

Università degli studi di Salerno

Corso di Laurea in Informatica

INGEGNERIA DEL SOFTWARE

System Design Document

“UNI-AirLines”

Studenti:

<i>Nome</i>	<i>Matricola</i>
Santoro Mario	0512104850
Marino Raffaele	0512104508
Pastore Matteo	0512104724
Fortunato Angelo	0512104532

Anno Accademico: 2018/19

SOMMARIO

1. Introduction.....	1
1.1. Purpose of the system.....	1
1.2. Design goals.....	1
1.3. Definitions, acronyms and abbreviations.....	4
1.4. References.....	4
2. Current system architecture.....	4
3. Proposed software architecture.....	5
3.1. Subsystem decomposition.....	5
3.2. Hardware/software mapping.....	8
3.3. Persistent data management.....	9
3.4. Access control and security.....	11
3.5. Global software control.....	11
3.6. Boundary conditions.....	12
4. Glossary.....	13

1.Introduction

1.1 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 o 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.

1.2 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 di login e di password non criptate.
- **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 requisiti 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

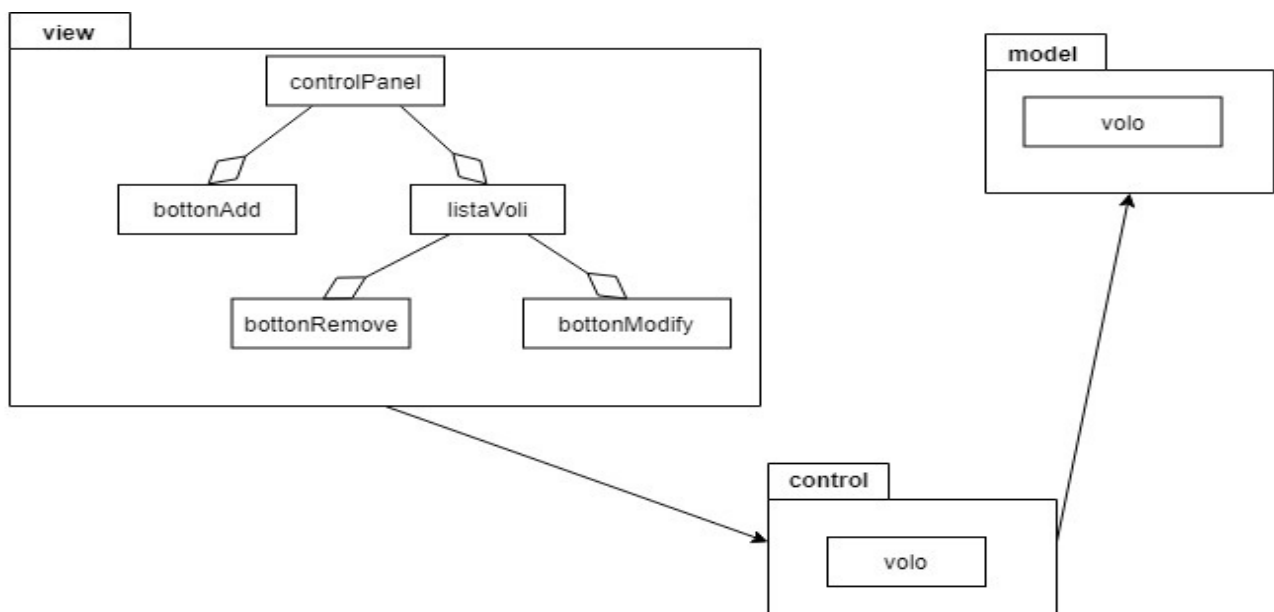
3. Proposed software architecture

3.1 Subsystem decomposition

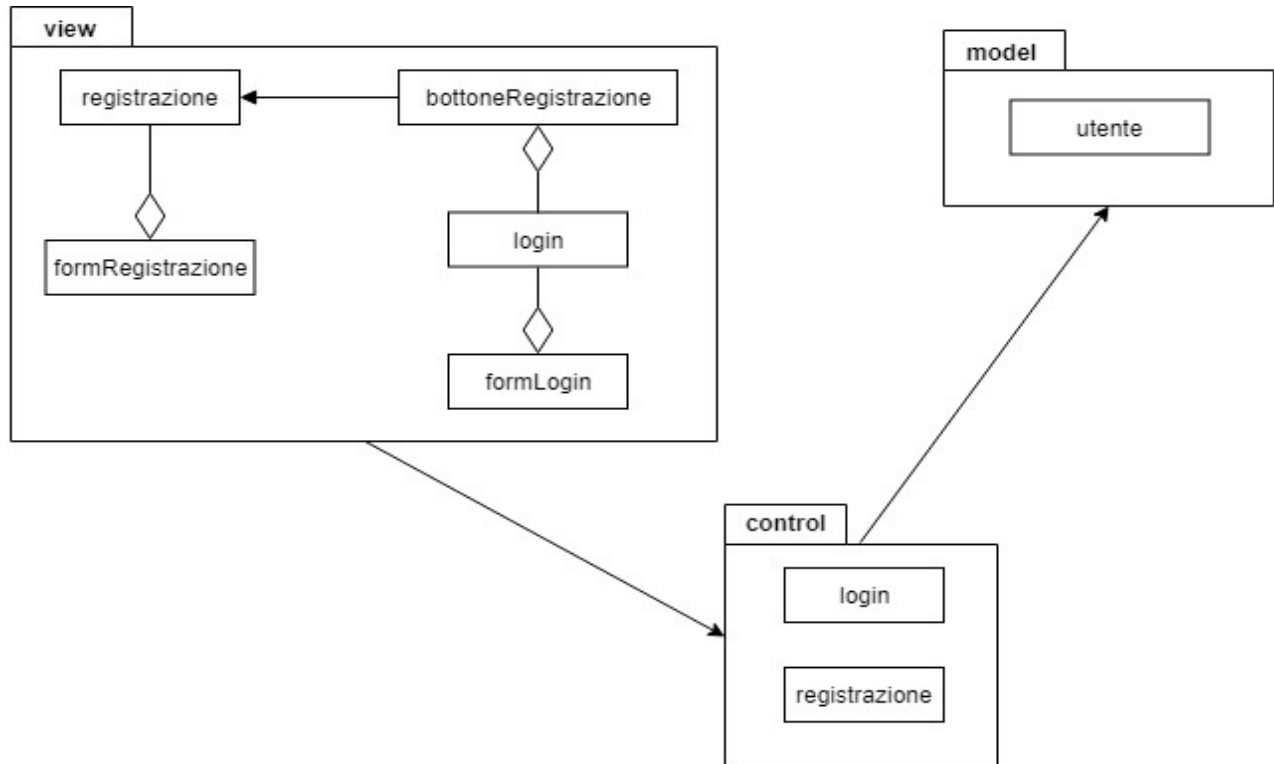
Overview:

1. Il sottosistema “Gestione admin” prevede la gestione da parte dell’admin dei voli da inserire, modificare e cancellare.
2. Il sottosistema “Gestione autenticazione” prevede la parte di registrazione, login di un utente (login se registrato precedentemente).
3. Il sottosistema “Gestione ricerca” prevede la ricerca di un volo da parte di un utente tramite la form di ricerca.
4. Il sottosistema “Gestione utente” prevede la possibilità di visualizzazione da parte di un utente registrato di vedere i voli da lui precedentemente acquistati e la possibilità di modifica dei dati utente.
5. Il sottosistema “Gestione prenotazione” prevede la possibilità da parte di un utente registrato di vedere i voli da lui acquistati nei quali è possibili effettuare il check-in e la procedura di ricerca ed acquisto di nuovi voli.

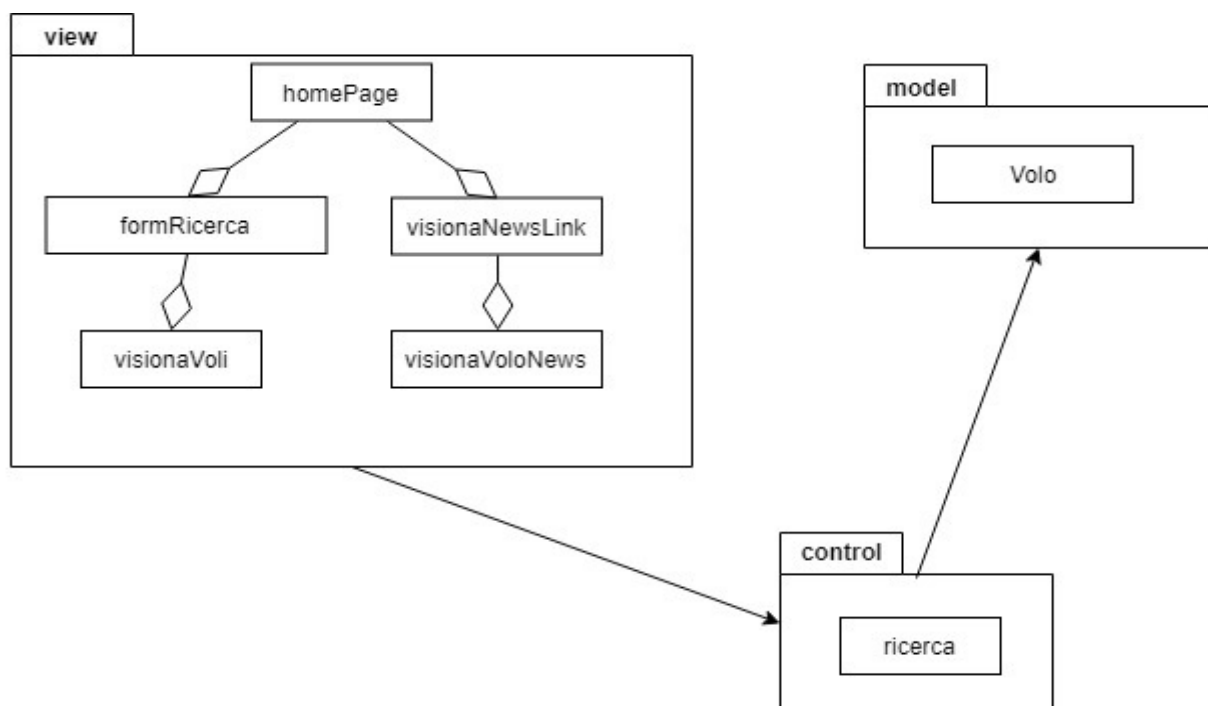
• Gestione admin



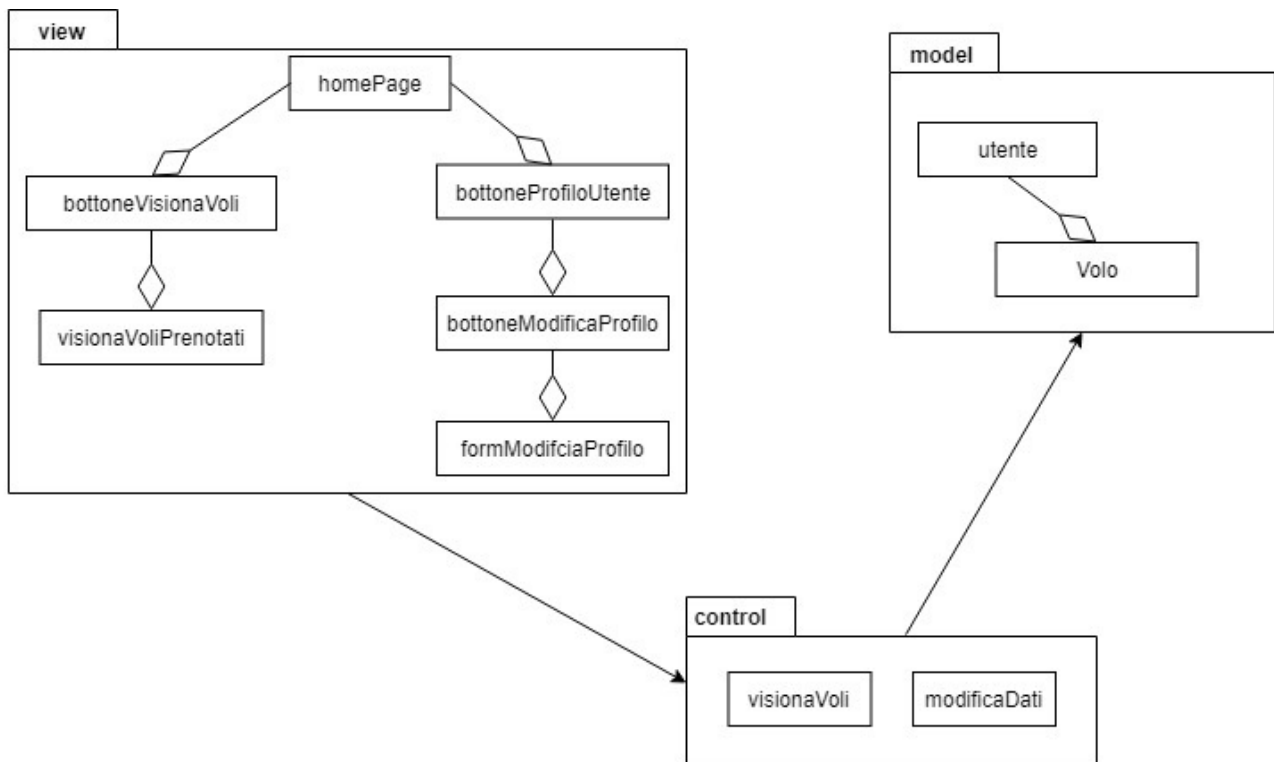
- ## Gestione autenticazione



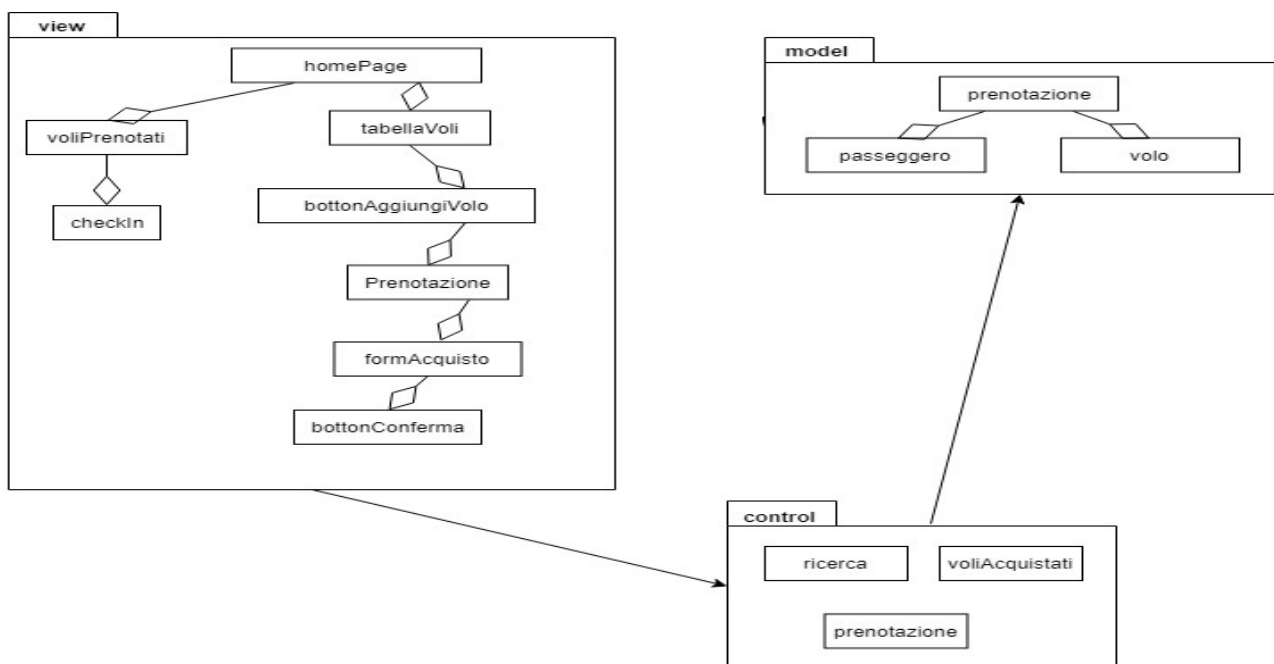
- ## Gestione ricerca



- **Gestione utente**

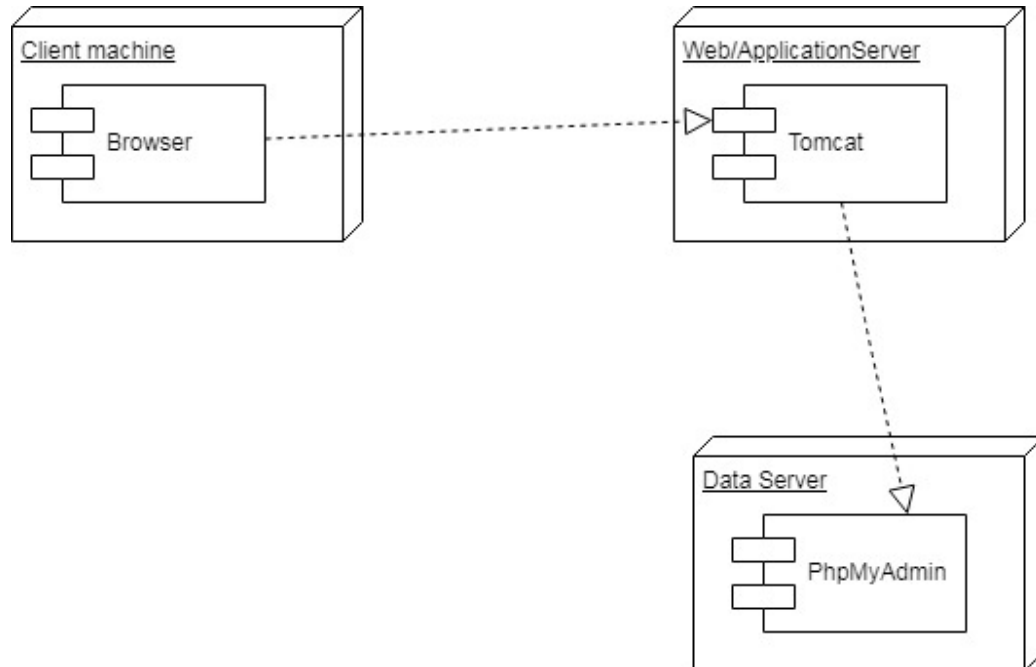


- **Gestione prenotazione**



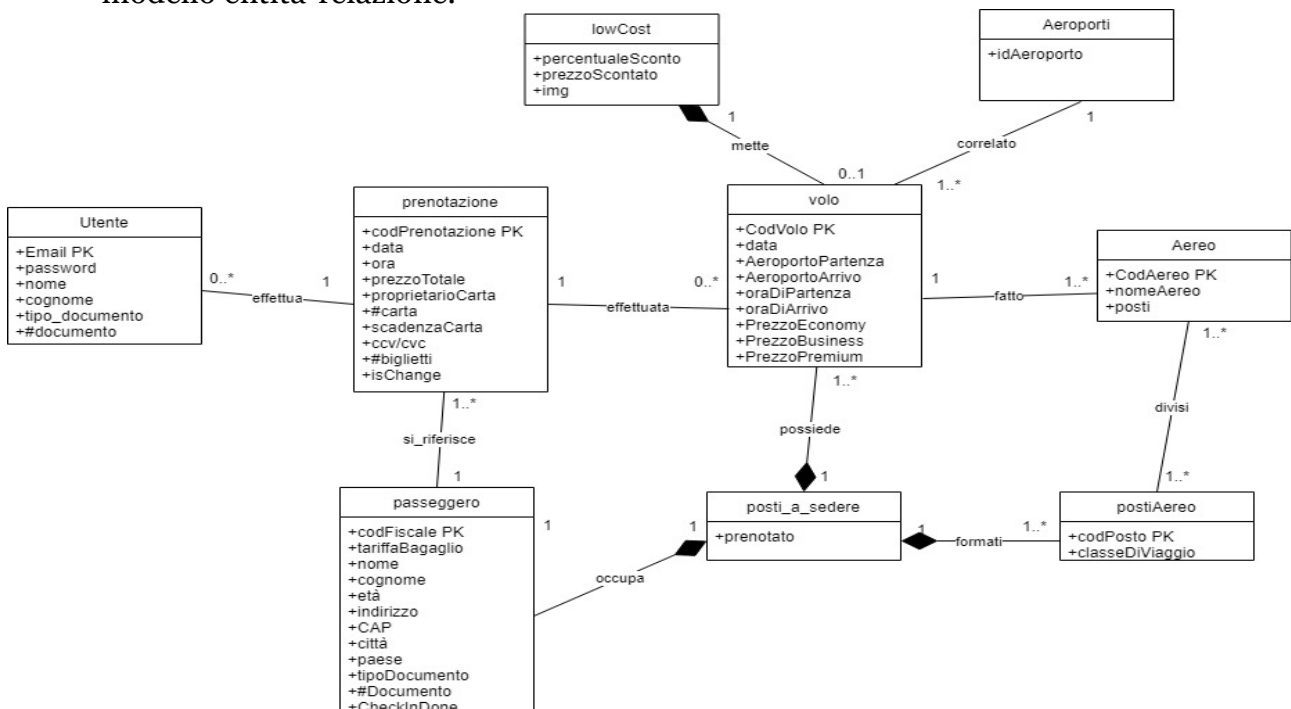
3.2 Hardware/software mapping

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.



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
isChange	boolean	1

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

Aeroporti

Attribute Name	Attribute Type	Attribute Size
idAeroporto	String	30

Aereo

Attribute Name	Attribute Type	Attribute Size
codAereo	int	200
Posti	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

LowCost

Attribute Name	Attribute Type	Attribute Size
percentualeSconto	int	3
PrezzoScontato	int	4
Img	img	1

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, in cui inserire i dati richiesti (email-password), e tramite la pressione di un bottone submit.

3.5 Global software control

Il sistema sarà accessibile tramite browser ed un WebServer (Apache Tomcat), che si occuperà di gestire in maniera concorrenziale gli accessi degli utenti. Nel momento in cui l'utente sottometterà i propri dati d'accesso per effettuare il login, il DBMS(MySql) eseguirà una query d'interrogazione per verificarne l'esistenza. Avendo esito positivo, il sistema metterà a disposizione dell'utente una serie di operazioni.

VisualizzaVolo consiste nel ricercare uno o più voli che rispettano i criteri che interessano l'utente e dato che l'accesso ai dati è in sola lettura, non abbiamo bisogno di gestire la concorrenza.

AcquistaBiglietto ha un flusso di operazioni più complesso in quanto può richiamare sia lo use case VisualizzaVolo, che abbiamo appena descritto, e sia VisionaNews che consiste nella visualizzazione di una serie di offerte che possono potenzialmente interessare l'utente. Prima della conferma dell'acquisto si entra in una sezione critica dove il DBMS lato server gestisce la concorrenza degli utenti per evitare che vi siano accavallamenti di decremento di disponibilità.

Check-in è un operazione molto importante in quanto in essa convergono tutte quelle effettuate precedentemente. Tramite questo use case l'utente può dare conferma della sua (ed eventualmente quella di altri passeggeri per cui ha prenotato il biglietto) salita a bordo dell'aereo. Qui non c'è bisogno di gestire la concorrenza in quanto non si entra in nessuna sezione critica.

Il visitatore, invece, può accedere al sistema solo per visualizzare offerte o ricercare voli senza però poter effettuare acquisti.

Va distinto, inoltre, il flusso di controllo legato all'attività del gestore della piattaforma. Costui tramite requisiti d'accesso particolari avrà a sua disposizione una serie di operazioni diverse rispetto all'utente, infatti, queste avranno lo scopo di manipolare i dati riguardanti i voli (InserimentoVolo, ModificaVolo, CancellaVolo). Tuttavia, il gestore è unico e pertanto il solo a poter manipolare questi dati, non c'è bisogno perciò di gestire concorrentemente queste operazioni, nonostante l'accesso ai dati sia in scrittura.

3.6 Boundary conditions

Inizializzazione

Dal momento in cui viene lanciato sul server, il sistema deve essere sempre acceso dato che il servizio fornito è sempre attivo.

Per quanto riguarda il sistema lato client, questo viene avviato ogni qual volta un utente accede al portale tramite browser.

Terminazione

Il sistema lato server a meno di guasti non può terminare.

Lato client, invece, il sistema è terminato alla chiusura del browser con il conseguente rilascio di tutte le variabili legate alla sessione.

Fallimento

In casi eccezionali, come la mancanza di elettricità o guasti hardware ecc., il sistema lato server può fallire. Talvolta per recuperare un fallimento il server deve essere riavviato, ma nel frattempo gli utenti non potranno collegarsi per usufruire dei suoi servizi.

Lato client il sistema può fallire a causa di vari guasti, come ad esempio la caduta della linea telefonica, tuttavia, ciò non influenza in alcun modo il server.

Eccezioni

Il web server può andare in crash e non essere agibile per un periodo di tempo non stimato. Al sistema lato client apparirà il messaggio "Http 404", ossia che la pagina non è disponibile.

4 Glossary

- Utente: rappresenta l'utilizzatore del sistema.
- Prenotazione: rappresenta la prenotazione del volo da parte di un utente.
- Passeggero: rappresenta l'utente che ha effettuato la prenotazione.
- HTTP: protocollo di trasferimento di ipertesti che consente a due macchine, client e server, di interagire attraverso un meccanismo di richiesta/risposta.
- Client: componente che accede a servizi e risorse offerte dal server.
- Server: componente che offre servizi e risorse al client.
- Model: componente del modello MVC che si occupa dall'interazione tra l'applicazione e il database.
- View: componente del modello MVC scritto in linguaggio HTML e visualizzato nella pagina web dall'utente che permette all'utente di interagire col sistema.
- Controller: componente del modello MVC che si occupa di elaborare le richieste di un utente e di comunicare con il model.
- MVC: modello architetturale che si basa sull'utilizzo di 3 componenti fondamentali (model,view e controller) per lo sviluppo di web application.
- HTML: linguaggio di markup utilizzato per la formattazione e l'impaginazione di documenti ipertestuali.
- Web host: servizio di rete che permette di allocare su un server web le pagine di un'applicazione web.
- PhpMyAdmin: è un'applicazione web scritta in PHP, che consente di amministrare un database MySQL tramite un qualsiasi browser.