Università degli studi di Salerno

***Corso di Laurea in Informatica***

******

***INGEGNERIA DEL SOFTWARE***

***System Design Document***

***“UNI-AirLines”***

# Studenti:

##### Nome Matricola

Santoro Mario 0512104850

Marino Raffaele 0512104508

Pastore Matteo 0512104724

Fortunato Angelo 0512104532

*Anno Accademico: 2018/19*

***SOMMARIO***

**1. Introduction……………………………………………………………..**

1.1. Purpose of the system………………………………………………………………….

1.2. Design goals……………………………………………………………………………….

1.3. Definitions, acronyms and abbreviations……………………………………..

1.4. References…………………………………………………………………………………

**2. Current software architecture……………………………………**

**3. Proposed software architecture………………………………….**

3.1. Subsystem decomposition…………………………………………………………..

3.2. Hardware/software mapping………………………………………………………

3.3. Persistent data management……………………………………………………….

3.4. Access control and security………………………………………………………….

3.5. Global software control……………………………………………………………….

3.6. Boundary conditions…………………………………………………………………..

**4. Subsystem services……………………………………………………**

1. **Introduction**
   1. **Purpose of the system**

L'SDD documenta e tiene traccia delle informazioni necessarie per definire efficacemente l'architettura e la progettazione del sistema per fornire al team di sviluppo una guida sull'architettura dello stesso.L'intento dell'implementazione proposta è quello di fornire un sito web semplice ed efficiente alla compagnia aerea UniAirlines che permetta all'utenza l'acquisto dei biglietti aerei di quest'ultima. Questa soluzione dovrà provvedere ad incrementare il numero di acquisti dei biglietti,riducendo così a 0 i tempi di attesa per l'acquisto e fornire un'assistenza più efficace. Di seguito sono riportare le principali funzionalità che verranno incluse:

-Pannello amministratore,che permette l'inserimento o la cancellazione dei voli;

-Registrazione e Login utente;

-Ricerca voli;

-Visualizzazione news e annunci;

-Check-In online.

* 1. **Design goals**

Gli obiettivi di progettazione rappresentano le qualità desiderate di Bumper e forniscono un insieme coerente di criteri da prendere in considerazione quando si prendono decisioni di progettazione. I seguenti obiettivi di progettazione sono identificati.

* ***Utilizzo di Java*** . Poiché UnisAir deve essere eseguiti su più browser viene scelto Java per l'implementazione. Utilizzando Java, lo stesso codice di UnisAir può essere eseguito su più piattaforme che supportano la Java Virtual Machine.

* ***Nessuna dipendenza dai componenti software commerciali*** . Poiché UnisAir deve essere distribuito gratuitamente da qualsiasi addebito o licenza, non deve fare affidamento su alcun prodotto commerciale.

* ***Usabilità*** . L'uso di UnisAir deve essere intuitivo. L'interfaccia utente grafica deve essere progettata in modo che l’utente comprenda come effettuare le operazioni di interazione con il sito.

* ***Robustezza*** .  Tutte le interazioni devono essere progettate affinché l’utente non sia in grado di inserire dati non validi.

* ***Tempo di risposta*** .  Massimo 30 secondi di attesa nell’interazione con il sistema quando si acquista un biglietto.
* **Multiutente.** Il sistema dovrebbe supportare operazioni che sono effettuate da utenti multipli contemporaneamente.

**• Sicurezza**: La sicurezza è garantita nei limiti da una login e di password non criptata.

**• Estendibilità**: E' consentito, in quanto è possibile aggiungere in futuro, nuove funzionalità al sistema, oppure creare nuove classi, con l’estensione di quelle già esistenti.

**• Attendibilità**: I risultati prodotti dalle pagine dinamiche (le servlet) riguardo la disponibilità o meno di voli sono attendibili nel senso che rispecchiano istante per istante la reale situazione dei voli stessi; non si deve mai verificare che venga visualizzata la disponibilità di un determinato volo se questo non c’è effettivamente.

**1.3. Definitions, acronyms and abbreviations**

* SDD: Software Design Document;
* RAD: Requirements analysis document.
* BROWSER: Explorer, Chrome, Mozilla.
* WebBrowser: Client (utente che accede al sistema)
* WebServer: Server su cui sono memorizzate le risorse.

**1.4. References**

-Problem Statement

-RAD

-https://www.easyjet.com/it

-Object-Oriented Software Engineering Using UML, Patterns, and Java™ Third Edition Bernd Bruegge & Allen H. Dutoit

-http://java.sun.com

**2. Current system architecture**

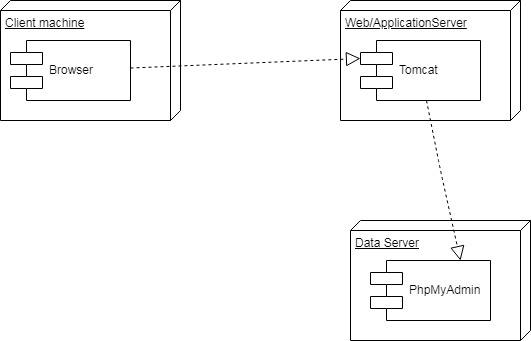
Il sistema è implementato dal nulla in quanto nessun sistema esiste in precedenza, per cui questa fase è di “Greenfield Engineering” e la raccolta dei requisisti viene fatta esclusivamente colloquiando con il cliente e cercando di estrarre i requisiti dalle sue richieste. Esempi simili è possibile individuarli ai seguenti siti: [www.easyjet.com/it](http://www.easyjet.com/it)

**3. Proposed software architecture**

**3.1. Subsystem decomposition**

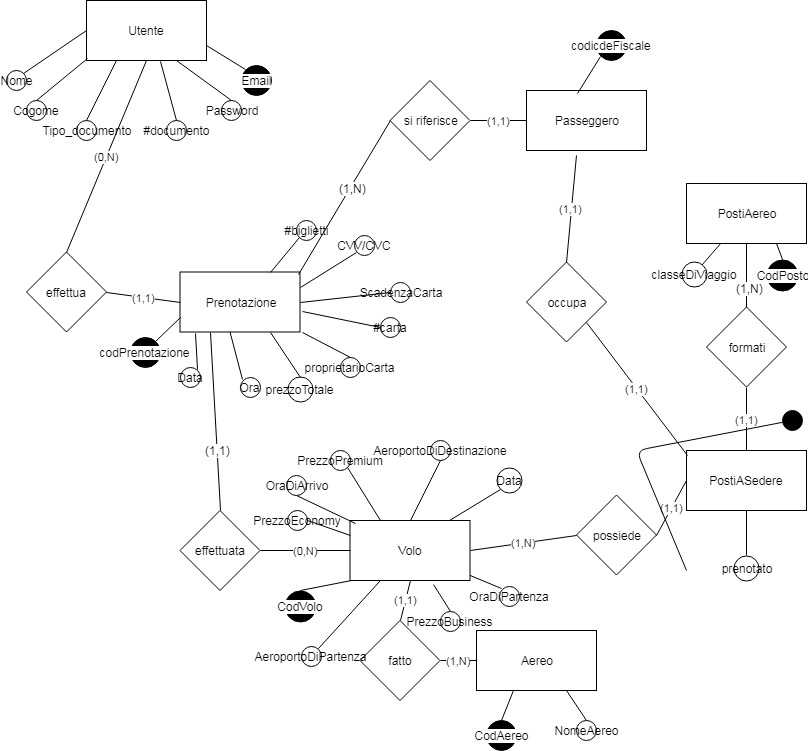
**3.2. Hardware/software mapping**

Il seguente diagramma di implementazione UML illustra la mappatura hardware / software per UnisAir.

****

**3.3. Persistent data management**

Per la persistenza dei dati useremo PhpMyAdmin, il diagramma seguente è il modello entità-relazione.

****

Per la chiarezza del diagramma si è preferito omettere gli attributi dell’entità Passeggero (eccetto la chiave primaria) verrà esaminato in dettaglio in seguito.

**Utente**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| Email | String | 20 |
| Password | String | 20 |
| Nome | String | 30 |
| Cognome | String | 30 |
| Tipo\_documento | String | 30 |
| Num\_documento | String | 20 |

**Prenotazione**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| codPrenotazione | int | 999 |
| Data | date | 10 |
| Ora | String | 30 |
| PrezzoTotale | double | 30 |
| ProprietarioCarta | String | 30 |
| NumCarta | int | 16 |
| ScadenzaCarta | date | 10 |
| CVV/CVC | int | 3 |
| NumBiglietti | int | 200 |

**Volo**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| codVolo | int | 999 |
| aeroportoDiPartenza | String | 30 |
| aeroportoDiDestinazione | String | 30 |
| Data | date | 10 |
| oraDiPartenza | String | 30 |
| oraDiArrivo | String | 30 |
| prezzoEconomy | Double | 200 |
| prezzoBusiness | Double | 200 |
| prezzoPremium | double | 200 |

**Passeggero**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| codFiscale | String | 16 |
| tariffaBagaglio | String | 20 |
| nome | String | 30 |
| cognome | String | 30 |
| età | Int | 3 |
| indirizzo | String | 30 |
| CAP | Int | 5 |
| città | String | 30 |
| paese | String | 30 |
| tipoDocumento | String | 30 |
| numDocumento | String | 20 |
| CheckInDone | boolean | 1 |

**Aereo**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| codAereo | int | 200 |
| nomeAereo | String | 30 |

**PostiASedere**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| prenotato | boolean | 1 |

**PostiAereo**

|  |  |  |
| --- | --- | --- |
| Attribute Name | Attribute Type | Attribute Size |
| codPosto | String | 2 |
| classeDiViaggio | String | 30 |

**3.4. Access control and security**

Il sistema presenta diversi use-case, di cui sette accessibili senza autenticazione.

Il visitatore può ricercare i voli interessati e scegliere tra le opzioni di voli disponibili o selezionarne uno tra le offerte.

Le operazioni di acquisto biglietto, check-in e gestione di dati richiedono il processo di autenticazione tramite la apposite email e password, con email strettamente univoche per ogni utente. È necessario l’autenticazione anche per il gestore del sito (tramite email e password fornite dalla compagnia) dove è possibile effettuare le operazione di gestione dei voli.

Gli utenti (così come il gestore) possono autenticarsi tramite un form apposito nel Pop-up che appare premendo su Login nella barra del menù presente su ogni pagina del sito, n cui inserire i dati richiesti (email-password), e tramite la pressione di un bottone submit.