Corso di Ingegneria del Software – Seconda Verifica Intermedia del 26 Maggio 2016

V. Gervasi, L. Semini - Dipartimento di Informatica, Università di Pisa, a.a. 2015/16

Nome_____ Cognome_____

Matricola_____ Corso _____

Aula_____ Posizione nell'aula_____

(senza contare i banchi vuoti)

			cattedra			
A1	A2	A3	A4	…	…	…
B1						
C1						
….						

Si consideri il seguente caso d'uso: PRENOTAZIONE TRAMITE APP
Attore primario: Cliente
Attori secondari: Autista, Istituto di credito
Precondizioni: Cliente autenticato tramite l'app
Postcondizioni: Prenotazione effettuata oppure richiesta al Cliente di riprovare più tardi.
Sequenza principale degli eventi:

1. Il Cliente chiede, tramite l'app, l'invio di un'auto, specificando tipo, indirizzo, orario e carta di credito con cui pagare la corsa.
2. Il Sistema trasmette la richiesta a tutti gli autisti in servizio, dotati di un'auto del tipo richiesto, e che non siano già assegnati ad altre corse nell'orario indicato
3. Il sistema raccoglie le segnalazioni di disponibilità degli autisti nell'arco di 1 minuto
4. **Se** (l'insieme degli autisti disponibili è vuoto ed esistono auto di tipo più costoso)
    4.1.1. Il Sistema trasmette la richiesta a tutti gli autisti in servizio, dotati di un'auto di tipo superiore e che non siano già assegnati ad altre corse nell'orario indicato, specificando nella richiesta che per la corsa si offre il prezzo corrispondente al tipo di auto della richiesta originale
    4.1.2. Il sistema raccoglie le segnalazioni di disponibilità degli autisti nell'arco di 1 minuto
5. **Se** (l'insieme degli autisti disponibili non è vuoto)
    5.1. Il Sistema sceglie uno degli autisti disponibili (in base a diverse euristiche)
    5.2. Il Sistema informa il Cliente sui tempi di attesa
    5.3. Il Cliente accetta la corsa
    5.4. Il Sistema assegna la corsa all'autista e lo informa
    5.5. Il Sistema richiede all'Istituto di credito la pre-autorizzazione ad addebitare sulla carta di credito l'importo della corsa.
    5.6. Il sistema conferma la corsa all'utente, fornendo i dettagli di contatto dell'autista a cui è stata assegnata.
6. **Altrimenti** il sistema chiede al cliente di riprovare dopo qualche minuto

Sequenze alternative degli eventi: Non ci sono autisti disponibili. L'istituto di credito rifiuta la pre-autorizzazione.

Domanda 1. Si consideri, per ora, il sistema come composto di due sotto-sistemi, la componente app_cliente e il resto (sottosistema lato server e app usata dall'autista). Si dia un diagramma di sequenza che descrive le interazioni tra un utente, la sua app, il resto del sistema, tre autisti in ruoli diversi, e l'istituto di credito.

Domanda 2. Si assuma che sia stata definita un'architettura con i seguenti componenti:

1. app_cliente (che implementa l'app usata dal cliente)
2. sistema_prenotazione (che implementa la business logic del sistema)
3. db_anagrafica (che raccoglie dati anagrafici su autisti e clienti)
4. db_corse (che raccoglie dati su prenotazioni, corse assegnate, corse effettuate)
5. app_autista (che implementa l'app usata dall'autista)

Si disegni il diagramma C&C, specificando gli stereotipi corrispondenti al tipo di interazione che si immagina sui connettori.

Domanda 3. Sotto le stesse assunzioni dell'esercizio 2, e assumendo un artefatto (omonimo) per componente, si disegni un diagramma di dislocazione (deployment) per il sistema. Si assuma, che il db_anagrafica venga aggiornato poco frequentemente (solo quando si iscrivono nuovi autisti e nuovi clienti), e che venga interrogato con query semplici, mentre il db_corse sia soggetto ad aggiornamenti più frequenti, e debba eseguire query complesse (per esempio, per verificare quali autisti saranno liberi a un dato orario occorre incrociare le ore di inizio e fine servizio di tutte le corse attualmente prenotate e delle altre richieste pendenti).

Domanda 4. Si assuma che il componente sistema_prenotazione offra un'interfaccia basata su REST alle app, e che sullo stesso porto voglia accettare anche le operazioni richieste via SMS o via sito web. Si dettagli la struttura interna del componente con un diagramma di struttura composita.

Domanda 5. Si consideri il seguente metodo Java che implementa l'euristica di selezione dell'autista:

```
public Autista seleziona(Set<Autisti> disponibili, Richiesta r)
{
     int score=1000, d=0;
     Autista prescelto=null;
      Location aLoc=null, rLoc=r.getLocation();
     for (Autista a : disponibili) {
          aLoc=a.getLocation();
          d=aLoc.distance(rLoc);
          if (d<score) {
               score=d;
               prescelto=a;
          }
     }
     return a;
}
```

Si definiscano:

1. Un insieme minimo di valori di input che garantisca una copertura del 100% dei comandi.
2. Una partizione del dominio degli argomenti del metodo seleziona: si ripartisca in classi di equivalenza rispetto al criterio di copertura delle decisioni;
3. Si svolga infine una ispezione strutturata del codice, definendo una checklist che includa la verifica del trattamento dei `null`. Si evidenzia qualche problema nel codice dato sopra?