

SAPIENZA Università di Roma
Facoltà di Ingegneria dell'Informazione, Informatica e Statistica
Corso di Laurea in Ingegneria Informatica ed Automatica
Corso di Progettazione del Software
Esame del **24 luglio 2020**
Tempo a disposizione: 3:30 ore

Requisiti. L'applicazione da progettare riguarda un sistema per gestire campagne pubblicitarie.

Delle campagne pubblicitarie interessano il nome, una descrizione (una stringa) e i creativi che ci lavorano. Di questi interessano il nome e l'email. Tra i creativi che lavorano ad una campagna pubblicitaria, uno è il responsabile della campagna. Ogni campagna pubblicitaria ha esattamente un responsabile ed un numero arbitrario di altri creativi che ci lavorano. Le campagne pubblicitarie sono costituite da diverse pubblicità. Ogni pubblicità è associata ad esattamente una campagna pubblicitaria. Ad una campagna pubblicitaria possono venire associate delle "idee", costituite da un titolo (una stringa) ed una descrizione (una stringa). Ogni idea è associata ad esattamente una campagna pubblicitaria ed è "creata" da esattamente un creativo, che può o meno lavorare alla campagna pubblicitaria stessa. Ogni creativo può creare al più 100 idee. Alcune idee sono dirompenti in quanto richiedono un cambiamento profondo della campagna pubblicitaria stessa ed in tal caso devono indicare una descrizione aggiuntiva che giustifichi il cambiamento proposto (una stringa).

Siamo interessati allo gestione delle idee. Una idea si trova inizialmente nello stato *creata*. Il suo creatore può decidere di chiuderla oppure di sostenerla. Nel primo caso va nello stato *chiusa*, altrimenti va nello stato *sostenuta*. Quando è nello stato *sostenuta*, un lavoratore della campagna pubblicitaria può decidere di chiuderla, oppure può mandare una richiesta di approfondimenti al suo creatore. In questo caso, l'idea passa nello stato *attesa*. Il creatore dell'idea può decidere di chiuderla oppure di aggiornare la descrizione dell'idea stessa, in tal caso l'idea manda una notifica al lavoratore che lo ha richiesto e torna nello stato *sostenuta*.

Siamo interessati all'attività di rilascio di una campagna pubblicitaria. In questa attività viene prima verificato che non ci siano idee dirompenti. Se questo controllo fallisce, viene stampato un messaggio di errore e l'attività termina. Se invece il controllo riesce, vengono eseguite in parallelo due sottoattività (i cui dettagli non interessano); nella prima, la campagna pubblicitaria viene pubblicata per avere nuove idee per un certo periodo al termine del quale viene preparato un elenco delle idee proposte (una stringa); nella seconda, si inseriscono le nuove pubblicità associate ad essa, stilandone un elenco delle pubblicità inserite (una stringa). Quando entrambe le sottoattività sono concluse, viene stampato un rapporto finale con entrambi gli elenchi proposti.

Domanda 1. Basandosi sui requisiti riportati sopra, effettuare l'analisi producendo lo schema concettuale in UML per l'applicazione, comprensivo di: diagramma delle classi (inclusi eventuali vincoli non esprimibili in UML); diagramma stati e transizioni per la classe *Idea*; diagramma delle attività di rilascio; specifica del diagramma stati e transizioni, riportando solo gli stati, e le variabili di stato ausiliarie, ma non la specifica delle transizioni; la segnatura delle attività complesse, delle attività atomiche e dei segnali di input/output. Motivare, qualora ce ne fosse bisogno, le scelte di progetto.

Domanda 2. Effettuare il progetto, illustrando i prodotti rilevanti di tale fase e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare definire SOLO le responsabilità sulle associazioni del diagramma delle classi (nella tabella, inserire anche il motivo di ognuna delle responsabilità).

Domanda 3. Effettuare la realizzazione, producendo un programma JAVA e motivando, qualora ce ne fosse bisogno, le scelte di progetto. In particolare realizzare in JAVA SOLO i seguenti aspetti dello schema concettuale:

- La classe *Idea* con la classe *IdeaFired*, le eventuali sottoclassi e le classi JAVA per rappresentare le *associazioni* di cui la classe *Idea* ha responsabilità.
- L'*attività di rilascio*, ma non le sue sotto-attività, e l'*attività atomica di verifica*.