

Università degli Studi di Salerno

Corso di Ingegneria del Software

**ClickFly
System Design Document
Versione 1.5**



Data: 21/01/2026

Coordinatore del Progetto:

Nome	Matricola
Fabio Pennarella	

Revision History

Partecipanti:

Outline

1. Introduzione

ClickFly è una piattaforma web per la ricerca e prenotazione di voli, progettata per supportare utenti guest e utenti registrati attraverso un'interfaccia intuitiva e un'architettura modulare.

1.1 Scopo del sistema

Gli utenti del sistema ClickFly si distinguono nelle seguenti categorie:

- **Utenti guest**, ovvero utenti non registrati che possono accedere liberamente alla piattaforma, navigare nella homepage e utilizzare la funzionalità di ricerca voli, senza però poter effettuare prenotazioni o pagamenti;
 - **Utenti registrati**, che possono effettuare il login e accedere a tutte le funzionalità del sistema, tra cui la gestione del carrello, il checkout, la visualizzazione delle prenotazioni effettuate, la gestione del portafoglio e delle informazioni del proprio account.
-

1.2 Obiettivi di design

Durante la fase di **System Design** si è deciso di dare maggiore enfasi ai seguenti **design goals**, ritenuti fondamentali per garantire la qualità e l'affidabilità della piattaforma ClickFly.

1.2.1 Usabilità

Obiettivo:

Garantire un'esperienza utente intuitiva e piacevole che faciliti la navigazione e l'utilizzo della piattaforma di prenotazione voli, migliorando la soddisfazione degli utenti e incentivando il completamento delle operazioni di ricerca e acquisto.

Requisiti chiave:

1. Struttura della navigazione:

- Progettare un'interfaccia di navigazione chiara e ben organizzata che permetta agli utenti di accedere facilmente alle principali funzionalità del sistema, come homepage, ricerca voli, login/registrazione, carrello e area personale.
- Consentire all'utente di tornare rapidamente alla homepage da qualsiasi sezione del sito.
- Organizzare in modo intuitivo il flusso di navigazione che conduce dalla ricerca del volo alla prenotazione.

2. Design dell'interfaccia utente:

- Progettare un'interfaccia utente coerente con l'identità del brand ClickFly, utilizzando colori, font e stili visivi che richiamino il tema del viaggio e del volo.
- Assicurare che tutti gli elementi interattivi (bottoni, form, link) siano facilmente riconoscibili e accessibili, con un comportamento chiaro e prevedibile.

3. Funzionalità di ricerca:

- Implementare una funzionalità di **ricerca voli** efficiente e precisa, basata su parametri fondamentali quali città di partenza, città di arrivo, data del volo e numero di passeggeri.
- Garantire che i risultati della ricerca siano pertinenti, aggiornati e ordinati in modo logico, facilitando il confronto tra le offerte disponibili.

4. Feedback e gestione degli errori:

- Fornire feedback visivo immediato per le azioni degli utenti, come conferme di aggiunta al carrello, rimozione di un volo, completamento del pagamento, login o registrazione riusciti.
- Gestire gli errori in modo chiaro e comprensibile, mostrando messaggi esplicativi e suggerimenti utili per la risoluzione del problema, ad esempio in caso di dati inseriti non validi o operazioni non consentite.

1.2.2 Robustezza

Obiettivo:

Garantire che la piattaforma ClickFly sia stabile, sicura e in grado di gestire situazioni impreviste, offrendo un'esperienza utente continua e affidabile anche in condizioni di carico elevato o in presenza di errori.

Requisiti chiave:

1. Stabilità tecnica:

- Implementare un'infrastruttura server affidabile, garantendo un elevato uptime del sistema e riducendo al minimo i tempi di inattività.
- Eseguire test approfonditi per individuare e correggere bug, adottando pratiche di sviluppo sicure e test automatici.

2. Sicurezza:

- Proteggere i dati sensibili degli utenti utilizzando protocolli di sicurezza come **HTTPS** per le comunicazioni.
- Salvare le password degli utenti tramite tecniche di hashing sicure.

3. Ottimizzazione delle prestazioni:

- Ottimizzare le query SQL per migliorare le prestazioni del database e ridurre i tempi di risposta durante la ricerca dei voli e la gestione delle prenotazioni.
- Monitorare e gestire le risorse del server per prevenire colli di bottiglia e garantire un'esperienza fluida agli utenti.

4. Testing e manutenzione:

- Eseguire test continui e integrati per assicurare che il sistema rimanga robusto anche con l'introduzione di nuove funzionalità.
 - Applicare aggiornamenti regolari per correggere vulnerabilità, migliorare le prestazioni e garantire la sicurezza complessiva della piattaforma.
-

1.2.3 Manutenzione

Obiettivo:

Assicurare che la piattaforma ClickFly rimanga aggiornata, efficiente e priva di bug attraverso attività di manutenzione regolari e proattive, migliorando nel tempo l'esperienza utente e la sicurezza del sistema.

Requisiti chiave:

1. Aggiornamenti regolari:

- Applicare aggiornamenti periodici al software di base (ad esempio Java, server applicativo e database) per correggere vulnerabilità e migliorare le prestazioni.
- Mantenere aggiornate le librerie e i framework utilizzati, garantendo compatibilità e sicurezza.

2. Revisione del codice:

- Eseguire revisioni del codice regolari per individuare e correggere potenziali problemi, migliorare la qualità del codice e assicurare il rispetto degli standard di sviluppo moderni.

Documentazione

- Tenere aggiornata la documentazione del sistema ClickFly, includendo guide per sviluppatori, documentazione dell'architettura e manuali utente, al fine di facilitare lo sviluppo, la manutenzione e l'utilizzo della piattaforma.
 - Mantenere registri delle modifiche dettagliati (changelog) per documentare aggiornamenti, correzioni di bug e l'introduzione di nuove funzionalità, garantendo tracciabilità e trasparenza nell'evoluzione del sistema.
-

1.2.4 Affidabilità

Obiettivo:

Garantire che la piattaforma di prenotazione voli ClickFly sia altamente affidabile, offrendo un'esperienza continua e senza interruzioni agli utenti, riducendo al minimo i tempi di inattività e assicurando l'integrità dei dati e delle transazioni.

Requisiti chiave:

1. Tolleranza ai guasti:

- Progettare il sistema affinché sia in grado di degradare in modo graduale e controllato in caso di problemi (ad esempio indisponibilità temporanea di servizi esterni o del database), evitando il fallimento completo della piattaforma.

2. Integrità dei dati:

- Assicurarsi che tutte le transazioni legate alla prenotazione dei voli, alla gestione del carrello e ai pagamenti siano gestite correttamente, utilizzando proprietà **ACID** (Atomicità, Coerenza, Isolamento, Durabilità) per garantire l'integrità dei dati.
 - Implementare meccanismi robusti di gestione degli errori per prevenire la perdita di dati e mantenere la coerenza del database, anche in presenza di errori o interruzioni improvvise.
-

1.3 Definizioni, abbreviazioni e acronimi

- **temporaneo:** si riferisce, nel contesto della gestione del carrello di un utente guest, a dati non persistenti che esistono solo per la durata della sessione. Alla chiusura della sessione, tali dati vengono eliminati poiché l'utente non è registrato.
 - **proprio:** indica che un attore del sistema (ad esempio un utente registrato) agisce sulla propria istanza di una determinata entità, come il proprio carrello o le proprie prenotazioni.
 - **tutti:** indica che un attore agisce su tutte le istanze di una determinata classe del sistema, ad esempio su tutti i voli o tutte le prenotazioni.
 - **SDD (System Design Document):** documento tecnico che descrive in dettaglio l'architettura, i componenti e le interazioni del sistema ClickFly. Serve come guida per sviluppatori, ingegneri e stakeholder.
 - **RAD (Requirement Analysis Document):** documento che descrive in dettaglio i requisiti funzionali e non funzionali del sistema. Costituisce la base per la progettazione, lo sviluppo e il testing del software.
 - **Problem Statement:** dichiarazione chiara e concisa che definisce il problema che il progetto ClickFly intende risolvere, orientando lo sviluppo dell'intero sistema.
-

1.4 Riferimenti

- Riferimento al **Requirements Analysis Document (RAD)** di ClickFly
 - Riferimento al **Problem Statement** del progetto ClickFly
-

1.5 Panoramica

In questo documento, il **System Design Document (SDD)**, vengono illustrati i dettagli tecnici del design del sistema **ClickFly**.

Per una panoramica generale sul sistema si rimanda al documento del **Problem Statement**, mentre per quanto riguarda le funzionalità e i requisiti del sistema si rimanda al **RAD (Requirement Analysis Document)**.

Nel presente documento viene fornita una descrizione dell'architettura corrente del sistema e di quella proposta, vengono illustrati gli obiettivi di design che si intendono raggiungere e spiegati nel dettaglio attraverso i requisiti chiave.

Viene inoltre descritta la suddivisione del sistema in sottosistemi, definendo il mapping **Hardware/Software** per assegnare ciascun sottosistema a uno specifico hardware.

Il documento include anche una descrizione del controllo degli accessi, delle problematiche di sicurezza e delle **Boundary Conditions**, enfatizzando il controllo generale del software e il corretto funzionamento del sistema.

2. Architettura software attuale

L'architettura software attuale di ClickFly è basata sulla suddivisione del sistema in **tre strati principali**, ciascuno responsabile di specifiche funzionalità. Questo approccio garantisce una chiara separazione delle responsabilità e migliora la manutenibilità e la scalabilità del sistema.

1. Presentation Layer

È responsabile della visualizzazione delle pagine richieste dai client web. Questo strato include:

- **View Components:** componenti utilizzati per la visualizzazione dei dati, come HTML, CSS e pagine JSP.
 - **Controllers:** Servlet Java che ricevono le richieste dagli utenti, le elaborano e determinano quale vista restituire.
 - **Client-Side Logic:** logica lato client per migliorare l'interattività dell'utente, come validazioni JavaScript e utilizzo di AJAX per aggiornamenti asincroni delle informazioni (ad esempio aggiornamento del carrello).
-

2. Service Layer

Funge da intermediario tra il Presentation Layer e il Model del sistema, seguendo il pattern architettonico **MVC**. Le sue responsabilità includono:

- **Service Classes:** classi che implementano la logica di business del sistema ClickFly, come la gestione delle prenotazioni, del carrello, del login e della registrazione.
 - **Transaction Management:** gestione delle transazioni per garantire che le operazioni critiche, come il checkout e il pagamento, siano eseguite in modo coerente e sicuro.
 - **Data Transfer Objects (DTOs):** oggetti utilizzati per il trasferimento dei dati tra il Presentation Layer e il Persistence Layer.
-

3. Persistence Layer

Si occupa della gestione dei dati persistenti del sistema. I suoi principali compiti sono:

- **Data Access Objects (DAOs)**: classi che forniscono metodi per l'interazione con il database MySQL, gestendo operazioni di lettura e scrittura.
- **Database Connection Management**: gestione delle connessioni al database, includendo pool di connessioni per migliorare le prestazioni.
- **Sessioni**: utilizzo di `sessionId` per mantenere informazioni volatili, come il carrello dell'utente o i dati dell'utente autenticato.

3. Architettura software proposta

3.1 Panoramica

A livello architettonico, il sistema software proposto per **ClickFly** coincide con quello attualmente implementato.

L'obiettivo principale della fase di progettazione non è quello di introdurre una nuova architettura, bensì **raffinare e migliorare il sistema esistente**, mantenendo invariate le scelte architettoniche e l'impostazione generale già adottate.

In particolare, si intende consolidare l'architettura a **tre livelli (Presentation Layer, Service Layer, Persistence Layer)**, migliorandone la chiarezza, la modularità e la manutenibilità, senza alterare le tecnologie di base o il modello architettonico utilizzato.

Le migliori proposte riguardano principalmente:

- una migliore separazione delle responsabilità tra i diversi livelli del sistema;
- una gestione più robusta delle transazioni legate alla prenotazione dei voli e al checkout;
- un'ottimizzazione della comunicazione tra i componenti del sistema;
- un miglioramento della gestione delle sessioni e della sicurezza.

In questo modo, ClickFly mantiene una struttura architettonica solida e coerente, già valida per gli obiettivi del progetto, rendendola però più affidabile e pronta ad eventuali estensioni future.

3.2 Scomposizione in sottosistemi

Service Layer

Il Service Layer costituisce il cuore logico dell'applicazione e media tra il Presentation Layer e il Persistence Layer, implementando la logica di business del sistema ClickFly.

Componenti principali:

- **Authentication Service:** responsabile dell'autenticazione degli utenti registrati. Gestisce il login, la registrazione e il controllo delle credenziali, distinguendo tra utenti guest e utenti registrati.
- **Security Service:** garantisce la protezione dei dati sensibili, occupandosi dell'hashing delle password, del controllo degli accessi e della sicurezza delle operazioni critiche.
- **Flight Search Service:** gestisce la ricerca dei voli in base ai criteri inseriti dall'utente (città di partenza, destinazione, data, numero di passeggeri), fornendo risultati aggiornati e coerenti con la disponibilità.
- **Cart Service:** gestisce le operazioni sul carrello, consentendo l'aggiunta, la modifica e la rimozione dei voli selezionati, oltre al calcolo del totale e alla verifica della disponibilità.
- **Booking Service:** gestisce le prenotazioni dei voli, occupandosi della creazione delle prenotazioni, della conferma del pagamento e della gestione dello stato delle prenotazioni.
- **Payment Service:** responsabile della gestione delle operazioni di pagamento, inclusa la verifica del saldo del portafoglio virtuale e la corretta esecuzione delle transazioni.

Presentation Layer

Il *Presentation Layer* è responsabile dell'interazione diretta con l'utente finale e della visualizzazione delle informazioni restituite dal sistema.

Componenti principali:

- **User View:** rappresenta l'interfaccia utente del sistema ClickFly e comprende tutte le viste accessibili dagli utenti (guest e utenti registrati).
Include le pagine per:
 - homepage e presentazione del servizio;

- ricerca dei voli;
- visualizzazione del carrello;
- gestione dell'account utente;
- visualizzazione delle prenotazioni effettuate;
- autenticazione e registrazione.

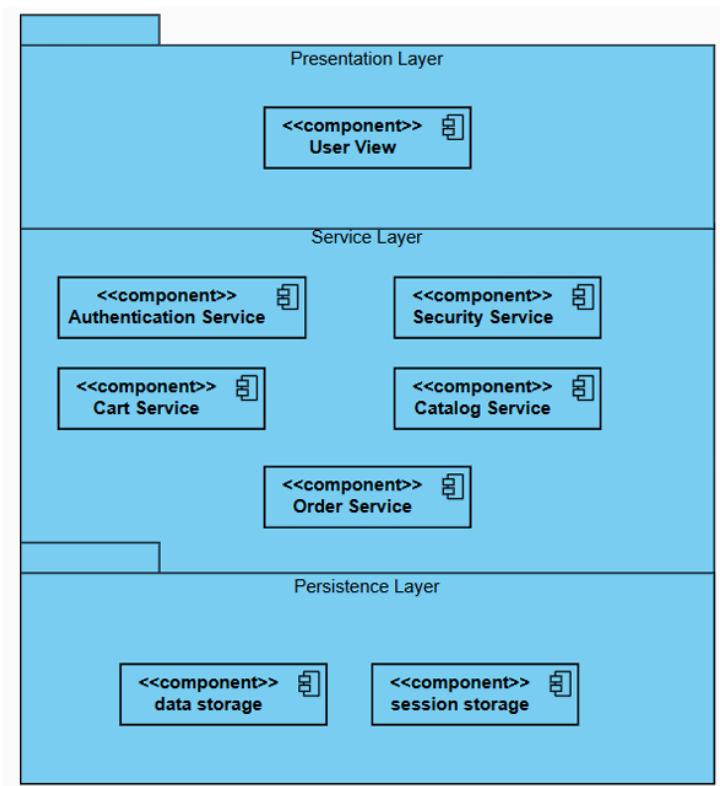
Le viste si occupano esclusivamente della presentazione dei dati, demandando ogni logica di business ai livelli sottostanti.

Persistence Layer

Il **Persistence Layer** si occupa della gestione dei dati persistenti del sistema ClickFly, garantendo integrità e coerenza delle informazioni memorizzate.

Componenti principali:

- **Data Storage:** gestisce la memorizzazione e il recupero persistente dei dati relativi a utenti, voli, prenotazioni, pagamenti e carrello, tramite l'uso di un database relazionale.
- **Session Storage:** gestisce le informazioni temporanee legate alla sessione utente, come l'utente autenticato e il contenuto del carrello, utilizzando identificatori di sessione (*sessionId*).



Oggetti	Catalogo	Ordini	Carrello
Utente non registrato	Visualizza	crea,visualizza	aggiunge,rimuove,modifica,visualizza
Guest	Visualizza,aggiunge,modifica,elimina,visualizza	gestisce stati	aggiunge,rimuove,modifica(temporaneo)

\\\\\\foto schemi

Considerazione finale

La scomposizione in sottosistemi consente una chiara separazione delle responsabilità tra i vari componenti del sistema, migliorando la manutenibilità, la scalabilità e la comprensione complessiva dell'architettura di ClickFly.

Interfacce utente e controllo degli accessi

Ogni utente del sistema ClickFly ha a disposizione una propria interfaccia grafica dedicata, che gli consente di accedere esclusivamente alle funzionalità previste in base allo stato di autenticazione.

L'interfaccia per **utente guest** consente l'accesso ai servizi di base del sistema, come la visualizzazione della homepage e la ricerca dei voli disponibili. Tuttavia, l'utente guest non può completare operazioni di prenotazione o pagamento senza effettuare il login o la registrazione.

L'interfaccia per **utente registrato** consente l'accesso completo alle funzionalità del sistema, tra cui la gestione del carrello, delle prenotazioni, dei pagamenti e delle informazioni personali associate al proprio account.

La schermata di login è unica per tutti gli utenti: una volta inserite le credenziali corrette, il sistema riconosce automaticamente l'utente come autenticato e abilita l'accesso alle funzionalità riservate.

Le password e i dati sensibili relativi ai pagamenti vengono protetti tramite tecniche di hashing sicuro.

3.6 Controllo Software Globale

Il controllo globale del software in **ClickFly** è affidato alla componente **Server**, che gestisce tutte le richieste provenienti dal Client.

Attraverso le **Servlet Java**, il server riceve le richieste HTTP del client, le instrada verso i componenti appropriati e ne delega l'elaborazione ai servizi del *Service Layer*.

Quando necessario, il sistema interagisce con il **DBMS** per recuperare, salvare o modificare dati persistenti; tali operazioni sono gestite tramite le classi **DAO** e i **Java Beans**.

La visualizzazione delle pagine è affidata alle **JSP (Java Server Pages)**, che presentano all'utente i dati elaborati dal sistema.

Il **Web Container Apache Tomcat** gestisce il ciclo di vita delle richieste e integra i protocolli **HTTP** e **HTTPS**, garantendo comunicazioni sicure e affidabili.

3.7 Boundary Conditions

Errore di autenticazione durante la prenotazione

Descrizione:

Se un utente non autenticato tenta di completare una prenotazione, il sistema deve richiedere il login o la registrazione.

Condizioni di ingresso:

- L'utente ha selezionato uno o più voli e accede alla fase di prenotazione senza essere autenticato.

Condizioni di uscita:

- L'utente viene reindirizzato alla pagina di login/registrazione con un messaggio esplicativo.
 - I voli selezionati rimangono salvati nel carrello, pronti per essere prenotati dopo l'accesso.
-

Errore durante il pagamento

Descrizione:

Se si verifica un errore durante il pagamento (ad esempio saldo insufficiente o pagamento rifiutato), il sistema deve garantire che la prenotazione non venga confermata.

Condizioni di ingresso:

- L'utente avvia il processo di pagamento fornendo i dati richiesti.
- Il sistema di pagamento restituisce un errore.

Condizioni di uscita:

- L'utente viene informato dell'errore con un messaggio chiaro e suggerimenti per la risoluzione.
 - La prenotazione non viene confermata e il carrello rimane invariato.
-

Errore di aggiornamento dei dati utente

Descrizione:

Se un utente tenta di aggiornare i propri dati personali e si verifica un errore, il sistema deve preservare le informazioni già salvate.

Condizioni di ingresso:

- L'utente modifica i dati del profilo.
- Un errore impedisce il salvataggio (es. problema di connessione).

Condizioni di uscita:

- L'utente riceve una notifica dell'errore.
 - I dati precedenti rimangono invariati.
 - Le modifiche possono essere ripetute successivamente.
-

Errore di disponibilità dei posti

Descrizione:

Se un volo diventa non disponibile durante il processo di prenotazione, il sistema deve gestire correttamente la situazione.

Condizioni di ingresso:

- L'utente seleziona un volo e procede alla prenotazione.
- I posti vengono esauriti prima del completamento dell'operazione.

Condizioni di uscita:

- Il sistema informa l'utente dell'indisponibilità.
 - L'utente può scegliere un volo alternativo o modificare la ricerca.
-

Errore durante la modifica di una prenotazione

Descrizione:

Se un utente tenta di modificare una prenotazione in lavorazione e si verifica un errore, lo stato della prenotazione deve rimanere coerente.

Condizioni di ingresso:

- Una prenotazione è in fase di elaborazione.
- Viene richiesta una modifica.

Condizioni di uscita:

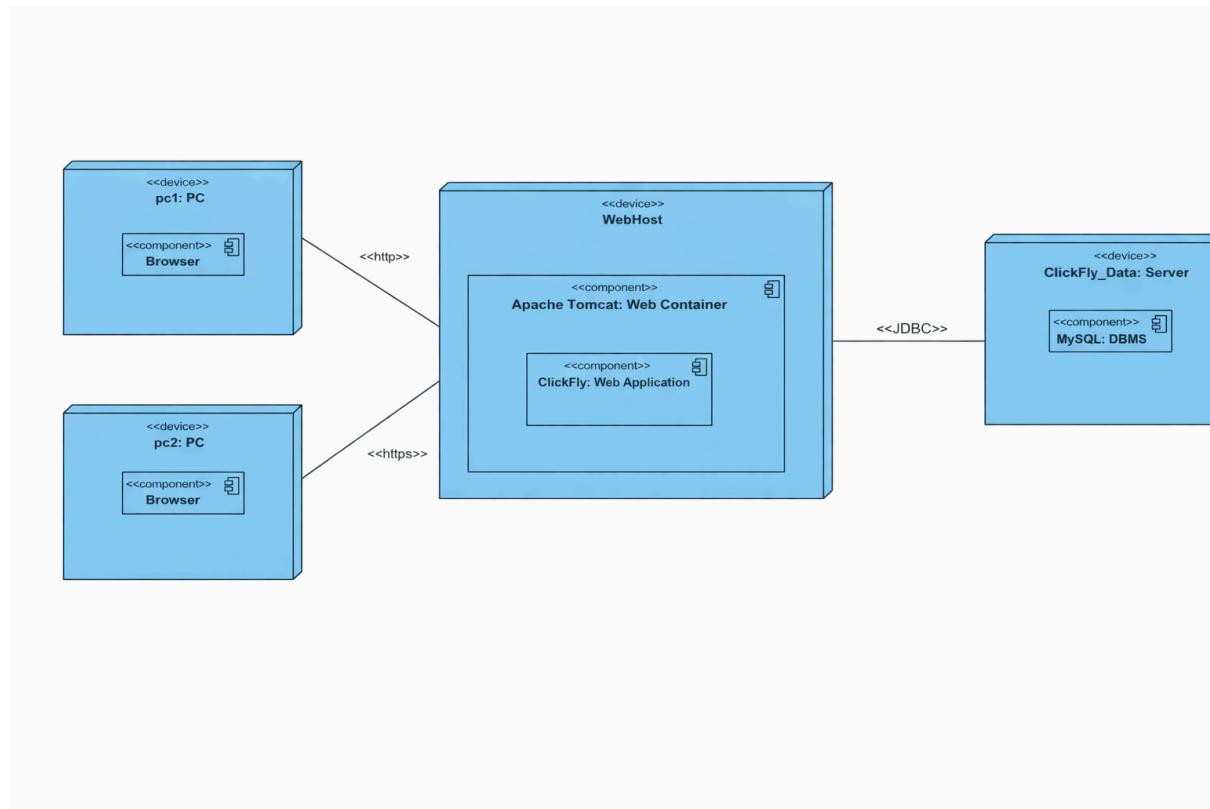
- L'utente viene notificato dell'insuccesso.
 - Lo stato originale della prenotazione viene mantenuto.
-

4. Servizi del Sottosistema

In questa sezione vengono elencati, in forma tabellare, i servizi offerti dal **Service Layer** del sistema ClickFly.

Per evitare ripetizioni, la parola *Service* è omessa accanto al nome di ciascun componente.

(Qui puoi inserire la tabella con: Authentication, FlightSearch, Cart, Booking, Payment, ecc.)



Service Component	Servizi offerti
Authentication	Verifica delle credenziali utente Gestione login/logout Assegnazione ruoli(utente,guest)
Catalog	Gestione dei dettagli dei voli (nome,descrizione,prezzo)
Security	Protezione dei dati sensibili Controllo degli accessi ai sottosistemi
Validation	Controllo della validità degli input utente Validazione dati durante il checkout
Cart	Aggiunta,rimozione e modifica degli articoli nel carrello Visualizzazione del contenuto del carrello
Prenotation	Creazione di nuove prenotazioni Aggiornamento dello stato delle prenotazioni Tracciamento delle prenotazioni e cronologia

\glossario

Termino	Descrizione
ACID	Insieme di proprietà delle transazioni nei database che garantiscono: Atomicità (le operazioni sono eseguite completamente o annullate), Coerenza (il database rimane in uno stato valido), Isolamento (le transazioni non interferiscono tra loro) e Durabilità (i dati persistono anche in caso di guasti).
AJAX	Asynchronous JavaScript and XML: tecnica che consente di aggiornare parti di una pagina web in modo asincrono, senza ricaricare l'intera pagina, migliorando reattività e interattività dell'applicazione.
BEANS	Componenti Java che encapsulano dati e logica applicativa, seguendo convenzioni standard come l'uso di metodi getter e setter per l'accesso controllato agli attributi.
BOUNDARY CONDITIONS	Condizioni limite o scenari eccezionali che il sistema deve gestire correttamente per garantire affidabilità, robustezza e coerenza del comportamento (ad esempio errori di pagamento o indisponibilità dei servizi).
CLIENT SIDE LOGIC	Logica implementata lato client (browser) per migliorare l'interattività e l'esperienza utente, includendo validazioni dei dati, gestione degli eventi e comunicazioni asincrone con il server.
DAO	Data Access Object: pattern di progettazione che separa l'accesso ai dati dalla logica di business, semplificando la gestione delle operazioni sul database e migliorando la manutenibilità del codice.
DATABASE	Raccolta strutturata di dati organizzati in modo da consentire un accesso, una gestione e un aggiornamento efficienti delle

	informazioni.
DBMS	Database Management System: software che consente la creazione, la gestione e l'interrogazione dei database, garantendo sicurezza, integrità e prestazioni efficienti.
Design Goals	Obiettivi di progettazione che guidano lo sviluppo del sistema, includendo aspetti quali usabilità, affidabilità, robustezza e manutenibilità.
FrameWorks	Insieme di strumenti e librerie che forniscono una struttura di base per lo sviluppo software, facilitando l'adozione di best practice e pattern architetturali.
Java Servlet	Classe Java che gestisce richieste e risposte HTTP lato server, utilizzata per implementare la logica di controllo nelle applicazioni web dinamiche.
JavaScript	Linguaggio di programmazione utilizzato nello sviluppo web per creare interazioni dinamiche lato client e migliorare l'esperienza utente.
JSP	JavaServer Pages: tecnologia che consente la generazione di contenuti web dinamici combinando codice Java ed elementi HTML.
MVC	Model-View-Controller: pattern architettonico che separa la logica di business (Model), la gestione delle interazioni (Controller) e la presentazione dei dati (View).
Web Container	Componente del server applicativo che gestisce il ciclo di vita delle applicazioni web, in particolare delle Servlet e delle JSP . Si occupa dell'elaborazione delle richieste HTTP/HTTPS, della gestione delle sessioni, della sicurezza e dell'instradamento delle richieste verso i componenti appropriati.

