



UNIVERSITÀ DEGLI STUDI
DI SALERNO



HOTEL COLOSSUS

YOUR RELAX OUR HAPPINESS

TEST

**SPECIFICATION
DOCUMENT**

Test Specification Document

Hotel colossus

Coordinatore del progetto:

Nome	Matricola
Stefano Santoro	0512120778

Partecipanti:

Nome	Matricola
Giovanni Riccardi	0512119392
Renato Natale	0512119641
Samuele Valiante	0512119125

Scritto da:	Giovanni Riccardi
-------------	-------------------

Revision History

Data	Versione	Descrizione	Autore
21/12/2025	1.0	Prima versione TestSpecification	Giovanni Riccardi
08/01/2026	2.0	Seconda versione TestSpecification	Giovanni Riccardi

Sommario

<i>TEST CASE SPECIFICATION TABLE</i>	3
GOVERNANTE	3
Test suite: Cambio stato Camera	3
Test suite: Check-Out	6
FUNZIONALITÀ AUTENTICAZIONE	16
FUNZIONALITÀ REGISTRAZIONE PRENOTAZIONE.....	20

TEST CASE SPECIFICATION TABLE

GOVERNANTE**Test suite: Cambio stato Camera**

TC	INPUT	RAZIONALE CASO DI TEST	ORACOLO
TC1	Nome stato= in servizio Stato camera corrente = Out of order, Environment="Database online".	Test della funzionalità della governante, sono stati scelti questi input per testare la relazione camera – modifica stato camera per garantire la corretta adempienza della mansione. In questo caso la governante deve modificare la camera da stato "out of order" in stato "in servizio "	Stato finale camera="in servizio"
TC2	Nome stato= in servizio Stato camera corrente= in pulizia, Environment="Database	Test della funzionalità della governante, sono stati scelti questi input per testare la relazione camera – modifica stato camera per garantire la corretta adempienza della mansione. In questo caso la governante modifica lo stato da "in pulizia"	Stato finale = "in servizio"

	online".	a "in servizio"	
TC3	Nome stato= Out of Order Stato camera corrente= in pulizia, Environment="Database online".	Test della funzionalità della governante, sono stati scelti questi input per testare la relazione camera – modifica stato camera per garantire la corretta adempienza della mansione. In questo caso la governante modifica lo stato da " Out of Order" in stato " in pulizia"	Stato finale = in pulizia
TC4	Nome stato= In Pulizia Stato camera corrente= Out of Order Environment="Database online".	Test della funzionalità della governante, sono stati scelti questi input per testare la relazione camera – modifica stato camera per garantire la corretta adempienza della mansione. In questo caso la camera si trova in uno stato di manutenzione per poi passare a uno stato di pulizia	Stato finale = "Out of Order".
TC5	Stato attuale = "In servizio" Nuovo stato richiesto= "Out of Order" Environment="Database online".	Test della funzionalità della governante , sono stati scelti questi input per testare la relazione camera – modifica stato camera per garantire la corretta adempienza della mansione. In questa caso la camera si trova nello stato in Servizio e si vuole passare allo stato in Out of Order	Stato Finale = "Out of Order".
TC6	Stato attuale="In servizio" Nuovo stato richiesto = "In pulizia", Environment = "Database online".	Test della funzionalità della governante , sono stati scelti questi input per testare la relazione camera - modifica stato camera per garantire la corretta adempienza della mansione. In questa caso la camera si trova nello stato in Servizio e si vuole passare allo stato In pulizia	Stato Finale = "In pulizia"
TC7	Stato attuale="Out Of Order" Nuovo stato richiesto = "In pulizia" Environment = "Database offline".	Test della funzionalità della governante , sono stati scelti questi input per testare la relazione camera - modifica stato camera per garantire la corretta adempienza della mansione. In questo caso ci sta un messaggio di errore di connessione	Errore di connessione: SQLException(" Lo stato della camera non viene modificato").

Test Specification Document
Hotel colossus



Test Specification Document
Hotel colossus



Test suite: Check-Out

TC	INPUT	RAZIONALE CASO DI TEST	ORACOLO
TC1	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00€", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="Match", Match servizi="Match", Match camere="Match", Environment="Database online"</p>	<p>Il test verifica la corretta esecuzione della funzionalità di check-out in condizioni operative ordinarie. Gli input forniti rappresentano un flusso standard di utilizzo del sistema, corrispondente a uno scenario tipico di check-out senza servizi aggiuntivi.</p> <p>L'obiettivo del test è accertare che il sistema completi correttamente il processo di check-out quando si ritrova un componente della prenotazione vuoto, come il servizio, che non dia quindi errori dovuti al fatto di avere un oggetto vuoto o null</p>	<p>Check-out completato con successo, ricevuta fiscale generata correttamente, tutte le camere sono impostate ad uno stato diverso da "libera", prenotazione non più attiva.</p>
TC2	<p>Numero clienti="1", Numero camere="1", Numero servizi="1", Metodo di pagamento="Carta di credito", Costo totale="315.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="Match", Match servizi="Match", Match camere="Match", Environment="Database online"</p>	<p>Il test verifica la corretta esecuzione della funzionalità di check-out in condizioni operative ordinarie. Gli input forniti rappresentano un flusso standard di utilizzo del sistema, corrispondente a uno scenario tipico di check-out senza servizi aggiuntivi.</p> <p>L'obiettivo del test è accertare che il sistema completi correttamente il processo di check-out quando tutti i dati sono coerenti e correttamente inseriti, garantendo la corretta gestione dello stato della prenotazione e delle camere.</p>	<p>Check-out completato con successo, ricevuta fiscale generata correttamente (incluso un solo servizio), tutte le camere sono impostate ad uno stato diverso da "libera", prenotazione non più attiva.</p>

TC3	<p>Numero clienti="1", Numero camere="1", Numero servizi="2", Metodo di pagamento="Carta di credito", Costo totale="340.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="Match", Match servizi="Match", Match camere="Match", Environment="Database online"</p>	<p>Il test verifica la corretta esecuzione della funzionalità di check-out in presenza di servizi aggiuntivi, infatti una persona può avere più servizi nella prenotazione posso usufruire di più servizi accessori associati alla prenotazione. Gli input rappresentano uno scenario ordinario in cui, oltre alla camera, vengono contabilizzati più servizi accessori.</p> <p>Il check-out viene completato con successo. Il sistema genera correttamente la ricevuta fiscale, includendo tutti i servizi associati alla prenotazione. Al termine dell'operazione, le camere risultano impostate in uno stato diverso da "libera" e la prenotazione viene correttamente marcata come non più attiva.</p>	<p>Check-out completato con successo, ricevuta fiscale generata correttamente (inclusi più servizi), tutte le camere sono impostate ad uno stato diverso da "libera", prenotazione non più attiva.</p>
TC4	<p>Numero clienti="2", Numero camere="1", Numero servizi="2", Metodo di pagamento="Carta di credito", Costo totale="340.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="Match", Match servizi="Match", Match camere="Match", Environment="Database online"</p>	<p>Il test verifica la corretta esecuzione della funzionalità di check-out in uno scenario in cui più clienti sono associati alla stessa camera e alla medesima prenotazione. Sono inoltre presenti servizi aggiuntivi contabilizzati nel costo totale.</p> <p>Il check-out viene completato con successo. Il sistema genera correttamente la ricevuta fiscale includendo tutti i clienti associati alla camera e i servizi utilizzati. Al termine dell'operazione, le camere vengono impostate allo stato "libera" e la prenotazione viene correttamente aggiornata allo stato "completata".</p>	<p>Check-out completato con successo, ricevuta fiscale generata correttamente (incluso più clienti in una camera), tutte le camere impostate a "libera", prenotazione impostata a "completata"</p>

TC5	<p>Numero clienti="2", Numero camere="2", Numero servizi="2", Metodo di pagamento="Carta di credito", Costo totale="640.00 €", Stato camere="Libera,Libera", Stato prenotazione="Non attiva", Match clienti="Match", Match servizi="Match", Match camere="Match", Environment="Database online"</p>	<p>Il test verifica la corretta esecuzione della funzionalità di check-out in uno scenario che coinvolge più clienti associati a più camere all'interno della stessa prenotazione, con la presenza di servizi aggiuntivi contabilizzati nel costo totale.</p> <p>Al termine dell'operazione, tutte le camere vengono impostate allo stato "libera" e la prenotazione viene correttamente aggiornata allo stato "completata".</p>	<p>Check-out completato con successo, ricevuta fiscale generata correttamente (incluso più clienti in più camere), tutte le camere impostate a "libera", prenotazione impostata a "completata"</p>
TC6	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Assegno", Costo totale="300.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out in presenza di un metodo di pagamento non supportato dal sistema. Gli input rappresentano un flusso operativo formalmente corretto, ad eccezione del metodo di pagamento selezionato.</p>	<p>Visualizzazione messaggio di errore "Metodo di pagamento non valido"</p>

TC7	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="350.00 € (incoerente)", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out in presenza di un costo totale formalmente valido (positivo) ma non coerente con i dati associati alla prenotazione, come numero di camere e servizi selezionati.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare un messaggio di errore esplicativo: "Costo totale positivo ma non coerente con i dati della prenotazione".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Costo totale positivo ma non coerente con i dati della prenotazione")</p>
TC8	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="-50.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando il costo totale della prenotazione è negativo, situazione formalmente non valida.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare un messaggio di errore esplicativo: "Costo totale non valido"</p>	<p>Visualizzazione messaggio di errore "Costo totale non valido"</p> <p>throw new RuntimeException("Costo totale non valido")</p>

TC9	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Occupata", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando una o più camere associate alla prenotazione risultano ancora occupate, impedendo la conclusione regolare del processo.</p> <p>Il sistema deve essere in grado di bloccare il check-out</p>	<p>Visualizzazione messaggio di errore "Almeno una camera risulta ancora occupata"</p> <p>throw new RuntimeException("Almeno una camera risulta ancora occupata")</p>
TC10	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Libera", Stato prenotazione="Attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando la prenotazione risulta ancora attiva. In questa situazione, il sistema non deve consentire il completamento del check-out fino a quando lo stato della prenotazione non viene aggiornato correttamente.</p> <p>Il sistema deve bloccare il check-out e visualizzare un messaggio</p>	<p>Visualizzazione messaggio di errore "</p> <p>throw new RuntimeException("Prenotazione ancora attiva")</p>

TC11	<p>Numero clienti="2", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Libera,Libera", Stato prenotazione="Non attiva", Match clienti="Ricevuta contiene 1 cliente, prenotazione contiene 2 clienti (no match)", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando vi è un disallineamento tra i clienti indicati nella ricevuta fiscale e quelli effettivamente presenti nella prenotazione. Gli input simulano una situazione in cui almeno un cliente della prenotazione non è incluso nella ricevuta.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare un messaggio di errore esplicativo: "Almeno un cliente della prenotazione non è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Almeno un cliente della prenotazione non è presente nella ricevuta fiscale")</p>
TC12	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Camera", Stato prenotazione="Non attiva", Match clienti="Ricevuta non contiene nessun cliente, prenotazione contiene 1 cliente (no match)", Match servizi="match", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando nessuno dei clienti presenti nella prenotazione è incluso nella ricevuta fiscale. Gli input simulano un caso in cui la ricevuta generata è completamente disallineata rispetto ai dati della prenotazione.</p> <p>Il sistema deve bloccare il completamento del check-out e visualizzare un messaggio di errore esplicativo: "Nessun cliente della prenotazione è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Nessun cliente della prenotazione è presente nella ricevuta fiscale")</p>

TC13	<p>Numero clienti="1", Numero camere="1", Numero servizi="2", Metodo di pagamento="Carta di credito", Costo totale="348.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="Ricevuta contiene 1 servizio, prenotazione contiene 2 servizi (no match)", Match camere="match", Environment="Database online"</p>	<p>Il test verifica il comportamento della funzionalità di check-out quando la ricevuta fiscale non include tutti i servizi associati alla prenotazione. Gli input simulano uno scenario in cui almeno un servizio della prenotazione non è riportato correttamente nella ricevuta.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare un messaggio di errore esplicativo: "Almeno un servizio della prenotazione non è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Almeno un servizio della prenotazione non è presente nella ricevuta fiscale")</p>
TC14	<p>Numero clienti="1", Numero camere="1", Numero servizi="1", Metodo di pagamento="Carta di credito", Costo totale="318.00 €", Stato camere="Libera", Stato prenotazione="non attiva", Match clienti="match", Match servizi="Ricevuta non contiene nessun servizio, prenotazione contiene 1 servizio (no match)", Match camere="match", Environment="match"</p>	<p>Verifica il comportamento della funzionalità di check-out quando nessun servizio della prenotazione è presente nella ricevuta fiscale.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare il messaggio: "Nessun servizio della prenotazione è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Nessun servizio della prenotazione è presente nella ricevuta fiscale")</p>

TC15	<p>Numero clienti="1", Numero camere="2", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="600.00 €", Stato camere="Libera, Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="Ricevuta contiene 1 camera, prenotazione contiene 2 camere (no match)", Environment="Database online"</p>	<p>Verifica il comportamento della funzionalità di check-out quando almeno una camera della prenotazione non è inclusa nella ricevuta fiscale.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare il messaggio: "Almeno una camera della prenotazione non è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>throw new RuntimeException("Almeno una camera della prenotazione non è presente nella ricevuta fiscale")</p>
TC16	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="Ricevuta non contiene nessuna camera, prenotazione contiene 1 camera (no match)", Environment="Database online"</p>	<p>Verifica il comportamento della funzionalità di check-out quando nessuna camera della prenotazione è presente nella ricevuta fiscale.</p> <p>Il sistema deve impedire il completamento del check-out e visualizzare il messaggio: "Nessuna camera della prenotazione è presente nella ricevuta fiscale".</p>	<p>Visualizzazione messaggio di errore</p> <p>RuntimeException("Nessuna camera della prenotazione è presente nella ricevuta fiscale")</p>

TC17	<p>Numero clienti="1", Numero camere="1", Numero servizi="0", Metodo di pagamento="Carta di credito", Costo totale="300.00 €", Stato camere="Libera", Stato prenotazione="Non attiva", Match clienti="match", Match servizi="match", Match camere="match", Environment="Database offline"</p>	<p>Verifica il comportamento della funzionalità di check-out quando il database non è disponibile.</p> <p>Il sistema deve bloccare l'operazione e visualizzare il messaggio: "Database non disponibile".</p>	<p>Visualizzazione messaggio di errore "Database non disponibile" SQL Exception ("connection is null")</p>
------	--	--	--

FUNZIONALITÀ AUTENTICAZIONE

TC1	Username= Receptionist1" Newpassword="pluto123" PasswordTemporanea = "PWD-TMP24SSAF"	Il test verifica la corretta esecuzione della funzionalità di cambio password per un utente receptionist, utilizzando una password temporanea valida. Il sistema deve completare correttamente l'operazione	cambio password effettuato
TC2	Username = "Receptionist13454" , Newpassword = "pluto123" PasswordTemporanea = "PWD-TMP24SSAF"	Il test verifica il comportamento della funzionalità di cambio password quando l'username fornito non corrisponde a nessun utente presente nel database. Il sistema deve bloccare l'operazione e	Throw new SQLException
TC3	Username = "Receptionist13" , Newpassword = "pluto123" PasswordTemporanea = "PWD-TMP24SSAF"	Il test verifica il comportamento della funzionalità di cambio password quando la password temporanea fornita non corrisponde a quella registrata nel database.	Throw new SQLException
TC4	Username = "Receptionist13454" , Newpassword = "pluto123" PasswordTemporanea = "PWD-TMP24SSAF"	Il test verifica il comportamento della funzionalità di cambio password quando la password temporanea fornita è scaduta nel database. Il sistema deve impedire il cambio password	throw new RuntimeException("password temporanea scaduta contattare manager")

TC5	Username = "3413454", Newpassword = "pluto123" PasswordTemporanea = "PWD-TMP24SSAF"	Il test verifica il comportamento della funzionalità di cambio password quando l'username fornito non rispetta il vincolo di formato alfanumerico. Il sistema deve impedire il cambio password	throw new RuntimeException("Inserire un username alfanumerico")
TC6	Username = "" , password = "minni23"	Il test verifica il comportamento della funzionalità di login quando il campo username non è compilato. Il sistema deve bloccare l'operazione	throw new RuntimeException("il campo Username deve essere riempito")
TC7	Username = "dcfsgagu6790_f2sf12xVggbrt3tywr4521edsqrfgSA3" , password = "minni23"	Il test verifica il comportamento della funzionalità di login quando il campo username supera il limite massimo di 40 caratteri. Il sistema deve bloccare l'operazione	throw new ArithmeticException("Errore Overflow Parameter: il campo Username non può avere più di 40 caratteri")
TC8	Username = "Recepationist1" , password = ""	Il test verifica il comportamento della funzionalità di login quando il campo password non è compilato. Il sistema deve bloccare l'operazione	throw new RuntimeException("Errore Missing Parameter: "il campo PASSWORD deve essere riempito")

TC9	Username = "FD1", password = "pass24sfar24vmio!dsf09i erv5fjo1"	Il test verifica il comportamento della funzionalità di login quando il campo password supera la lunghezza massima consentita. Il sistema deve bloccare l'operazione	throw new RuntimeException("Err ore Overflow Parameter: "il campo Password non può avere più di 30 caratteri")
-----	--	---	--

FUNZIONALITÀ REGISTRAZIONE PRENOTAZIONE

TC	INPUT	RAZIONALE CASO DI TEST	ORACOLO
TC1	Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="Nessuno", Trattamento="", Clienti="1", Camere="1", Documento="Patente", Numero documento = "CA123AA", Data Rilascio="10/01/2020", Data Scadenza="10/01/2030"	Il test simula il flusso di creazione di una prenotazione con: 1 cliente, 1 camera, 0 servizi e senza trattamento. Tutti i dati richiesti dall'utente sono corretti e coerenti con le regole del sistema. Il sistema deve completare correttamente la registrazione della prenotazione e deve memorizzarla nel database con tutti i dati inseriti di input inseriti dall'operatore.	Prenotazione registrata con successo nel sistema new prenotazione() inizializzata con i dati di input e inserita nel DB
TC2	Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="1", Trattamento="Mezza pensione", Clienti="1", Camere="1", Documento="Patente", Numero documento = "CA123AA", Data Rilascio="10/01/2020", Data Scadenza="10/01/2030"	Il test simula il flusso di creazione di una prenotazione con: 1 cliente, 1 camera, 1 servizio e con trattamento. Tutti i dati richiesti dall'utente sono corretti e coerenti con le regole del sistema. Il sistema deve completare correttamente la registrazione della prenotazione e deve memorizzarla nel database con tutti i dati inseriti di input inseriti dall'operatore.	Prenotazione registrata con successo nel sistema new prenotazione() inizializzata con i dati di input e inserita nel DB

TC3	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Pensione completa", Clienti="2", Camere="1", Documento="CID", Numero documento = "CA123AA", Data Rilascio="10/01/2020", Data Scadenza="10/01/2030"</p>	<p>Il test simula il flusso di creazione di una prenotazione con:</p> <p>2 clienti, 1 camera, 3 servizi e con trattamento.</p> <p>Tutti i dati richiesti dall'utente sono corretti e coerenti con le regole del sistema. Il sistema deve completare correttamente la registrazione della prenotazione e deve memorizzarla nel database con tutti i dati inseriti di input inseriti dall'operatore.</p>	<p>Prenotazione registrata con successo nel sistema</p> <p>new prenotazione() inizializzata con i dati di input e inserita nel DB</p>
TC4	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="2", Trattamento="Solo pernottamento", Clienti="3", Camere="2", Doc="Passaporto", Numero documento = "CA123AA", Data Rilascio="10/01/2020", Data Scadenza="10/01/2030"</p>	<p>Il test simula il flusso di creazione di una prenotazione con:</p> <p>2 clienti, 2 camere, 3 servizi e con trattamento.</p> <p>Tutti i dati richiesti dall'utente sono corretti e coerenti con le regole del sistema. Il sistema deve completare correttamente la registrazione della prenotazione e deve memorizzarla nel database con tutti i dati inseriti di input inseriti dall'operatore.</p>	<p>Prenotazione registrata con successo nel sistema</p> <p>new prenotazione() inizializzata con i dati di input e inserita nel DB</p>

TC5	Data arrivo ="10/01/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"	Il test simula il tentativo di registrare una prenotazione con data di arrivo passata (precedente alla data odierna) . il sistema deve verificare: <ol style="list-style-type: none"> 1. Se il parametro menzionato sia o meno precedente alla data odierna. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente 	Visualizzazione messaggio di errore "Data di arrivo non può essere passata" throw new InvalidInputExcpetion("Data di arrivo non può essere passata")
TC6	Data arrivo ="", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"	Il test simula il tentativo di registrare una prenotazione senza specificare la data di arrivo . Il sistema deve verificare: <ol style="list-style-type: none"> 1. Se il parametro menzionato sia inserito o meno. 2. Blocchi la registrazione della prenotazione in caso di errore. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente. 	Visualizzazione messaggio di errore "Data di arrivo è obbligatoria, il campo non può essere vuoto" throws new InvalidInputExcpetion("Data di arrivo è obbligatoria, il campo non può essere vuoto")

TC7	<p>Data arrivo="17/03/2026", Data partenza="17/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui la data di partenza coincide con la data di arrivo. il sistema deve</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia o meno precedente alla data di arrivo. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente 	<p>Visualizzazione messaggio di errore "Data di partenza deve essere successiva alla data di arrivo"</p> <pre>throw new InvalidInputExcpetion("Data di partenza deve essere successiva alla data di arrivo")</pre>
TC8	<p>Data arrivo="17/03/2026", Data partenza="10/01/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrare una prenotazione con data di partenza passata (precedente alla data odierna). il sistema deve verificare:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia o meno precedente alla data odierna. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente 	<p>Visualizzazione messaggio di errore "Data di partenza non può essere passata"</p> <pre>throws new InvalidInputException("Data di partenza non può essere passata")</pre>

TC9	<p>Data arrivo="17/03/2026", Data partenza="", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrare una prenotazione senza specificare la data di partenza. Il sistema deve verificare:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia inserito o meno. 2. Blocchi la registrazione della prenotazione in caso di errore. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente. 	<p>Visualizzazione messaggio di errore "Data di partenza è obbligatoria"</p> <p>Throws new InvalidInputException("Data di partenza è obbligatoria")</p>
TC10	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza Pensione", Clienti="", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrare una prenotazione senza scegliere alcun cliente. Il sistema deve verificare:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia inserito o meno. 2. Blocchi la registrazione della prenotazione in caso di errore. 3. In casi erronei, forniscia un messaggio di errore chiaro all'utente. 	<p>Visualizzazione messaggio di errore "Almeno un cliente deve essere selezionato"</p> <p>Throws new InvalidInputException("Almeno un cliente deve essere selezionato")</p>

TC11	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrare una prenotazione senza scegliere alcuna camera. Il sistema deve verificare:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia inserito o meno. 2. Blocchi la registrazione della prenotazione in caso di errore. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente. 	<p>Visualizzazione messaggio di errore "Almeno una camera deve essere selezionata"</p> <pre>Throws new InvalidInputException("Almeno una camera deve essere selezionata")</pre>
TC12	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di inserimento di una prenotazione valida sotto tutti gli altri aspetti, ma priva del tipo di documento. L'obiettivo è verificare che il sistema:</p> <ol style="list-style-type: none"> 1. Rilevi l'assenza del tipo di documento associato alla registrazione delle prenotazioni. 2. Blocchi la registrazione della prenotazione. 3. Fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Tipo documento è obbligatorio"</p> <pre>throws new InvalidInputException("Tipo documento è obbligatorio")</pre>

TC13	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Tessera Sanitaria", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di inserimento di una prenotazione valida sotto tutti gli altri aspetti, ad eccezione del tipo di documento (ovvero viene fornito un documento non accettabile). L'obiettivo è verificare che il sistema:</p> <ol style="list-style-type: none"> 1. Rilevi se il campo menzionato prima sia accettabile o meno. 2. Blocchi la registrazione della prenotazione. 3. Fornisca un messaggio di errore chiaro e immediato all'utente 	<p>Visualizzazione messaggio di errore "Tipo documento non valido"</p> <pre>throws new InvalidInputException("Tipo documento non valido")</pre>
TC14	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="4", Camere="1", Tipo documento="Tessera sanitaria", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui il numero di clienti inserito eccede la capacità totale della camera. L'obiettivo è verificare che il sistema:</p> <ol style="list-style-type: none"> 1. Rilevi questa incongruenza. 2. Blocchi la registrazione della prenotazione. 3. Fornisca un messaggio di errore chiaro e immediato all'utente 	<p>Visualizzazione messaggio di errore "Numero clienti non corrisponde alla capacità totale delle camere"</p> <pre>throws new InvalidInputException("Numero clienti non corrisponde alla capacità totale delle camere")</pre>

TC15	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="20/02/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrare una prenotazione con data di rilascio del documento futura (successiva alla data odierna). il sistema deve verificare:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia o meno successivo alla data odierna. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro all'utente 	<p>Visualizzazione messaggio di errore "Data rilascio documento non può essere futura"</p> <p>throws new InvalidInputException("Data rilascio documento non può essere futura")</p>
------	--	---	--

TC16	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui la data di rilascio del documento non è stata inserita. L'obiettivo è verificare che il sistema rilevi:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia stato inserito o meno. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Data rilascio documento è obbligatoria"</p> <pre>throws new InvalidInputException("Data rilascio documento è obbligatoria")</pre>
TC17	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza="10/01/2019"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui la data di scadenza del documento è successiva alla data di rilascio del documento. L'obiettivo è verificare che il sistema:</p> <ol style="list-style-type: none"> 1. Rilevi che il parametro menzionato non sia conforme come detto prima. 2. Blocchi la registrazione della prenotazione. 3. Fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Data scadenza documento deve essere successiva alla data di rilascio"</p> <pre>throws new InvalidInputException("Data scadenza documento deve essere successiva alla data di rilascio")</pre>

TC18	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2010", Data scadenza="10/01/2024"</p>	<p>Il test simula l'inserimento di un documento di identità formalmente valido nella struttura delle date, ma non più valido temporalmente, in quanto la data di scadenza è già passata rispetto alla data di sistema.</p> <p>L'obiettivo è verificare che il sistema:</p> <ol style="list-style-type: none"> 1. Confronti la data di scadenza del documento con la data corrente. 2. Identifichi correttamente il documento come scaduto. 3. Blocchi la procedura di prenotazione. 4. Fornisca un messaggio di errore chiaro e coerente all'utente. 	<p>Visualizzazione messaggio di errore "Data scadenza documento non può essere passata"</p> <pre>throws new InvalidInputException("Data scadenza documento non può essere passata")</pre>
TC19	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123AA", Data rilascio="10/01/2020", Data scadenza=""</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui la data di scadenza del documento non è stata inserita.</p> <p>L'obiettivo è verificare che il sistema rilevi:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia stato inserito o meno. 2. Blocchi la registrazione della prenotazione in casi erranei. 3. In casi erranei, fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Data scadenza documento è obbligatoria"</p> <pre>throws new InvalidInputException("Data scadenza documento è obbligatoria")</pre>

TC20	Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123!!", Data rilascio="10/01/2020", Data scadenza="10/01/2030"	Il test simula il tentativo di registrazione di una prenotazione in cui il numero di documento contiene caratteri non accettabili . L'obiettivo è verificare che il sistema rilevi: 1. Se il parametro menzionato contiene dei caratteri non accettabili per il campo inserito. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente.	Visualizzazione messaggio di errore "Numero documento può contenere solo caratteri alfanumerici" throws new InvalidInputException("Numero documento può contenere solo caratteri alfanumerici")
TC21	Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "", Data rilascio="10/01/2020", Data scadenza="10/01/2030"	Il test simula il tentativo di registrazione di una prenotazione in cui il numero di documento non è stato inserito . L'obiettivo è verificare che il sistema rilevi: 1. Se il parametro menzionato sia stato inserito o meno. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente.	Visualizzazione messaggio di errore "Campo numero documento obbligatorio" throws new InvalidInputException("Campo numero documento obbligatorio")

TC22	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123456789AA", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui in cui il numero di documento è inserito ma la sua lunghezza eccede la lunghezza massima consentita dal sistema. L'obiettivo è verificare che il sistema rilevi:</p> <ol style="list-style-type: none"> 1. Se il parametro menzionato sia di una lunghezza conforme o meno con il sistema. 2. Blocchi la registrazione della prenotazione in casi erronei. 3. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Campo numero documento obbligatorio"</p> <p>throws new InvalidInputException("Lunghezza numero documento deve essere minore di 10 caratteri")</p>
TC23	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "CA123!!", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui il numero di documento contiene caratteri non accettabili. L'obiettivo è verificare che il sistema rilevi:</p> <ol style="list-style-type: none"> 4. Se il parametro menzionato contiene dei caratteri non accettabili per il campo inserito. 5. Blocchi la registrazione della prenotazione in casi erronei. 6. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Numero documento può contenere solo caratteri alfanumerici"</p> <p>throws new InvalidInputException("Numero documento può contenere solo caratteri alfanumerici")</p>

TC24	<p>Data arrivo="17/03/2026", Data partenza="20/03/2026", Servizi="3", Trattamento="Mezza pensione", Clienti="1", Camere="1", Tipo documento="Patente", Numero documento = "", Data rilascio="10/01/2020", Data scadenza="10/01/2030"</p>	<p>Il test simula il tentativo di registrazione di una prenotazione in cui il numero di documento non è stato inserito. L'obiettivo è verificare che il sistema rilevi:</p> <ol style="list-style-type: none"> 4. Se il parametro menzionato sia stato inserito o meno. 5. Blocchi la registrazione della prenotazione in casi erronei. 6. In casi erronei, fornisca un messaggio di errore chiaro e comprensibile all'utente. 	<p>Visualizzazione messaggio di errore "Campo numero documento obbligatorio"</p> <p>throws new InvalidInputException("Campo numero documento obbligatorio")</p>
------	--	---	--