

<p align="center">Corso di Ingegneria del Software Corso di Laurea in INGEGNERIA INFORMATICA Prova di Esame del 23/10/2013</p>	<p align="center"><u>Allievo</u> Cognome e Nome:</p> <p align="center">Matricola:</p>
---	---

Descrizione del Problema

- *Si vuole realizzare un sistema software per realizzare una gestione automatizzata degli imbarchi dei passeggeri di un volo.*

Ogni volo è caratterizzato dal codice volo (alfanumerico di 5 caratteri), un aeroporto di partenza ed uno di destinazione (stringhe alfanumeriche di 15 caratteri), un orario di partenza ed uno di arrivo. I passeggeri possono acquistare un biglietto per uno specifico volo da effettuarsi in una determinata data. Ogni biglietto ha un proprio codice (alfanumerico di 10 caratteri).

Per accedere al volo i passeggeri devono effettuare preliminarmente la registrazione al volo (ossia il check-in) presso un chiosco automatico disponibile in aeroporto, fornendo il proprio codice biglietto. Il sistema accerta la validità del biglietto per la data corrente e in tal caso fornisce una ricevuta di imbarco dotata di un codice a barre (codice di imbarco). In caso di errore il chiosco visualizza un opportuno messaggio.

L'imbarco di ogni volo avviene presso una specifica porta di imbarco caratterizzata da un proprio identificativo numerico di 3 cifre e si effettua in un intervallo orario compreso fra quello di inizio imbarco e di fine imbarco. La porta è dotata di un tornello la cui apertura è pilotata da un lettore automatico di codice a barre. Il passeggero che si imbarca dovrà far rilevare al lettore automatico il codice a barre della ricevuta di imbarco per consentire al sistema di verificare la corrispondenza fra tale codice, la porta e l'orario. In caso di esito positivo, il sistema abiliterà l'apertura del tornello e aggiornerà la lista dei passeggeri imbarcati sul volo.

QUESITI

1. Si disegni il diagramma dei **casi d'uso** e quello delle **classi concettuali** (System Domain Model) per tale sistema, tenendo conto anche delle responsabilità da attribuire alle classi.
2. Si progetti una soluzione basata sul modello architetturale a più livelli (Multi-Layer) per la realizzazione di questo sistema. Al fine di descrivere la soluzione proposta, il candidato dovrà fornire:
 - 2.1) il Package diagram descrittivo dell'architettura del sistema contenente anche le classi di dettaglio necessarie per l'implementazione;
3. Si progettino inoltre i Sequence diagram di dettaglio per la realizzazione delle seguenti funzionalità:
 - a. **"Esegui check-in"** che riceve in input il codice del biglietto e verifica la corrispondenza fra il data del volo associato al biglietto e la data corrente. In caso di esito positivo, viene fornita una ricevuta di imbarco dotata di un codice a barre (codice di imbarco), altrimenti viene visualizzato un messaggio di errore.
 - b. **"Esegui imbarco"** che riceve in input il codice di imbarco e l'identificativo della porta e in base all'orario corrente abilita l'apertura del tornello della porta oppure no.

Seconda Parte della Prova

Data di Consegna: 26 ottobre 2013, entro le ore 18.00, per e-mail al docente: fasolino@unina.it (specificando come Oggetto della Mail: Prova Pratica IS- nome studente)

3. Implementare in linguaggio Java le classi del sistema necessarie a realizzare le due funzionalità delle quali è richiesta la definizione tramite sequence diagrams.

Dovrà inoltre essere definita la struttura del database (modello E/R e relazionale) per l'implementazione della persistenza e dovrà essere riempito con dati sufficienti ad eseguire i casi di prova.

Lo studente dovrà consegnare:

- Tutti i file del progetto Java (comprese le eventuali librerie – è consigliato l'uso di Eclipse per la gestione dei file del progetto). Come nome del progetto si indichi il proprio Cognome seguito dal numero di matricola completo (ad esempio: ROSSI_N46000000)
- Il database, il cui nome dovrà essere obbligatoriamente **NomeCognomeDB**. In dettaglio dovranno essere prodotti (e nominati anch'essi con il Cognome ed il numero di matricola):
 - o Un file mwb prodotto da MySQL Workbench, da cui si vede il modello Entità-Relazione;
 - o script .sql ottenuto con la funzionalità Data Export di MySQL Workbench, contenente i dati di riempimento del database eventualmente utilizzati nei test.

Lo studente deve tenere particolarmente in conto la portabilità del sistema realizzato e la completezza del materiale inviato. L'applicazione dovrebbe essere funzionante su una macchina target che abbia installate Java SDK, MySQL Workbench ed Eclipse, in versioni analoghe a quelle viste durante il corso.

4. Progettare un insieme di casi di test black box (usando la tecnica delle Classi di Equivalenza) per provare la funzionalità **Esegui imbarco** tenendo conto anche delle diverse pre-condizioni per tale funzionalità.

Si usi lo schema seguente per definire i casi di test

TC	Descrizione Test Case	Classi di Equivalenza coperte	Precondizioni	Input	Output Attesi	Post-Condizioni Attese	Output Ottenuti	Post-Condizioni Ottenute	Esito

Lo studente dovrà consegnare:

1. Un file di testo (formato .DOC , .RTF o .XLS) contenente le classi di equivalenza e la tabella dei casi di test progettati.

N.B.: Si consegnino sia il file di progetto VPP (Visual Paradigm) contenente tutti i diagrammi UML ritenuti utili a descrivere la soluzione proposta, sia un file di testo che spieghi e giustifichi le scelte progettuali fatte. Entrambi i file dovranno essere nominati con il Cognome del candidato seguito dal numero di matricola completo (ad esempio: ROSSI_M63000000).

NN.BB.: Eventuali similitudini rilevate fra elaborati di diversi autori comporteranno automaticamente la non ammissione alla prova orale.