

**Università degli Studi di Salerno**  
Corso di Ingegneria del Software

**MalTour**  
**System Design Document**  
**Versione 2.0**

*MalTour* 

Data: 25/12/2021

Progetto: MaITour	Versione: 2.0
Documento: SSD	Data: 25/11/2021

#### Coordinatore del progetto:

<b>Nome</b>	<b>Matricola</b>
Teresa Vitagliano	052105622

#### Partecipanti:

<b>Nome</b>	<b>Matricola</b>
Teresa Vitagliano	0512105622

<b>Scritto da:</b>	Teresa Vitagliano
--------------------	-------------------

## Revision History

Data	Versione	Descrizione	Autore
11/11/2021	1.0	Prima stesura del documento	Teresa Vitagliano
22/11/2021	1.1	criteri	Teresa Vitagliano
23/11/2021	1.2	Proposta architettura software	Teresa Vitagliano
24/11/2021	2.0	Review	Teresa Vitagliano

1.	Introduction.....	4
1.1	Purpose of the system.....	4
1.2	Design goals.....	4
1.3	Definition, acronyms and abbreviations.....	4
2.	Current software architecture.....	4
3.	Proposed software architecture .....	4
3.1	Overview.....	Errore. Il segnalibro non è definito.
3.2	Subsystem decomposition.....	Errore. Il segnalibro non è definito.
3.3	Hardware/software mapping.....	5
3.4	Persistent data management .....	6
3.5	Access control and security.....	6
3.6	Global software control .....	7
3.7	Boundary condition .....	Errore. Il segnalibro non è definito.
4.	Subsystem services .....	7

## 1. Introduction

Nel mercato dei viaggi ci occuperemo del processo di ricerca, calcolo del prezzo, prenotazione e rilascio di biglietti, della gestione degli ordini.

Con servizio, vengono indicati le categorie dei servizi offerti dall'agenzia. Ad esempio "minicrociere".

Con offerta, viene indicato un pacchetto compreso di mezzo di trasporto e pernottamento, oppure un pacchetto compreso solo di pernottamento o di solo mezzo di trasporto. Ad esempio, "MSC mediterraneo con partenza da Napoli, 7 notti in cabina.", farà parte del servizio minicrociere.

Il sistema Maltour offre ai clienti la funzionalità di registrazione, visualizzare le offerte e funzionalità per poterli acquistare, aggiornamento delle offerte, aggiornamento dei servizi, gestione degli ordini e la gestione degli utenti.

### 1.1 Purpose of the system

Il sistema "Maltour" viene sviluppato per agevolare un'agenzia di viaggi in cui è possibile prenotare biglietti d'aereo, hotel, automobili a noleggio, crociere, pacchetti vacanza e vari servizi.

### 1.2 Design goals

#### 1.2.1 Criteri di Performance

Il sistema dovrà garantire tempi di risposta di al massimo 5 secondi.

Il sistema deve contenere l'estendibilità delle offerte.

#### 1.2.2 Criteri di affidabilità

Il sistema dovrà gestire le eccezioni, per input errati e verrà notificato all'utente.

La password deve essere criptata.

#### 1.2.3 Criteri di Manutenzione

Le funzionalità del sistema devono essere debolmente accoppiate (loose coupling).

#### 1.2.4 Criteri and User

Il sistema dovrà avere un menù con non più di sei elementi.

Il sistema dovrà essere responsive.

## 2. Current software architecture

Il sistema proposto è implementato da zero. Esistono altri sistemi simili come ad esempio Vento Viaggi.

## 3. Proposed software architecture

Il sistema proposto è un'applicazione web Based, verrà suddiviso tra client e server, il controllo del formato che si trova nella logica applicativa sarà metà sul server e parte sul client, la logica di controllo invece starà tra il client e il server. Le funzionalità saranno divise in layer logici in base alle differenti funzionalità: presentation, application e data storage.

### 3.1 Decomposizione in sottosistemi

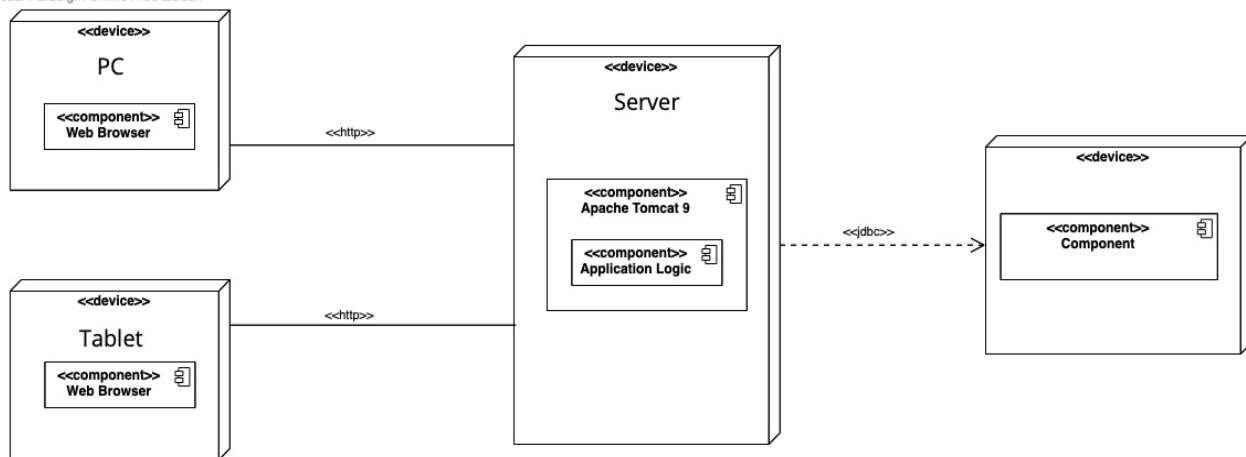
Il modulo Presentation Layer sarà composto da oggetti Bunday e Control.

Il modulo Application Logic Layer sarà composto da oggetti Bean e Model.

Il modulo Data Storage Layer sarà composto dal DBMS.

1. Gestione Utente
2. Gestione Offerte
3. Gestione Servizi
4. Gestione Ordini

Il sistema verrà realizzata su architettura MVC. La struttura hardware è composta da un server centrale e vari client che possono collegarsi. Sul server ci sarà un DBMS per la gestione dei dati persistenti. Il client conterrà presentation layer e business logic (JavaScript), il server conterrà la logica di business (Servlet e JSP) e il layer di data storage. La comunicazione tra i nodi è rappresentata da richieste e risposte http tra client e server, e da query in JDBC tra server e database.



### 3.3 Persistent data management

Per gestire i dati persistenti è stato preferito l'utilizzo di un database relazionale. Poiché serve un ampio spazio di memorizzazione. Inoltre, i database consentono l'accesso concorrente da parte di utenti. Tutte le classi presenti nel Class Diagram diventeranno tabelle nel database. Per la gestione del database si sceglie DBMS MySQL.

Si manda al documento DatiPersistenti\_Maltour.

### 3.4 Access control and security

È stata disegnata una matrice per il controllo degli accessi che mostra le operazioni consentite da ogni attore su ogni oggetto. Il controllo degli accessi è garantito tramite l'utilizzo di credenziali personale univoche. Queste devono essere inserite ogni volta che si desidera utilizzare il sistema per iniziare una nuova sessione, la quale terminerà quando l'utente effettuerà il logout o quando chiuderà l'applicazione o il Web Server.

	Gestione Utente	Gestione Offerte	Gestione Servizi	Gestione Ordini
<b>Visitatore</b>	<ul style="list-style-type: none"> <li>Registrazione</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaOfferta</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaServizi</li> </ul>	
<b>Cliente</b>	<ul style="list-style-type: none"> <li>Login</li> <li>Logout</li> <li>VisualizzaProfilo</li> <li>RimuoviProfilo</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaOfferta</li> <li>RicercaOfferta</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaServizi</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaCarrello</li> <li>AggiungiAlCarrello</li> <li>RimuoviDalCarrello</li> <li>EffettuaOrdine</li> <li>AggiungiComponenti</li> <li>Pagamento</li> </ul>
<b>Gestore Offerte</b>	<ul style="list-style-type: none"> <li>Login</li> <li>Logout</li> <li>VisualizzaProfilo</li> <li>RimuoviProfilo</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaOfferta</li> <li>RicercaOfferta</li> <li>AggiungiOfferta</li> <li>RimuoviOfferta</li> <li>ModificaOfferta</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaServizi</li> <li>AggiungiServizio</li> <li>RimuoviServizio</li> <li>ModificaServizio</li> </ul>	
<b>Gestore Ordini</b>	<ul style="list-style-type: none"> <li>Login</li> <li>Logout</li> <li>VisualizzaProfilo</li> <li>RimuoviProfilo</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaOfferta</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaServizi</li> </ul>	<ul style="list-style-type: none"> <li>VisualizzaOrdini</li> <li>ConvalidaOrdine</li> </ul>

### 3.5 Global software control

Il controllo del flusso globale del sistema è di tipo Event-Driven, Controllo Centralizzato, in quanto fornisce funzionalità che richiedono una continua interazione con l'utente.

Il controllo risiede nel dispatcher che chiama le funzioni mediante callback. (facile da estendere)

## 4. Subsystem Services

**Presentation Layer:** Interfacce che gestiscono l'interfaccia grafica e gli eventi generati dall'interazione dell'utente con il sistema.

### Gestione Utente

Application Logic Layer	Servizio
Registrazione	Inserisce un nuovo cliente all'interno del sistema.
Login	Effettua l'accesso al sistema come utente iscritto.
Logout	Effettua l'uscita dal sistema.
VisualizzaProfilo	Mostra i dati dell'utente quali, nome, username, email.
RimuoviProfilo	Elimina l'account dell'utente all'interno del sistema.

### Gestione Offerte

Application Logic Layer	Servizio
VisualizzaOfferta	Il sistema deve permettere all'utente di visualizzare un'offerta, composta da destinazione, descrizione, data_partenza, ora_partenza, data_ritorno, ora_ritorno, partenza_da, arrivo_a, pernottamento, posti_disponibili, prezzo.
RicercaOfferta	Il sistema deve permettere all'utente di ricercare un'offerta tramite il nome della destinazione.
AggiungiOfferta	Il sistema deve permettere al Gestore Offerte di aggiungere un'offerta.
ModificaOfferta	Il sistema deve permettere al Gestore Offerte di modificare un'offerta.
RimuoviOfferta	Il sistema deve permettere al Gestore Offerte di rimuovere un'offerta.

### Gestione Servizi

Application Logic Layer	Servizio
VisualizzaServizi	Il sistema deve permettere all'utente di visualizzare un servizio, composta da nome e descrizione.
AggiungiServizio	Il sistema deve permettere al Gestore Offerte di aggiungere un servizio.
ModificaServizio	Il sistema deve permettere al Gestore Offerte di modificare un servizio.
RimuoviServizio	Il sistema deve permettere al Gestore Offerte di rimuovere un servizio.

## GestioneOrdini

Application Logic Layer	Servizio
VisualizzaCarrello	Il sistema deve permettere all'utente di visualizzare le offerte che si vorrebbero acquistare.
AggiungiAlCarrello	Il sistema mostra la possibilità all'utente di aggiungere un'offerta all'interno del carrello.
RimuoviDalCarrello	Il sistema mostra la possibilità all'utente di rimuovere un'offerta all'interno del carrello.
EffettuaOrdine	Il sistema permette all'utente di creare un ordine, su un'offerta che si vuole acquistare
AggiungiComponenti	Il sistema permette di aggiungere i componenti di viaggio, inserendo nome, cognome e dati_nascita.
Pagamento	Il sistema permette di inserire il metodo di pagamento
VisualizzaOrdini	Il sistema deve permettere al Gestore Ordini di visualizzare gli ordini effettuati dai clienti.
ConvalidaOrdine	Il sistema deve permettere al Gestore Ordini di convalidare un ordine e segnalare al sistema che l'ordine ha cambiato stato.

**Data Storage Layer:** immagazzina e gestisce i dati persistenti.



