SPECIFICHE SUPPLEMENTARI

Sommario

[1. Introduzione 2](#_Toc364774026)

[1.1. Scopo 2](#_Toc364774027)

[1.2. Portata 2](#_Toc364774028)

[1.3. Definizioni, acronimi e abbreviazioni 2](#_Toc364774029)

[1.4. Riferimenti 2](#_Toc364774030)

[1.5. Panoramica 2](#_Toc364774031)

[2. Presupposti e dipendenze 2](#_Toc364774032)

[3. Usabilità 2](#_Toc364774033)

[3.1. Facilità d'uso 2](#_Toc364774034)

[3.2. Supporto multi-lingua 2](#_Toc364774035)

[4. Affidabilità 2](#_Toc364774036)

[4.1. Disponibilità 2](#_Toc364774037)

[4.2. Caching 2](#_Toc364774038)

[4.3. Degrado dolce delle funzionalità 3](#_Toc364774039)

[5. Performance 3](#_Toc364774040)

[5.1. Reattività 3](#_Toc364774041)

[5.2. Capacità 3](#_Toc364774042)

[5.3. Degrado dolce delle prestazioni 3](#_Toc364774043)

[6. Supportabilità 3](#_Toc364774044)

[6.1. Scalabilità adattabile 3](#_Toc364774045)

[6.2. Rilascio continuo 3](#_Toc364774046)

[7. Vincoli di progettazione 4](#_Toc364774047)

[7.1. Linguaggio OO 4](#_Toc364774048)

[7.2. Framework applicazioni web 4](#_Toc364774049)

[7.3. Client dinamici 4](#_Toc364774050)

[8. Sicurezza 4](#_Toc364774051)

[9. Documentazione in linea per l'utente e requisiti del sistema di aiuto 4](#_Toc364774052)

[10. Interfacce 4](#_Toc364774053)

[10.1. Interfacce utente 4](#_Toc364774054)

[10.2. Interfacce hardware 4](#_Toc364774055)

[10.3. Interfacce software 5](#_Toc364774056)

[10.4. Interfacce di comunicazione 5](#_Toc364774057)

[11. Standards applicabili 5](#_Toc364774058)

# Introduzione

## Scopo

In questo documento vengono introdotte delle richieste aggiuntive per l'implementazione del progetto.

## Portata

## Definizioni, acronimi e abbreviazioni

Verranno utilizzati i seguenti acronimi:

* Tour Operator = TO
* Travel Agency = TA

## Riferimenti

## Panoramica

# Presupposti e dipendenze

# Usabilità

## Facilità d'uso

Ѐ importante che il sistema sia facile da usare, in quanto è quasi totalmente orientato all’uso da parte dei clienti. Particolare attenzione va prestata alla facilità e intuitività delle funzioni di costruzione di un itinerario da parte di un cliente.

## Supporto multi-lingua

La lingua usata nell’interfaccia utente del sistema deve poter essere configurata. Inizialmente devono essere disponibili almeno le seguenti lingue principali: Italiano, Inglese, Francese, Tedesco.

# Affidabilità

## Disponibilità

Il sistema deve avere un alto grado di disponibilità, in modo da permettere ai clienti di accedervi in qualunque momento, specie per quanto riguarda la prenotazione e il pagamento di viaggi. Un blocco del sistema dovrebbe sempre essere risolto entro un massimo di tre ore.

## Caching

Per aumentare il grado di disponibilità del sistema anche in caso di problemi di connessione con i Tour Operator, occorre mantenere una cache locale delle informazioni già ottenute o comunque di potenziale interesse di tali Tour Operator. Tale cache dovrebbe essere aggiornata in modo *batch* abbastanza frequentemente.   
Normalmente, l’accesso alle informazioni dei TO avviene su tale cache locale. E’ possibile prevedere anche che, in caso un’informazione non sia presente nella cache locale, si possa accedere direttamente al sistema del Tour Operator (se disponibile).

## Degrado dolce delle funzionalità

Quando una funzionalità del sistema non è disponibile per cause esterne (es. connessione con banca, accesso ai siti di servizi di trasporto), tutte le altre funzionalità che non dipendono in modo critico da essa dovrebbero continuare ad essere disponibili.

# Performance

## Reattività

Le operazioni effettuate sul sistema sono interattive, e devono quindi avere tempi di risposta rapidi. Con una connessione Internet veloce, la risposta alle azioni degli utenti dovrebbe essere immediata per operazioni semplici (come visualizzare i dettagli di un viaggio) e comunque non superare qualche secondo di attesa anche per le operazioni più complesse (come la ricerca di un itinerario o viaggio).

## Capacità

Si prevede che, a regime, il sistema debba essere capace di gestire un numero di sessioni contemporanee dell’ordine di diverse centinaia.

## Degrado dolce delle prestazioni

In caso di guasto hardware o blocco software, le prestazioni dovrebbero degradare il meno possibile, e soprattutto il sistema non dovrebbe bloccarsi. Occorre quindi prevedere una certa ridondanza delle varie risorse.

# Supportabilità

## Scalabilità adattabile

Dato che non è possibile stimare in modo preciso a priori il numero di utenti (contemporanei) del sistema e la quantità di dati da gestire, deve essere possibile aumentare le risorse del sistema (CPU, RAM, disco, banda di comunicazione) in modo trasparente agli utenti, con un tempo di blocco del sistema nullo o comunque estremamente breve.

## Rilascio continuo

Dato che il sistema è un’applicazione web, si prevede che i nuovi rilasci (bug fix, nuove funzionalità) debbano poter essere attivati con un tempo di blocco del sistema nullo o comunque estremamente breve.

# Vincoli di progettazione

## Linguaggio OO

Al momento non è stato deciso il linguaggio per l’implementazione del sistema, ma esso dovrà certamente essere Object Oriented, in quanto si è deciso di eseguire un’analisi e una progettazione OO.

## Framework applicazioni web

Il linguaggio OO sarà scelto anche in base alla disponibilità per lo stesso di un framework per la costruzione di applicazioni Web che sia efficiente ed affidabile. Per esempio, tecnologie *servlet* e *EJB* in Java oppure *Ruby-on-Rails* in Ruby.

## Client dinamici

Per avere dei client dinamici su browser si ipotizza l’uso di tecniche AJAX.

# Sicurezza

Nella versione corrente ogni accesso al sistema deve essere preceduto da autenticazione, in modo che si sappia sempre chi ha effettuato un’operazione.  
Particolare attenzione va prestata alla memorizzazione e alla comunicazione sicura delle *password* e di dati sensibili dei clienti, quali il numero di carta di credito.

# Documentazione in linea per l'utente e requisiti del sistema di aiuto

Il sistema deve mettere a disposizione un ricco help on-line (anch’esso multi lingua) che aiuti l’utente (specialmente il cliente) a risolvere i dubbi sull’utilizzo del sistema stesso. Oltre alla descrizione delle singole funzionalità, tale help dovrebbe contenere anche dei tutorial di tipo *how-to* su come raggiungere i principali obiettivi degli utenti (es. creazione di un itinerario).

# Interfacce

## Interfacce utente

Occorre prevedere che il sistema sarà usato sia da PC che da tablet e smart-phone. Per ciascuna di questa categoria di device, occorre studiare un’opportuna Interfaccia Utente.

## Interfacce hardware

N/A

## Interfacce software

Il sistema deve comunicare con le seguenti interfacce software di servizi esterni (accessibili via Internet):

* servizi dei Tour Operator
* servizi di trasporto
* servizi di gestione pagamenti con carta di credito

## Interfacce di comunicazione

N/A

# Standards applicabili