

<p align="center"><b>Corso di INGEGNERIA DEL SOFTWARE</b>  <b>Corsi di Laurea in INGEGNERIA INFORMATICA e in</b>  <b>INGEGNERIA DELL'AUTOMAZIONE</b>  <b>A.A. 2014-2015</b></p> <p align="center"><b>ESAME DEL 21 LUGLIO 2015</b>  <b>DURATA DELLA PROVA: 2,5 ORE</b></p>	<p align="center"><b><u>STUDENTE</u></b></p> <p><b>Cognome e Nome:</b></p>    <p><b>Matricola:</b></p>
---	--

## Prima Parte della Prova

### *1. Si consideri il seguente documento di descrizione informale dei requisiti:*

Si vuole realizzare un social network per il turismo, che permette di connettere persone interessate a vacanze in barca a vela, e i proprietari degli scafi, che offrono il noleggio dei natanti.

Tutti gli utenti possono registrarsi al sistema fornendo il proprio username, la propria password, l'email, e l'età. Inoltre, in caso di registrazione di un proprietario di barca, l'utente deve specificare il numero di patente nautica, la sua data di rilascio e di scadenza. Dopo aver completato la registrazione, gli utenti si devono autenticare per accedere alle altre funzionalità del sistema.

I proprietari di barca possono aggiungere, rimuovere, e modificare i dati relativi alle barche a vela. Ciascuna unità è caratterizzata dal nome, dalla lunghezza (espressa in metri), dalla targa, e numero di posti disponibili. Le unità sono messe a disposizione per un *piano di vacanza*.

Un *piano di vacanza* è caratterizzato dalla data e dal porto di partenza, dalla data e dal porto di arrivo, dal numero massimo di passeggeri, dal costo della vacanza. I proprietari possono inserire, modificare ed eliminare i piani di vacanza. Considerando una barca, non è possibile inserire un nuovo piano vacanza o modificarne uno esistente, se le date di inizio e fine si sovrappongono con almeno un altro piano vacanza registrato nel sistema, che coinvolge la stessa imbarcazione. Gli utenti possono prenotare un piano di vacanza. All'atto della prenotazione, il sistema memorizza l'utente, la data di prenotazione e il numero di passeggeri.

Gli utenti autenticati al sistema possono utilizzare la funzionalità "ricerca" per ricercare vacanze in base alla data e al porto di partenza. Il sistema restituirà tutti i piani di vacanza che rispettano i criteri scelti dall'utente ed hanno ancora posti disponibili prenotabili. Una volta terminata la ricerca, l'utente può prenotare un piano di vacanza tra i risultati.

Inoltre, il sistema permette agli utenti di inviare messaggi ai proprietari delle barche.

*Per tale sistema, si realizzino in linguaggio UML:*

*- il diagramma dei casi d'uso*

*- il diagramma delle classi raffinato, che mostri anche attributi e responsabilità attribuite alle classi;*

*Suggerimento: si progetti applicando le buone norme dell'ingegneria del software, utilizzando in maniera opportuna le classi «boundary», «control».*

*- il diagramma di sequenza raffinato della funzionalità "ricerca".*

## Seconda Parte della Prova

Lo studente produca autonomamente i seguenti artefatti, da presentare sia in forma di relazione stampata sia in formato elettronico per la discussione alla successiva prova orale.

2. Si implementi in linguaggio Java la parte del sistema necessaria a realizzare la funzionalità **“ricerca”**.

- Si realizzi con JDBC una classe DAO per una delle entità persistenti del sistema.

A corredo della relazione, lo studente dovrà consegnare per la discussione alla prova orale:

- I file del progetto (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)
- Una base di dati che permetta di esercitare la classe DAO implementata. È consigliato scegliere come DBMS di supporto H2 (<http://www.h2database.com/>) o Microsoft Access.

3. Si progetti ed esegua un insieme di casi di test black box per la funzionalità **“ricerca”**, tenendo conto anche delle diverse pre-condizioni per tale funzionalità.

Si usi lo schema seguente per definire i casi di test.

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

Lo studente dovrà consegnare:

- Un file di testo (formato .DOC, .RTF o .XLS) contenente la tabella dei casi di test progettati;
- Un Progetto di Test in JUnit contenente i casi di test necessari a testare la funzionalità richiesta;
- Esito dei test eseguiti con JUnit e opzionalmente i livelli di copertura del codice raggiunti.
- il *Class diagram* di progettazione del sistema; si può trascurare in questo diagramma la gestione della persistenza (perché si suppone sia realizzata da un framework esterno), l’interfacciamento con gli utenti e le politiche di security (autenticazione con gli utenti);
- il *Sequence diagram* di progettazione per la realizzazione della funzionalità **“ricerca”**.