

Si consideri il caso di studio *SkiSystemSupport*. Si ignori tutto quanto riguarda le prenotazioni.

**Domanda 1. Requisiti.** Si fornisca il diagramma dei casi d'uso: Noleggio e RegolazioneAttacchi. Dare la narrativa del caso d'uso Noleggio

**Domanda 2. Analisi del dominio.** Si fornisca un diagramma delle classi che modella gli sci. Il modello costruito deve permettere di modellare i costi di noleggio.

**Domanda 3. Analisi del dominio.** Si fornisca un diagramma di macchina a stati che mostri l'evoluzione dello stato di un elemento di attrezzatura dopo che è stato consegnato al cliente. Ricordarsi che può essere restituito in tempo o in ritardo. In caso di ritardo nella riconsegna (dopo una settimana dall'inizio del noleggio nel caso di quota minima, o dopo una settimana dalla scadenza nel caso sia stata pagata una settimana), il sistema invia un sms al cliente chiedendo un nuovo anticipo. Se il cliente non ottempera, viene inviato un sms al giorno per la settimana successiva. Infine, si avvia una pratica per il recupero e il cliente è dichiarato inaffidabile.

\*\*\*

Per realizzare il caso d'uso Noleggio sono state individuate le componenti nella tabella seguente.

GestioneNoleggio	Realizza la funzionalità del caso d'uso, inclusa l'interazione col lettore di codici a barre.
InterfacciaAddetto	Realizza l'interfaccia utente.
DBclienti	Mantiene i clienti di <i>SkiSystemSupport</i> .
DBattrezzatura	Mantiene il catalogo dell'attrezzatura.

**Domanda 4. Architettura.** Si fornisca la vista ibrida C&C e deployment della parte dell'architettura di *SkiSystemSupport* relativa al caso d'uso Noleggio, sapendo che ogni componente è manifestata da un artefatto omonimo.

\*\*\*\*\*

Per implementare parte dell'algoritmo di noleggio, è stato realizzato il codice seguente, che verifica le condizioni del cliente che vuole noleggiare, ed eventualmente prepara l'attrezzatura richiesta.

```
public boolean concediNoleggio(Client cliente, Client[] clienti) {  
    if (! DBclienti.getStatus(cliente).equals("affidabile")) // 1  
        return false; // 2  
  
    preparaSci(clienti[0]); // 3  
  
    if (cliente.Maggiorenni()) // 4  
        for (int i = 1; i < clienti.length; i++) // 5  
            preparaSci(clienti[i]);  
  
    return true; // 6  
}
```

**Domanda 5. Verifica.** Si disegni il grafo di flusso per il codice dato, etichettando i nodi, per semplicità, con i numeri associati alle linee di codice. In aggiunta, si definisca un numero minimo di casi di test (senza considerare l'ambiente) per testare il metodo `concediNoleggio` e avere il 100% di copertura per il criterio delle decisioni. Si giustifichi la risposta indicando i nodi del grafo attraversati.