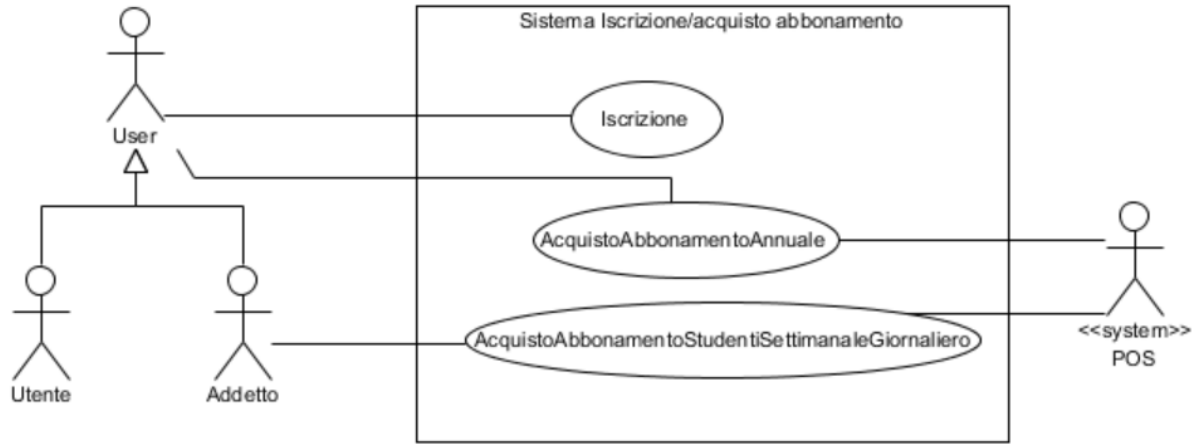
Corso di Ingegneria del software – Seconda verifica – 27 maggio 2015

F. Gadducci, L. Semini - Dipartimento di Informatica, Università di Pisa, a.a. 2014/15

La prova si svolge a libri chiusi (non è permessa la consultazione di materiale didattico)

L'analisi dei requisiti ha portato alla definizione del seguente diagramma dei casi d'uso relativo al sottosistema iscrizione/acquisto abbonamenti, e alla seguente narrativa di uno specifico caso d'uso, ovvero, AcquistoAbbonamentoStudentiSettimanaleGiornaliero.

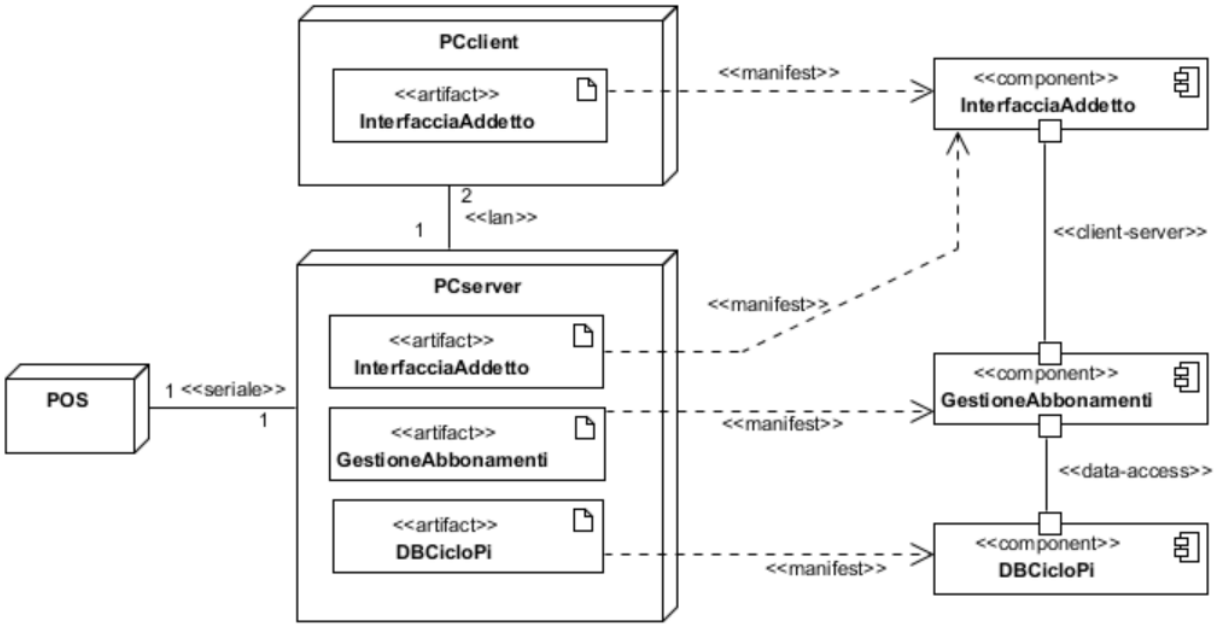


Caso d'Uso: AcquistoAbbonamentoStudentiSettimanaleGiornaliero	
Breve descrizione	Procedura per la registrazione al servizio CicloPi.
Attori primari	Addetto.
Attori secondari	POS.
Precondizioni	Cliente iscritto a CicloPi.
Sequenza eventi principale	1 L'addetto inserisce i dati del Cliente che vuole acquistare un nuovo abbonamento. 2 L'addetto indica il tipo di abbonamento. 3 Il sistema verifica se un eventuale credito residuo copre la ricarica obbligatoria. 4 Se (credito residuo non copre ricarica obbligatoria) 4.1 al costo dell'abbonamento base viene aggiunto quanto manca. 5 Il sistema calcola il prezzo da pagare e lo comunica all'addetto. 6 L'addetto indica al sistema se la somma è stata pagata in contanti o se il Cliente vuole pagare con il POS. 7 Se (il cliente sceglie il POS) 7.1 il sistema invia al POS la cifra che deve esser pagata; 7.2 il POS conferma che il pagamento è stato effettuato. 8 Il sistema aggiorna lo stato del Cliente. 9 Il sistema conferma all'addetto.
Postcondizione	Abbonamento acquistato, credito pari o superiore alla ricarica obbligatoria.
Sequenze eventi alternative	Il cliente rinuncia. Il POS non conferma il pagamento.

Per realizzare questo caso d'uso si prevede una architettura a 3 livelli, avendo individuato le seguenti componenti: InterfacciaAddetto, GestioneAbbonamenti, DBCicloPi. La dotazione hardware consiste in 3 PC (uno per ciascuno dei 3 addetti alla registrazione), uno dei quali funge anche da server per il sistema. A quest'ultimo è collegato il POS.

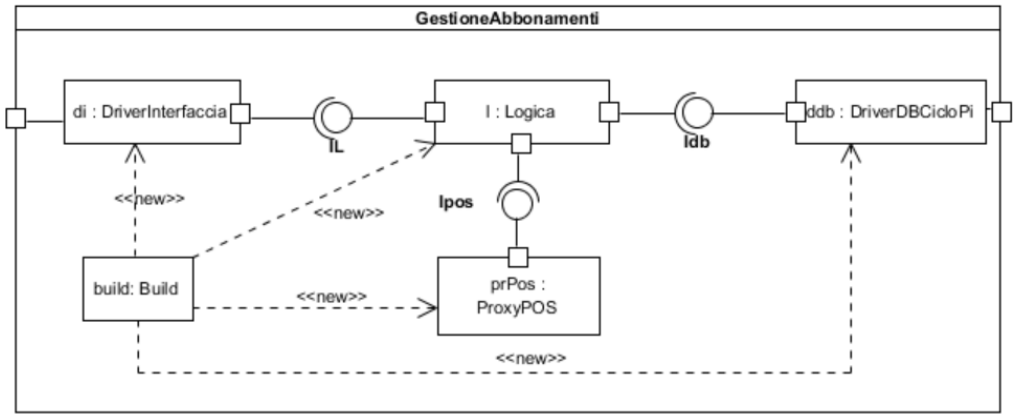
Domanda 1. *Architettura*.
Si fornisca la vista ibrida (C&C più dislocazione) del sottosistema considerato, sapendo che ogni componente è manifestata da un artefatto omonimo. GestioneAbbonamenti e DB sono dislocati sul PC server, le interfacce su ciascun PC.

Risposta.



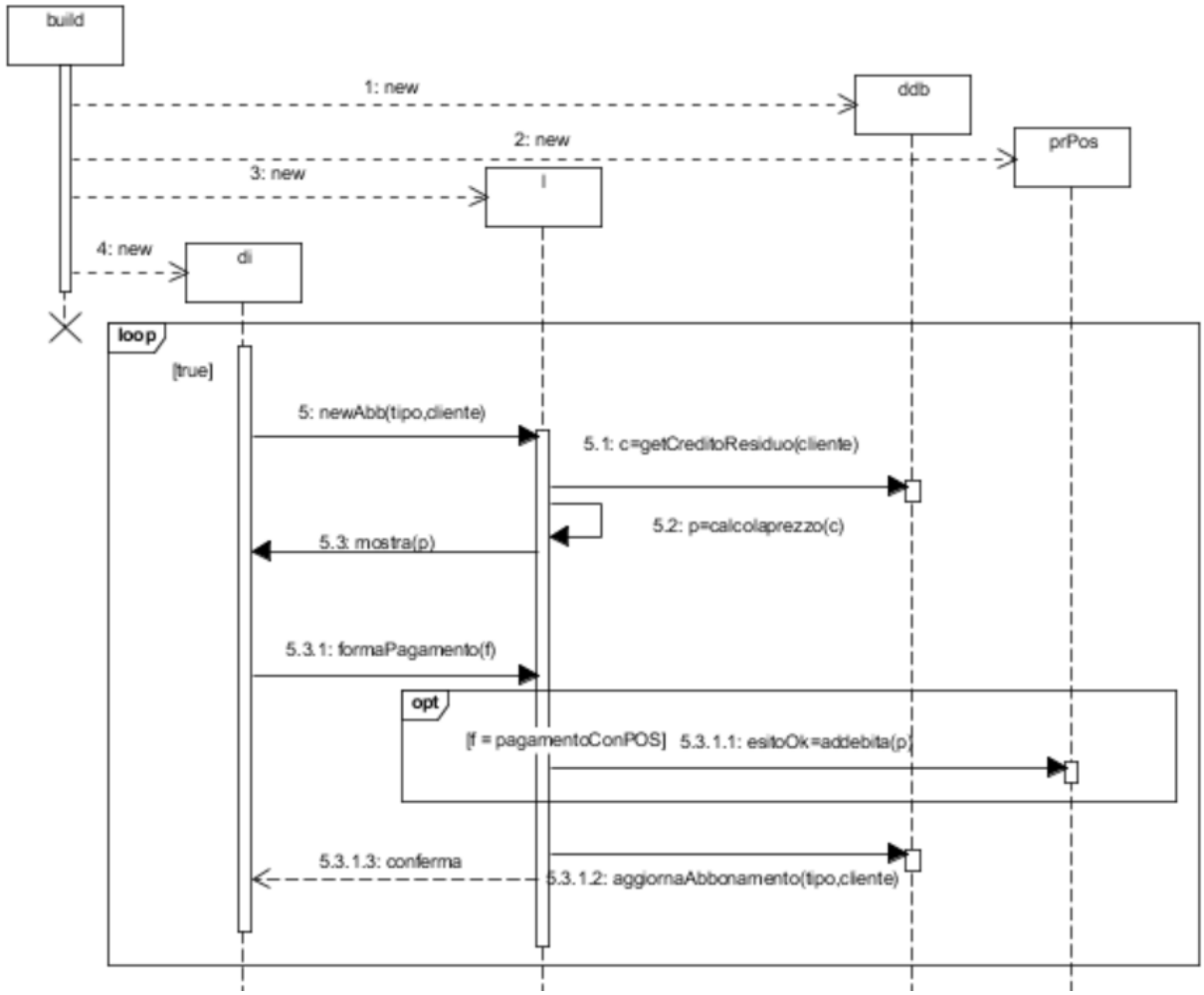
Domanda 2. *Progettazione di dettaglio*.
Si fornisca un diagramma di struttura composita della componente GestioneAbbonamenti. Si dettagli maggiormente il "diagramma minimo", supponendo che esista una parte di tipo Build che crea tutte le altre.

Risposta.

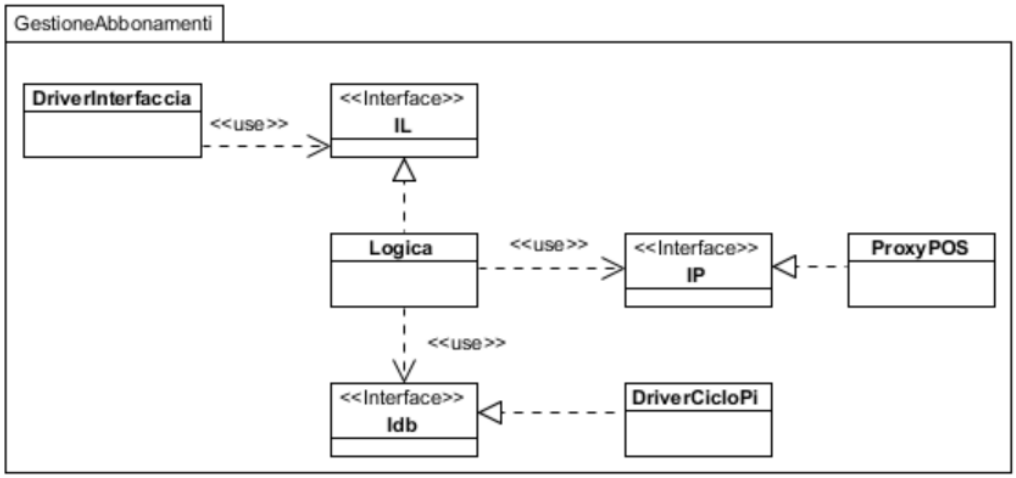


Domanda 3. *Realizzazione dei casi d'uso*.
Si fornisca un diagramma di sequenza che mostri come le parti individuate nell'esercizio 2 collaborano per realizzare il caso d'uso AcquistoAbbonamentoStudentiSettimanaleGiornaliero.

Risposta.



Domanda 4. Architettura.
Si fornisca la vista strutturale che mostra le relazioni d'uso tra le parti individuate nell'esercizio 2, incluse le interfacce. Per semplicità ignorare la *build*.
Risposta.



****

Per calcolare quanto deve essere pagato al momento dell'acquisto di un abbonamento, è stata definita la classe enumerazione TipoAbbonamento e scritto il codice seguente (per semplicità sono stati considerati solo due tipi di abbonamento).

```
private final double COSTO_SETTIMANALE = 8.00;
private final double RICARICA_SETTIMANALE = 2.00;
private final double COSTO_GIORNALIERO = 4.00;
private final double RICARICA_SETTIMANALE = 1.00;


private double credito;

public double DRIVERcalcolaCosto(TipoAbbonamento abb, double credito){
   double costoAbbonamento = 0;
   this.credito = credito;                                          // 1

   switch(abb){
      case SETTIMANALE:
         costoAbbonamento = calcolaCostoSettimanale(); break;
      case GIORNALIERO:
         costoAbbonamento = calcolaCostoGiornaliero();
   }
   return costoAbbonamento;                                         // 2
}

private double calcolaCostoSettimanale( ) {
   double costo = COSTO_SETTIMANALE;                                // 3

   if (credito < RICARICA_SETTIMANALE)
     costo = costo + (RICARICA_SETTIMANALE - credito);              // 4

   return  costo;                                                   // 5
}

private double calcolaCostoGiornaliero( ) {
   double costo = COSTO_GIORNALIERO;                                // 6

   if (credito < RICARICA_GIORNALIERA)
     costo = costo + (RICARICA_GIORNALIERA - credito);              // 7

   return  costo;                                                   // 8
}
```

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.

Si definisca un numero minimo di casi di test (senza considerare l'ambiente) per testare il metodo `DRIVERcalcolaCosto` (e i metodi che invoca) e avere 100% di copertura per il criterio dei cammini. Si giustifichi la risposta indicando i nodi del grafo attraversati.

Risposta.

Input		Output	Giustificazione
abb	credito		
SETTIMANALE	1.0	9.0	1,3,4,5,2
SETTIMANALE	3.0	8.0	1,3,5,2
GIORNALIERO	0.5	4.5	1,6,7,8,2
GIORNALIERO	2.0	4.0	1,6,8,2