMINIO	17
2.3 SSD E CONTRATTI DELLE OPERAZIONI	18
3. PROGETTAZIONE	23
3.1 DIAGRAMMA DELLE CLASSI	23
3.2 DIAGRAMMI DI SEQUENZA.....	24
4. TESTING.....	30
4.1 Introduzione.....	30
4.2 Individuazione dei casi di test	30

1. IDEAZIONE E ANALISI DEI REQUISITI

1.1 INTRODUZIONE

Il dirigente di una nuova e ambiziosa compagnia di crociere richiede lo sviluppo di un innovativo sistema software, finalizzato a perfezionare la gestione delle prenotazioni e assicurare un'esperienza di soggiorno impeccabile ai clienti a bordo delle sue lussuose navi da crociera. La compagnia, che inaugura la sua attività con un'esclusiva rotta marittima, comprendente Catania, Santorini, Mykonos, La Valletta e ritorno a Catania, si impegna a offrire servizi di lusso durante la crociera dalla durata di 5 notti, con partenze ogni lunedì e rientri il sabato. Il software dovrà occuparsi, non solo di gestire le prenotazioni delle cabine ma, anche di registrare i clienti che utilizzano i servizi esclusivi offerti a bordo, quali la Spa e le variegiate esperienze di escursione. Il software facilita la registrazione dettagliata dei servizi selezionati dai clienti, permettendo così alla compagnia di crociere di offrire un servizio su misura di ineguagliabile qualità.

1.2 REQUISITI

Requisiti del Tour Operator:

- Deve poter registrare i dati personali dei clienti e la settimana di prenotazione sulla nave.
- Deve avere la capacità di cancellare le prenotazioni esistenti.

Requisiti del Receptionist:

- Deve poter gestire richieste specifiche provenienti dalle cabine, come servizio in cabina.
- In base alle esigenze dei clienti, può mostrare servizi aggiuntivi e registrarli sul conto degli stessi.
- Può visualizzare i clienti a bordo della crociera e i clienti che stanno usufruendo dei servizi aggiuntivi
- Può effettuare il check-in dei clienti all'arrivo e il check-out alla fine della crociera.
- Se durante la crociera il cliente ha usufruito di servizi aggiuntivi e/o servizio in cabina, deve essere in grado di calcolare il conto finale di quest'ultimo

Requisiti dell'Admin

- Deve poter aggiungere o rimuovere servizi offerti dalla crociera.
- Può rendere una cabina non prenotabile per manutenzione
- Può variare i prezzi delle cabine in base alla stagionalità
- Gestisce il menù del servizio in cabina, aggiungendo o rimuovendo piatti.

Requisiti dei Clienti

- I clienti possono scegliere tra diverse tipologie di cabina: interna, con oblò e suite
- I clienti possono acquistare servizi extra a bordo della crociera come spa ed escursioni
- I clienti possono prenotare pasti da consumare in cabina tramite il servizio in cabina

1.3 OBIETTIVO DEI CASI D'USO

Analizzando i requisiti riportati nel paragrafo precedente, sono stati individuati l'attore principale a cui è destinato il sistema e gli obiettivi che egli intende portare a termine.

Da queste informazioni infine sono stati ricavati i casi d'uso principali:

ATTORE	OBIETTIVO	CASO D'USO
Tour Operator	Inserimento di una nuova prenotazione relativa ad una tipologia di cabina scelta	UC 1: Inserisci prenotazione cabina
Tour Operator	Cancellazione di una prenotazione esistente	UC 2: Elimina prenotazione cabina
Receptionist	Effettuare check-in e check-out	UC 3: Gestisci check-in/check-out
Receptionist	Vendita di escursioni e ingressi alla Spa	UC 4: Vendi servizi
Receptionist	Calcolare il conto finale del cliente in base ai servizi scelti a bordo della crociera	UC 5: Calcola conto cliente
Receptionist	Visualizza le informazioni relative ai clienti a bordo della crociera	UC 6: Visualizza clienti
Receptionist	Inserimento nuovo ordine relativo al servizio in cabina, previsto nel menù	UC 7: Crea nuovo ordine servizio in cabina
Admin	Rendere una cabina non prenotabile per manutenzione.	UC 8: Gestisci cabine
Admin	Modifica del prezzo delle cabine in un determinato periodo.	UC 9: Gestisci prezzo cabine
Admin	Aggiunta o rimozione di una portata nel menù.	UC 10: Gestisci portata
Admin	Aggiunta o rimozione di servizi offerti dalla crociera	UC 11: Gestisci servizi

1.4 MODELLI DEI CASI D'USO

Tra i casi d'uso individuati si è scelto di fornire una descrizione dettagliata per i casi d'uso 1 e 7, i rimanenti, invece, vengono descritti nel formato breve.

UC 1: Inserisci prenotazione cabina

Nome del caso d'uso	Inserisci prenotazione cabina
Portata	Applicazione
Livello	Obiettivo Utente
Attore primario	Tour Operator
Parte interessate e interessi	<ul style="list-style-type: none">• Tour Operator: vuole registrare le prenotazioni dei clienti.• Cliente: vuole effettuare una prenotazione.
Pre-condizioni	Il Cliente sa già la data di partenza della crociera, la tipologia di cabina da prenotare
Garanzia di successo	La prenotazione si conclude con la conferma del Cliente
Scenario principale di successo	<ol style="list-style-type: none">1. Il Cliente desidera prenotare una crociera e una tipologia di cabina per una determinata settimana2. Il Tour Operator verifica la disponibilità delle cabine3. Il Sistema mostra tutte le cabine disponibili corrispondenti alla categoria richiesta dal cliente, indicando anche il prezzo di ciascuna.4. Il Tour Operator informa il cliente sul prezzo della categoria di cabina scelta.5. Il cliente conferma la volontà di procedere con la prenotazione.6. Il Tour Operator richiede al cliente i dati personali necessari per la prenotazione e un metodo di pagamento valido (es. carta di credito).7. Il Cliente fornisce i propri dati personali e i dettagli della carta di credito.8. Il Tour Operator inserisce nel sistema i dati personali del cliente (nome, cognome, documento, codice fiscale, numero telefono) e i dettagli del metodo di pagamento.

	<p>9. Il sistema conferma la registrazione dei dati del cliente</p> <p>10. Il Tour Operator conferma nel sistema la prenotazione della crociera</p> <p>11. Il sistema finalizza la registrazione della prenotazione, associando al cliente la cabina selezionata</p>
Estensioni	<p>3a La tipologia di cabina richiesta non è disponibile.</p> <ol style="list-style-type: none"> 1. Il Tour Operator comunica al Cliente che la tipologia di cabina richiesta non è disponibile. 2. Il Cliente non vuole proseguire con la prenotazione e va via. 3. Il Tour Operator chiede al Sistema di annullare la prenotazione della cabina. 4. Il Sistema conferma l'annullamento della prenotazione della cabina. <p>3b La tipologia di cabina richiesta non è disponibile.</p> <ol style="list-style-type: none"> 1. Il Tour Operator comunica al Cliente che la tipologia di cabina richiesta non è disponibile. 2. Il Cliente richiede un'altra tipologia di cabina. 3. Il Tour Operator chiede al Sistema di annullare la prenotazione della cabina. 4. Il Sistema conferma l'annullamento della prenotazione della cabina 5. Lo scenario riprende dal passo 1. <p>5a Il Cliente non conferma la tipologia di cabina richiesta a causa del prezzo.</p> <ol style="list-style-type: none"> 1. Il Cliente non vuole proseguire con la prenotazione e va via. 1. Il Tour Operator chiede al Sistema di annullare la prenotazione della cabina. 2. Il Sistema conferma l'annullamento. <p>5b Il Cliente richiede un'altra tipologia di cabina a causa del prezzo.</p> <ol style="list-style-type: none"> 1. Il Cliente vuole cambiare tipologia di cabina. 2. Il Tour Operator chiede al Sistema l'annullamento della prenotazione della cabina. 3. Il Sistema conferma l'annullamento. 4. Lo scenario riprende dal passo 1.
Frequenza di ripetizioni	Legata all'affluenza dei clienti e alla prenotazione che i clienti vogliono effettuare.

UC 7: Crea nuovo ordine servizio in cabina

Nome del caso d'uso	Crea nuovo ordine servizio in cabina
Portata	Applicazione
Livello	Obiettivo Utente
Attore primario	Receptionist
Parte interessate e interessi	<ul style="list-style-type: none">• Receptionist: vuole prendere l'ordinazione.• Cliente: vuole effettuare un ordine di servizio in cabina.
Pre-condizioni	-
Garanzia di successo	La prenotazione si conclude con la conferma del Cliente
Scenario principale di successo	<ol style="list-style-type: none">1. Il Cliente richiede al Receptionist di effettuare un ordine di servizio in cabina, fornendo il numero della propria cabina2. Il Receptionist avvia la procedura di creazione di un nuovo ordine nel Sistema, inserendo il numero della cabina del Cliente3. Il Sistema verifica la validità del numero della cabina e conferma la creazione di un nuovo ordine per il servizio in cabina4. Il Receptionist richiede al sistema l'elenco aggiornato delle portate disponibili.5. Il Sistema mostra l'elenco delle portate disponibili con i rispettivi prezzi6. Il Receptionist comunica al Cliente l'elenco delle portate disponibili e i relativi prezzi7. Il Cliente sceglie le portate desiderate e specifica la quantità per ciascuna8. Il Receptionist inserisce nel Sistema le portate e la quantità richieste dal Cliente9. Il Sistema conferma l'inserimento di ogni portata all'ordine10. Il Receptionist registra l'ordine nel Sistema.11. Il Sistema crea un nuovo ordine e lo associa alla cabina del Cliente.

	12. Il Receptionist conferma l'ordine al Cliente.
Estensioni	<p>*a In qualsiasi momento il sistema fallisce e ha un arresto improvviso</p> <ol style="list-style-type: none"> 1. L'Admin riavvia il software e richiede il ripristino dello stato precedente del Sistema. 2. Il Sistema ripristina lo stato precedente. <p>5a Il Cliente non vuole più ordinare a causa dei prezzi.</p> <ol style="list-style-type: none"> 1. Il Cliente non vuole proseguire e va via 2. Il Receptionist chiede al Sistema di annullare l'ordine in corso 3. Il Sistema Conferma l'annullamento dell'ordine
Frequenza di ripetizioni	Legata ai clienti che vogliono usufruire del servizio in cabina

UC 2: Elimina prenotazione cabina

1. Il Tour Operator seleziona la prenotazione del Cliente che intende cancellare.
2. Il Sistema mostra i dettagli della prenotazione selezionata
3. Il Tour Operator comunica al Cliente la decisione di procedere con la cancellazione della prenotazione.
4. Il Cliente conferma la volontà di cancellare la prenotazione.
5. Il Tour Operator conferma nel Sistema la cancellazione della prenotazione.
6. Il Sistema registra la cancellazione della prenotazione e aggiorna la disponibilità della cabina.
7. Il Tour Operator informa il Cliente della cancellazione avvenuta

UC 3: Gestisci check-in/check-out

1. Il Receptionist richiede al Cliente le proprie generalità.
2. Il Cliente fornisce le generalità al Receptionist.
3. Il Receptionist inserisce le generalità del Cliente all'interno del Sistema e verifica la presenza della prenotazione da parte del Cliente.
4. Il Sistema conferma la presenza della prenotazione del Cliente al Receptionist.
5. Il Receptionist inserisce nel Sistema l'arrivo del Cliente (check-in).
6. Il Sistema registra l'arrivo del Cliente.
7. Il Sistema conferma l'avvenuta registrazione

Scenario alternativo

5.a) Il Receptionist inserisce nel Sistema la partenza del Cliente (check-out).

1. Il Sistema registra la partenza del Cliente.
2. Il Sistema conferma la fine del pernottamento del Cliente.

UC 4: Vendi servizi

1. Il Cliente chiede di conoscere i servizi offerti dalla crociera.
2. Il Receptionist richiede al Sistema i servizi disponibili con i relativi prezzi.
3. Il Receptionist comunica al Cliente i servizi disponibili e i relativi prezzi.
4. Il Cliente sceglie un servizio indicando il giorno in cui ne vuole usufruire.
5. Il Receptionist inserisce all'interno del Sistema il servizio scelto dal Cliente e la relativa data.
6. Il Sistema registra il servizio al Cliente.
7. Il Receptionist conferma l'avvenuta registrazione del servizio.

Scenario alternativo

3.a Prezzo del servizio elevato:

1. Il Cliente rifiuta il servizio richiesto.
2. Il Receptionist chiede al Sistema di annullare l'operazione di registrazione del servizio.
3. Il Sistema mostra la conferma dell'annullamento dell'operazione di registrazione.

UC 5: Calcola conto cliente

1. Il Receptionist richiede al Sistema di calcolare il conto del cliente.
2. Il Sistema effettua il calcolo del conto in base alla tipologia di cabina, ai servizi scelti (Spa e/o escursioni) e in base alle portate ordinate tramite il servizio in cabina.
3. Il Sistema mostra le portate ordinate tramite il servizio in cabina, i servizi acquistati durante il soggiorno e il conto da pagare.
4. Il Receptionist comunica al Cliente il conto.

UC 6: Visualizza clienti

1. Il Receptionist richiede al Sistema l'elenco dei clienti della crociera, con i relativi servizi che hanno acquistato.
2. Il Sistema mostra l'elenco dei clienti con i relativi servizi acquistati.

UC 8: Gestisci cabine

1. L'Admin richiede al Sistema le cabine non prenotate per un determinato periodo.
2. Il Sistema mostra le cabine non prenotate.
3. L'Admin inserisce nel Sistema la cabina da sottoporre a manutenzione.
4. Il Sistema rende non prenotabile la cabina inserita dall'Admin.
5. Il Sistema conferma l'avvenuta operazione.

Scenario alternativo

- 1.a L'Admin richiede al Sistema la riapertura di una cabina.
 1. Il Sistema mostra all'Admin le cabine sottoposte a manutenzione.
 2. L'Admin comunica al Sistema la cabina da riaprire.
 3. Il Sistema rende prenotabile la cabina comunicata dall'Admin.
 4. Il Sistema conferma l'avvenuta operazione.

UC 9: Gestisci prezzo cabine

1. L'Admin richiede al Sistema i prezzi delle cabine.
2. Il Sistema mostra all'Admin i prezzi delle cabine.
3. L'Admin comunica al Sistema la variazione di prezzo (in percentuale) e il periodo soggetto a modifiche del prezzo.
4. Il Sistema modifica il prezzo delle cabine per il periodo selezionato e conferma l'operazione

UC 10: Gestisci portata

1. L'Admin richiede al Sistema il menù delle portate.
2. Il Sistema mostra all'Admin il menù.
3. L'Admin comunica al Sistema la portata d'aggiungere con il relativo prezzo nel menù.
4. Il Sistema inserisce la portata nel menù.
5. Il Sistema conferma all'Admin l'operazione.

Scenario alternativo

- 3.a L'Admin comunica al Sistema la portata da rimuovere dal menu.
 1. Il Sistema elimina la portata dal menu.
 2. Il Sistema conferma all'Admin l'operazione.

UC 11: Gestisci servizi

1. L'Admin richiede al Sistema i servizi disponibili.
2. Il Sistema mostra all'Admin i servizi.
3. L'Admin inserisce nel Sistema un nuovo servizio con il relativo prezzo.
4. Il Sistema rende disponibile il servizio inserito.
5. Il Sistema conferma l'operazione.

Scenario alternativo

3.a L'Admin rimuove un servizio dal Sistema

1. Il Sistema effettua la rimozione del servizio.
2. Il Sistema conferma l'operazione

1.5 REGOLE DI DOMINIO

Per il corretto funzionamento dell'applicazione, devono essere rispettate le seguenti regole di dominio:

ID	REGOLA	MODIFICABILITÀ	SORGENTE
R1	Se la crociera si svolge nei periodi estivi (Giugno, Luglio, Agosto) il prezzo delle cabine sarà aumentato del 40%	Media, l'Admin potrebbe decidere di applicare l'incremento solo in alcuni periodi dell'anno.	Politica interna della crociera
R2	Se la crociera si svolge nei mesi invernali (Dicembre, Gennaio e Febbraio) il prezzo delle cabine sarà diminuito del 20%.	Alta, l'Admin può decidere di applicare lo sconto anche in altri periodi dell'anno.	Politica interna della crociera
R3	Se viene prenotata una suite, il Cliente ha diritto ad uno sconto del 20% sul servizio in cabina.	Alta, l'Admin può decidere di applicare lo sconto anche in altre occasioni.	Politica interna della crociera
R4	I servizi extra possono essere acquistati solo a bordo della crociera a partire dal secondo giorno di navigazione	Bassa	Politica interna della crociera
R5	I servizi acquistati non possono essere rimborsati	Bassa	Politica interna della crociera

1.6 GLOSSARIO

Di seguito sono riportate le parole più rilevanti e la relativa semantica:

NaviGator: sistema dedicato alla gestione delle prenotazioni e all'amministrazione di tutti i servizi offerti durante la crociera.

Admin: responsabile della gestione e configurazione del sistema

Tour Operator: professionista o agenzia che gestisce le prenotazioni per conto dei clienti

Receptionist: personale addetto alla reception sulla nave da crociera, responsabile per il check-in/check-out dei passeggeri e la gestione dei servizi aggiuntivi come la Spa e le escursioni

Cliente: persona che effettua una prenotazione e che utilizza i servizi offerti dalla compagnia di crociere

Prenotazione: accordo formalizzato tramite il quale un cliente riserva una cabina su una nave da crociera per un determinato periodo di tempo.

Cabina: alloggio il cui utilizzo è riservato ai clienti, vincolato dalla prenotazione effettuata. Ogni cabina è dotata di un letto matrimoniale standard, assicurando comfort e intimità ai viaggiatori.

Servizio: prestazione offerta al cliente vincolata dal contratto stipulato. I servizi offerti sono Spa ed escursioni.

Escursione: attività organizzata che si svolge durante una crociera, offrendo ai passeggeri l'opportunità di esplorare destinazioni specifiche mentre la nave è attraccata

Spa: centro benessere situato all'interno di una nave da crociera, che offre una varietà di trattamenti e servizi per il relax e la cura personale dei passeggeri

Servizio in Cabina: Servizio offerto ai passeggeri che consente di ordinare cibo e bevande da consumare privatamente nella propria cabina.

Ordine: richiesta da parte del cliente di una o più portate.

Portata: vivanda offerta al cliente su richiesta.

1.7 SPECIFICHE SUPPLEMENTARI

Usabilità

- L'interfaccia grafica deve essere semplice e intuitiva
- Sebbene sia consigliabile consultare le istruzioni di base per familiarizzare con il software, gli utenti dovrebbero essere in grado di navigarlo efficacemente anche senza questa premessa.

Vincoli di Sviluppo Software

- Il software è stato sviluppato utilizzando il linguaggio di programmazione C#

2 ANALISI ORIENTATA AGLI OGGETTI

2.1 INTRODUZIONE

L'implementazione dell'applicazione è stata guidata da un processo iterativo ed evolutivo basato sul modello UP (Unified Process), suddiviso in tre fasi. Grazie a questo metodo è stato possibile sviluppare gradualmente l'architettura del software NaviGator.

Ciascuna iterazione ha affrontato diverse problematiche gestendo:

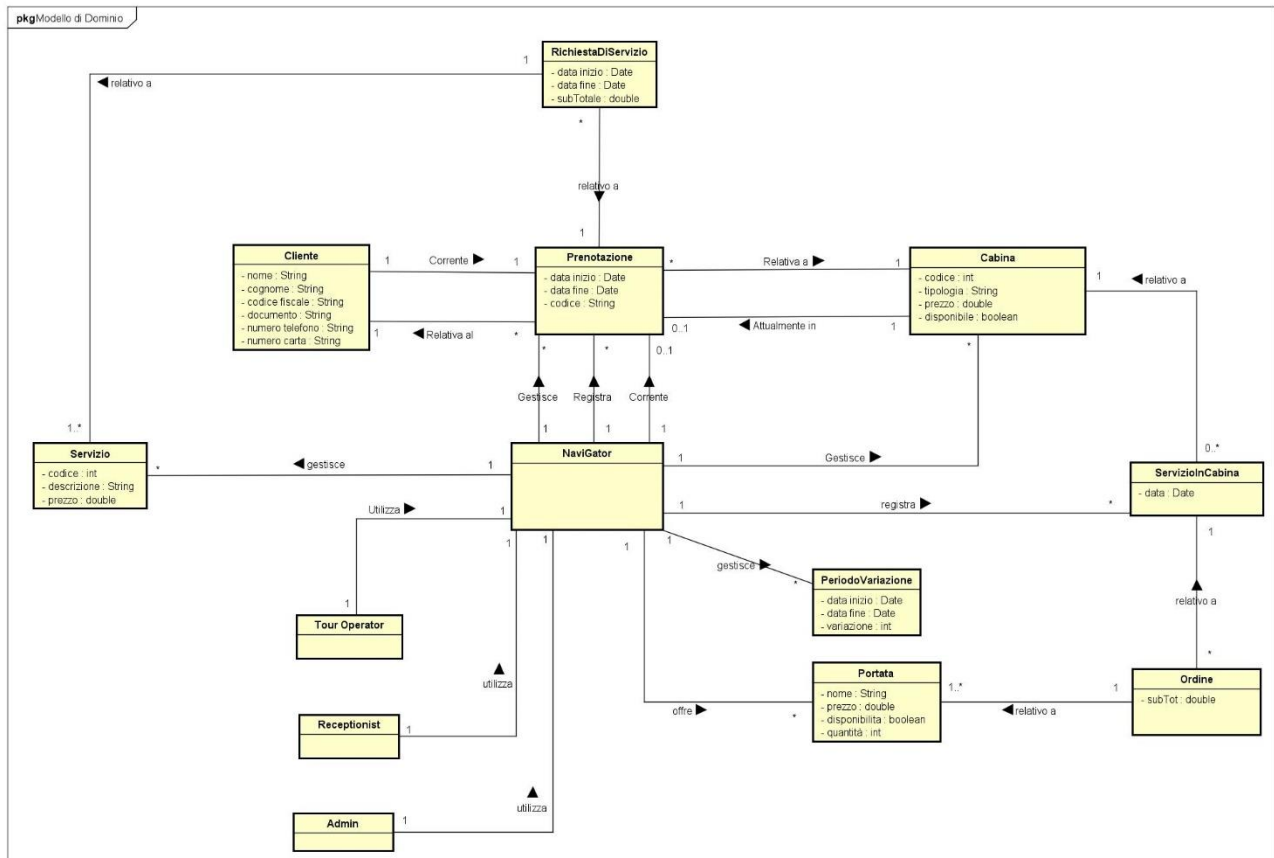
- Iterazione 1:
 - Tratta l'implementazione dello UC1: Inserisci prenotazione cabina.
- Iterazione 2:
 - Viene affrontato l'UC7: Crea nuovo ordine servizio in cabina
- Implementazione 3:
 - Viene esaminato l'UC9: Gestisci prezzo Cabine.

I rimanenti casi d'uso sono stati trattati in maniera non dettagliata.

All'inizio di ogni iterazione, viene affrontata l'analisi dei requisiti Orientata agli oggetti, che consiste nella stesura del modello di dominio, costituito dalle classi concettuali utili per la risoluzione del problema e il corretto funzionamento del sistema, e dei relativi diagrammi di sistema e contratti delle operazioni.

2.2 MODELLO DI DOMINIO

La Modellazione del Business è la disciplina che si occupa di fornire una visione completa del dominio. In particolare, essa include la creazione del Modello di Dominio, un elaborato grafico che individua concetti, attributi e associazioni rilevanti. A seguire il Modello di dominio finale di NaviGator, elaborato utilizzando il software Astah Professional:



Sono state individuate le seguenti classi concettuali:

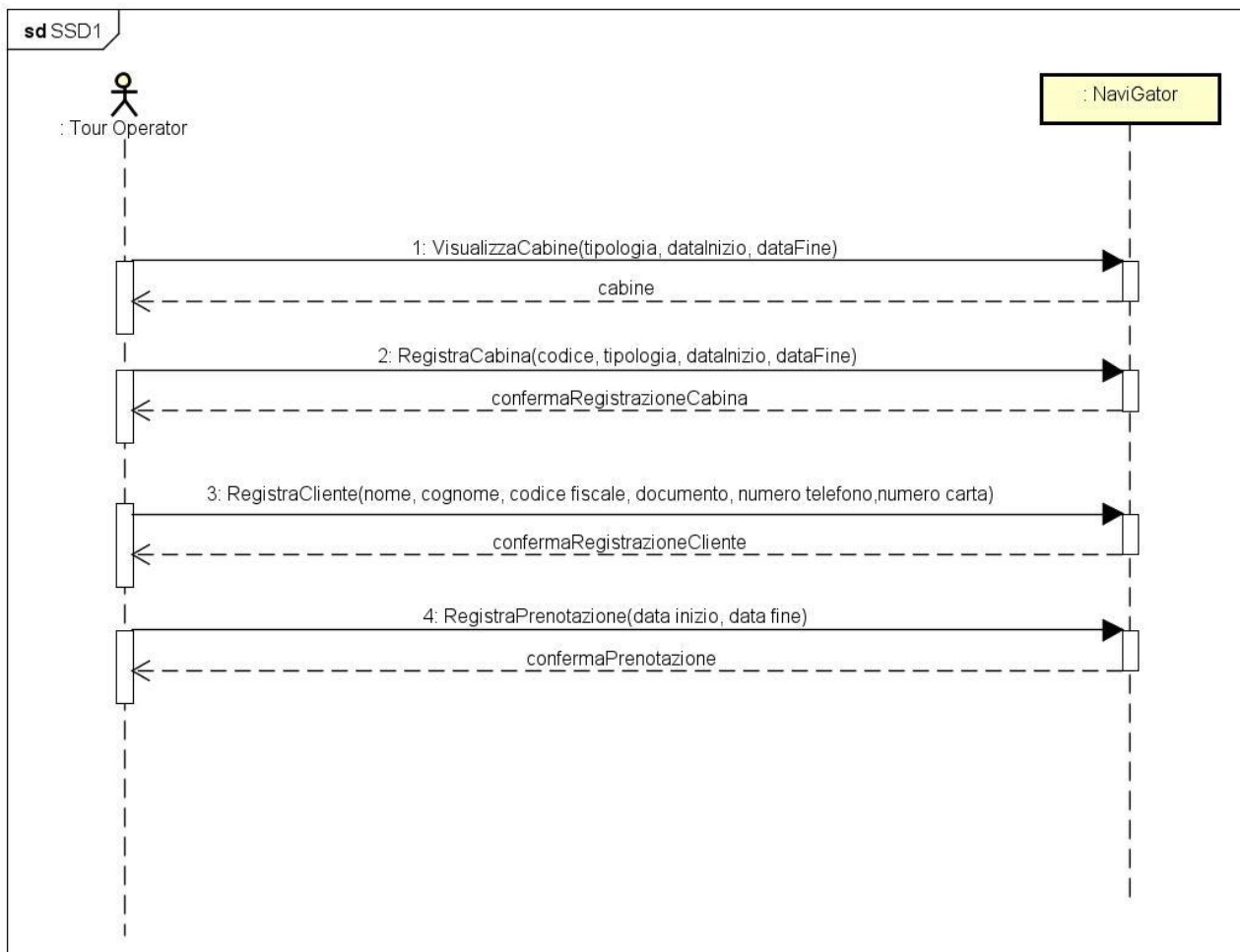
- **Cliente**: persona che vuole effettuare una prenotazione per una crociera.
- **Cabina**: rappresenta l'oggetto prenotabile.
- **Tour Operator**: attore primario, che interagisce direttamente con il Sistema.
- **NaviGator**: rappresenta il Sistema.
- **Prenotazione**: rappresenta la prenotazione effettuata dal cliente.
- **Receptionist**: attore primario, che interagisce direttamente con il Sistema;
- **Portata**: rappresenta le varie portate che il sistema offre ai clienti tramite il servizio in cabina;
- **ServiziInCabina**: contiene informazioni relative al servizio in cabina;
- **Ordine**: rappresenta l'ordine specifico richiesto dal cliente per il servizio in cabina.
- **Admin**: attore primario, che interagisce direttamente con il Sistema;
- **RichiestaDiServizio**: contiene informazioni relative alla prenotazione di servizi da parte del cliente;
- **Servizio**: rappresenta i vari servizi che il sistema offre ai clienti;
- **PeriodoVariazione**: rappresenta il periodo soggetto a variazione di prezzo

2.3 SSD E CONTRATTI DELLE OPERAZIONI

Continuando con l'approccio Orientato agli Oggetti il passaggio successivo coinvolge la realizzazione dei Diagrammi di Sequenza di Sistema (SSD) per delineare il flusso degli eventi di input e output relativi ai diversi casi d'uso esaminati in ogni iterazione. Le principali funzionalità del sistema, identificate nei SSD, saranno espresse attraverso i Contratti.

Iterazione 1:

UC1: Inserisci prenotazione cabina



Contratto CO2: RegistraCabina

Operazione:	RegistraCabina(codice, tipologia, dataInizio, dataFine)
Riferimenti:	Caso d'uso: Inserisci prenotazione cabina
Pre-condizioni:	nessuna
Post-condizioni:	<ul style="list-style-type: none">- è stata creata una nuova istanza <i>prenotazioneInCorso</i> di Prenotazione- è stata selezionata l'istanza c di Cabina riferita a codice- l'istanza c di Cabina è stata associata a <i>prenotazioneInCorso</i>- sono state settate le date di inizio e fine prenotazione in <i>prenotazioneInCorso</i>.

Contratto CO3: RegistraCliente

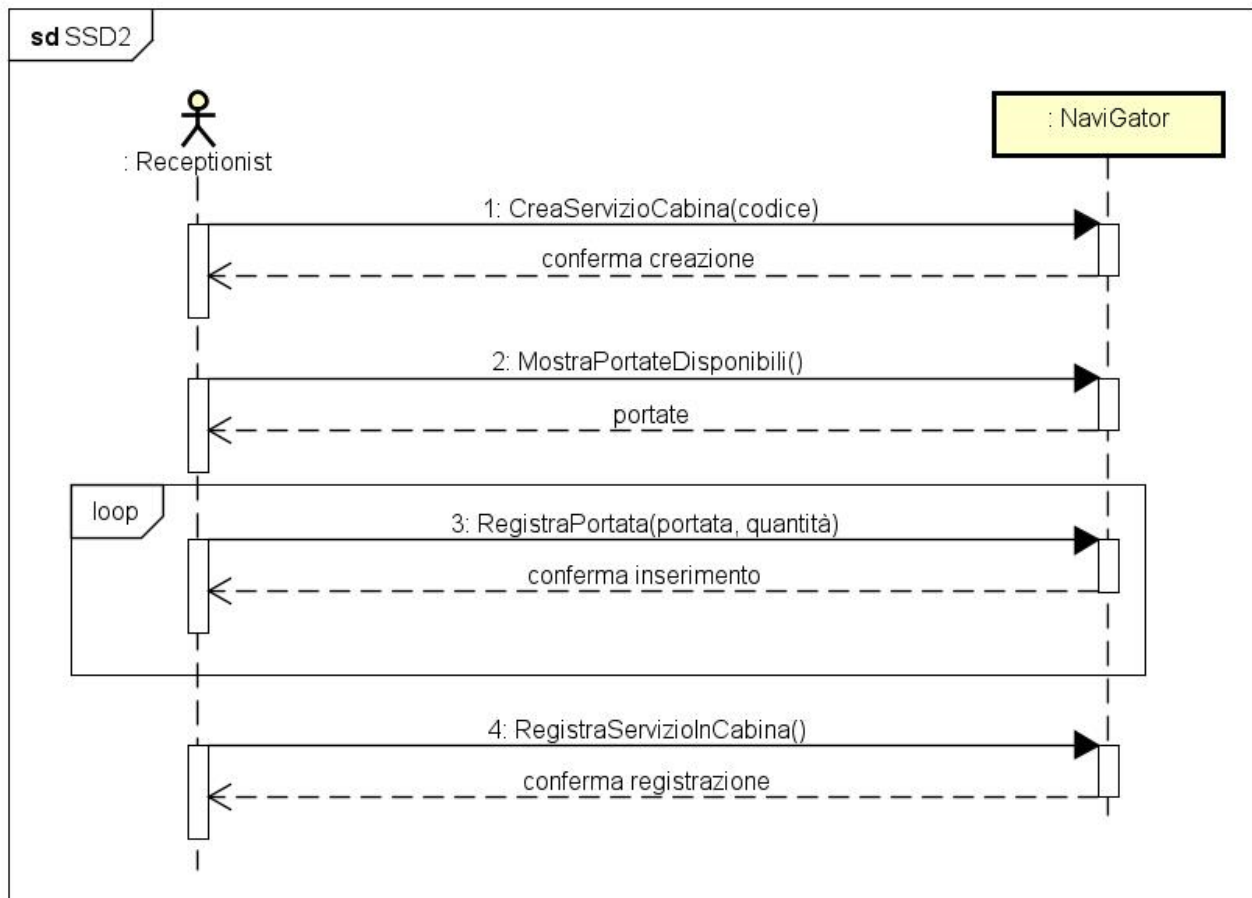
Operazione:	RegistraCliente(nome, cognome, codice fiscale, documento, numero telefono, numero carta)
Riferimenti:	Caso d'uso: Inserisci prenotazione cabina
Pre-condizioni:	è in corso una prenotazione <i>prenotazioneInCorso</i>
Post-condizioni:	<ul style="list-style-type: none">- è stata creata una nuova istanza c di Cliente- l'istanza c di Cliente è stata associata a <i>prenotazioneInCorso</i>

Contratto CO4: RegistraPrenotazione

Operazione:	RegistraPrenotazione()
Riferimenti:	Caso d'uso: Inserisci prenotazione cabina
Pre-condizioni:	<ul style="list-style-type: none">- è in corso una prenotazione <i>prenotazioneInCorso</i>- <i>prenotazioneInCorso</i> contiene un'istanza c di Cliente
Post-condizioni:	- <i>prenotazioneInCorso</i> è stata aggiunta all'elenco delle prenotazioni <i>elencoPrenotazioni</i>

Iterazione 2:

UC7: Crea nuovo ordine servizio in cabina



Contratto CO1: CreaServizioCabina

Operazione: CreaServizioCabina(codice, data)

Riferimenti: caso d'uso: Crea nuovo ordine servizio in cabina

Pre-condizioni: -

Post-condizioni: - è stata creata una nuova istanza sc di ServizioInCabina

Contratto CO2: MostraPortate

Operazione: MostraPortateDisponibili()

Riferimenti: caso d'uso: Crea nuovo ordine servizio in cabina

Pre-condizioni: - è in corso un ServizioInCabina sc

Post-condizioni: -

Contratto CO3: RegistraPortata

- Operazione:** RegistraPortata(portata, quantità)
- Riferimenti:** caso d'uso: Crea nuovo ordine servizio in cabina
- Pre-condizioni:** - è in corso un ServizioInCabina sc
- Post-condizioni:** - è stata selezionata l'istanza p di Portata
- l'istanza p di Portata è stata associata all'Ordine o

Contratto CO4: RegistraServizioCabina

- Operazione:** RegistraServizioCabina()
- Riferimenti:** caso d'uso: Crea nuovo ordine servizio in cabina
- Pre-condizioni:** - è in corso un ServizioInCabina sc
- Post-condizioni:** - è stata associata l'istanza sc alla lista di ServizioInCabina
(*elencoServiziCabina*)

Iterazione 3:

UC9: Gestisci prezzo Cabine



Contratto CO1: InserisciVariazione

Operazione: InserisciVariazione(dataInizio, dataFine, variazione)

Riferimenti: Caso d'uso: Gestisci prezzo Cabine

Pre-condizioni: -

Post-condizioni:

- È stata creata un'istanza v di PeriodoVariazione
- è stata associata l'istanza v alla lista dei periodi di variazione *listaPeriodiVariazione*

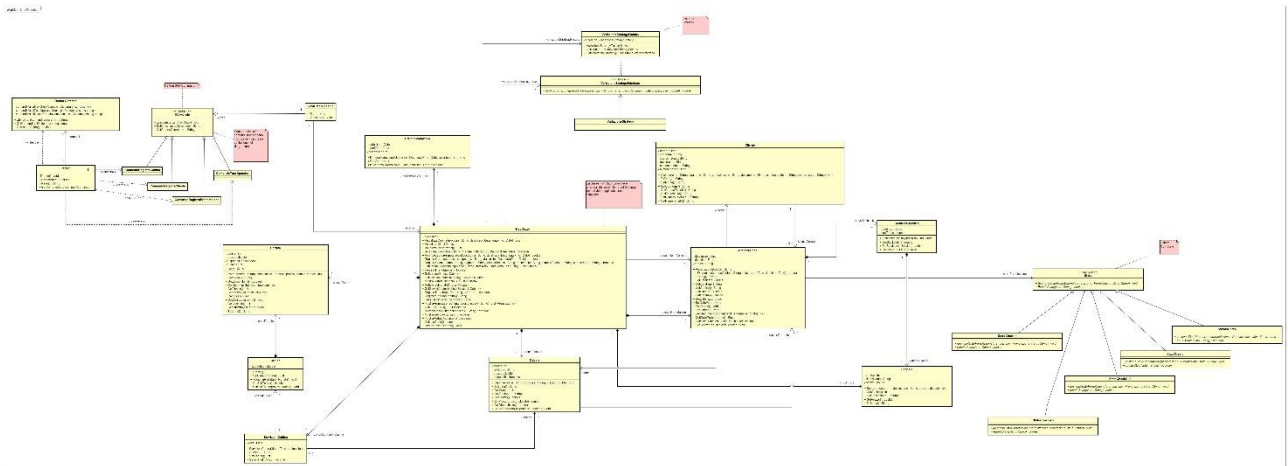
3. PROGETTAZIONE

3.1 DIAGRAMMA DELLE CLASSI

Durante la fase di progettazione, il focus è sul Modello di Progetto, che rappresenta una serie di diagrammi che descrivono la progettazione logica:

- Dinamicamente, attraverso i Diagrammi di Iterazione
- Staticamente, attraverso il Diagramma delle Classi

Il Diagramma delle Classi è:



Si è deciso di utilizzare i seguenti pattern GoF:

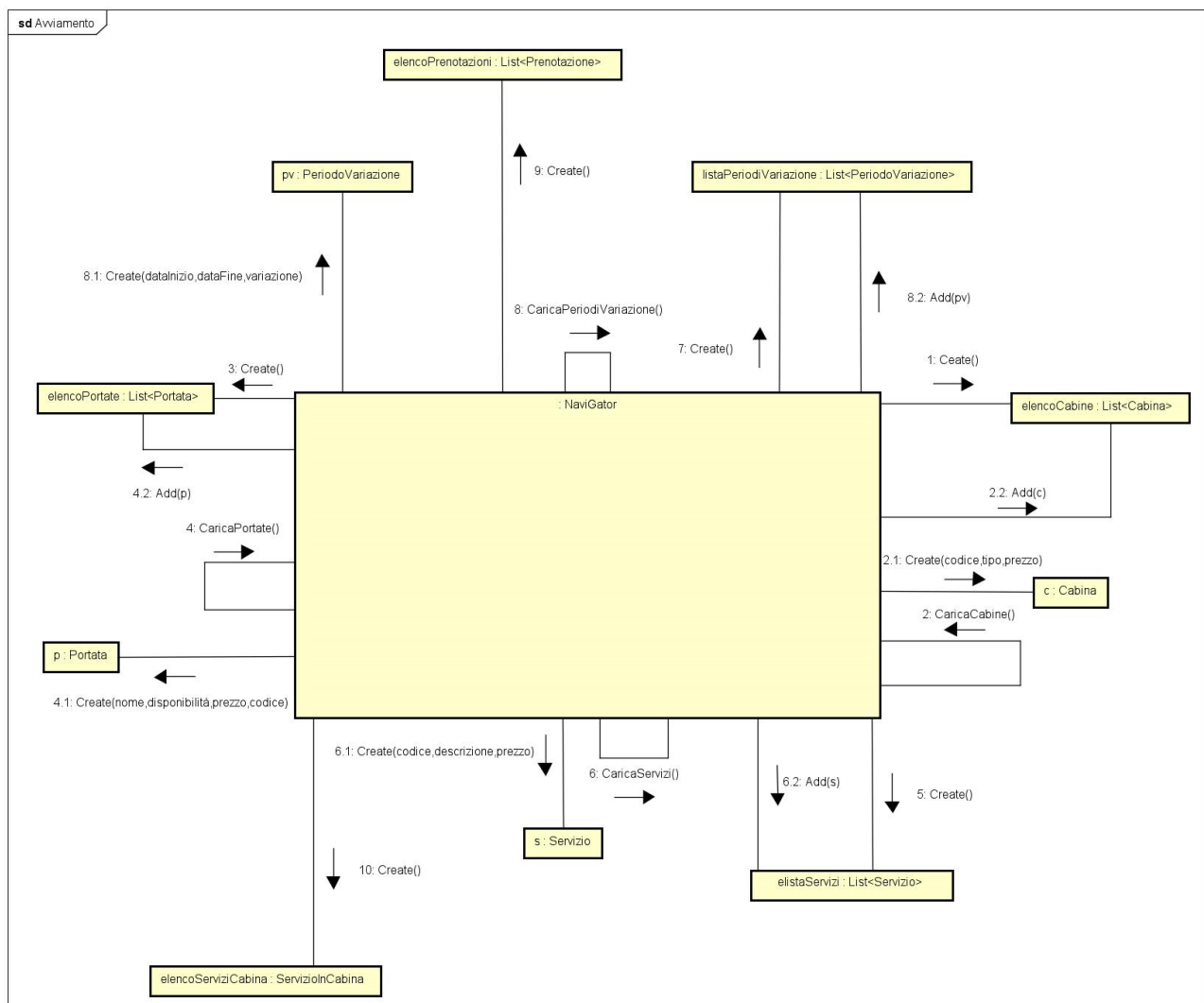
- **Command:** utilizzato per gestire le diverse azioni che possono essere eseguite dall'utente attraverso l'interfaccia console. Questo pattern è particolarmente utile quando si hanno diverse operazioni che possono essere eseguite in modo indipendente e si vuole separare il codice che innesca un'operazione dal codice che effettivamente la esegue. Inoltre, il pattern Command è anche utile per implementare callback, cioè operazioni che devono essere eseguite in risposta a determinati eventi, e per gestire code di richieste, dove le operazioni possono essere schedulate, messe in coda ed eseguite in momenti diversi.
- **State:** utilizzato per gestire i diversi stati della prenotazione: Creata, Check-In, In corso, Check-Out e Conclusa.
- **Strategy:** utilizzato per aggiungere o cambiare in futuro l'algoritmo di calcolo della variazione del prezzo in modo dinamico a runtime

3.2 DIAGRAMMI DI SEQUENZA

Attinenti alla fase di progettazione, vi sono i Diagrammi di Sequenza di Sistema, i quali mostrano le interazioni tra gli oggetti, rispettando l'ordine sequenziale in cui avvengono gli scambi dei messaggi tra gli stessi

CASO D'USO DI AVVIAMENTO

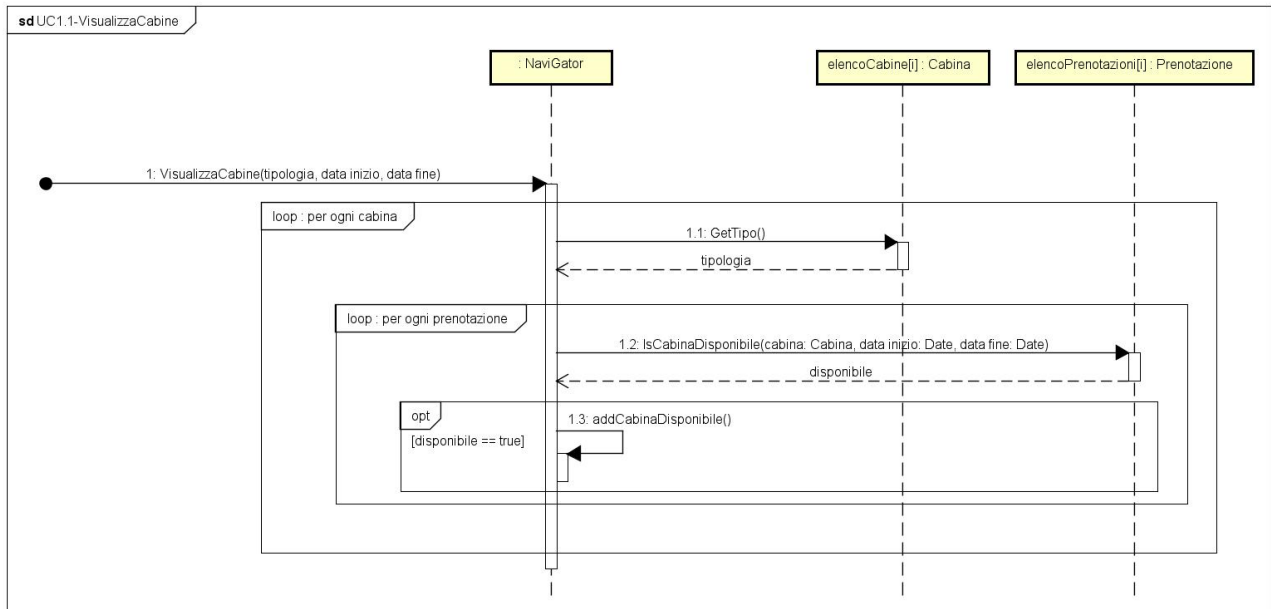
Per garantire che il sistema sia operativo fin dalla prima fase di sviluppo, è stato progettato un caso d'uso di avviamento che consentisse di creare una lista di: cabine, portate, servizi e periodi di variazione del prezzo



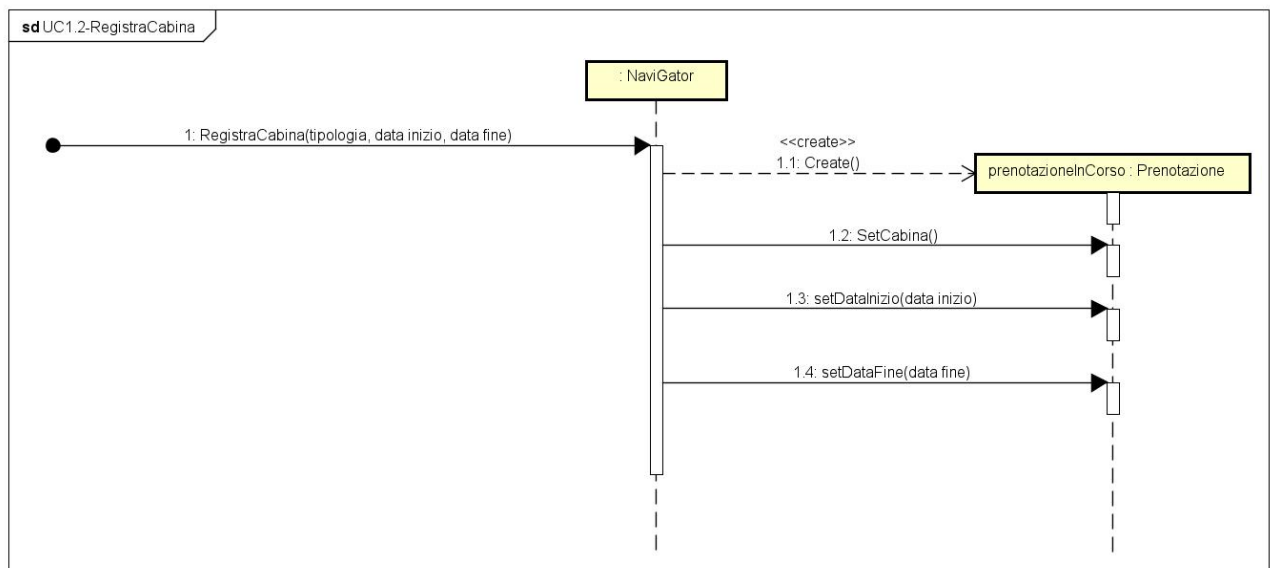
ITERAZIONE 1:

UC1: Inserisci prenotazione cabina

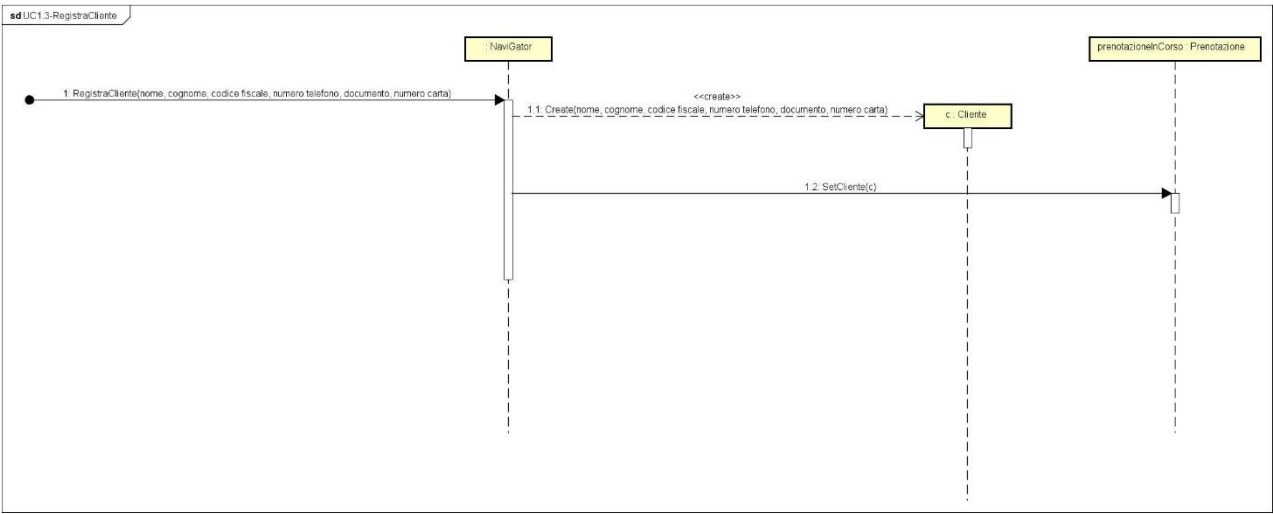
SD VisualizzaCabine



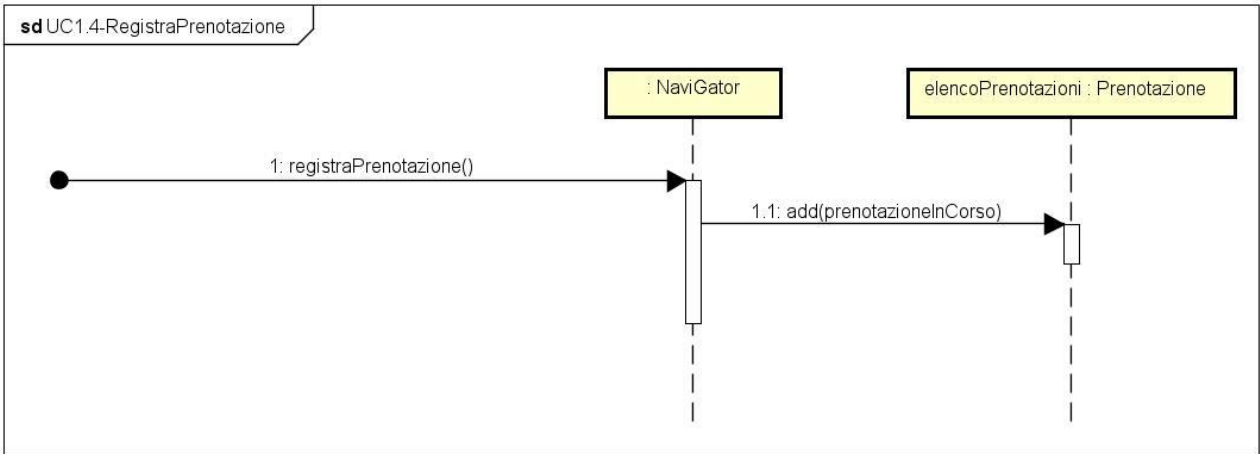
SD RegistraCabina



SD RegistraCliente



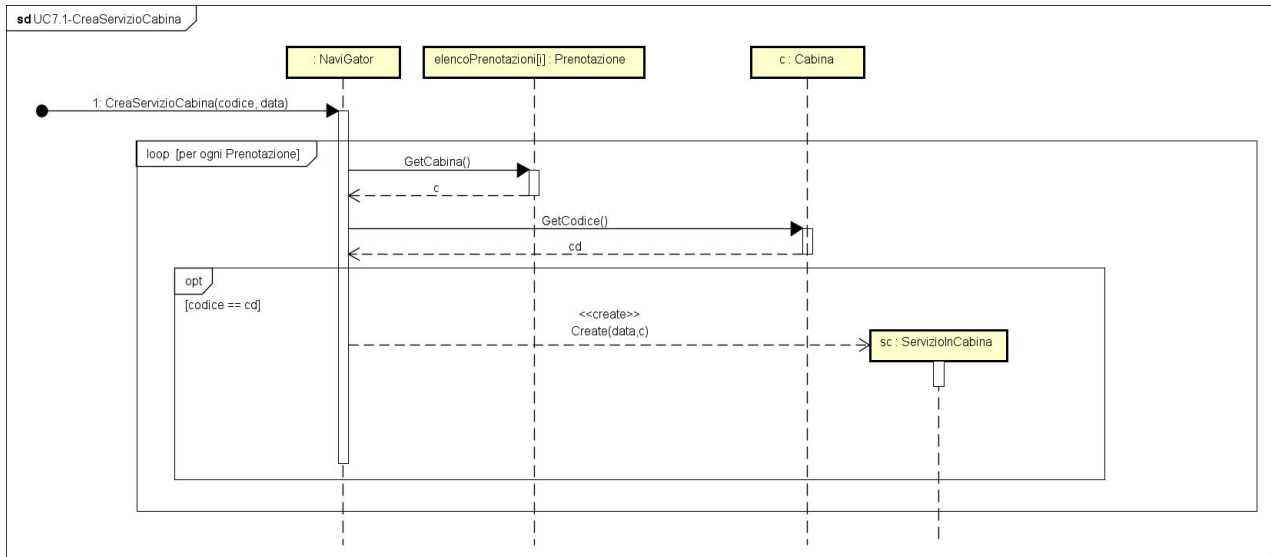
SD RegistraPrenotazione



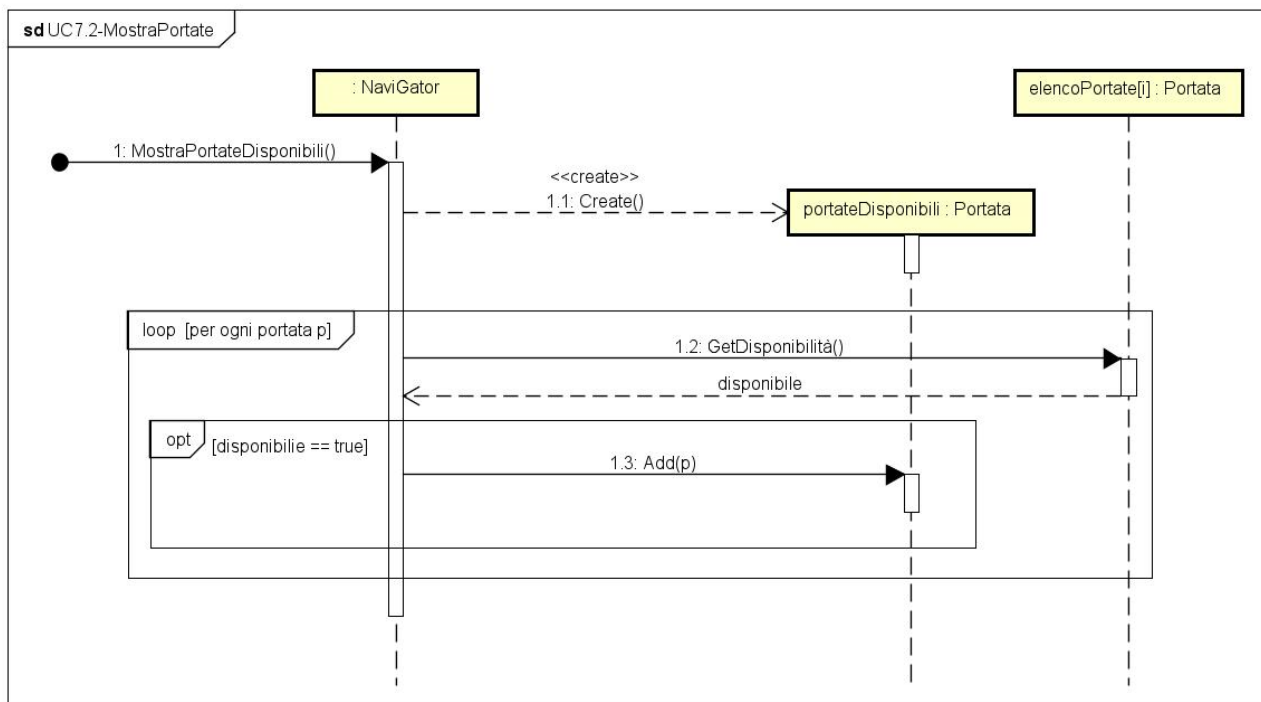
ITERAZIONE 2:

UC7: Crea nuovo ordine servizio in cabina

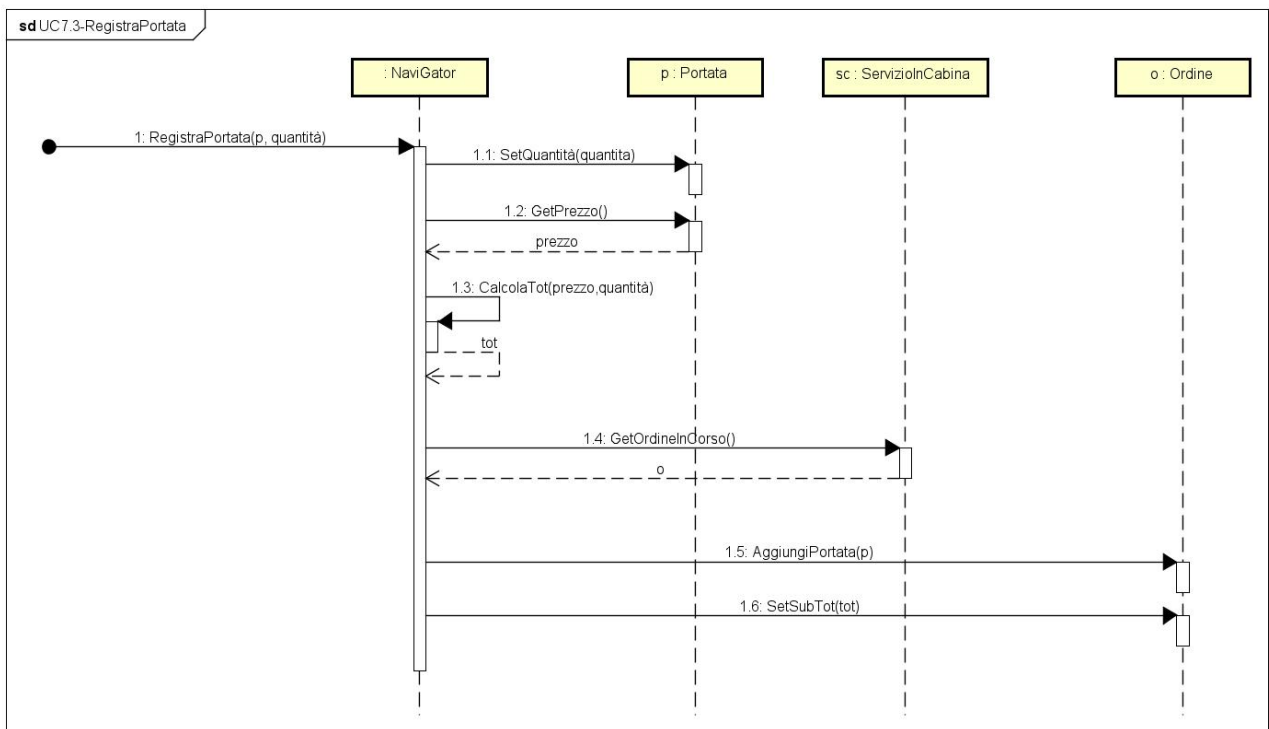
SD CreaServizioCabina



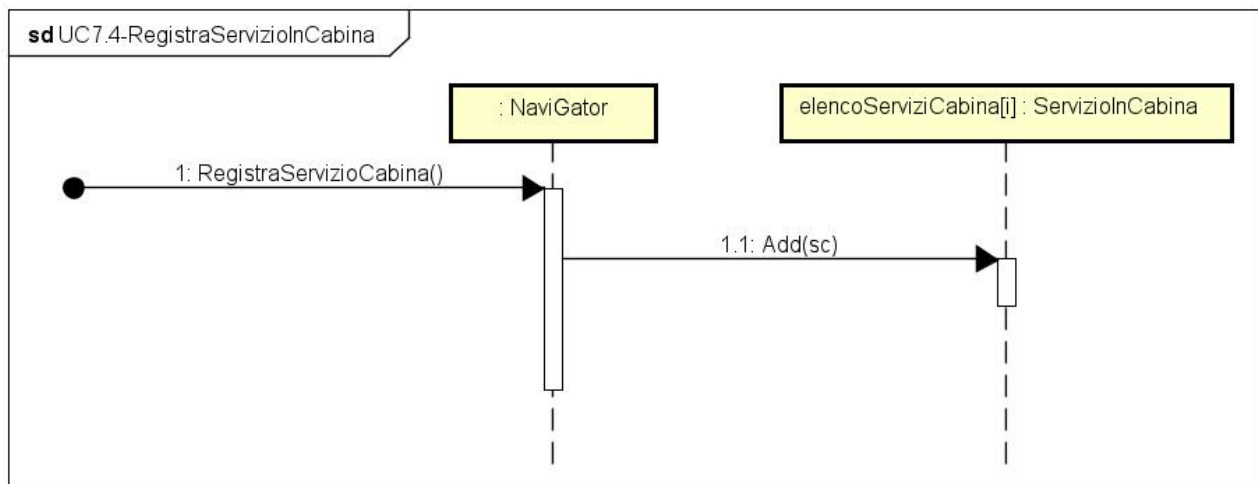
SD MostraPortate



SD RegistraPortata



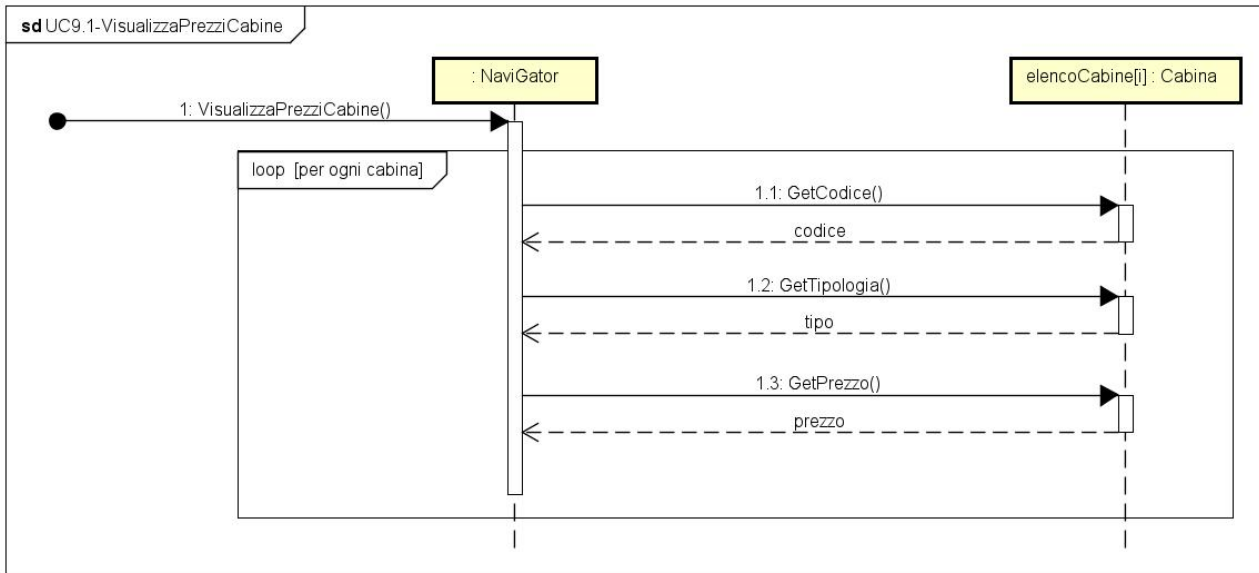
SD RegistraServizioInCabina



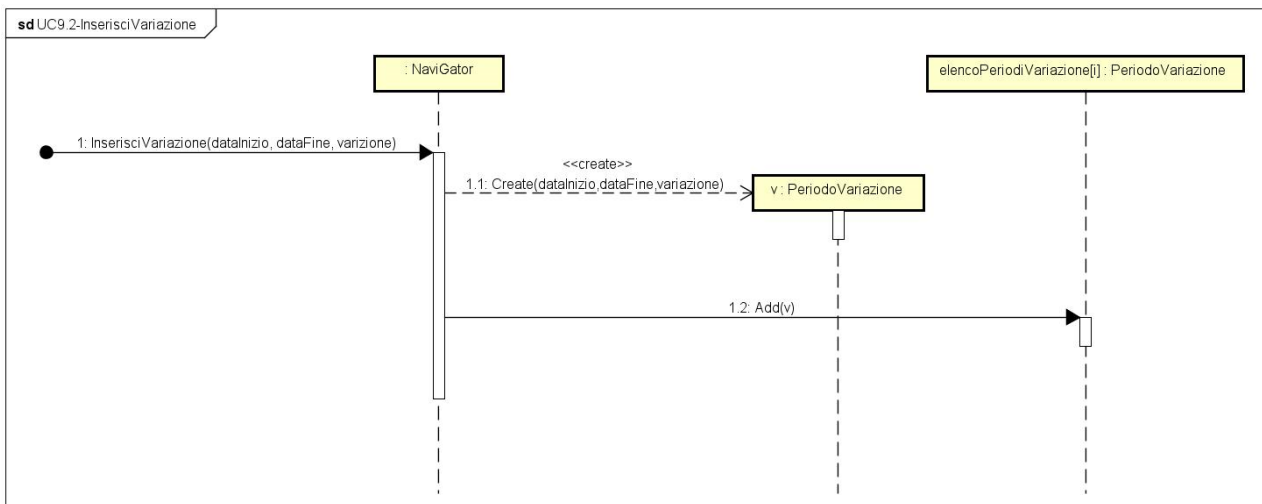
Iterazione 3:

UC9: Gestisci prezzo Cabine

SD VisualizzaPrezziCabine



SD InserisciVariazione



4. TESTING

4.1 Introduzione

Il processo di sviluppo di un programma robusto ed efficiente dipende in gran parte dalla fase di testing, la quale aumenta la probabilità di individuare eventuali difetti. Un testing ben progettato può individuare comportamenti anomali, assicurando che il software soddisfi le esigenze del cliente.

I test possono essere suddivisi principalmente in due categorie:

- Test unitari: questi test valutano la correttezza del codice a livello di singole componenti, spesso verificando il comportamento di ogni metodo rispetto ai valori attesi. XUnit è uno dei framework più utilizzati per l'automazione di questi test, specialmente nel contesto di programmazione C#.
- Test funzionali: Questi test esaminano il corretto funzionamento del sistema software nel suo complesso, trattandolo come una "scatola nera" e verificando l'output in base agli input forniti.

Nel caso specifico dell'applicazione in questione, si è deciso di concentrarsi principalmente sui test unitari utilizzando XUnit.

4.2 Individuazione dei casi di test

Durante la fase preliminare del processo di testing, si è condotta un'analisi e una revisione del codice al fine di selezionare le classi e i metodi da sottoporre ai test. Il criterio utilizzato per individuare tali metodi consiste nel dare maggiore importanza a quelli che sono responsabili dell'introduzione, dell'aggiornamento e della modifica di nuove istanze.

I test citati sono:

TestRegistraCabina:

- Scopo: Verificare che il metodo inserisca correttamente una cabina nella prenotazione in corso.
- Realizzazione: Iniziamo caricando le cabine disponibili e registrando una cabina specifica. Successivamente, recuperiamo la prenotazione in corso. Infine, verifichiamo che la cabina inserita nella prenotazione in corso non sia nulla, il che indica che la cabina è stata inserita correttamente

TestRegistraCliente:

- Scopo: Verificare che il metodo inserisca correttamente un cliente nella prenotazione in corso.
- Realizzazione: Iniziamo caricando le cabine e registrando una cabina e un cliente con dati specifici. Successivamente, recuperiamo la prenotazione in corso. Infine, verifichiamo che venga restituito un oggetto cliente non null dalla prenotazione in corso, il che indica che il cliente è stato inserito correttamente

TestRegistraPrenotazione:

- Scopo: Verificare che il metodo registri correttamente una prenotazione nel sistema.
- Realizzazione: Iniziamo caricando le cabine disponibili, registrando una cabina specifica e un cliente con dati specifici. Successivamente, registriamo una prenotazione. Infine, recuperiamo l'elenco delle prenotazioni e verifichiamo che il cliente della prima prenotazione non sia nullo, il che indica che la prenotazione è stata inserita correttamente.

TestCreaNuovoServizioInCabina:

- Scopo: Verificare che il metodo crei correttamente un nuovo servizio in cabina
- Realizzazione: Iniziamo caricando le cabine disponibili, registrando una cabina specifica, un cliente con dati specifici e una prenotazione. Successivamente, effettuiamo il check-in per la prenotazione e creiamo un nuovo servizio in cabina. Infine, verifichiamo che il servizio in cabina in corso non sia nullo, il che indica che il servizio in cabina è stato creato correttamente.

TestRegistraPortate:

- Scopo: Verificare che il metodo registri correttamente una portata nel sistema
- Realizzazione: Iniziamo caricando le cabine disponibili, registrando una cabina specifica, un cliente con dati specifici e una prenotazione. Successivamente, effettuiamo il check-in per la prenotazione e creiamo un nuovo servizio in cabina. Creiamo una portata e la registriamo. Infine, verifichiamo che la portata sia stata correttamente registrata verificando che l'elenco delle portate del servizio in cabina in corso non sia vuoto

TestRegistraServizioInCabina:

- Scopo: Verificare che il metodo registri correttamente un servizio in cabina
- Realizzazione: Iniziamo registrando le portate. Successivamente, registriamo un servizio in cabina. Infine, verifichiamo che il servizio in cabina in corso sia nullo (indicando che il servizio è stato registrato e non è più "in corso") e che l'ordine in corso del primo servizio in cabina non sia nullo, il che indica che il servizio in cabina è stato registrato correttamente

TestPeriodoVariazioniPeriodoValido:

- Scopo: Verificare che il metodo aggiunga correttamente un periodo di variazione
- Realizzazione: Iniziamo salvando il numero attuale di periodi di variazione. Successivamente, aggiungiamo un nuovo periodo di variazione con una data di inizio e di fine valide. Infine, verifichiamo che il conteggio dei periodi di variazione sia aumentato di uno, il che indica che il nuovo periodo di variazione è stato aggiunto

TestDisabilitaCabina:

- Scopo: Verificare che il metodo disabiliti correttamente una cabina
- Realizzazione: Creiamo una cabina e la aggiungiamo alla lista delle cabine. Successivamente, chiamiamo il metodo per disabilitare la cabina. Infine, verifichiamo che il metodo restituisca true e che la disponibilità della cabina sia impostata su false, il che indica che la cabina è stata correttamente disabilitata

TestAbilitaCabina:

- Scopo: Verificare che il metodo abiliti una cabina precedentemente disabilitata
- Realizzazione: Iniziamo disabilitando una cabina. Successivamente, chiamiamo il metodo per abilitare la cabina. Infine, verifichiamo che il metodo restituisca true e che la disponibilità della cabina sia impostata su true, il che indica che la cabina è stata correttamente abilitata

TestRimuoviPortata:

- Scopo: Verificare che il metodo modifichi correttamente la disponibilità di una portata nel sistema
- Realizzazione: Creiamo una portata e la aggiungiamo all'elenco delle portate. Successivamente, chiamiamo il metodo RimuoviPortata con il codice della portata. Infine, verifichiamo che il metodo restituisca true e che la disponibilità della portata sia impostata su false, il che indica che la portata è stata correttamente rimossa

TestInserisciPortate:

- Scopo: Verificare che il metodo aggiunga correttamente una nuova portata all'elenco delle portate
- Realizzazione: Otteniamo la dimensione corrente dell'elenco delle portate e la salviamo in expected. Successivamente, chiamiamo il metodo AggiornaMenu per aggiungere una nuova portata all'elenco. Incrementiamo expected di uno. Infine, verifichiamo che la dimensione dell'elenco delle portate sia uguale a expected, il che indica che la nuova portata è stata aggiunta correttamente

TestCalcolaConto:

- Scopo: Verificare che il metodo calcoli correttamente il totale del conto considerando sia le portate che i servizi
- Realizzazione: Iniziamo caricando le cabine e registrando una cabina e un cliente. Successivamente, registriamo una prenotazione e facciamo il check-in. Creiamo un servizio in cabina e registriamo due portate con quantità diverse. Calcoliamo il totale atteso delle portate. Registriamo il servizio in cabina e aggiungiamo un servizio alla prenotazione. Infine, chiamiamo il metodo CalcolaConto e verifichiamo che il risultato sia uguale al totale atteso delle portate più il prezzo del servizio, il che indica che il conto è stato calcolato correttamente

TestCalcolaContoSconto:

- Scopo: Verificare che il metodo calcoli correttamente il totale del conto considerando sia le portate che i servizi, e applicando correttamente lo sconto quando il cliente ha prenotato una suite
- Realizzazione: Iniziamo caricando le cabine e registrando una cabina e un cliente. Successivamente, registriamo una prenotazione e facciamo il check-in. Creiamo un servizio in cabina e registriamo una portata con una quantità specifica. Calcoliamo il totale atteso delle portate, considerando lo sconto. Registriamo il servizio in cabina e aggiungiamo un servizio alla prenotazione. Infine, chiamiamo il metodo CalcolaConto e verifichiamo che il risultato sia uguale al totale atteso delle portate più il prezzo del servizio, il che indica che il conto è stato calcolato correttamente con lo sconto applicato