Esame di Ingegneria del Software – 9 CFU – 21 Giugno 2022

- Scrivere immediatamente, su ogni foglio che vi è stato consegnato, cognome, nome, numero di matricola.
- Non è consentito consultare appunti, libri, colleghi, né qualunque dispositivo elettronico, pena l'immediato annullamento della prova.
- Tempo a disposizione: 3 ore.

Esercizio 1

Si vuole realizzare una piattaforma social per permettere a studenti universitari di conoscersi e organizzare gruppi di studio, eventualmente anche in modalità a distanza.

In fase di registrazione, l'utente deve compilare un form indicando le proprie informazioni e preferenze. In particolare, l'utente deve indicare un nickname univoco, una email, una password, un'università e un corso di laurea cui è iscritto/a.

Il nickname viene di default generato automaticamente dal sistema sfruttando le API REST del servizio esterno `FunnyNickGen`, ma l'utente può sovrascrivere il valore di default indicando un nuovo nome utente. Quanto alla selezione di università e corso di laurea, l'utente deve dapprima selezionare una delle università presenti nel sistema. Dopo aver selezionato l'università, il sistema permette di scegliere un corso di laurea tra quelli correntemente associati a quella università.

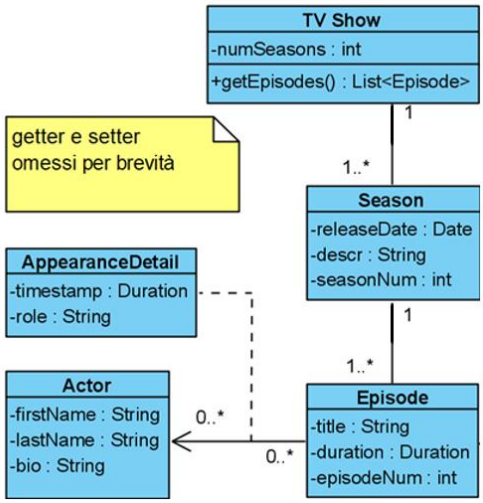
Dopo la registrazione, l'utente può visualizzare i gruppi di studio correntemente attivi per il suo corso di laurea, ed entrare in contatto con altri studenti.

Dettagliare il caso d'uso relativo alla funzionalità di registrazione di uno studente per mezzo di mock-up e descrizioni testuali strutturate secondo il formalismo di Cockburn. Usare la propria conoscenza del dominio per derivare dettagli non definiti nei requisiti.

a) Definire un class diagram di analisi del sistema, inteso come modello di dominio, relativo al caso d'uso della registrazione di un utente. È possibile rifarsi alle euristiche *EBC*, e di *Abbott*.
b) Fornire un sequence diagram di analisi per il caso d'uso relativo alla registrazione di un utente.

Esercizio 2

Si scriva tutto il codice Java che è possibile desumere dal seguente class diagram. È possibile omettere il codice relativo a metodi getter e setter.



Esercizio 3

```
1  public int evaluate(String s, int n) {
2      int res = n;
3      while(true) {
4              if(res%7==0) {
5                      break;
6              }
7              res = res - s.length();
8              if(res < 0) {
9                      break;
10             }
11     }
12     return res;
13 }
```

Si consideri il metodo Java riportato sopra.

a) Rappresentare il CFG del metodo;
b) Scrivere quattro test JUnit con strategia White Box per il metodo `evaluate`, indicando per ciascuno di essi quale cammino copre nel CFG. Ove possibile, si richiede che i test JUnit coprano cammini distinti nel CFG.

Esercizio 4

Le luci di cortesia di un'automobile hanno un interruttore che può assumere tre posizioni: ON, OFF, e DEFAULT. Quando l'interrutore è in posizione ON, le luci di cortesia sono sempre accese. Al contrario, quando è in posizione OFF, le luci di cortesia sono sempre spente. Quando l'interruttore è in posizione DEFAULT, le luci si accendono soltanto quando una delle portiere è aperta, e restano spente altrimenti. Inoltre, quando il motore è spento e l'interruttore è in posizione ON, le luci si spengono in ogni caso dopo 30 minuti per evitare di consumare la batteria, e l'interruttore si sposta su OFF.

Si rappresenti il comportamento delle luci di cortesia sopra descritte utilizzando il formalismo degli StateChart.