

# Ingegneria del Software A.A. 2017/2018

## Esame 2018-04-23

---

### Esercizio 1 (6 punti)

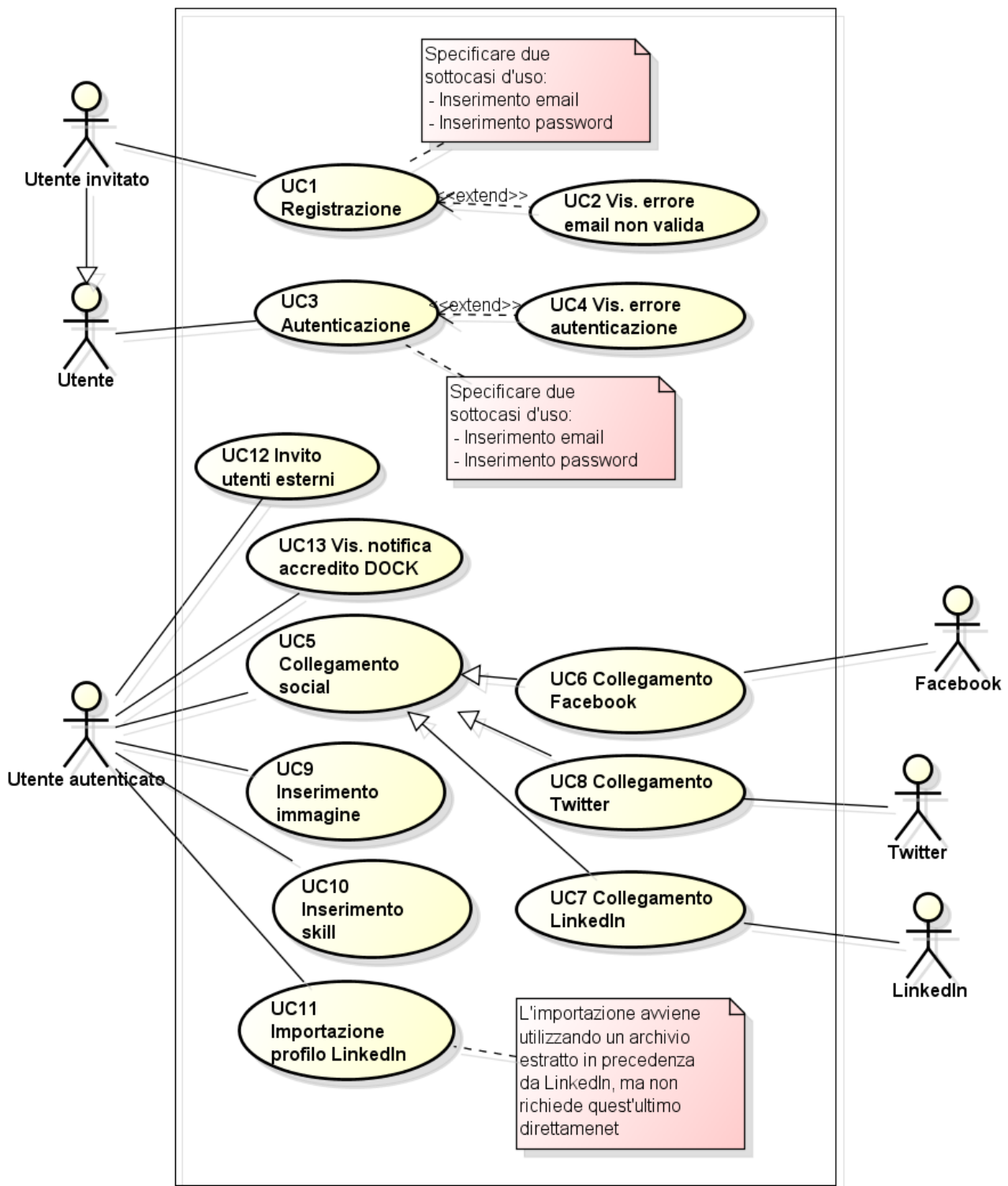
#### Descrizione

Dock.io è un nuovo progetto che prevede l'utilizzo di una Blockchain per condividere e validare le esperienze social. Per ora la registrazione al sito è possibile unicamente tramite invito e richiede l'inserimento di un'email valida e di una password. Le medesime informazioni sono richieste in fase di autenticazione. Per validare il proprio profilo è necessario collegare almeno uno dei seguenti social: LinkedIn, Facebook o Twitter. Inoltre è necessario inserire un'immagine del proprio profilo ed inserire almeno 5 *skill*, con indicazione del livello di esperienza associato: junior, middle e senior. L'inserimento di queste informazioni rende il profilo "verificato". E' inoltre possibile importare le proprie informazioni dal proprio profilo LinkedIn, utilizzando un archivio compresso precedentemente esportato dal social network esterno. Infine, è possibile invitare altre persone ad iscriversi a Dock.io. Per ogni nuovo account invitato, che completa l'iscrizione fino allo stato di profilo "verificato", vengono accreditati 44 DOCK, un *crypto token* appositamente creato. L'informazione viene notificata all'utente con un apposito messaggio interno.

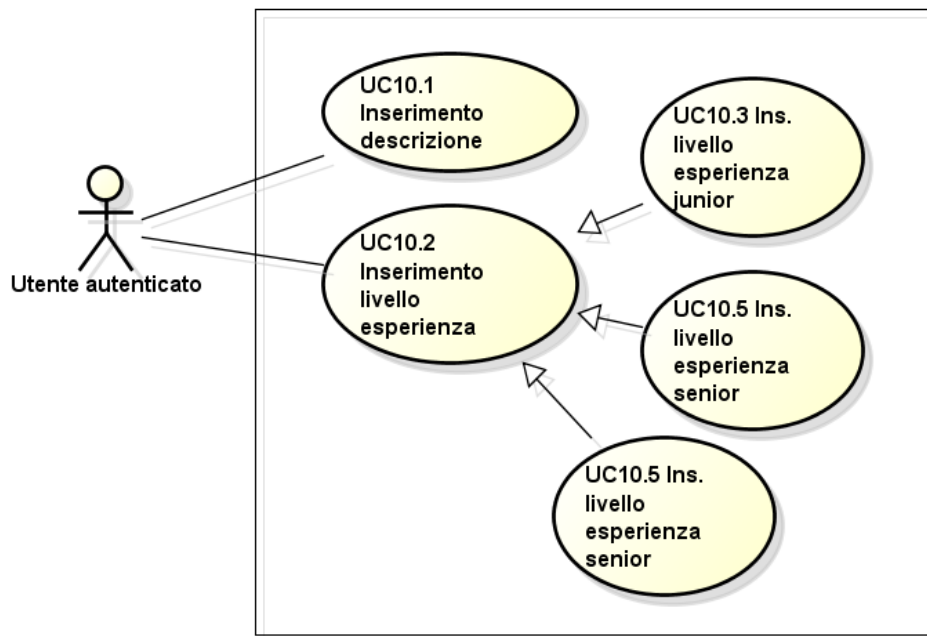
Si utilizzino i diagrammi dei casi d'uso per modellare gli scenari descritti. Non è richiesta la descrizione testuale dei casi d'uso individuati.

#### Soluzione

Il una possibile soluzione per il diagramma principale è la seguente.



Inserimento di una skill può essere dettagliato utilizzando il seguente diagramma.



## Esercizio 2 (7 punti)

Amazon distribuisce i propri libri sui dispositivi Kindle utilizzando una metodologia *publish-subscribe*. Per ogni utente, Amazon mantiene una libreria dedicata nei propri server. Ogni qualvolta un utente compra un libro in formato Kindle, tutti i suoi dispositivi vengono notificati e, se disponibile una connessione, scaricano il nuovo libro. Per ottimizzare la memoria utilizzata, il libro non viene caricato tutto in memoria, ma oggetti pesanti come le immagini vengono caricate unicamente quando visualizzate a schermo. Ne deriva che ogni pagina del libro è renderizzata *just-in-time*, ossia solo all'occorrenza. Per questo un libro può essere visto come una collezione di pagine, dove l'algoritmo di attraversamento è particolare per ogni tipologia di Kindle. Il tutto è implementato utilizzando un pattern MVC sul dispositivo.

Si modelli il sistema descritto utilizzando un diagramma delle classi e gli opportuni *design pattern*. Inoltre, si descriva utilizzando un diagramma di sequenza l'operazione di scaricamento su un dispositivo di un nuovo libro e la visualizzazione della sua prima pagina.

## Soluzione

Il diagramma delle classi corrispondente è il seguente, nel quali sono stati utilizzati i pattern:

- Observer
- MVC
- Iterator
- Proxy

È possibile individuare anche un pattern Abstract factory per la creazione delle coppie libro / dispositivo.



### Esercizio 3 (3 punti)

#### Descrizione

Si fornisca un esempio di diagramma di attività all'interno del quale sia utilizzata una "regione di espansione". Si ricorda che la regione di espansione è la struttura che permette la ripetizione di un insieme di azioni su una collezione di elementi.

#### Soluzione

Riprendendo l'esempio visto a lezione, una possibile soluzione all'esercizio è la seguente. Dovranno ovviamente essere individuate opportunamente le azioni iniziali e finali.

