

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 **26 giugno 2020**
Tempo a disposizione: 3 ore

Requisiti. L'applicazione da progettare riguarda un sistema di sviluppo software cooperativo.

Gli elementi centrali del sistema sono gli utenti e i progetti software che sviluppano. Degli utenti interessa il nome e l'email. Dei progetti interessa il nome e la data di creazione. Ogni utente può creare un numero arbitrario di progetti (anche nessuno). Ogni progetto ha un unico creatore ma può avere un arbitrario di altri manutentori. Il creatore è egli stesso un manutentore. I manutentori di un progetto sono al massimo venti. Un progetto contiene dei file (inizialmente nessuno); ogni file appartiene a un solo progetto. A un progetto possono venire fatte delle richieste di modifica. Ognuna di queste richieste è costituita da un codice (un intero) e una descrizione (una stringa). Una richiesta viene creata da un utente qualsiasi (che può o meno essere un manutentore del progetto), ed è associata a uno o più file; dato un file, non è di interesse sapere in quali richieste è coinvolto. Le richieste possono essere bloccanti o meno. Le richieste bloccanti hanno una priorità (un intero).

Una parte specifica dell'applicazione riguarda il meccanismo delle richieste. Un utente crea la richiesta, che si trova inizialmente nello stato di *creata*. Lo stesso utente può decidere di chiuderla oppure di inviarla. Nel primo caso va nello stato *chiusa*, altrimenti *inviata*. Quando è nello stato *inviata*, un manutentore del progetto può decidere di chiuderla (non interessa sapere se viene accettata o meno), oppure può mandare un suggerimento di modifica all'autore della richiesta. In questo caso, la richiesta passa nello stato *attesa*. L'autore della richiesta può decidere di chiuderla oppure di effettuare la modifica e far tornare la richiesta nello stato di *inviata*; in entrambi i casi, viene informato il manutentore che ha mandato il suggerimento.

Siamo interessati all'attività di rilascio di una versione del progetto software. In questa attività viene prima verificato che non ci siano richieste bloccanti. 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; nella prima, il progetto viene compilato e poi testato; nel secondo, il progetto viene assemblato in un singolo archivio. Quando entrambe le sottoattività sono concluse, se il test è andato a buon fine il pacchetto viene rilasciato e viene stampato un messaggio di conferma. In caso contrario viene stampato un messaggio di errore.

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 *Richiesta*; diagramma delle attività di rilascio; specifica del diagramma stati e transizioni; specifica completa dell'attività di rilascio, sottoattività non atomiche, atomiche e 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 *Richiesta* con la classe *RichiestaFired*, le classi JAVA per rappresentare le *associazioni* di cui la classe *Richiesta* ha responsabilità.
- L'*attività di rilascio* e le sue eventuali sottoattività NON atomiche.