

# Ingegneria del Software A.A. 2016/2017

## Esame 2017-05-15

---

### Esercizio 1 (6 punti)

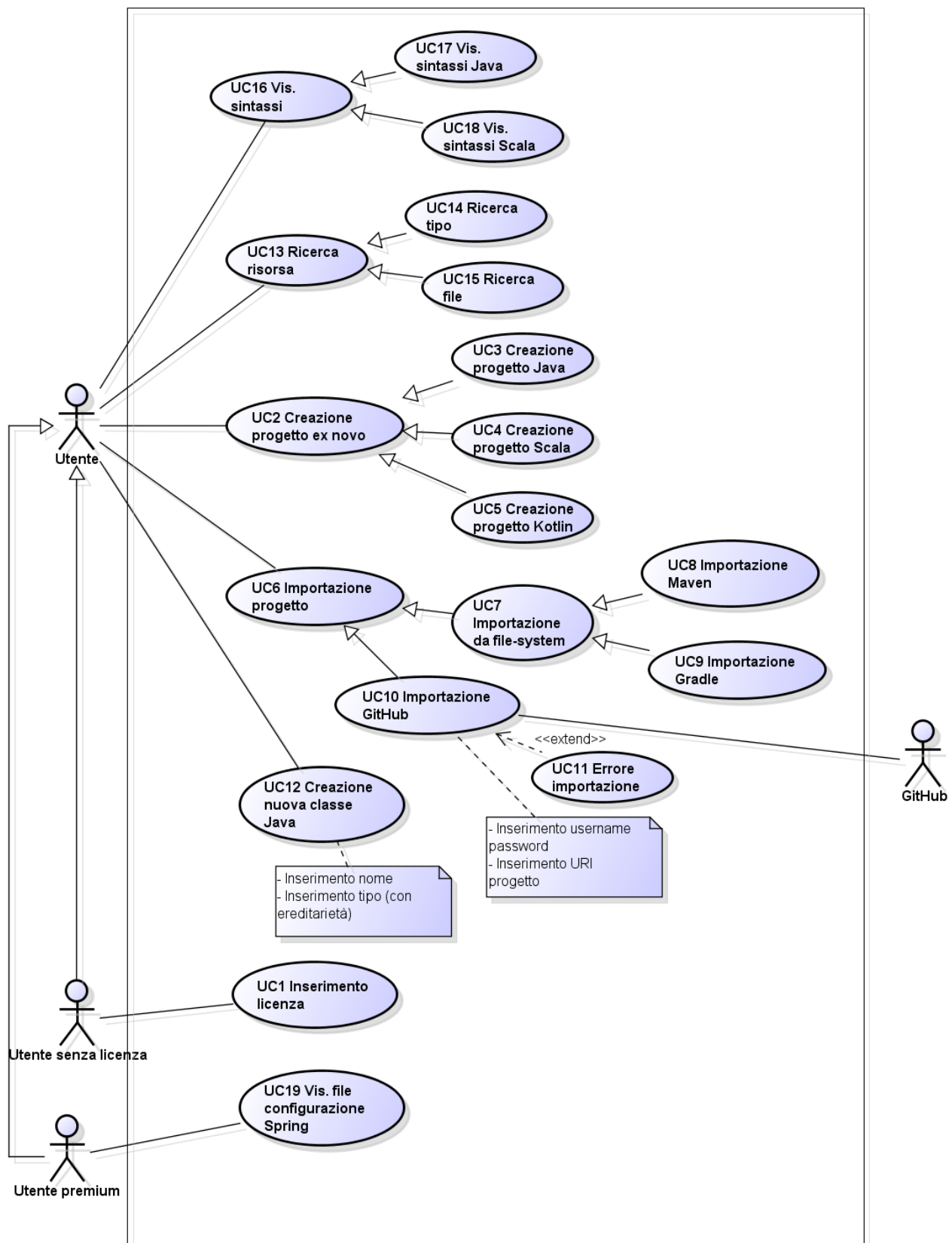
#### Descrizione

Uno dei più famosi programmi IDE per lo sviluppo Java è indubbiamente IntelliJ IDEA. Mentre nei suoi primi anni di vita il software era esclusivamente a pagamento, ora, essendo l'IDE di riferimento per lo sviluppo di applicazioni Android, prevede anche una versione CE non a pagamento. Le due versioni differiscono fra loro per le funzionalità offerte. Per accedere alla funzione premium è necessario disporre di una licenza valida, da inserire opportunamente all'interno del programma. Gli utenti non a pagamento possono di base creare diversi tipi di progetto: Java, Scala, Kotlin, ecc...E' possibile scegliere se creare un progetto *ex novo*, se importarlo da un progetto Maven o Gradle esistente su filesystem o se importarlo dal repository esterno GitHub. In questo caso, è necessario effettuare l'autenticazione al proprio account GitHub (inserendo username e password) e fornire l'URI del progetto da clonare in locale. La creazione di una nuova classe Java richiede l'inserimento del Nome della classe e del suo tipo (classe, interfaccia o enumerazione). La sintassi all'interno di un file viene evidenziata opportunamente dal programma, sia essa Java, Scala, ecc... Sono disponibili diversi tipi di ricerca delle risorse all'interno dell'IDE. La pressione di Ctrl+N permette di cercare direttamente tra i tipi (classi) del linguaggio scelto, mentre la pressione di Maiusc+Maiusc permette la ricerca di qualsiasi tipo di risorsa gestita. Gli utenti Premium, hanno un'integrazione maggiore con alcuni framework Java, come ad esempio con Spring. Nella versione Premium è possibile infatti risalire al file di configurazione Spring per una classe Java direttamente selezionandola.

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

La soluzione fa ampio uso dell'ereditarietà fra casi d'uso. Ove riportate le caratteristiche dei sottocasi d'uso come commento, è necessario effettivamente fornirli come diagramma.



## Esercizio 2 (7 punti)

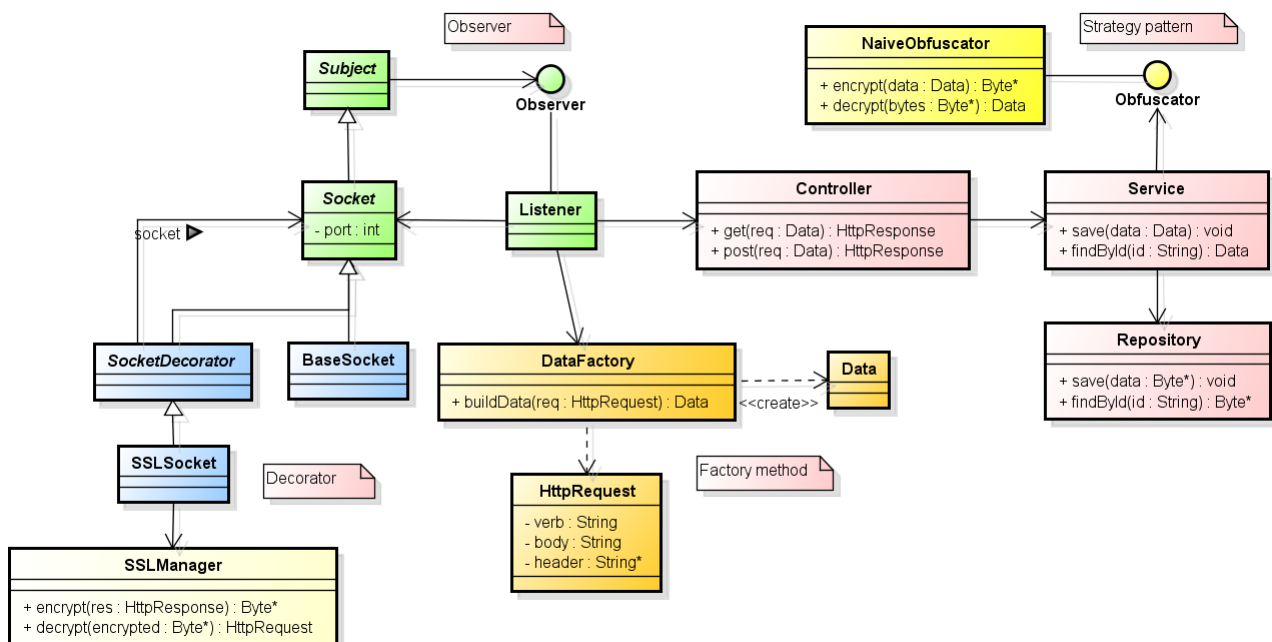
### Descrizione

Una giovane *startup* italiana vuole creare un servizio web che consenta ad un qualsiasi utente di salvare una stringa associandola ad una chiave. Il salvataggio avviene nel cloud della *startup*. Il servizio deve esporre un'interfaccia REST che attraverso l'utilizzo di richieste HTTP con i verbi HTTP POST e GET permettano rispettivamente di salvare e recuperare successivamente la coppia (key, value). La struttura dati fornita con le richieste (HTTP body) è simile al seguente JSON `{'key': 'chiave', 'value': 'valore'}`. Esso viene processato per creare una struttura dati opportuna che ne consente una più semplice elaborazione. Il servizio REST è implementato utilizzando un *listener* che resta in ascolto di connessioni sulla porta 80 nel caso di connessioni in HTTP. Esiste inoltre un'estensione del listener che utilizza la crittazione (HTTPS), restando in ascolto sulla porta 443. I dati crittati necessitano di essere ovviamente decifrati prima di essere elaborati ulteriormente. Le informazioni relative alla coppia (key, value) vengono infine salvate all'interno di un database in formato binario. Un algoritmo proprietario, ancora in via di miglioramento, comprime e decompone le informazioni verso e dal database.

Si modelli il sistema descritto utilizzando un diagramma delle classi e gli opportuni *design pattern*. Inoltre, si descriva utilizzando un diagramma di sequenza una richiesta di lettura del valore associato alla chiave pippo.

### Soluzione

La soluzione prevede l'utilizzo dei seguenti *design pattern*: Observer, Decorator, Factory Method e Strategy. La struttura è inoltre utilizza una classica struttura a *tier* Controller – Service – Repository.



### Esercizio 3 (3 punti)

#### Descrizione

Apache Spark è una libreria che consente di effettuare analisi sui Big Data, ossia su volumi di dati considerevoli e solitamente non strutturati. La componente *streaming* della libreria può essere utilizzata per eseguire analisi su grandi moli di dati il cui arrivo è continuo. Spark Streaming utilizza la tecnica del *micro batching*: anziché elaborare i dati appena arrivati uno alla volta, li accumula in gruppi di  $n$  (*chunk*), elaborandoli successivamente con una sola esecuzione.

Utilizzando un diagramma di attività e focalizzandosi sull'arrivo di un singolo *chunk*, si modelli una sua elaborazione, che consista nella trasformazione e scrittura su database di questi dati. Si preveda inoltre un *timeout* di 10 secondi, che faccia terminare prematuramente l'elaborazione nel caso in cui non arrivi alcun *chunk* da elaborare.

#### Soluzione

La soluzione prevede l'utilizzo di una regione di espansione.

