Esercizi di Ingegneria del Software

V. Ambriola, C. Montangero e L. Semini

Appunti per il corso di Ingegneria del Software

Corso di Laurea in Informatica
Corso di Laurea in Informatica Applicata

Dipartimento di Informatica
Università di Pisa

INDICE

AVVERTENZE

Nei diagrammi di questa dispensa, compaiono talvolta alcuni simboli che non appartengono allo standard e dipendono dallo strumento utilizzato. Per quanto deprecabile, con questo fatto della vita bisogna convivere. Si tratta:

- della frecciolina in basso a sinistra di alcune icone: viene usata per indicare che l'elemento di modellazione rappresentato è stato introdotto in un diagramma diverso da quello in cui compare l'icona;
- di un 'lollipop' ridondante quando si indica un'interfaccia richiesta, di cui non si indica il realizzatore;
- di alcune ombreggiature 'estetiche' nelle icone.

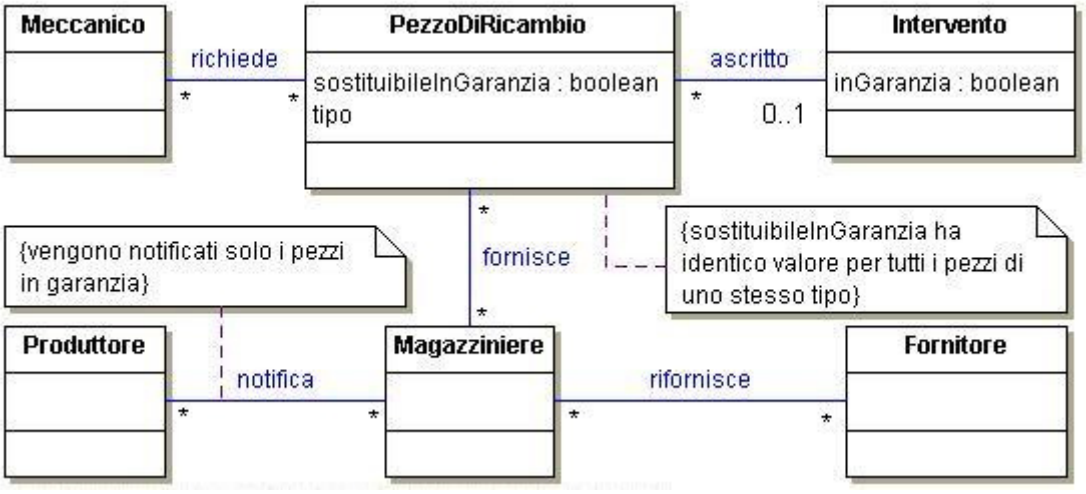
Per quanto riguarda gli esercizi di progettazione di dettaglio, utilizzeremo la seguente convenzione, relativa alle parti di una componente che comunicano con l'esterno: una parte che ha la delega di un porto avrà un nome che inizia per Driver; una parte che invece incapsula una comunicazione con l'esterno del sistema avrà un nome che inizia per Proxy.

CAP 1. Officina meccanica

Ci troviamo all'interno di un'officina meccanica, autorizzata da varie case automobilistiche a effettuare interventi di riparazione su automobili e autocarri. All'interno dell'officina incontriamo il proprietario che ci illustra le procedure seguite dal personale per rispettare le regole di comportamento introdotte all'inizio dell'anno, in seguito a un'azione di miglioramento della qualità del servizio.
«Il primo problema che abbiamo affrontato è relativo al controllo dei **pezzi di ricambio** usati in un **intervento**. Abbiamo introdotto un protocollo tra il **meccanico** che ripara un veicolo e il **magazziniere** che gestisce i pezzi di ricambio, per tenere sotto controllo le scorte e per notificare ai **produttori** la sostituzione di pezzi di ricambio che sono ancora in garanzia.»
Incuriositi da questa procedura chiediamo al proprietario maggiori dettagli su questo protocollo.
«L'idea di base è molto semplice. Ogni veicolo in riparazione è assegnato a un meccanico che si occupa di identificare le operazioni da effettuare e i pezzi di ricambio necessari per l'intervento. Anziché rendere disponibile a tutti i meccanici il magazzino abbiamo assunto un magazziniere che si occupa di analizzare le richieste provenienti dai meccanici e di fornire loro i pezzi di ricambio se sono disponibili in giacenza. Qualora un pezzo di ricambio non fosse disponibile, il magazziniere si preoccupa di richiederlo a uno dei nostri **fornitori** locali, eventualmente ripristinando il livello di scorta.»
«Come sono gestiti gli interventi in garanzia?»
«Non è così semplice come può sembrare! Per prima cosa è necessario stabilire se l'intervento è in garanzia, e questo lo decide l'amministrazione quando il veicolo entra in officina. Inoltre, a complicare il tutto, alcuni pezzi di ricambio sono sempre considerati di consumo e, quindi, sono esclusi dalla garanzia. Il magazziniere, sulla base di queste due informazioni, notifica ai produttori la sostituzione dei pezzi di ricambio in garanzia, per evitare che il loro costo sia attribuito al proprietario del veicolo.»
«In effetti è proprio diverso da come me l'ero immaginato.»

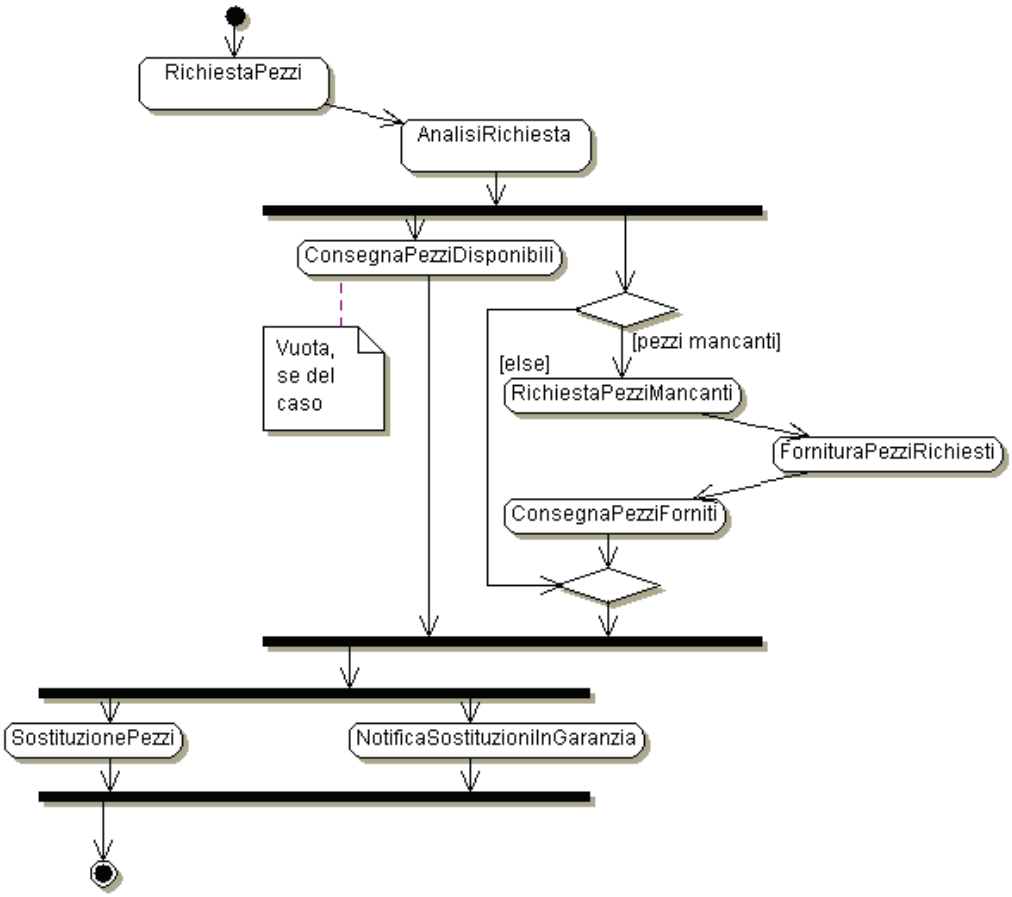
Domanda. (Analisi del dominio) Disegnare un diagramma delle classi relativo ai termini: pezzi di ricambio, intervento, meccanico, **magazziniere, produttori, fornitori**.

Una possibile **risposta** è la seguente:



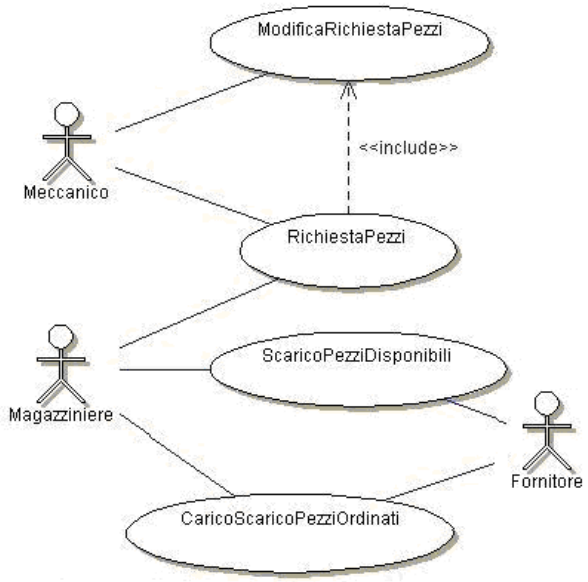
Si noti che anche quando nel testo il termine compare al plurale, lo si utilizza al singolare per nominare la classe. Questo permette di dire frasi del tipo "l'oggetto *dalmine* è un Fornitore".

Domanda. (Analisi del dominio) Disegnare un diagramma di attività che descriva la richiesta dei pezzi, come presentata nell'enunciato.
Una possibile **risposta** è la seguente:



La disposizione dei nodi azione suggerisce una divisione in partizioni: come?

Si consideri il seguente diagramma dei casi d'uso:



Le brevi descrizioni dei casi d'uso presentati nel diagramma sono date nel seguito:

Nome: ModificaRichiestaPezzi
Descrizione: Il meccanico modifica una richiesta già esistente.
Nome: RichiestaPezzi
Descrizione: Il meccanico fornisce al magazziniere l'elenco dei pezzi di ricambio da sostituire in un intervento.
Nome: ScaricoPezziDisponibili
Descrizione: Il magazziniere ottiene la lista dei pezzi di ricambio disponibili e quella dei pezzi di ricambio ordinati al fornitore.
Nome: CaricoScaricoPezziOrdinati
Descrizione: Il magazziniere ottiene la lista dei pezzi di ricambio forniti.

Domanda (Analisi dei requisiti)**.** Completare la descrizione del caso d'uso "RichiestaPezzi".

Una possibile **risposta** è la seguente:

Nome: RichiestaPezzi
Breve descrizione: Il meccanico fornisce al magazziniere l'elenco dei pezzi di ricambio da sostituire in un intervento.
Attore principale: Meccanico.
Attori secondari: Magazziniere.
Precondizioni: NA
Postcondizioni: La richiesta di pezzi di ricambio è stata notificata al Magazziniere.
Sequenza principale degli eventi:
1. il meccanico richiede al sistema l'elenco degli interventi affidatigli. Il sistema risponde. Il meccanico sceglie quello cui addebitare i pezzi richiesti;
2. il sistema recupera le informazioni necessarie a identificare i pezzi pertinenti (produttore, modello, ecc.);
3. per ogni pezzo da sostituire:
   3.1. il sistema fornisce la lista delle opzioni per la sostituzione
   3.2. il meccanico ne sceglie una, indicando il numero di pezzi necessari;
4. il sistema presenta la lista per l'approvazione finale;
5. il meccanico approva;
6. il sistema informa il magazziniere della nuova richiesta.

Sequenze alternative degli eventi:
1. Il meccanico non approva la lista (si esegue il caso d'uso "ModificaRichiestaPezzi").

Domanda. (Analisi dei requisiti) Completare la descrizione del caso d'uso "ScaricoPezziDisponibili", tenendo presente che prima di scaricare i pezzi disponibili dal magazzino e fornirne la lista, il magazziniere deve confermare l'operazione. Il magazziniere deve anche confermare la lista dei pezzi mancanti, da inviare al fornitore.

Una possibile **risposta** è la seguente:

Nome: ScaricoPezziDisponibili
Breve descrizione: Il magazziniere ottiene la lista dei pezzi di ricambio disponibili e quella dei pezzi di ricambio ordinati al fornitore, per una data richiesta di pezzi da sostituire.
Attore principale: Magazziniere.
Attori secondari: Fornitore.
Precondizioni: NA
Postcondizioni: I pezzi disponibili utili per un intervento sono stati scaricati, eventuali pezzi indisponibili sono stati ordinati.

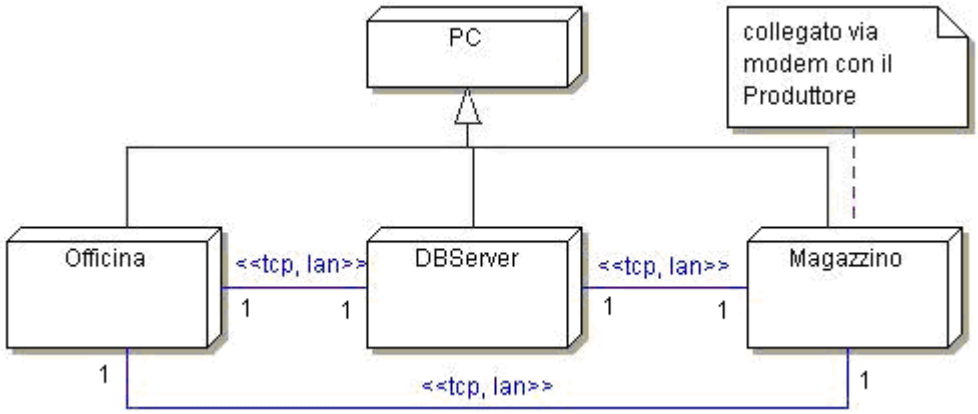
Sequenza principale degli eventi:

1. il magazziniere richiede al sistema l'elenco delle richieste pendenti; il sistema risponde
2. il magazziniere comunica al sistema quale servire;
3. il sistema recupera le informazioni necessarie a identificare i pezzi presenti e presenta la lista al magazziniere;
4. il magazziniere approva;
5. il sistema stampa la lista (che il magazziniere userà per andare a cercare i pezzi), e scarica i pezzi dall'inventario;
6. il sistema presenta la lista dei pezzi mancanti;
7. il magazziniere approva;
8. il sistema trasmette al fornitore le richieste.

Sequenze alternative degli eventi:

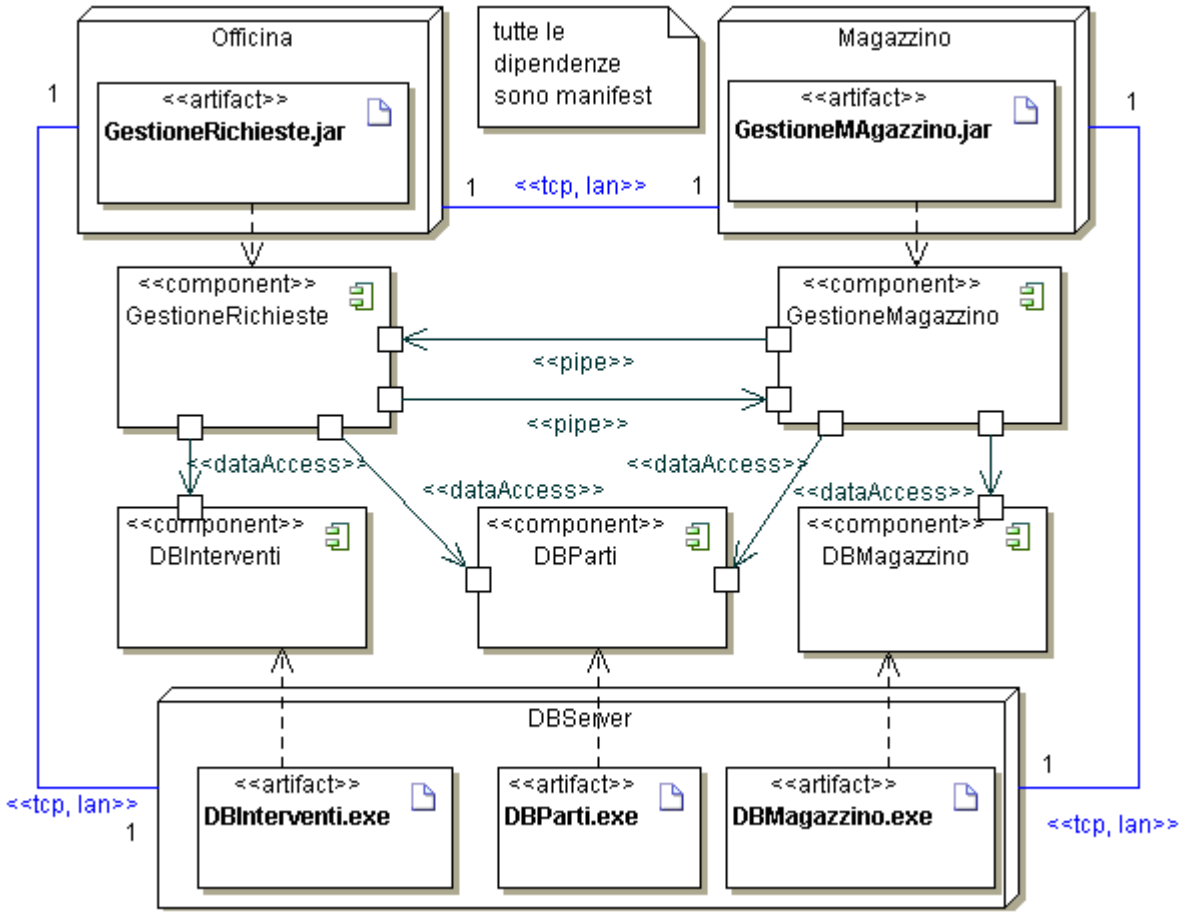
1. il magazziniere non approva la lista dei pezzi presenti (il caso d'uso termina senza conseguenze);
2. il magazziniere non approva la lista dei pezzi mancanti (la lista viene reinserita come una nuova richiesta pendente, associata all'originale).

Domanda (Architettura)**.** L'hardware a disposizione per realizzare il sistema di gestione dell'officina è descritto nella figura seguente.



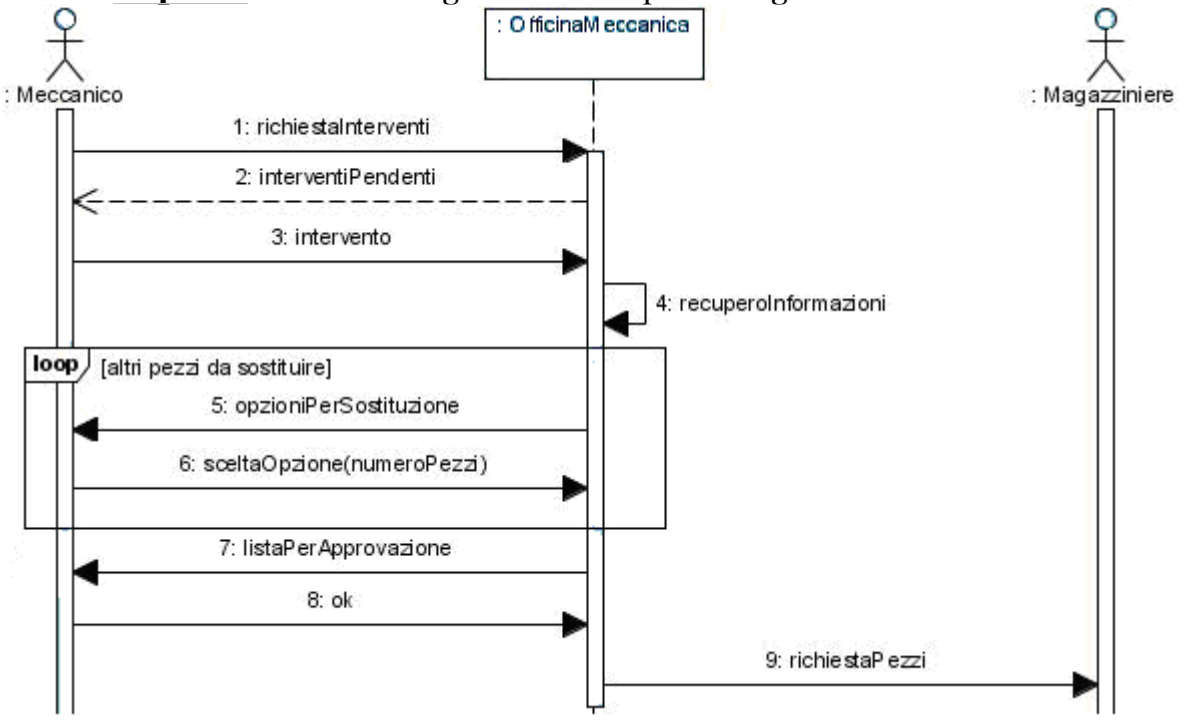
Il nodo DBServer è dedicato ai data base. Dare una vista ibrida sull'architettura software relativa ai casi d'uso presentati che mostri:
a) una vista comportmentale C&C;
b) la relativa vista logistica di dislocazione.
La soluzione deve introdurre una componente separata di gestione del magazzino, e utilizzare, per la connessione tra le componenti che realizzano la logica e quelle che memorizzano i dati, un connettore <<dbAccess>>. La connessione tra le parti che realizzano la logica, data la natura delle interazioni, può essere di tipo pipe. Si assuma che il tipo degli artefatti dislocati sia jar, per le parti che realizzano la logica dell'applicazione, e exe per quelle che gestiscono i dati.

Possibile **risposta.** Si veda la figura seguente, dove tutte le dipendenze sono <<manifest>>, e abbiamo tenuto conto che, per realizzare i casi d'uso, possiamo introdurre due componenti per la gestione della logica dell'applicazione, una sul lato del magazzino e una sul lato officina, che sfruttano dei gestori di dati, relativi alle Parti (per la presentazione al meccanico, per sapere se un pezzo e' sostituibile, e per decidere se un pezzo è in garanzia), agli Interventi (per associarlo alla richiesta) e al Magazzino, per decidere gli eventuali ordini e per scaricare gli elementi consegnati.



Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che mostri come l'architettura realizza la sequenza principale del il caso d'uso RichiestaPezzi.

Una possibile **risposta** è data dal diagramma di sequenza seguente.

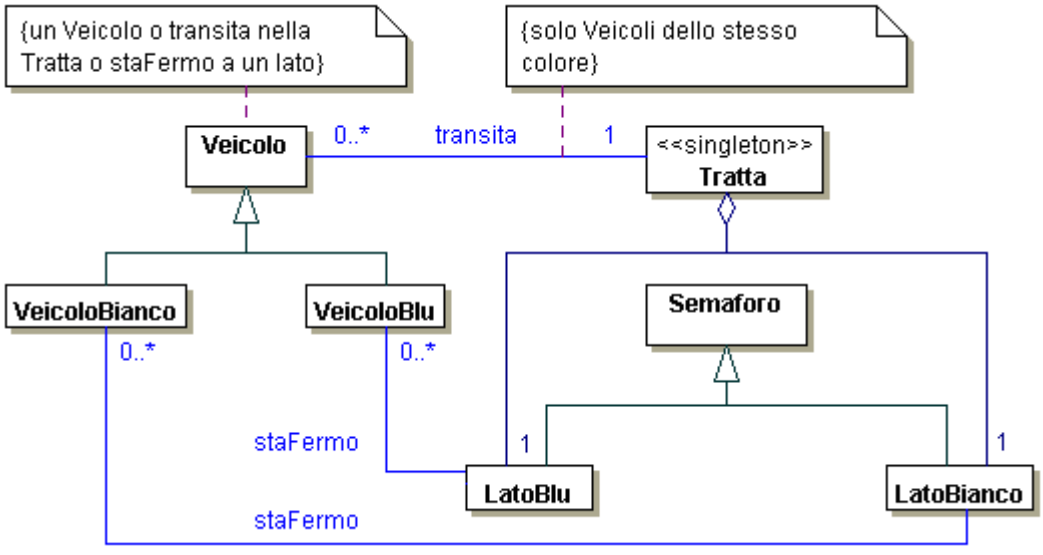


CAP 2. Semafori 1

Spett. Ditta,
In seguito all'incontro tenuto presso la nostra sede vi trasmettiamo le informazioni richieste dal vostro analista.
Il sistema di controllo del traffico veicolare sulle **tratte** stradali interessate a lavori di manutenzione straordinaria, per i quali non è possibile provvedere in anticipo all'incanalamento standard, deve essere in grado di operare in due modalità: connessa e autonoma. In entrambe le modalità la coppia di **semafori** a due luci (l'uso della luce gialla è sostituito dalla temporizzazione a doppia luce rossa) provvede a bloccare il traffico da uno dei due lati (convenzionalmente chiamati blu e bianco) e a consentire il deflusso dei **veicoli** accodati dall'altro. In pratica, i **veicoli blu** (accodati dal lato del rispettivo colore) attendono l'accensione della luce verde e, ottenuto il consenso, procedono all'attraversamento del tratto stradale interessato. Dopo un periodo di tempo, configurabile dall'operatore al momento della predisposizione del sistema sul campo, il **semaforo del lato blu** spegne la luce verde e accende quella rossa per consentire lo sgombero del tratto stradale. La durata del periodo a doppia luce rossa è calcolata dal sistema in base alla lunghezza del tratto stradale e alla velocità media di percorrenza (dati configurabili dall'operatore). Al termine di questo periodo, il **semaforo del lato bianco** spegne la luce rossa e accende quella verde, iniziando per i **veicoli bianchi** un ciclo analogo a quello precedente. Per motivi di sicurezza, il sistema è dotato di un meccanismo di blocco/sblocco che interrompe il ciclo di luce verde e attiva la condizione di doppia luce rossa. Lo stato bloccato è esplicitamente liberato dall'operatore mediante il meccanismo di blocco/sblocco.
Restiamo in attesa di un vostro riscontro per procedere alla fase successiva del progetto.
Cordiali saluti,
Ing. Franco Strada

Domanda. (Analisi del dominio) Dare un diagramma delle classi, relativo ai concetti evidenziati in neretto nell'enunciato.

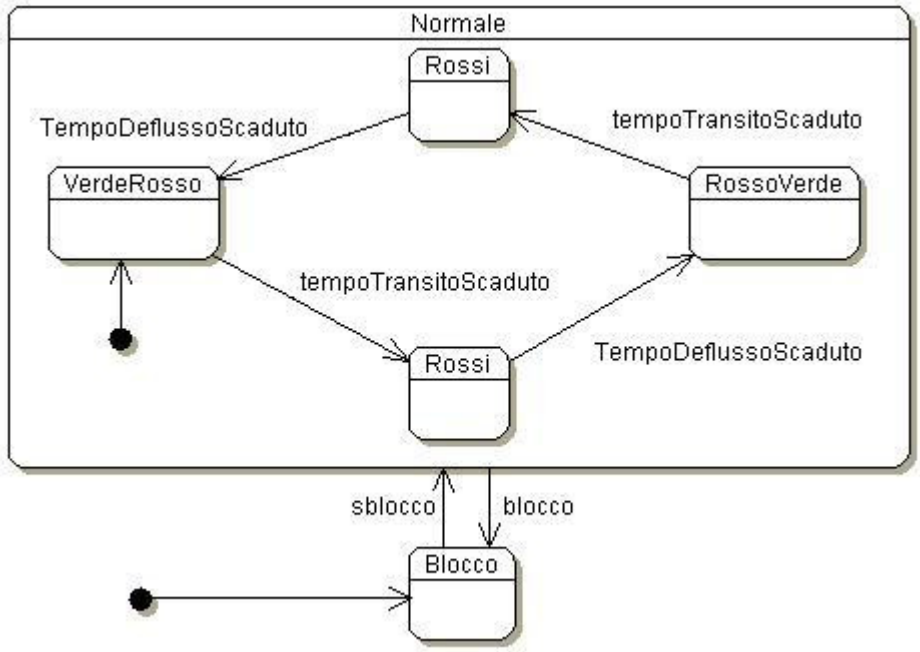
Una possibile **risposta** è la seguente:



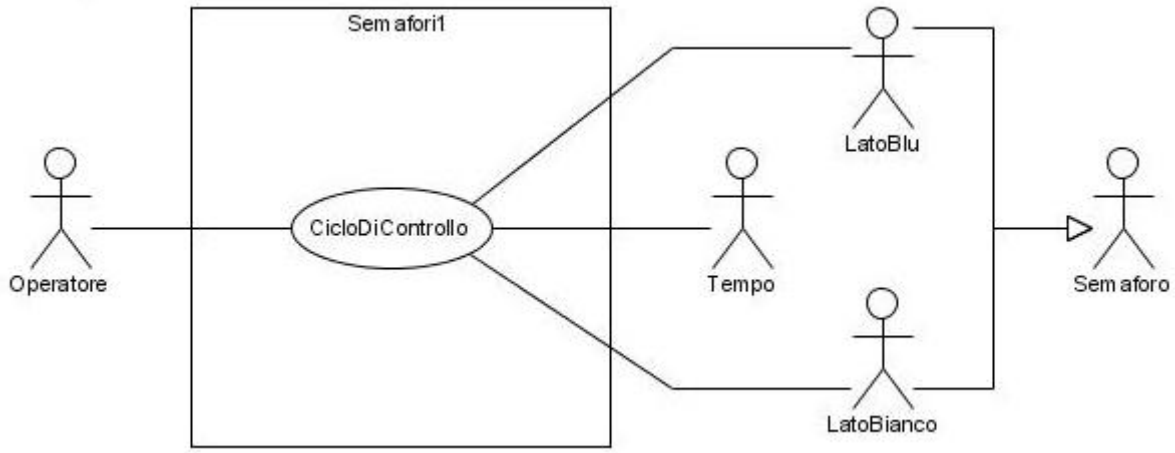
Nota. Lo sterotipo <<singleton>> indica che della classe si può avere una sola istanza.

Domanda. (Analisi del dominio) Dare una macchina a stati che descriva l'evoluzione nel tempo del sistema costituito dai due semafori.

Una possibile **risposta** è la seguente:



L'analisi dei requisiti ha portato al seguente diagramma dei casi d'uso:



Le brevi descrizioni dei casi d'uso presentati nel diagramma sono date nel seguito:

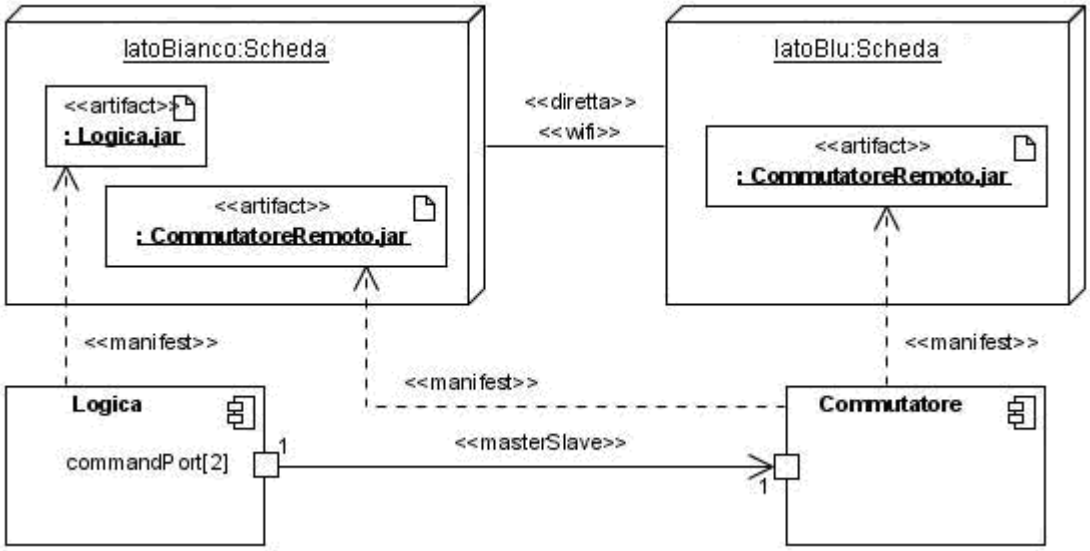
Nome: CicloDiControllo
Breve descrizione: Il sistema, una volta avviato dall'operatore, guida i semafori nel ciclo rosso-verde, in base ai segnali dall'orologio Tempo. Mediante il pulsante *blocco* (sul latoBianco) l'operatore può bloccare il ciclo e portare i semafori in posizione di sicurezza (entrambi rossi).

Sulla base di un'analisi dei casi d'uso si è arrivati a decidere un'architettura con due componenti, *Logica*, che realizza la logica di controllo, e *Commutatore* che riceve e applica i

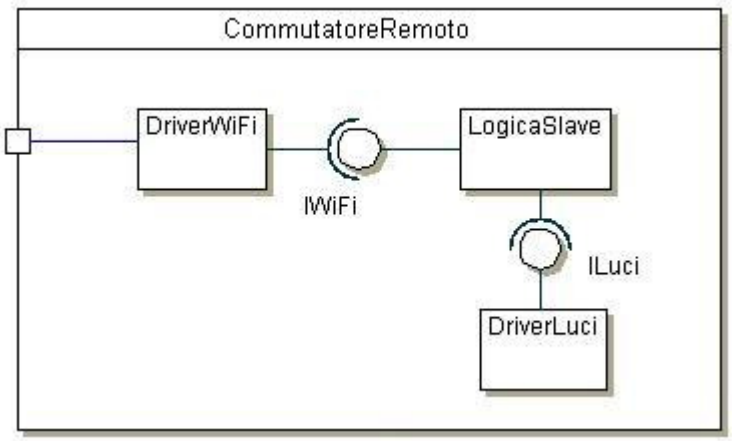
comandi di commutazione generati da Logica per un semaforo. Le manifestazioni delle componenti (una per Logica e una per ciascun semaforo) sono da dislocare su due dispositivi di calcolo (essenzialmente due schede madri), uno per lato, connessi da un collegamento diretto senza fili, con protocollo WiFi. Le apparecchiature hardware da utilizzare sono descritte dalla seguente figura:

Domanda. (Architettura) Dare una vista ibrida (dislocazione e C&C) dell'architettura del sistema che rappresenti le scelte indicate sopra, assumendo di dislocare file di tipo jar.

Una possibile **risposta**, in forma istanza, è la seguente, dove si può osservare che le manifestazioni di Commutatore sono diverse, a seconda che si trovino sul lato master o su quello client: la realizzazione delle connessioni è in effetti diversa nei due casi: via wifi nel caso remoto, via comunicazioni tra processi nel caso locale.

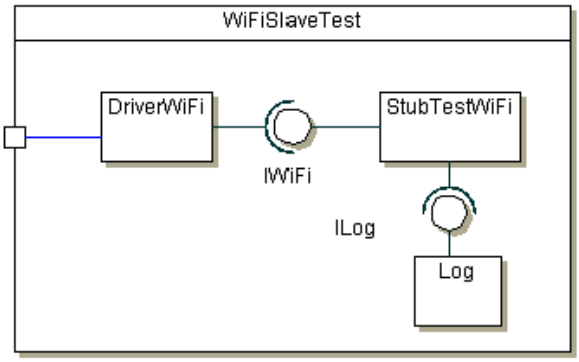


Il seguente diagramma definisce la struttura della componente *CommutatoreRemoto*. Il compito della parte LogicaSlave è di convertire il formato dei segnali in ingresso in comandi per le luci.

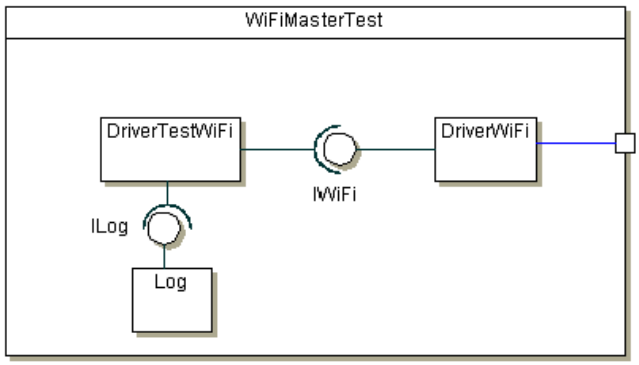


Per verificare il corretto funzionamento della comunicazione WiFi si vuole eseguire una batteria di test in una configurazione descritta dal seguente diagramma:

La struttura della componente *WiFiSlaveTest* è data dal diagramma che segue. Il diagramma, oltre alla parte da verificare, DriverWiFi, mostra uno *stub* che implementa l'interfaccia IwiFi raccogliendo i dati ricevuti dalla componente e registrandoli sul registro *Log*, per il successivo confronto con i dati spediti, raccolti in un simile file nell'altra componente. Si noti che la componente master ha un solo porto, sufficiente per la configurazione di test.



Domanda. (Progettazione di dettaglio) Dare la struttura della componente *WiFiMasterTest*.
Risposta: si veda la figura seguente:



Domanda. (Verifica) Il processo del produttore, come primo passo per la costruzione delle prove, prevede la definizione informale dei casi di prova, nella forma [descrizione dei valori di input, descrizione dei risultati attesi].
Definire il minimo insieme di casi di prova per la convalida della comunicazione WiFi, nella configurazione indicata sopra.

Risposta. Una possibile selezione di casi di test è data dalla seguente tabella, che riflette il fatto che le informazioni sul canale WiFi riguardano solo il colore da accendere sul lato *Slave*.

Casi di test: comunicazioni WiFi		
No	Input: colore da accendere	Output: colore da accendere
1	Rosso	Rosso
2	Verde	Verde

CAP 3. Semafori 2

Spett. Ditta,

In seguito alla sperimentazione sul campo del sistema di controllo del traffico veicolare sono emerse alcune situazioni che richiedono un'evoluzione del sistema. In particolare, si è osservato che quando la lunghezza della tratta stradale controllata è maggiore di 500 metri spesso i tempi di attesa sono eccessivi e, di conseguenza, rallentano il traffico. Per mitigare questo fenomeno, i nostri tecnici hanno proposto l'uso di quattro **sensori** di passaggio direzionale dei veicoli, per consentire al sistema di rilevare l'assenza di veicoli in transito su entrambi i lati della tratta. Un **sensore in** sarà posizionato prima del semaforo blu (la distanza di posizionamento dipende dalle caratteristiche della tratta) per rilevare il passaggio dei veicoli che arrivano dal lato blu. Un **sensore out** sarà posizionato all'altezza del semaforo bianco, per rilevare il passaggio dei veicoli arrivati dal lato blu che hanno completato l'attraversamento della tratta. Altri due sensori sono posizionati simmetricamente, per rilevare il passaggio dei veicoli bianchi. Il comportamento del sistema deve essere modificato in modo da interrompere anticipatamente il ciclo verde in assenza di veicoli in transito. Per evitare fenomeni di oscillazione, il ciclo verde deve avere una durata minima *tempoMin*, configurabile dall'operatore al momento della predisposizione del sistema sul campo.

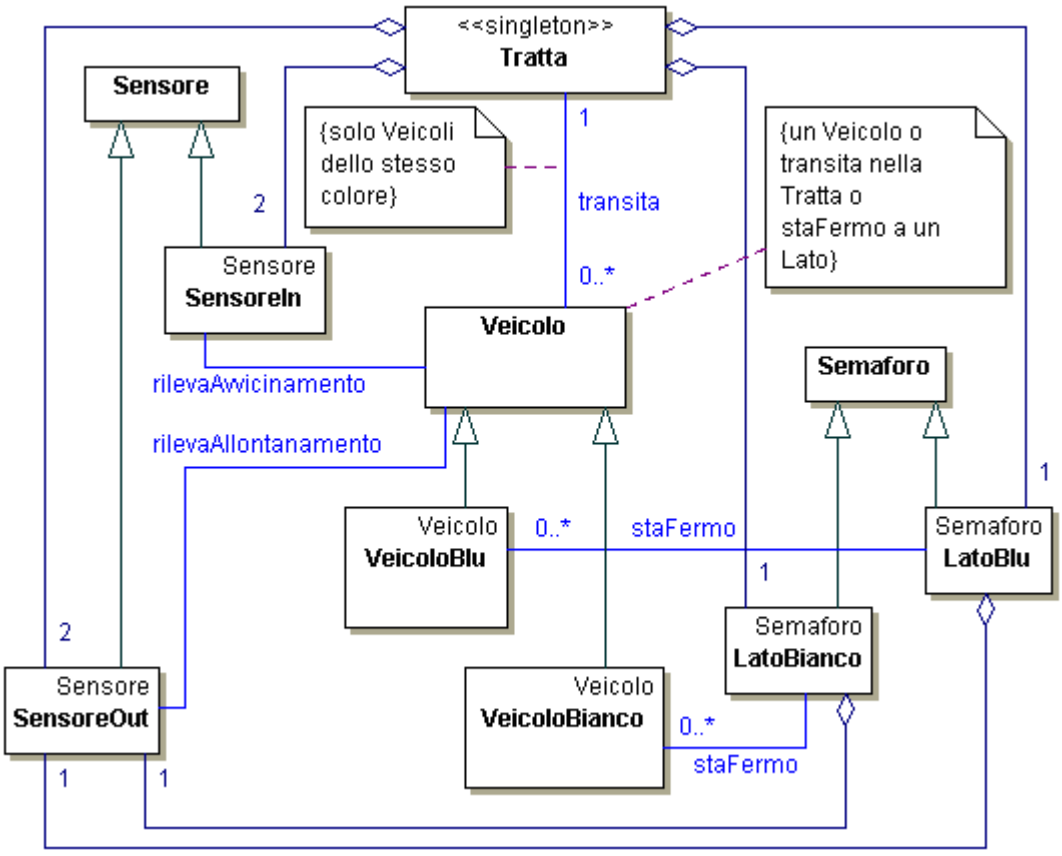
Restiamo in attesa di un vostro riscontro.

Cordiali saluti,

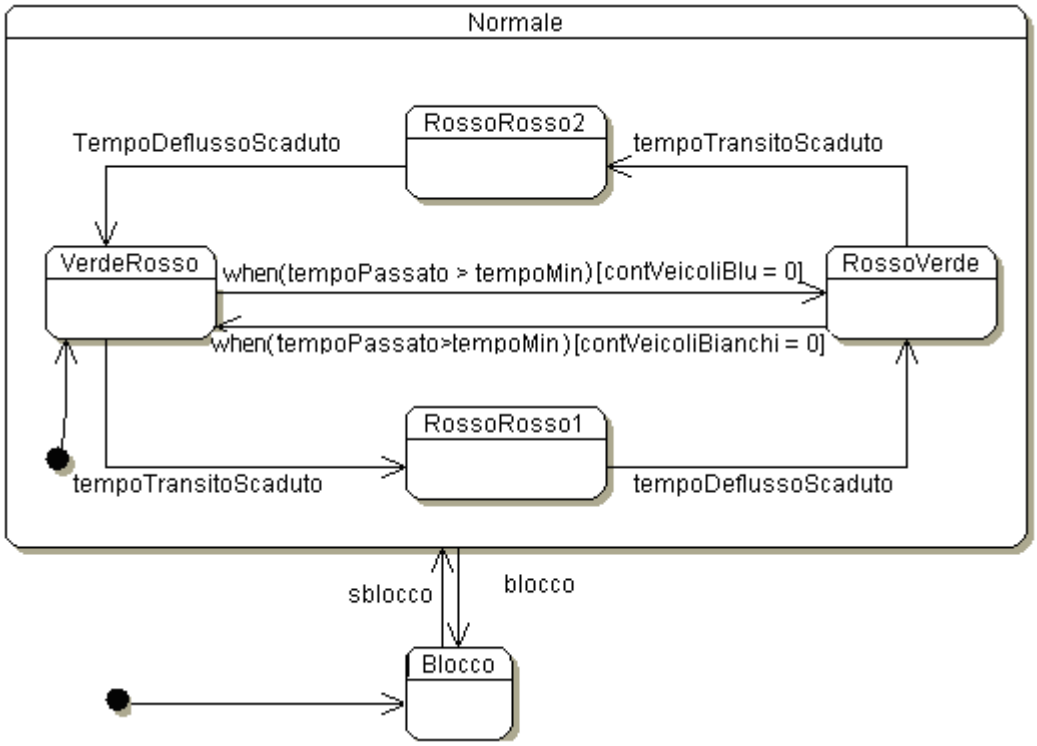
Ing. Franco Strada

Domanda. (Analisi del dominio) Dare un diagramma di classi, relativo ai concetti evidenziati in neretto negli enunciati Semaforo1 e Semaforo2.

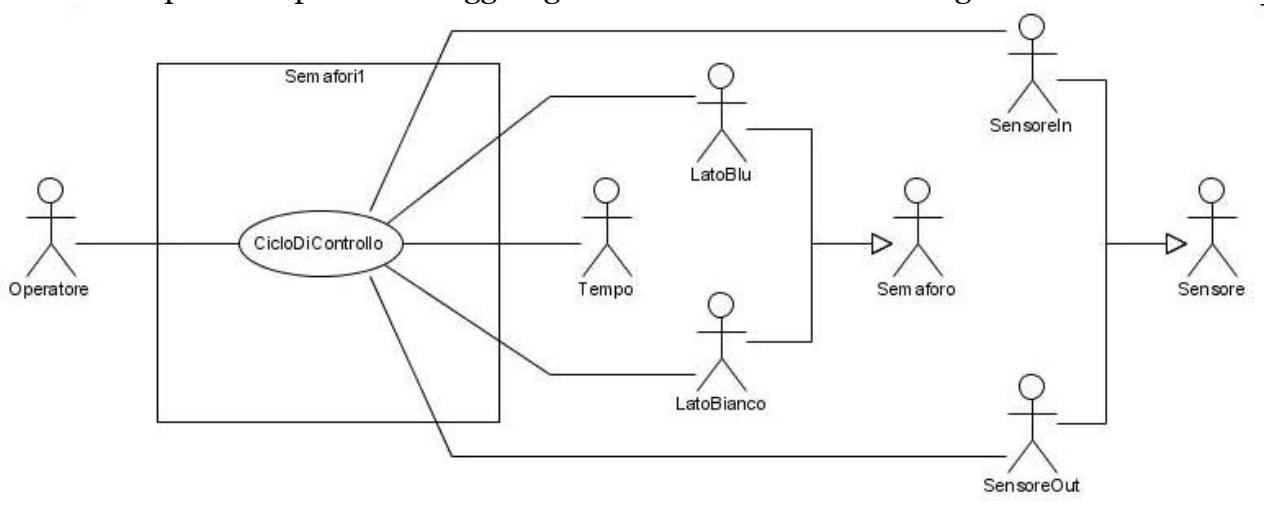
Una possibile **risposta** è la seguente:



Domanda. (Analisi del dominio) Dare una macchina a stati che descriva l'evoluzione nel tempo del sistema costituito dai due semafori.
Una possibile **soluzione** è la seguente:



L'analisi dei requisiti ha portato ad aggiungere ai casi di Semaforo1 il seguente:



Le brevi descrizioni dei casi d'uso sono state modificate per adattarle alla nuova situazione nel seguito:

Nome: CicloDiControllo
Breve descrizione: Il sistema guida i semafori nel ciclo rosso-verde, in base ai segnali del Tempo e all'andamento dei contatori. Mediante il pulsante *blocco* (sul latoBianco) l'operatore può bloccare il ciclo e portare i semafori in posizione di sicurezza (entrambi rossi).

Domanda. (Analisi dei requisiti) Dare la narrativa del caso d'uso CicloDiControllo, considerando il blocco come situazione eccezionale.

Una possibile **soluzione** è la seguente.

Nome: CicloDiControllo.
Breve descrizione: Il sistema guida i semafori nel ciclo rosso-verde, in base ai segnali di Tempo e agli azzeramenti dei contatori dei veicoli presenti. Mediante il pulsante *blocco* (sul latoBianco) l'operatore può bloccare il ciclo e portare i semafori in posizione di sicurezza (entrambi rossi).
Attore principale: Operatore.
Attori secondari:LatoBlu, LatoBianco, SensoreIn, SensoreOut.
Precondizioni: NA
Postcondizioni: NA (Trascurando il blocco si tratta di un ciclo senza fine).
Sequenza principale degli eventi:

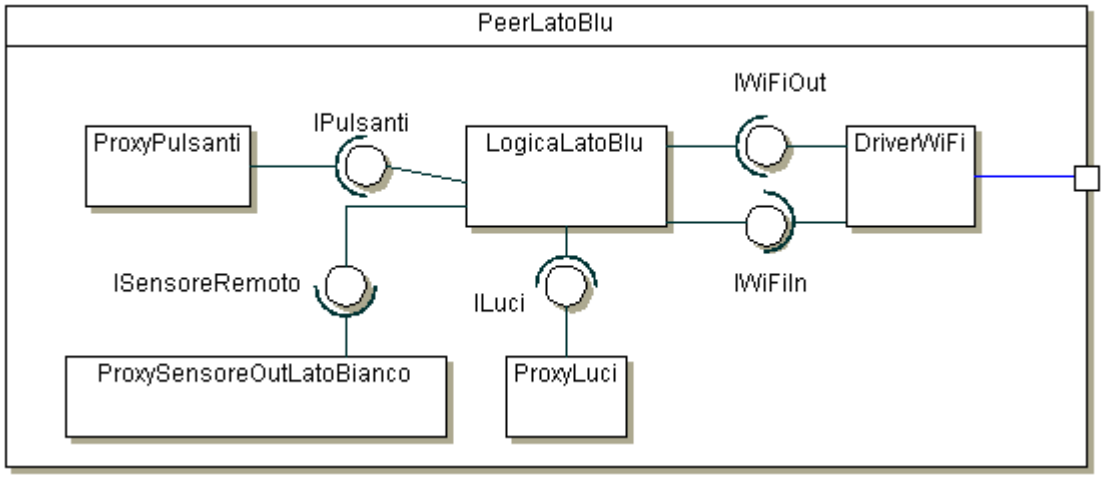
1. L'operatore, premendo il pulsante blocco, inizializza il sistema: latoBlu a verde, latoBianco a rosso, entrambi i contatori azzerati;
2. while(true)
   2.1. Tempo segnala che è passato tempoMin dalla transizione a verde
   2.2. se (contatore delle macchine in transito è nullo)
      2.2.1. il sistema commuta entrambi i lati;
   2.3. altrimenti
      2.3.1. Tempo segnala tempoTransitoScaduto
      2.3.2. Il sistema commuta il lato verde a rosso
      2.3.3. Tempo segnala tempoDeflussoScaduto
      2.3.4. Il sistema commuta l'altro lato (sempre rosso) a verde.

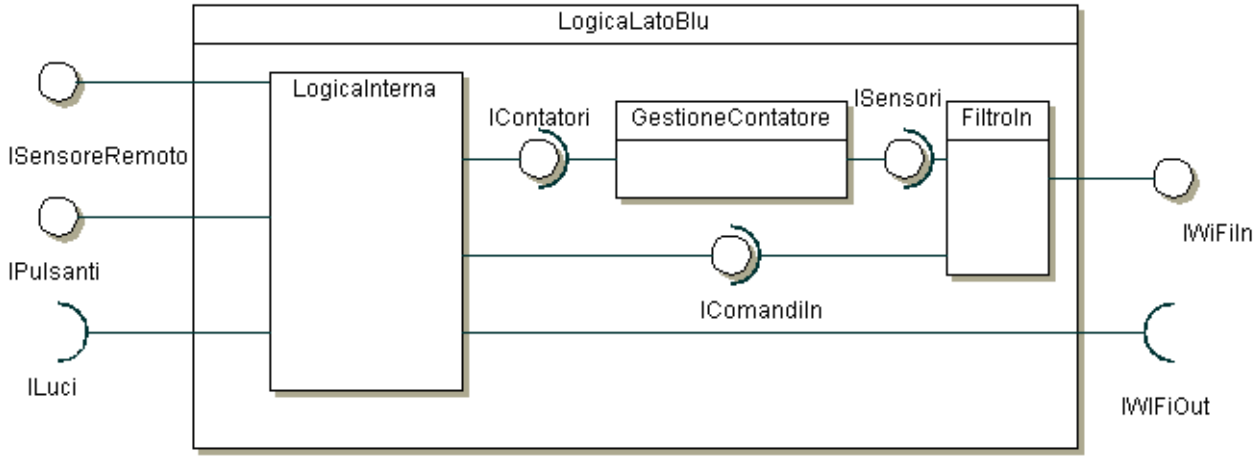
Situazioni eccezionali: se l'operatore preme il pulsante *blocco*, il sistema porta entrambi i lati a rosso.

Sulla base di un'analisi dei casi d'uso si è arrivati a decidere un'architettura con due componenti, in stile *peer to peer*, come mostrato nel seguente diagramma C&C:

che non richiede ulteriori commenti, se non che il connettore tra i due *peer* permette la comunicazione nei due versi, senza interferenze. Le componenti sono da dislocare su due dispositivi di calcolo (essenzialmente due schede madri), uno per lato, e sono connesse fisicamente da un collegamento WiFi. Il *sensore in* del LatoBlu è posto alcuni metri prima del semaforo del lato blu, ed è collegato al dispositivo di calcolo del LatoBlu con un collegamento diretto senza fili. Il *sensore out* del LatoBlu è posto in prossimità del semaforo del lato bianco ed è collegato al dispositivo di calcolo LatoBianco con un collegamento seriale. I sensori del LatoBianco sono collegati in modo simmetrico.

La progettazione di dettaglio della componente *PeerLatoBlu* ha portato ai seguenti diagrammi di struttura composita:





Domanda. (Architettura) Descrivere le responsabilità di ciascuna parte o classe nei diagrammi precedenti, tranne *LogicaLatoBlu* e *LogicaInterna*.

Risposta

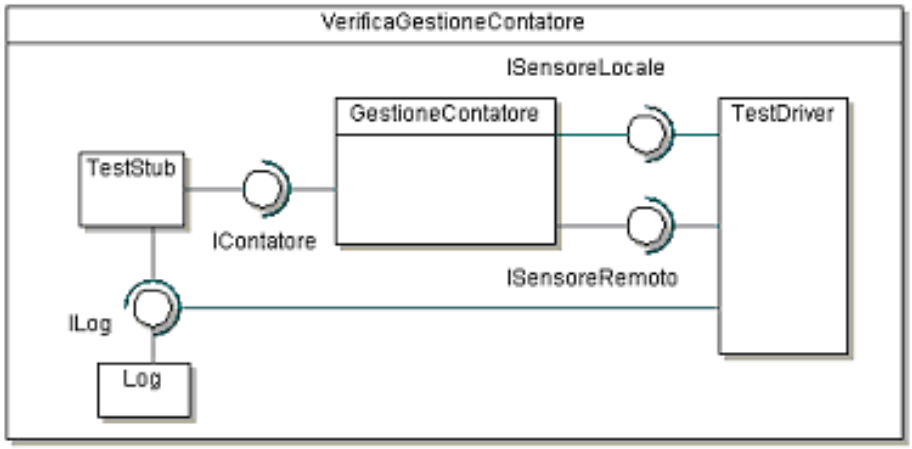
Nel primo diagramma:

1. *DriverPulsanti* gestisce l'interazione con i pulsanti di blocco e sblocco, trasformando i segnali generati premendoli in chiamate alla logica, secondo l'interfaccia *IPulsanti*.
2. *DriverLuci* gestisce l'interazione con le luci, trasformando le chiamate della logica, secondo l'interfaccia *ILuci* in segnali di accensione/spegnimento.
3. *DriverWiFi* gestisce l'interazione con il canale senza fili, trasformando
   3.1. le chiamate della logica, secondo l'interfaccia *IWiFiOut* in segnali verso l'altro *peer*;
   3.2. i segnali provenienti dall'altro *peer* o dal *sensore in* posizionato sul LatoBlu, in chiamate alla logica, secondo l'interfaccia *IWiFiIn*.
4. *DriverSensoreOutLatoBianco* gestisce l'interazione con il *sensore out* locale che rileva i veicoli bianchi in uscita, trasformando i segnali generati dal sensore in chiamate alla logica, secondo l'interfaccia *ISensoreLocale*.

Nel secondo diagramma:

1. *FiltroIn* filtra i segnali in ingresso sul canale senza fili, trasformandoli in chiamate alla logica, secondo l'interfaccia *IComandiIn*, se sono comandi di commutazione della luce, o in chiamate a *GestioneContatore*, secondo l'interfaccia *ISensoreRemoto*, se sono segnali provenienti dal sensore di traffico remoto.
2. *GestioneContatore* trasforma i segnali provenienti dai sensori, secondo l'interfaccia *ISensori*, in eventi che segnalano l'azzeramento del contatore sull'interfaccia *IContatore*.

Domanda. (Verifica) Per verificare la classe *GestioneContatore* si è progettata la componente descritta di seguito:



Definire una batteria di casi di prova per la convalida della gestione del contatore di veicoli in transito, nella configurazione indicata sopra.

Risposta. Una possibile selezione di casi di test è data dalla seguente tabella.
N.B: L'ambiente di prova prevede che il contatore gestito dalla classe sia a zero all'inizio di ogni sequenza di test.

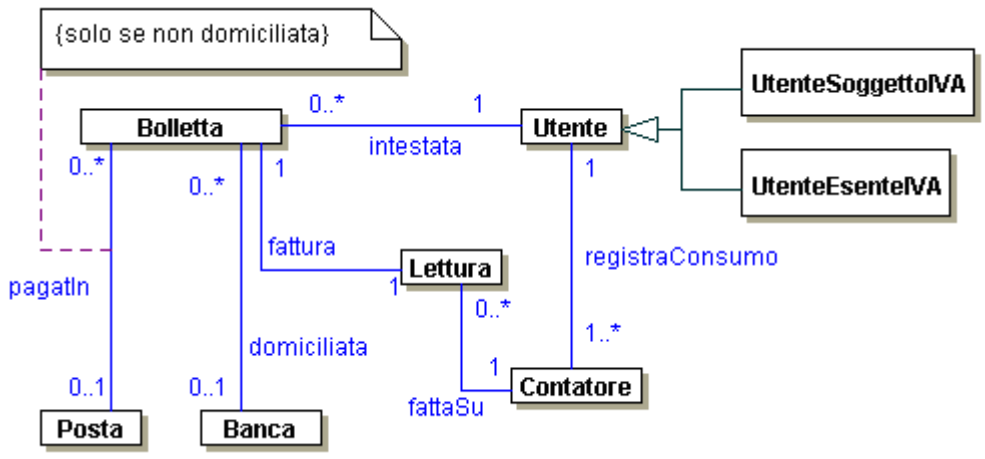
Casi di test: contatore veicoli in transito		
No	Input: sequenza segnali	Output: segnale di azzeramento del contatore
1	In, out	Si
2	In, in, out	No
3	In, in , out, out	Si
4	Out	No

Il quarto caso permette di verificare che i veicoli già in transito quando il sistema viene attivato siano trascurati.

CAP 4. Energia elettrica

Il sistema di fatturazione di una società di distribuzione di energia elettrica si basa sull'interazione tra diversi soggetti. Questo documento, destinato all'attenzione del responsabile sviluppo software, presenta un'analisi del problema basata su informazioni ottenute da interviste effettuate negli ultimi due mesi.
La società di distribuzione (Società) eroga energia elettrica a **utenti** collegati all'impianto di distribuzione mediante un allacciamento controllato da un dispositivo elettronico di misura (**contatore**) in grado di registrare il consumo di energia, di fornire a richiesta della centrale di bassa potenza il valore della **lettura** corrente dei consumi, di segnalare eventuali malfunzionamenti nell'impianto elettrico installato presso l'utenza. La centrale di bassa potenza periodicamente raccoglie le letture e le invia al sistema di fatturazione della Società che provvede al calcolo dell'importo relativo al consumo registrato dal contatore e all'emissione di una **bolletta** o fattura. Il calcolo dell'importo è effettuato considerando il regime fiscale applicato a ogni utente (esente IVA o soggetto a IVA) che prevede, quando applicabile, un'imposizione determinata da un'aliquota (aliquota IVA, attualmente pari al 20%). Inoltre, nel rispetto della normativa vigente, il calcolo dell'importo deve considerare una soglia (cosiddetta di consumo sociale) sotto la quale deve essere applicata una tariffa ridotta (costo unitario sociale). Per consumi che superano questa soglia, l'importo deve essere calcolato applicando la tariffa piena (costo unitario). Una volta emessa, la bolletta è spedita al domicilio dell'utente che può provvedere al pagamento presso un qualsiasi ufficio postale (Posta). Nel caso in cui l'utente abbia domiciliato le bollette presso un istituto bancario, la bolletta è inviata anche all'istituto (Banca) che provvede automaticamente al suo pagamento alla data di scadenza indicata. In questo caso, la bolletta inviata all'utente è stampata in modo che non possa essere pagata presso un ufficio postale. Un utente può essere intestatario di più **contratti**, ognuno associato a un contatore. Periodicamente, la Società provvede alla verifica dei pagamenti delle bollette, mediante un processo di accreditamento. I pagamenti effettuati presso la Banca e la Posta sono incrociati con le bollette emesse. In casi di morosità grave, la Società si riserva il diritto di sospendere l'erogazione dell'energia elettrica.

Domanda (Analisi del dominio). Dare un diagramma di classi relativo a **bolletta, utenti, lettura, contatore, Posta e Banca**.
Una possibile **risposta** è la seguente:

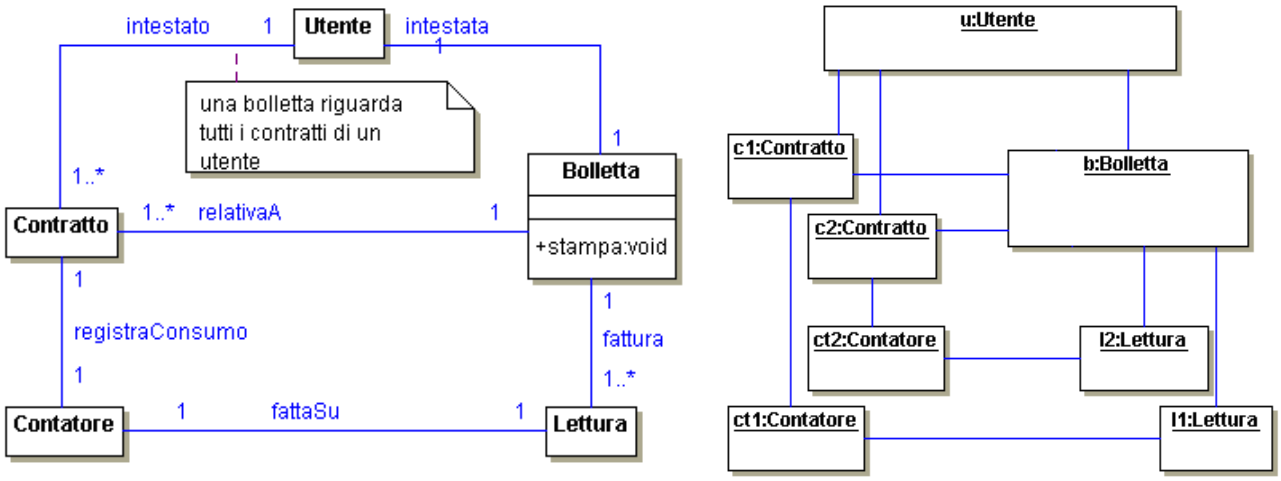


Il vincolo indica in realtà la mutua esclusione tra i legami pagataIn e domiciliata. Come è possibile esprimere in modo più esplicito tale vincolo?

Domanda Si estenda la descrizione del problema come segue: L'utente riceve una bolletta relativa ai consumi rilevati dai contatori associati ai suoi contratti.

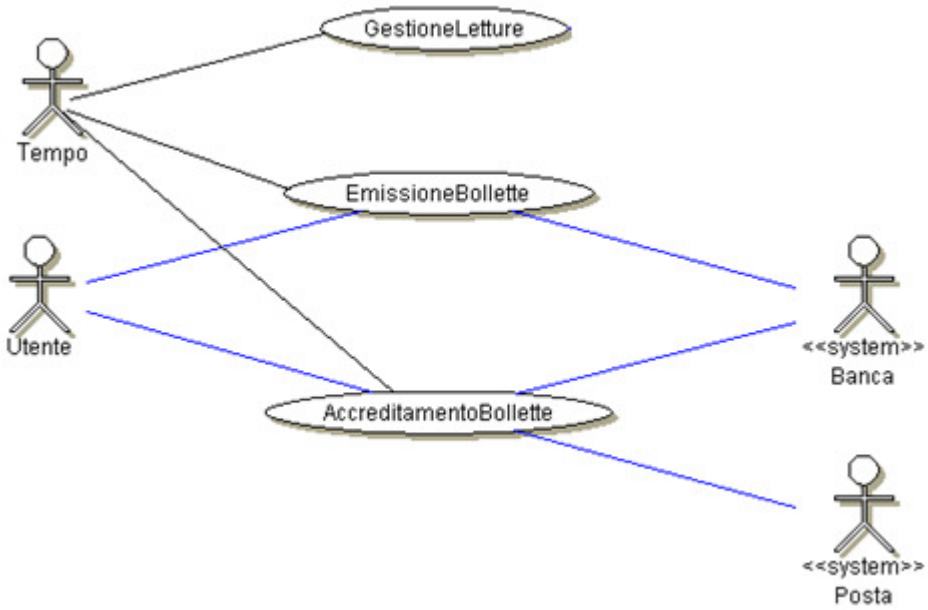
(a) dare un diagramma di classi, relativo a utente, bolletta, contratto, contatore, lettura;
(b) dare un diagramma degli oggetti relativo a un utente intestatario di due contratti.

Una possibile **soluzione**, relativa a una singola fatturazione, è la seguente:



Domanda (Analisi dei requisiti). Dare un diagramma dei casi d'uso che comprenda GestioneLetture, EmissioneBollette, Accreditamento Bollette. Fornire brevi descrizioni.

Risposta.

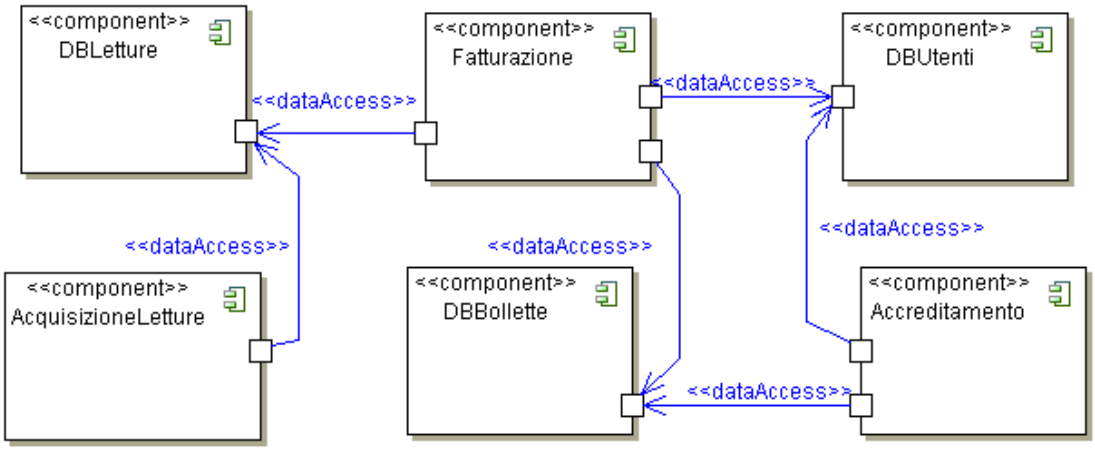


GestioneLetture: periodicamente, il sistema preleva dai Contatori le letture del consumo e le registra in una base di dati, DBLetture.

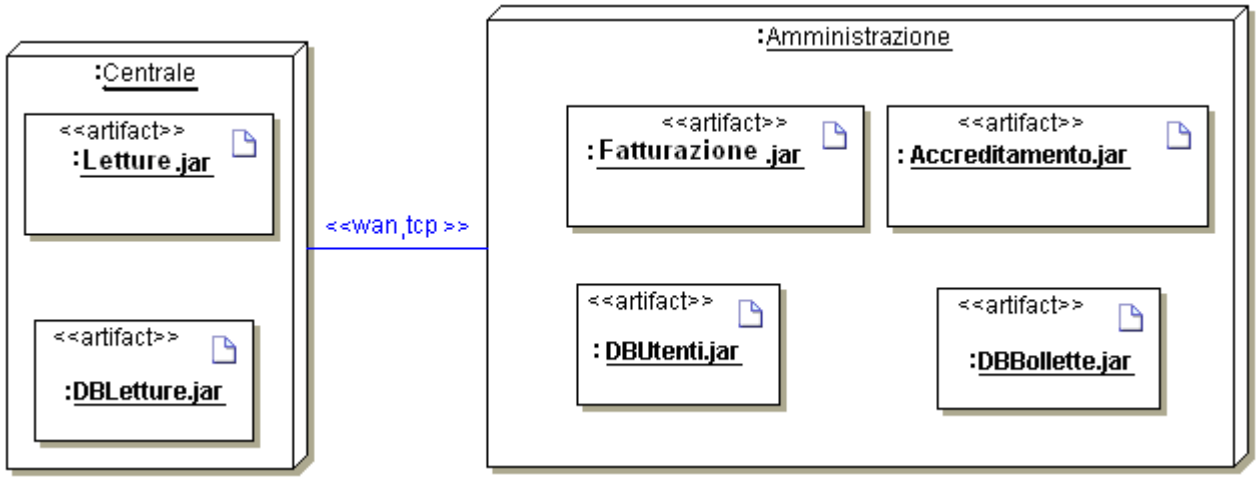
EmissioneBollette: periodicamente, il sistema stampa, per l'invio all'Utente, usando dati prelevati dal DBUtente, le bollette relative all'ultima lettura, e le registra nel DBBollette; nel caso di bollette domiciliate le invia pure elettronicamente alla Banca, per il pagamento.
AccreditamentoBollette: periodicamente, il sistema riceve elettronicamente dalla Banca e dalla Posta i pagamenti delle bollette, e li registra nel DBBollette; periodicamente, estraendo le informazioni dal DBBollette, stampa i solleciti per i clienti morosi, usando i dati del DBUtenti.

Il sistema software in dotazione della Società sfrutta tre basi di dati (*DBLetture*, *DBUtenti*, *DBBollette*) dislocate su due calcolatori distinti (uno presso la centrale e l'altro presso la sede amministrativa, collegati in rete geografica) per evitare che l'attività di *Acquisizione delle letture* interferisca con quelle di *Emissione* e *Accreditamento*.

Domanda. (Architettura) Dare una vista C&C che rappresenti il sistema software sopra descritto.
Una possibile **soluzione** è la seguente:

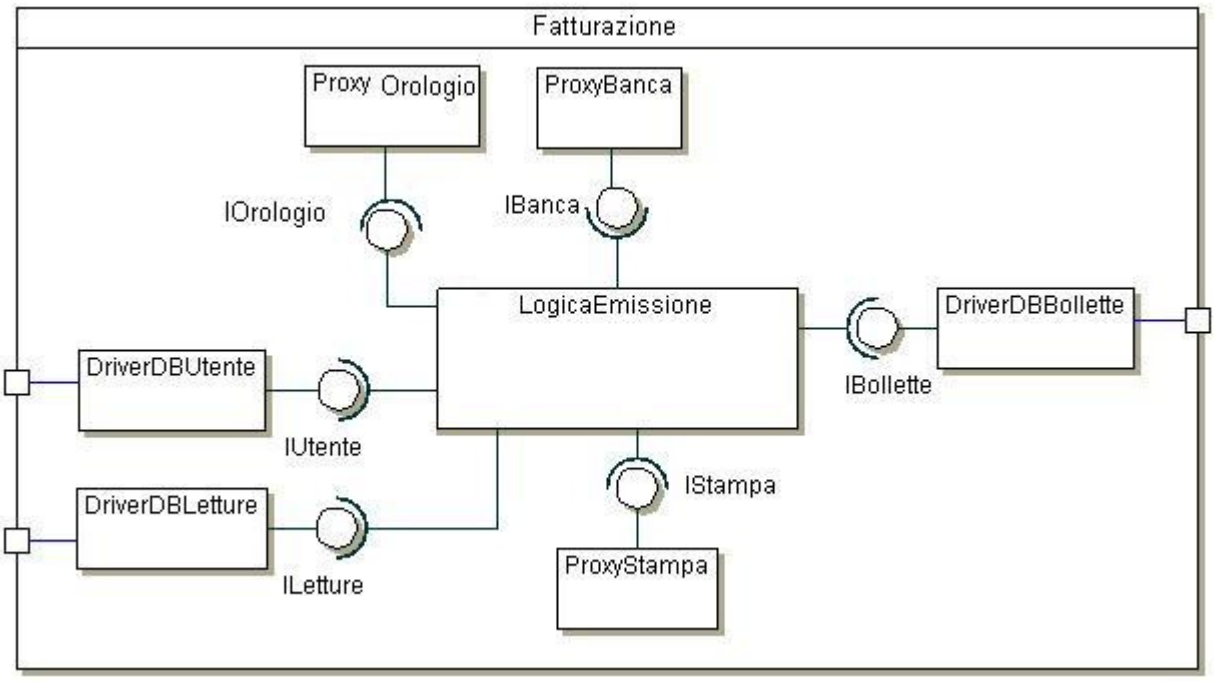


Domanda. (Architettura) Dare un diagramma di dislocazione che rappresenti il sistema software sopra descritto, assumendo che tutti gli artefatti siano di tipo jar, e che ciascuno manifesti la componente omonima. Dare un diagramma forma istanza.
Una possibile **soluzione** è la seguente:



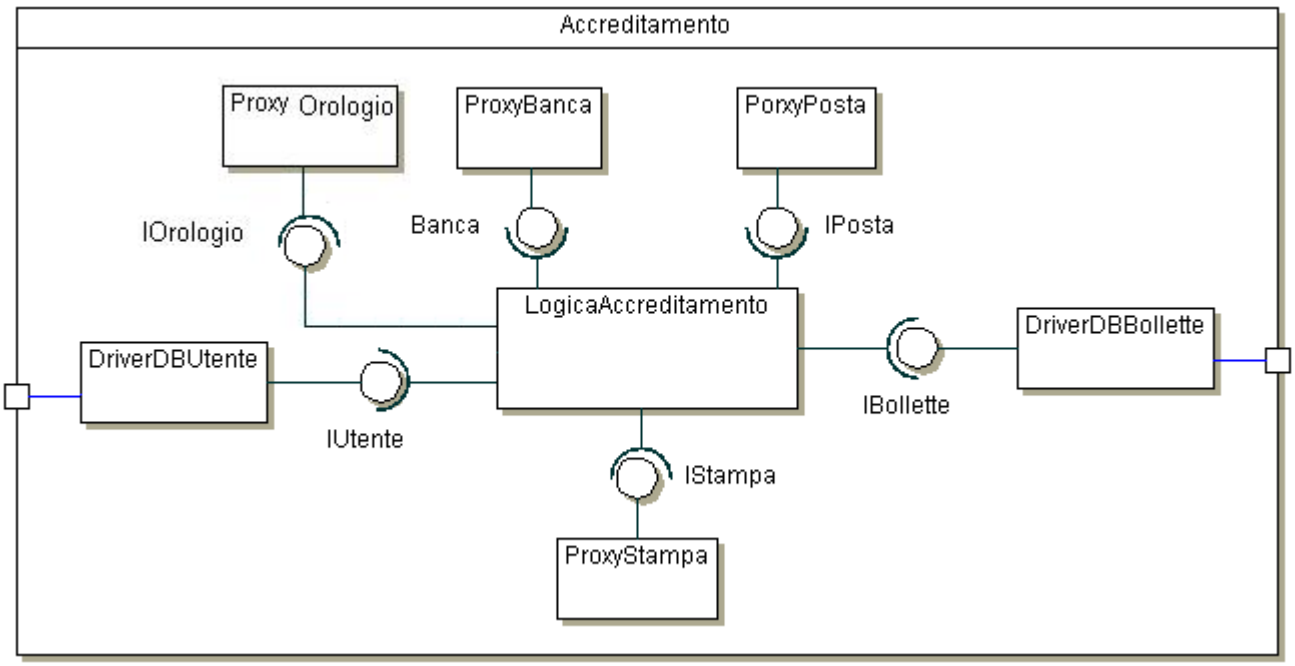
Domanda. (Progettazione di dettaglio) Dare il diagramma di struttura composito relativo relativo alla componente Fatturazione.

Una possibile **soluzione** è la seguente:



Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita della componente che realizza la componente Accreditamento.

Risposta.



In particolare, la parte *LogicaAccreditamento* ha due responsabilità:
1. in base alle informazioni provenienti dalla *Banca* e dalla *Posta*, marca le bollette pagate;
2. periodicamente, segnala agli utenti le bollette scadute e non pagate.

La componente che realizza l'attività di fatturazione è stata implementata in Java e contiene una classe con un metodo, chiamato *calcolaImporto*, che calcola l'importo relativo a una bolletta. Il codice del metodo è il seguente:

```
public double calcolaImporto() {
    double totale = 0.0;
    if (this.consumo <= SOGLIA_CONSUMO_SOCIALE) {
        totale = this.consumo * COSTO_UNITARIO_SOCIALE;
    } else {
        totale = this.consumo * COSTO_UNITARIO;
    }
    if (!this.esenteIVA) {
        totale = totale * (1 + ALIQUOTA_IVA);
    }
    return totale;
}
```

Domanda. (Verifica) Definire i valori di input di un insieme di casi di prova per la convalida del metodo *calcolaImporto*, applicando un criterio, a scelta, di progettazione delle prove a *scatola aperta*.

Si assuma che l'ambiente di prova preveda i seguenti valori per le costanti usate dal metodo:

```
private final double ALIQUOTA_IVA           = 0.20;
private final double COSTO_UNITARIO         = 0.20;
private final double COSTO_UNITARIO_SOCIALE = 0.10;
private final double SOGLIA_CONSUMO_SOCIALE = 50.0;
```

Risposta. Criterio scelto: valutazione di tutte le scelte. Una possibile selezione di valori di ingresso per i casi di prova è data dalla seguente tabella.

Input	
consumo	esenteIVA
40.0	true
60.0	false

Domanda. (Verifica) Descrivere informalmente un caso di prova (non banale) per la convalida del componente che realizza l'attività di *Emissione*.
Una possibile **soluzione** è la seguente:

Dati in ingresso:
- Il *DBUtenti* contiene un cliente intestatario di un solo contratto e un cliente intestatario di due contratti. Uno dei clienti ha richiesto la domiciliazione bancaria.
- Il *DBLetture* contiene tre letture relative ai contratti dei clienti presenti in *DBUtenti*.

Dati in uscita:

- Il *DBBollette* contiene due bollette intestate ai due clienti presenti in *DBUtenti*. La Banca ha ricevuto la bolletta intestata all'utente che ha la domiciliazione bancaria.
- Sono state stampate due bollette, intestate ai due utenti presenti in *DBUtenti*. La bolletta intestata all'utente che ha la domiciliazione bancaria è annullata.

CAP 5. Pub

Il proprietario di un pub ha deciso di introdurre un sistema per la raccolta e la gestione delle ordinazioni dei clienti, per migliorare il servizio diminuendo i tempi di attesa, e per semplificare le interazioni tra i dipendenti.
Il pub è situato nella zona centrale di una città di medie dimensioni. D'inverno il pub ha 25 tavoli nei locali interni, mentre d'estate può utilizzare la piazza antistante per altri 15 tavoli. I clienti sono serviti ai tavoli o al bancone. Il servizio ai tavoli prevede la raccolta delle ordinazioni da parte dei camerieri e la successiva consegna delle bevande e degli snack ordinati. Di norma l'ordinazione di un tavolo è raccolta una volta sola anche se, sempre più spesso, i clienti richiamano il cameriere per ordinare ulteriori bevande e snack. Il cameriere accetta solo ordinazioni che fanno riferimento a bevande e snack presenti sul menu giornaliero.
Raccolta l'ordinazione, il cameriere la consegna al bar e in cucina. Uno dei baristi prende un'ordinazione dalla lista di quelle da servire (rispettando l'ordine temporale di consegna), prepara le bevande indicate su un vassoio che appoggia sul bancone, a disposizione del cameriere. Lo stesso accade in cucina per gli snack, preparati da uno dei cuochi. Il cameriere preleva le bevande e gli snack e li consegna ai clienti.
Prima di lasciare il pub, il cliente passa dalla cassa e comunica al cassiere il numero del suo tavolo, ottenendo il conto che può pagare sia in contanti sia mediante carta di credito o bancomat. Il conto è calcolato utilizzando i prezzi indicati nel menu. Il cassiere rilascia uno scontrino fiscale, a prova dell'avvenuto pagamento. In seguito a un'ordinanza del sindaco, nel periodo estivo non è permesso servire bevande alcoliche dopo le 22, fino alla chiusura.
Per dare una veste accattivante al locale, il proprietario è disposto a dotare ogni cameriere di un palmare collegato senza fili al sistema di raccolta e gestione delle ordinazioni.

Domanda. (Analisi del dominio) Dare un modello per i principali concetti del dominio.
Risposta. I diagrammi delle classi nelle figure 1 e 2 presentano i principali concetti del dominio. Il diagramma delle attività di figura 3 descrive il comportamento del cameriere, quando serve ai tavoli. Non si fanno per ora ipotesi sull'ordine migliore con cui consegnare l'ordine ai preparatori. Si assume inoltre che le bevande siano la consumazione principale.

Data la centralità dell'ordine, e la sua propensione a passare attraverso stati successivi, in figura 4 viene dato un diagramma degli stati corrispondente. Questi stati saranno usati nelle narrative dei casi d'uso, per descrivere l'effetto di alcune interazioni. La figura 5 mostra come viene modificata l'attività ai tavoli, tenendo conto dell'uso dei palmari. Si noti come, in questo caso, l'interazione tra i vari attori sia resa possibile dal nuovo sistema di gestione.

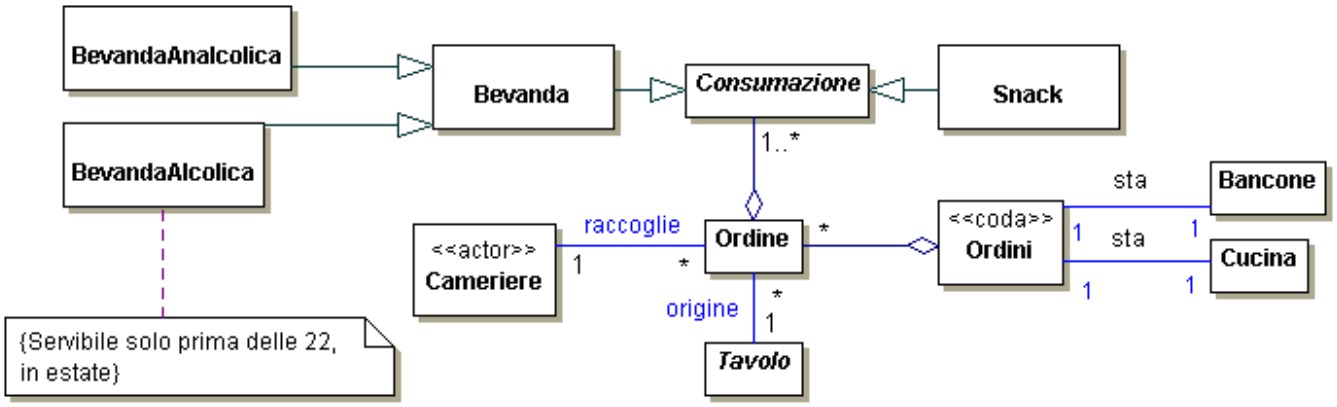


Figura 1. Dominio del pub: ordini

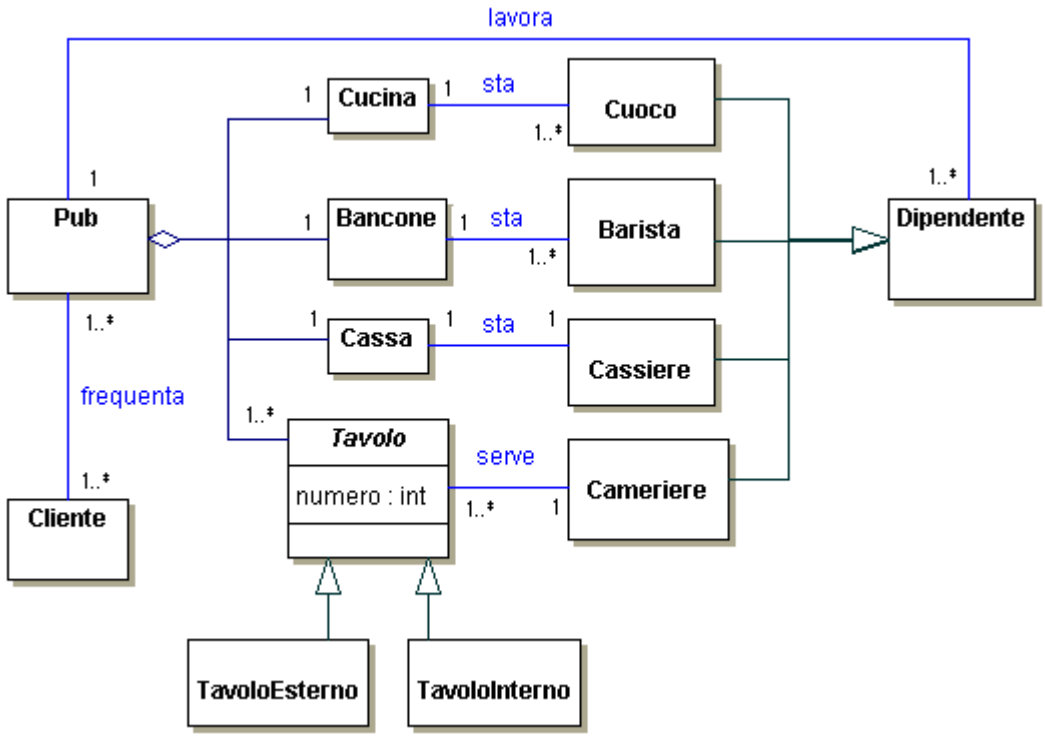


Figura 2. Dominio del pub: struttura

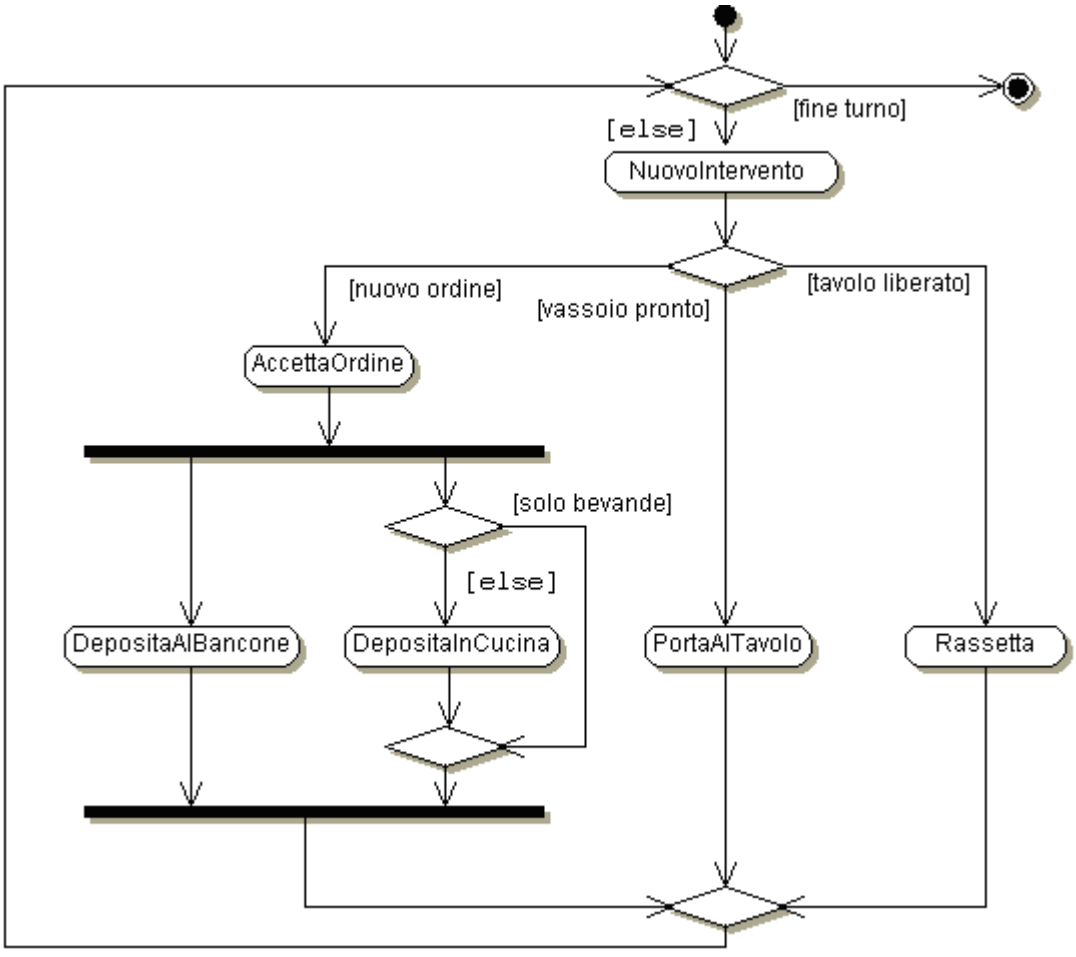


Figura 3. Attività di servizio al tavolo

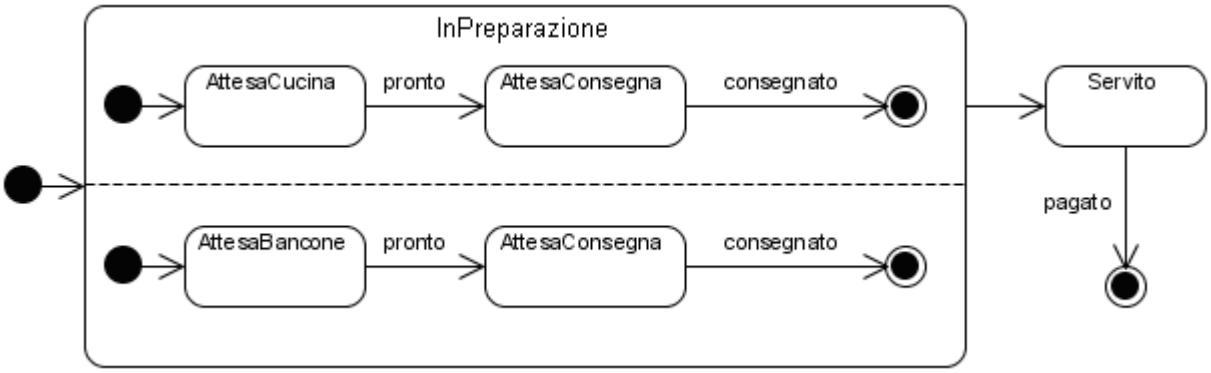


Figura 4. Stati di un ordine

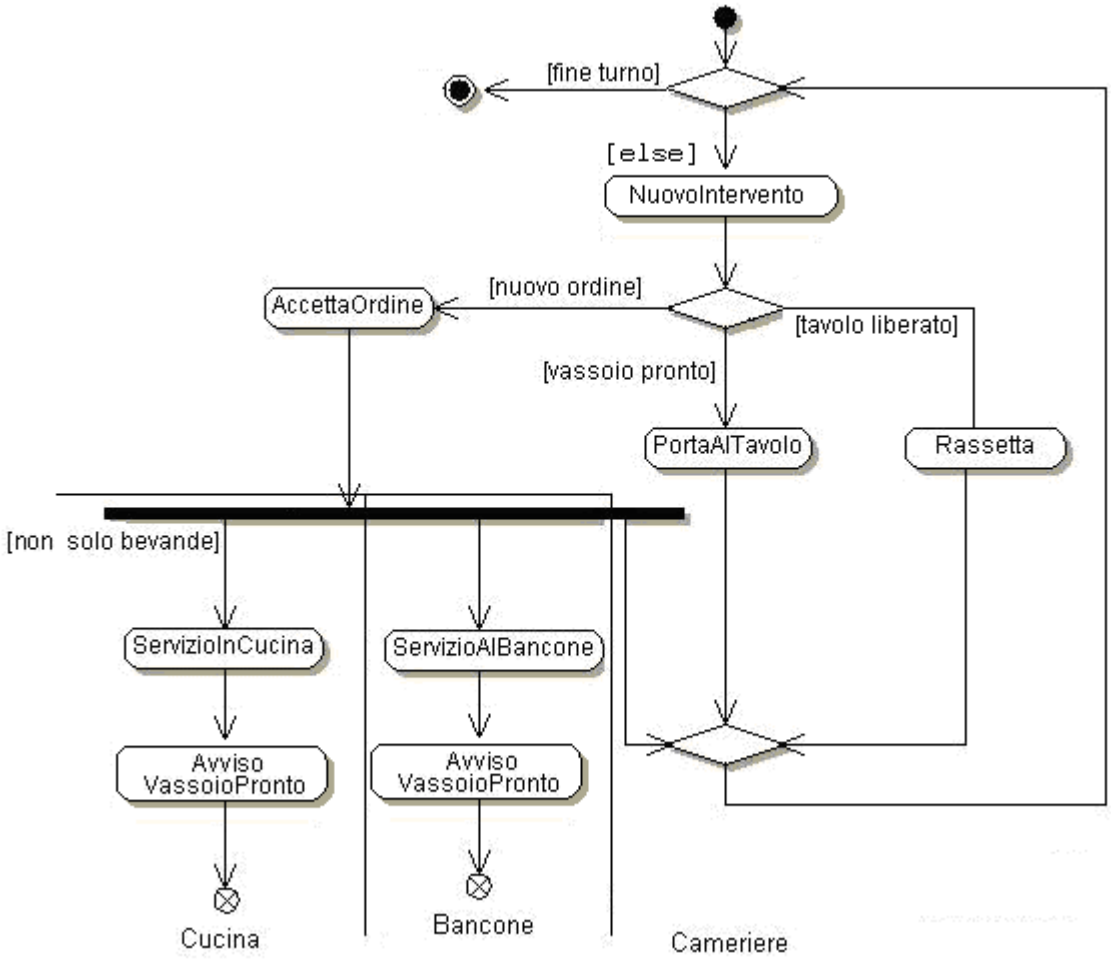


Figura 5. Servizio ai tavoli con palmari.

Si considerino le seguenti funzionalità del sistema per la raccolta e la gestione delle ordinazioni: il sistema riceve le ordinazioni dei clienti raccolte dal cameriere; il sistema comunica al cuoco (al barman) le ordinazioni ed è informato della loro preparazione; il sistema notifica al cameriere la preparazione di un'ordinazione ed è informato della sua consegna; il sistema riceve dal cassiere la richiesta del conto di un tavolo, emette lo scontrino e gestisce il relativo pagamento in contanti o mediante carta di credito o bancomat, interagendo in questo caso con la banca. La figura 6 presenta i casi d'uso che interessano al cliente.

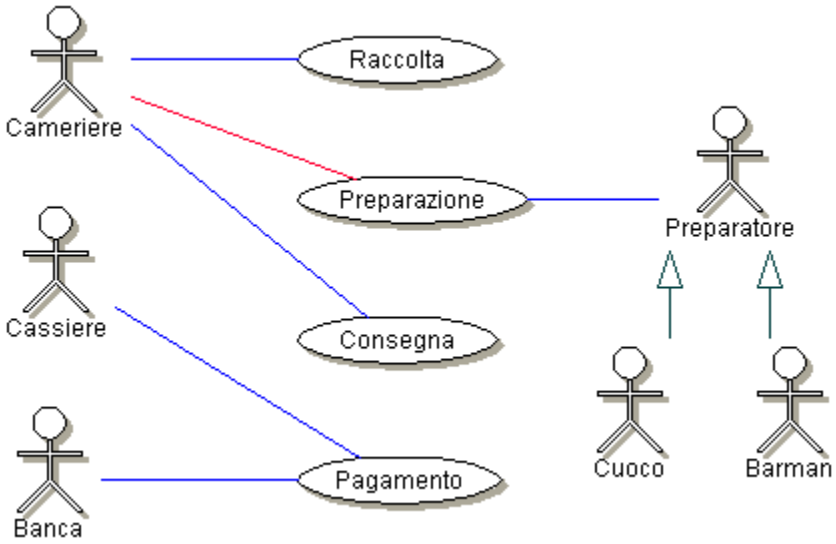


Figura 6. I casi d'uso che riguardano il cliente

Domanda. (Analisi dei requisiti) Definire la narrativa di due dei casi d'uso mostrati in figura, scegliendone uno tra Raccolta e Consegna, e un altro tra Preparazione e Pagamento.

Risposta. Diamo di seguito le narrazioni dei casi citati. Si noti come ci si focalizza sulle informazioni che vengono scambiate e non sui dettagli dell'interfaccia.

Caso d'uso: Raccolta.
Breve descrizione: Registrazione delle ordinazioni al tavolo.
Attore principale: Cameriere.
Attori secondari: NA
Precondizioni: NA
Postcondizioni: Ordine in attesa.
Sequenza principale degli eventi:
1. Il cameriere richiama un nuovo ordine e imposta il numero del tavolo.
2. Il sistema crea l'ordine.
3. while (ci sono consumazioni da registrare)
   3.1. il cameriere la introduce nell'ordine;
4. il cameriere chiude l'ordine;
5. il sistema aggiorna lo stato dell'ordine a 'InPreparazione'.

Caso d'uso: Preparazione
Breve descrizione: Registrazione della preparazione del vassoio.
Attore principale: Preparatore.
Attori secondari: Cameriere.
Precondizioni: Ordine in attesa.
Postcondizioni: Ordine in attesa consegna.
Sequenza principale degli eventi:
1. Il preparatore richiama il prossimo ordine da preparare;
2. while (ci sono consumazioni non pronte)
   2.1. una volta pronta, il prpearatore marca la consumazione come tale nell'ordine;
3. il sistema aggiorna lo stato dell'ordine a 'AttesaConsegna', e lo segnala al Cameriere.

Sequenza alternativa degli eventi: ArticoloMancante.

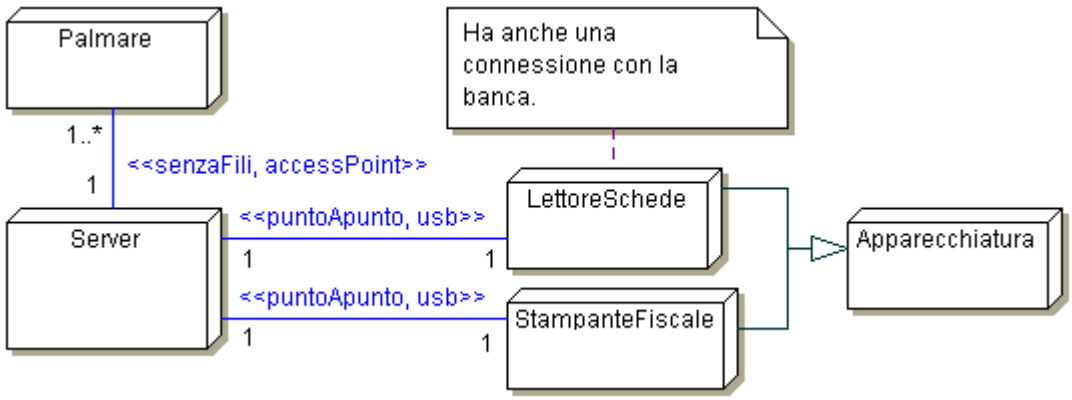
Caso d'uso: Consegna
Breve descrizione: Registrazione della consegna delle ordinazioni.
Attore principale: Cameriere.
Attori secondari: NA
Precondizioni: Ordine in attesa consegna.
Postcondizioni: Ordine conseganto.
Sequenza principale degli eventi:
1. Il cameriere richiama l'ordine relativo al tavolo appena servito e lo marca come consegnato;
2. il sistema aggiorna lo stato dell'ordine a 'Servito'.

Caso d'uso: Pagamento
Breve descrizione: Registrazione del saldo del conto.
Attore principale: Cassiere.
Attore secondario: Banca.
Sequenza principale degli eventi:
1. Il cassiere richiama l'ordine relativo al tavolo comunicatogli dal cliente;
2. il sistema calcola l'importo da pagare;
3. il cassiere comunica al sistema la forma di pagamento.
4. Se il pagamento è mediante carta di credito o bancomat,
   4.1. il cassiere inserisce i dati relativi (passando la carta nell'apposito lettore)
   4.2. il sistema contatta la banca e ottiene l'autorizzazione al pagamento.
5. Il cassiere comunica al sistema l'avvenuto pagamento,
6. il sistema emette lo scontrino.

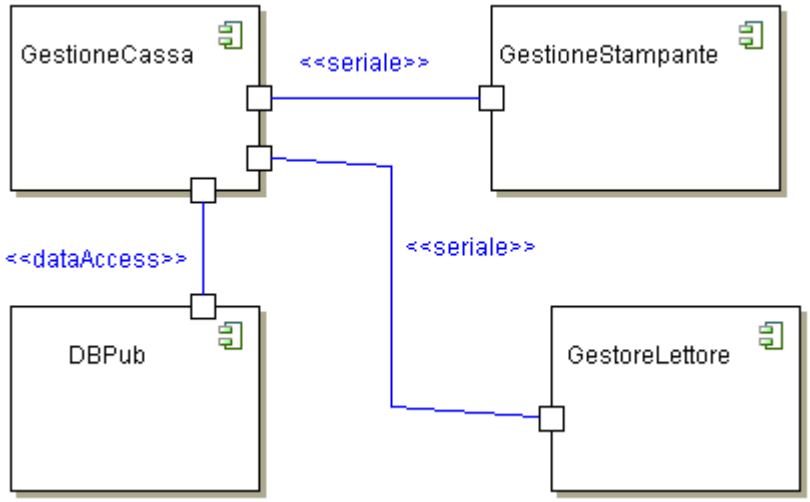
Sequenza alternativa degli eventi:
1. La banca non concede l'autorizzazione al pagamento.
2. Il cliente non è in grado di pagare.

Note.
In relazione al caso d'uso 'Raccolta'. Nella descrizione non si considera la ripetizione di ordini da parte degli stessi clienti a un tavolo. La situazione è risolta al 'Pagamento'.
In relazione al caso d'uso 'Preparazione'. Il sistema potrebbe impedire al cameriere di mettere in ordine gli articoli mancanti, rendendo in linea teorica inutile considerare la situazione eccezionale. Dato che in realtà può essere difficile fare un conto preciso di alcuni elementi minuti, o possono esserci dei disallineamenti tra inventario e disponibilità reali, conviene prevedere anche questa possibilità. Potrebbe essere utile anche un auto arco sullo stato 'AttesaBancone' nel diagramma di Figura 4, per registrare questa transizione.
In relazione al caso d'uso 'Pagamento'. La gestione degli ordini sospesi deve essere descritta in un caso d'uso apposito.

L'hardware previsto per il sistema è descritto nel seguente diagramma:



La vista architettonica in stile C&C relativa al caso d'uso 'Pagamento' è data di seguito:

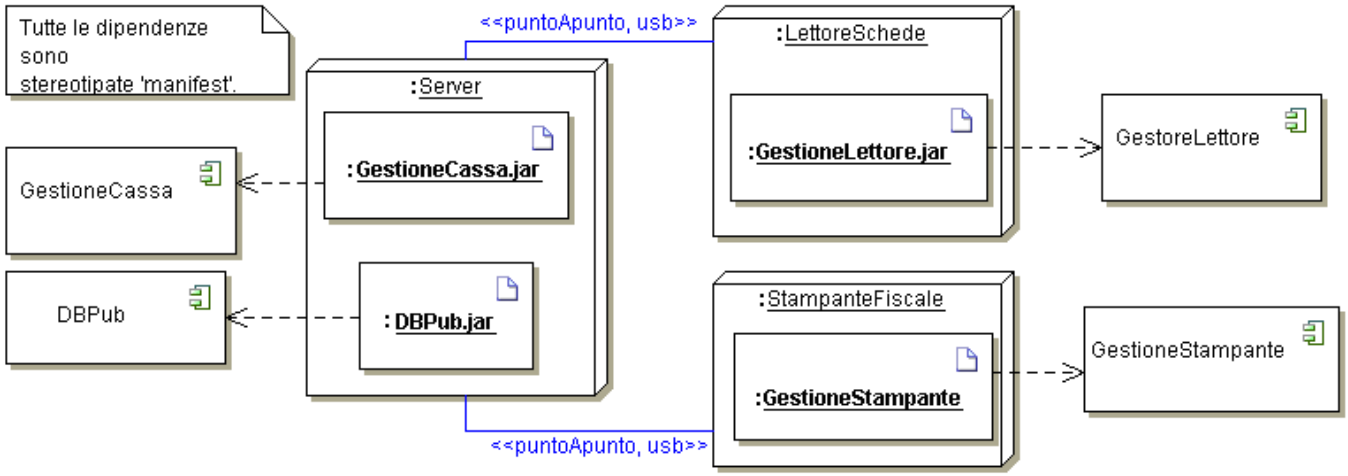


Domanda. (Realizzazione dei casi d'uso) Associare a ciascuna componente la parte di comportamento pertinente della narrazione del caso d'uso data sopra.

Risposta.

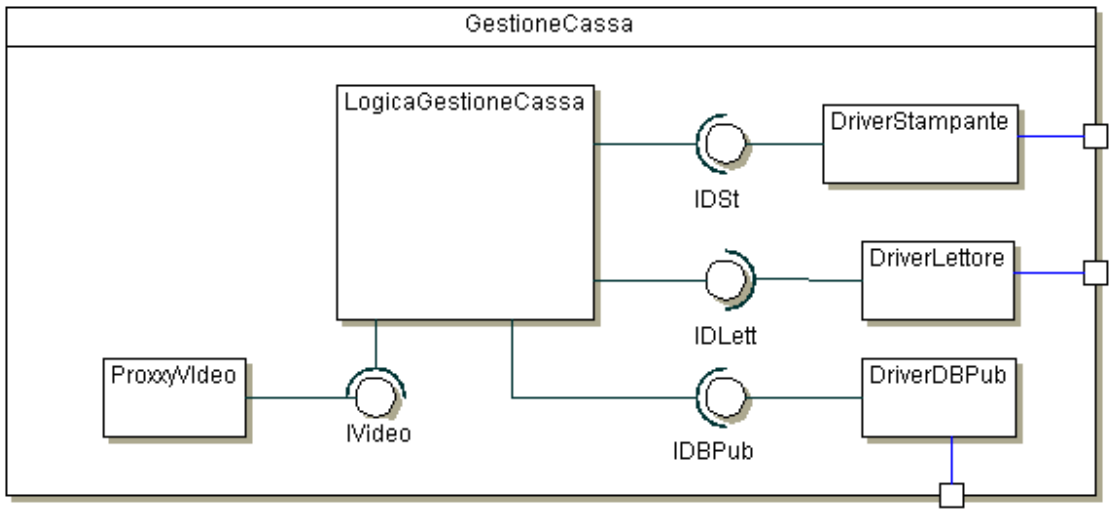
Componente	Responsabilità
GestioneCassa	Richiama l'ordine relativo al tavolo inserito dal cassiere e accetta la forma di pagamento; se il pagamento è mediante carta di credito o bancomat, accetta l'autorizzazione al pagamento; altrimenti accetta l'indicazione di avvenuto pagamento. In entrambi i casi, richiede l'emissione dello scontrino. *Situazioni Eccezionali.* La banca non concede l'autorizzazione al pagamento: informa il cassiere, accetta una nuova forma di pagamento e procede come nel caso normale. Il cliente non è in grado di pagare: pone l'ordine in uno stato di 'Sospeso'.
Gestione stampante	Stampa lo scontrino.
DBPub	Recupera l'ordine e calcola il totale.
GestoreLettore	Legge la scheda, contatta la banca, ottiene la risposta e la comunica alla componente GestioneCassa.

Domanda. (Architettura) Assumendo che il sistema sia sviluppato in Java, e che tutti gli eseguibili siano di tipo jar, dare una vista di dislocazione delle componenti sull'hardware, che mostri anche le dipendenze degli artefatti dalle relative componenti.
Risposta. In forma istanza:



Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita della componente GestioneCassa.

Risposta.



CAP 6. Supermercato

Spett. Ditta,

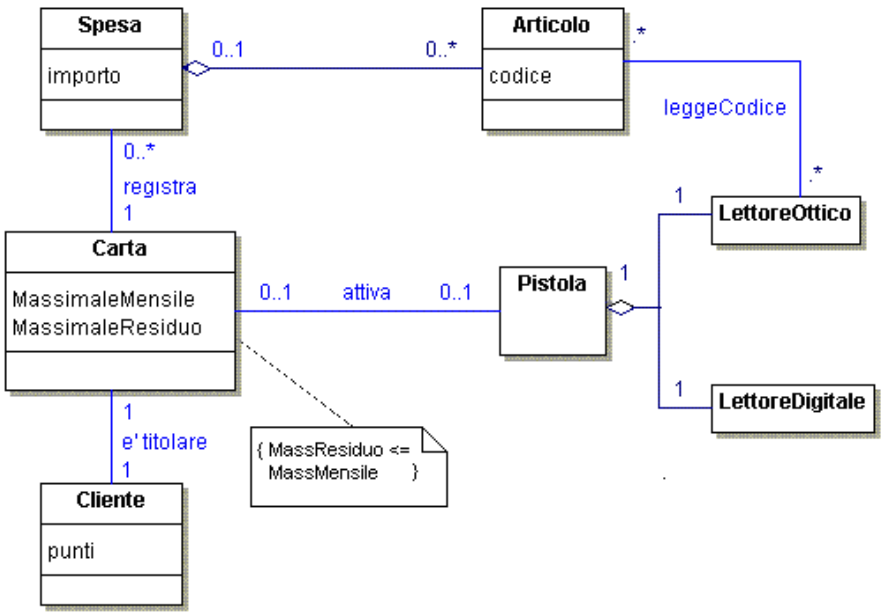
In seguito all'incontro relativo alle modalità di funzionamento del nuovo sistema di cassa, vi trasmettiamo le informazioni richieste dal vostro analista.
Un recente sondaggio ha evidenziato il problema dell'attesa in coda per il pagamento della spesa effettuata presso i nostri supermercati. Sempre nel sondaggio, l'utenza ha recepito positivamente la possibilità di utilizzare strumenti per la registrazione della **spesa** e il pagamento mediante **carte** fedeltà, tra l'altro già in uso presso la nostra catena di distribuzione. La proposta su cui stiamo lavorando prevede che il **cliente** utilizzi un dispositivo (**pistola**) dotato di un **lettore ottico** per leggere il **codice** a barre degli **articoli** aggiunti e tolti dal suo carrello. Prima di effettuare la spesa, il cliente attiva una pistola disponibile nell'isola di pagamento inserendo la sua carta nel **lettore digitale** della pistola e, successivamente, autenticandosi mediante un PIN. Se la fase di autenticazione termina con successo il cliente preleva la pistola. Sullo schermo a cristalli liquidi della base, nel frattempo, è comparso il **massimale residuo** che può spendere nel mese corrente. Al momento dell'attivazione della carta, infatti, il cliente concorda con l'ufficio vendite il **massimale mensile**, ovvero l'importo che può spendere in un mese nei nostri supermercati usando la carta. Al termine della spesa, il cliente torna all'isola di pagamento e ripone la pistola in una base libera. Così facendo, la pistola si collega al sistema centrale, invia l'elenco degli articoli della spesa e riceve l'**importo** della spesa da pagare. Il cliente preme il pulsante di accettazione del pagamento, presente nella base, ritira prima la carta e poi lo scontrino relativo alla spesa. Prima di consegnare la carta al cliente, si aggiorna il massimale residuo, il sistema centrale aggiorna i **punti** fedeltà del cliente e invia un ordine di pagamento alla banca indicata dal cliente. Se l'importo è superiore al massimale residuo, sullo schermo compare un messaggio che invita il cliente a prendere la pistola e a togliere alcuni articoli dalla spesa. Analogamente, il cliente può decidere, dopo aver visto l'importo della spesa, di prendere la pistola e modificare la spesa. Un caso particolare si presenta quando il carrello è vuoto e l'importo è pari a zero: il cliente riconsegna la pistola e ritira la carta senza che sia emesso lo scontrino.

In attesa di un vostro riscontro.

Cordiali saluti, Ing. Salvo La cassa

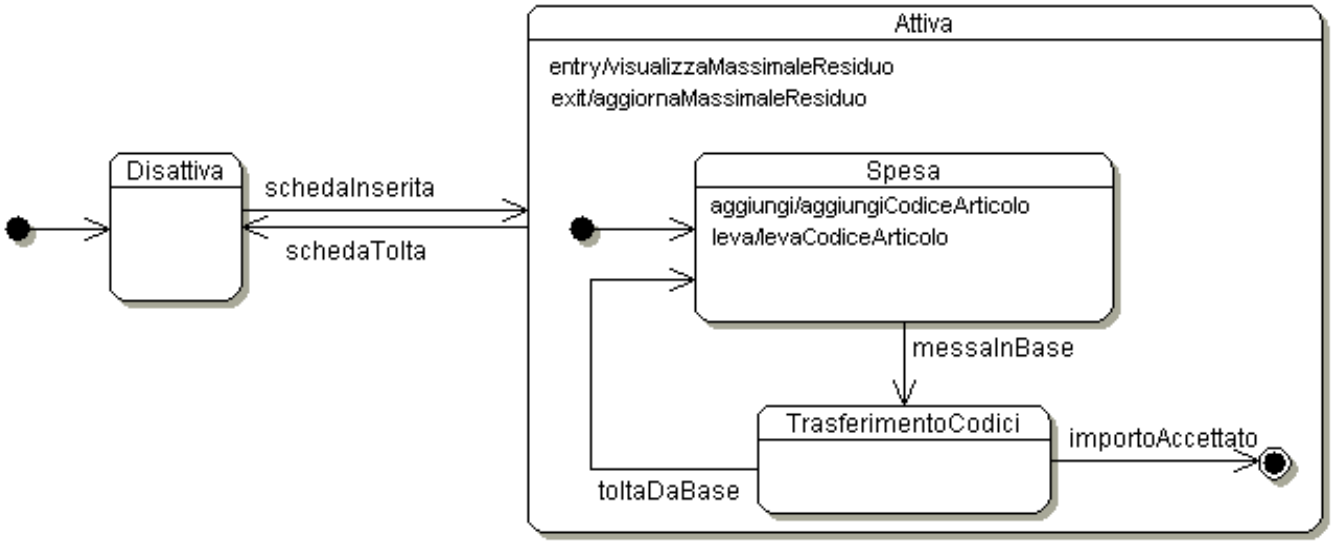
Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi o attributi i termini che nell'enunciato compaiono in grassetto.

Risposta. Un possibile diagramma delle classi è il seguente:



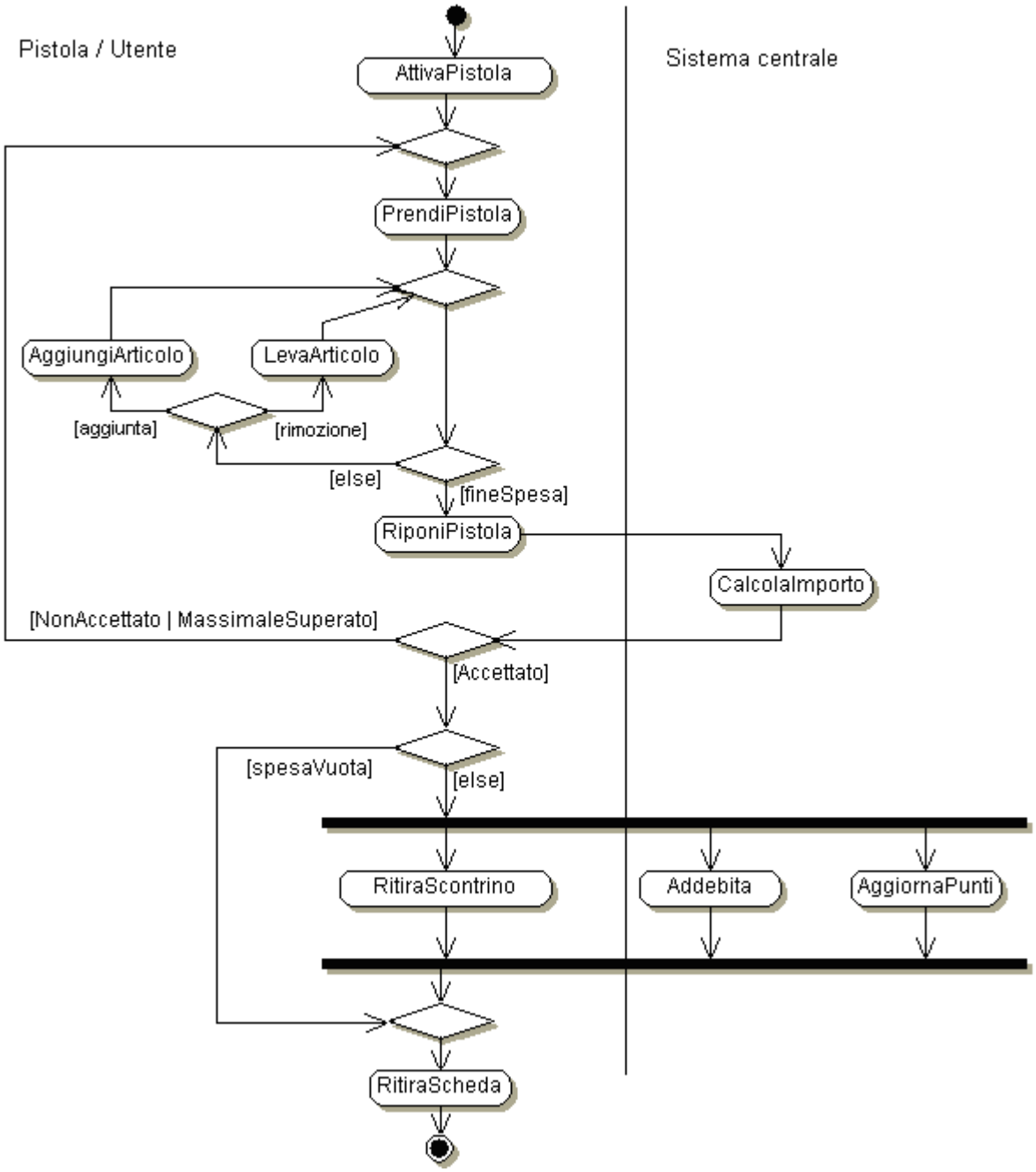
Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva il comportamento del sistema formato dalla pistola e dalla base, trascurando la fase di autenticazione.

Risposta. Un possibile diagramma di macchina a stati è il seguente:

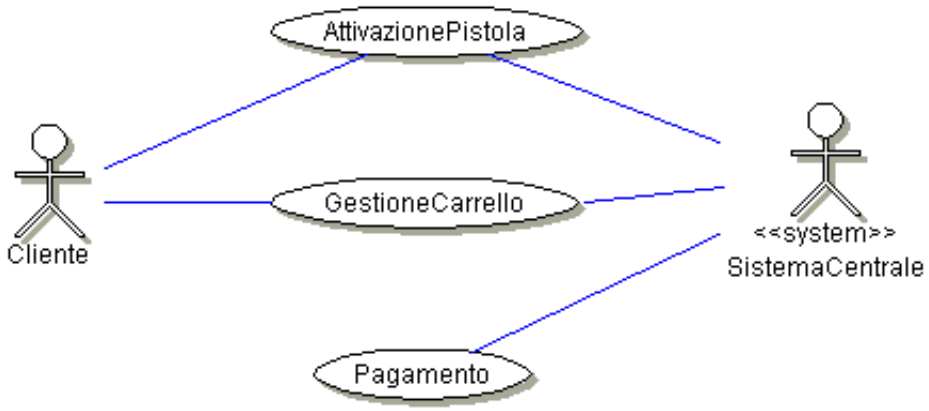


Domanda. (Analisi del dominio) Dare un diagramma di attività della procedura descritta nell'enunciato, trascurando la fase di autenticazione. Il diagramma deve mostrare due partizioni: una per il sistema centrale, una per la pistola e l'utente.

Risposta. Un possibile diagramma di attività è il seguente:



L'analisi dei requisiti ha portato al seguente diagramma dei casi d'uso:



con le seguenti descrizioni:

AttivazionePistola

Breve descrizione	Il cliente attiva la pistola per leggere il massimale residuo ed effettuare la spesa.
Attore principale	Cliente
Attore secondario	Sistema centrale.
PreCondizioni	Pistola disattivata nella base.
PostCondizioni	Pistola attivata nella base.

GestioneCarrello

Breve descrizione	Il cliente ripetutamente aggiunge/leva articoli dalla spesa, tramite la pistola.
Attore prncipale	Cliente.
Attore secondario	Nessuno.
PreCondizioni	Pistola attivata nella base.
PostCondizioni	Pistola attivata nella base, importo della spesa accettato.

Pagamento

Breve descrizione	Il sistema stampa lo scontrino; il sistema centrale registra le informazioni contabili sulla spesa.
Attore principale	Sistema centrale.
Attore secondario	Nessuno
PreCondizioni	Pistola attivata nella base, importo della spesa accettato.
PostCondizioni	Pistola disattivata nella base.

Domanda. (Analisi dei requisiti) Dare una narrativa del caso d'uso *GestioneCarrello*.
Risposta. Una possibile narrativa del caso d'uso *GestioneCarrello* è la seguente:

GestioneCarrello
Sequenza principale degli eventi

1. Il cliente preleva la pistola dalla base.
2. Ripetutamente, il cliente legge il codice di un articolo che mette o toglie dal carrello, finché non è soddisfatto della spesa.
3. Il cliente inserisce la pistola nella base; il sistema centrale calcola l'importo della spesa, che la pistola visualizza sullo schermo.
4. Il cliente accetta il pagamento.

Situazioni eccezionali

1. L'importo della spesa supera il massimale residuo.
2. Il cliente ha un ripensamento, prima di accettare la spesa.

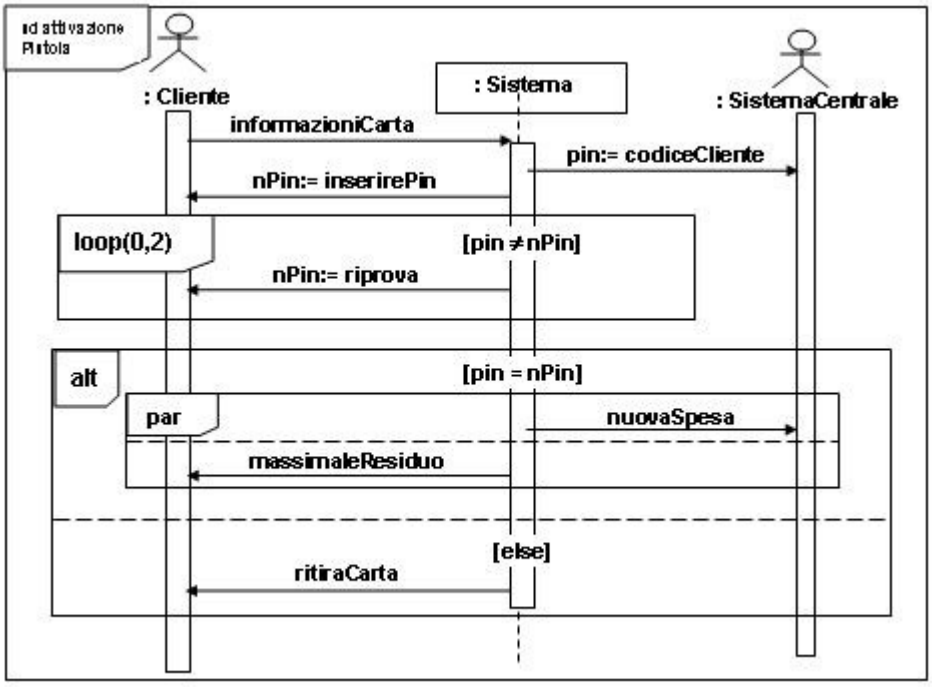
L'analisi dei requisiti ha portato alla definizione del seguente caso d'uso:

AttivazionePistola
Breve descrizione Il cliente attiva la pistola per leggere il massimale residuo ed effettuare la spesa.
Attore principale Cliente
Attore secondario Sistema centrale.
PreCondizioni Pistola disattivata nella base.
PostCondizioni Pistola attivata nella base.
Sequenza principale degli eventi

1. Il cliente inserisce la carta nella pistola.
2. Il sistema legge il codice del cliente e richiede il PIN al sistema centrale.
3. while (il cliente non ha digitato il PIN giusto e ha fatto meno di tre tentativi)
   3.1. il cliente digita il PIN;
   3.2. se il PIN è sbagliato
      3.2.1. il sistema chiede al cliente di riprovare
4. Se ci son stati tre tentativi sbagliati,
   4.1. il sistema chiede al cliente di ritirare la carta.
5. Altrimenti, il PIN è giusto e il sistema
   5.1. comunica il massimale residuo al cliente e
   5.2. segnala l'inizio di una nuova spesa al sistema centrale.

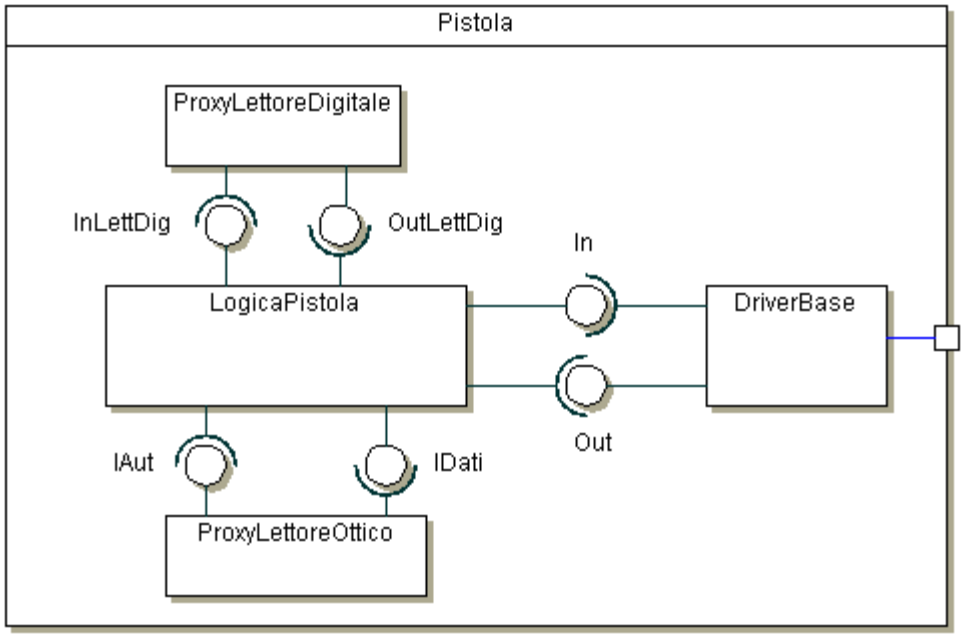
Domanda. (Realizzazione dei casi d'uso) Si dia un diagramma di sequenza che realizzi il caso d'uso AttivazionePistola.

Risposta. Un possibile diagramma di sequenza è il seguente:



La progettazione architettonica del sistema si esaurisce nella seguente vista comportamentale:

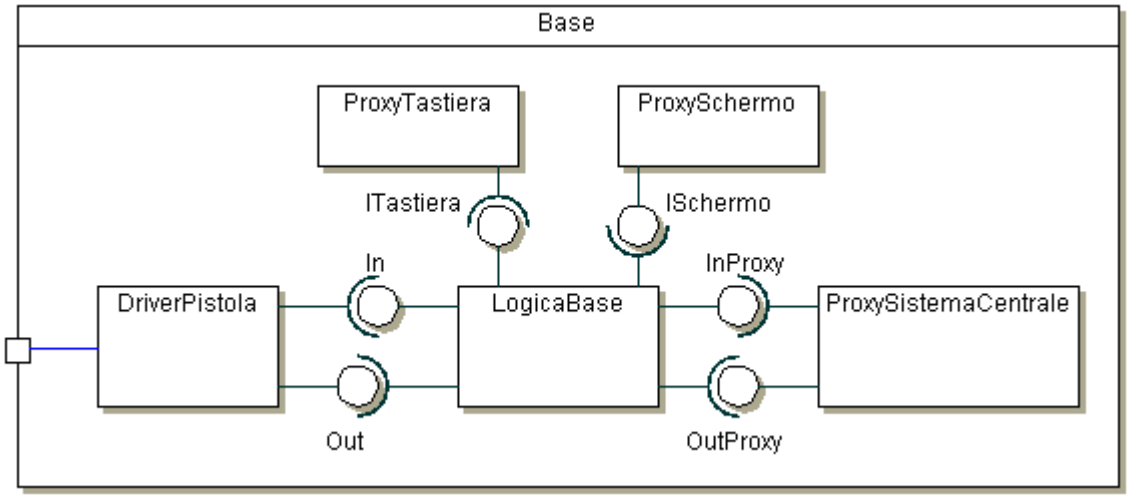
Il seguente diagramma di struttura composita mostra il progetto del componente *Pistola*, mettendo in evidenza le relazioni tra la parte di logica e i driver delle apparecchiature che esso controlla: il lettore digitale della scheda, il lettore ottico di codice a barre e la connessione verso il componente *Base*.



In particolare, il *ProxyLettoreOttico* controlla sia il lettore ottico di codice a barre, sia i pulsanti di comando (+ per aggiungere articoli, – per toglierli) che sono abilitati, al momento dell'attivazione della pistola, tramite l'interfaccia *IAut*. Attraverso l'interfaccia *IDati* il lettore ottico comunica sia il codice letto, sia il tipo di operazione selezionata dal cliente tramite i pulsanti di comando.

Domanda. (Progettazione di dettaglio) Dare un analogo diagramma di struttura composita per il componente *Base*, ricordando che la sua logica deve controllare sia la tastiera per l'immissione del PIN, sia lo schermo per visualizzare i massimali, e comunicare con il sistema centrale.

Risposta. Un possibile diagramma di struttura composita è il seguente:

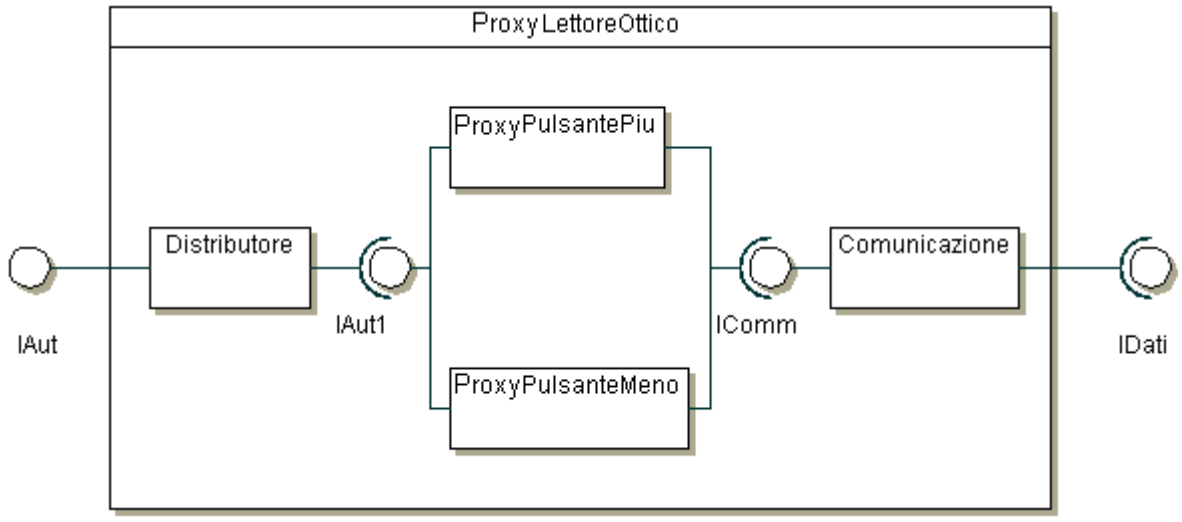


Domanda. (Progettazione di dettaglio) Ricordando che *ProxyLettoreOttico* controlla sia il lettore di codice a barre, sia i pulsanti di comando e che attraverso l'interfaccia *IDati* il lettore ottico comunica sia il codice letto, sia il tipo di operazione, la progettazione di dettaglio del *ProxyLettoreOttico* ha individuato un insieme di parti, con le seguenti responsabilità:

Iaut è l'interfaccia per bloccare e sbloccare i pulsanti.
Distributore ha il compito di distribuire ad entrambi i driver il blocco o lo sblocco dei pulsanti.
ProxyPulsantePiu richiede di aggiungere un articolo, emettendo il codice dell'articolo e il codice dell'operazione "aggiungi" in formato di basso livello, secondo l'interfaccia *IComm*.
ProxyPulsanteMeno, come sopra, ma con il codice "leva" per togliere un articolo.
Comunicazione riceve la richiesta di operazione e la converte dal formato di basso livello al formato richiesto dalla logica con l'interfaccia *IDati*, ed effettua alcuni controlli d'integrità.

Dare un diagramma di struttura composita corrispondente.

Risposta.



Si consideri la seguente classe Java:

```
import java.util.Date;

public class Massimali {
   public int massMese;
   public int massResiduo;
   public Date dataAcquisto;

   public void update(int importo) {
      massResiduo = massResiduo - importo;
      dataAcquisto = new Date();
      }
   public boolean checkAndSet(int importo) {
      if (importo <= 0) return false;                          //1
      Date oggi = new Date();
      if ((dataAcquisto.getYear() < oggi.getYear()) ||         //2
          (dataAcquisto.getMonth() < oggi.getMonth()))         //3
                massResiduo = massMese;
     if (importo <= massResiduo) return true;                  //4
           else return false;
      }

   public Massimali(int mm, int mr, Date d) {
     massMese = mm;
     massResiduo = mr;
     dataAcquisto = d;
     }
   }
```

Domanda. (Verifica) Dare i valori di input dei casi di prova per il metodo `checkAndSet`, assumendo che il massimale residuo sia 200, il massimale mensile sia 1000 e che la data odierna sia 15 dicembre 2005. I casi di prova devono garantire la copertura totale delle condizioni. Si ricorda che un comando condizionale ha un'unica decisione ma può avere più condizioni.

Risposta. Le prime due colonne della seguente tabella mostrano i valori di input dei casi di prova richiesti. Le altre quattro mostrano i corrispondenti valori delle condizioni, se valutate. Si vede che per tutte le condizioni si ha sia il valore true sia false. Si noti anche che, per le condizioni 1 e 4, si considerano esplicitamente i casi di eguaglianza.

importo	**acquisto**	1	2	3	4
-1000	14.12.05	true			
0	14.12.05	true			
190	14.12.05	false	false	false	true
200	14.12.05	false	false	false	true
210	14.12.05	false	false	false	false
190	15.12.04	false	true	false	true
190	15.11.05	false	false	true	true

In assenza di oracolo, non si possono completare i casi di test con i risultati attesi.

CAP 7. Teletaxi

Spett. Ditta,

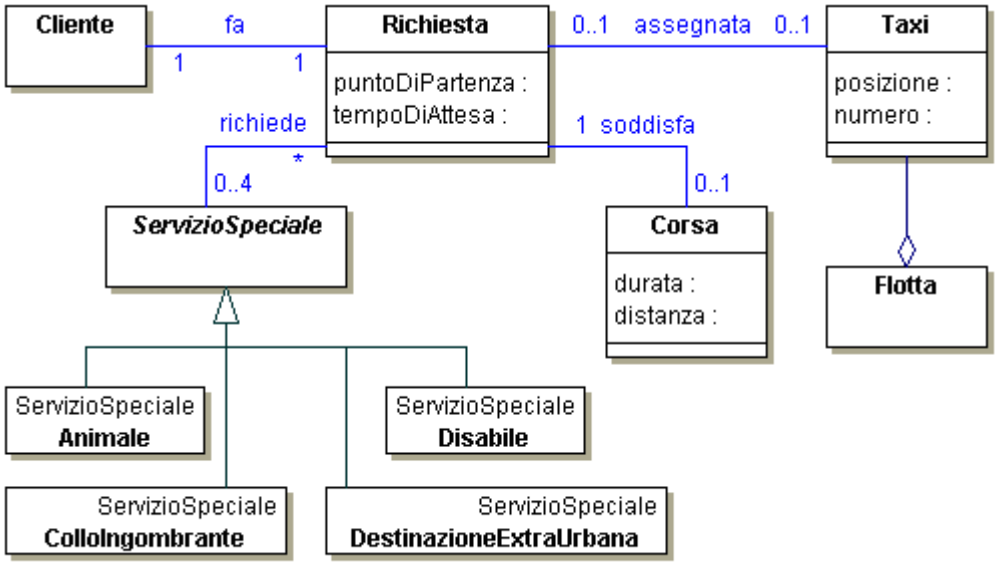
Dopo l'incontro relativo al sistema *TeleTaxi* vi inviamo le informazioni richieste dal vostro analista.

L'evoluzione tecnologica nel settore della geolocalizzazione ha permesso la realizzazione di *TeleTaxi*, un sistema che garantisce una diminuzione dei tempi di risposta alle richieste dei **clienti** e una migliore allocazione delle **corse** effettuate dai **taxi** della **flotta**. Per chiedere un taxi, il cliente può interagire con *TeleTaxi* in varie modalità (operatore tramite telefono, Internet, SMS), fornendo il **punto di partenza** della corsa e ottenendo in tempi rapidi e certi il **numero** di un taxi libero e il **tempo stimato d'attesa**. Il sistema consente al cliente di richiedere **servizi speciali** (**colli ingombranti**, **animali**, **disabili**, **destinazione extraurbana**) che saranno gestiti con taxi opportunamente attrezzati. La scelta del taxi da inviare al punto di partenza avviene in base a strategie basate sui taxi disponibili, sulle condizioni del traffico e sull'orario della chiamata. Dopo aver ricevuto la **richiesta**, il sistema la invia ai taxi potenzialmente interessati i cui autisti, tramite l'apposita consolle, potranno accettarla. Tra tutte le risposte ricevute, il sistema seleziona quella migliore assegnando la richiesta al taxi prescelto e cancellandola su tutti gli altri. La dimensione della flotta garantisce, nella maggior parte dei casi, una risposta in tempi ragionevoli. Se il tempo stimato d'attesa eccede le sue aspettative, il cliente può rinunciare al servizio. Anche gli autisti trarranno vantaggi dall'uso di *TeleTaxi*. La consolle visualizza le richieste (punto di partenza e servizi speciali) e, mediante un tasto dedicato (*AccettaRichiesta*), consente all'autista di accettarla. L'arrivo di una nuova richiesta è segnalato da un suono emesso da un cicalino. Giunto al punto di partenza, l'autista chiede al cliente la destinazione e accetta la corsa premendo un tasto dedicato (*Corsa*). Al termine della corsa, l'autista preme di nuovo il tasto *Corsa* per comunicare a *TeleTaxi* la sua disponibilità ad accettare altre richieste. In casi eccezionali la richiesta può essere cancellata premendo il tasto *CancellaRichiesta*. L'importo della corsa è calcolato dal tassametro, in base alle tariffe in vigore, alla **durata** della corsa e alla **distanza** percorsa. *TeleTaxi* mantiene aggiornato lo stato (fuori servizio, libero, assegnato, occupato) e la **posizione** dei taxi della flotta, con una precisione temporale di un minuto o spaziale di 200 metri, inviata via radio dopo essere stata rilevata da un dispositivo GPS, installato su ogni taxi. Sempre mediante un tasto dedicato (*Servizio*), l'autista può mettere il taxi in servizio o fuori servizio. *TeleTaxi* gestisce anche le richieste provenienti da clienti che si recano direttamente presso le aree di sosta dei taxi o che fermano "al volo" un taxi libero. In questi casi l'autista preme il tasto *RichiestaDiretta* per comunicare al sistema la sua intenzione di servire un cliente. Dopo aver premuto il tasto di *RichiestaDiretta*, l'autista si comporta come se si trattasse di una richiesta proveniente dal sistema.

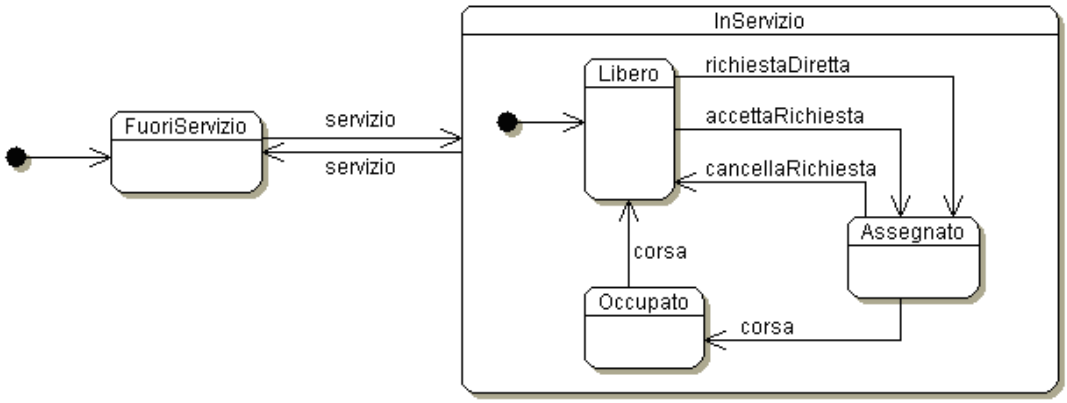
Cordiali saluti,

Ing. Vito Accorsio

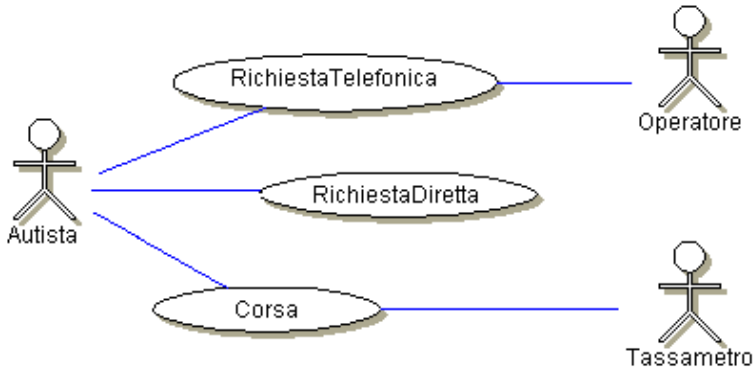
Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.
Risposta. Un possibile diagramma delle classi è il seguente:



Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva il taxi.
Risposta. Un possibile diagramma di macchina a stati è il seguente:



Si consideri il seguente diagramma (parziale) dei casi d'uso:



con le seguenti brevi descrizioni:

RichiestaTelefonica
Breve descrizione L'operatore, a fronte della chiamata di un cliente che richiede un taxi, contatta il sistema per trovare un taxi che accetti la richiesta.

RichiestaDiretta
Breve descrizione L'autista, a fronte della chiamata di un cliente che richiede un taxi per strada, contatta il sistema per registrare la richiesta.

Corsa
Breve descrizione L'autista segnala al sistema l'inizio e la fine della corsa. Il sistema comunica questi due eventi al tassametro.

Domanda. (Analisi dei requisiti) Dare la narrativa del caso d'uso *RichiestaTelefonica*, considerando le seguenti precondizioni (almeno un taxi è libero) e postcondizioni (la richiesta è assegnata a un taxi libero) per il caso normale.

Risposta. Una possibile narrativa è la seguente:

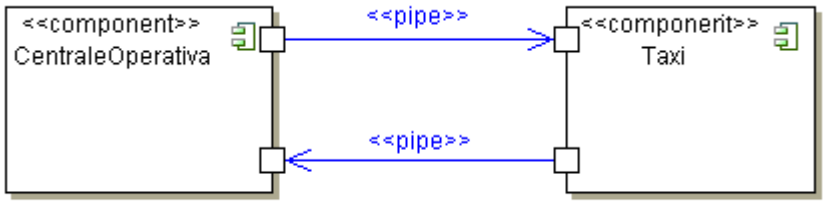
RichiestaTelefonica
Breve descrizione L'operatore, a fronte della chiamata di un cliente che richiede un taxi, contatta il sistema per trovare un taxi che accetti la richiesta.
Attore Principale Operatore.
Attore Secondario Autista.
Precondizioni Almeno un taxi è libero.
Postcondizioni La richiesta è assegnata a un taxi libero.
Sequenza principale degli eventi

1. L'operatore inserisce la richiesta nel sistema, indicando eventuali servizi speciali;
2. Il sistema invia la richiesta ai taxi liberi della flotta, selezionati secondo la strategia adottata;
3. Il sistema assegna la richiesta a uno dei taxi che hanno risposto;
4. Il sistema calcola il tempo stimato d'attesa;
5. Il sistema comunica all'operatore il numero del taxi prescelto e il tempo stimato d'attesa;
6. L'operatore, sentito il cliente, conferma e chiude la richiesta.
7. Il sistema informa l'autista del taxi prescelto.

Sequenze alternative degli eventi

1. Nessun taxi risponde alla chiamata. La richiesta è rifiutata.
2. Il cliente non accetta la proposta. La richiesta è cancellata.

L'architettura del sistema è data dalla seguente vista C&C:

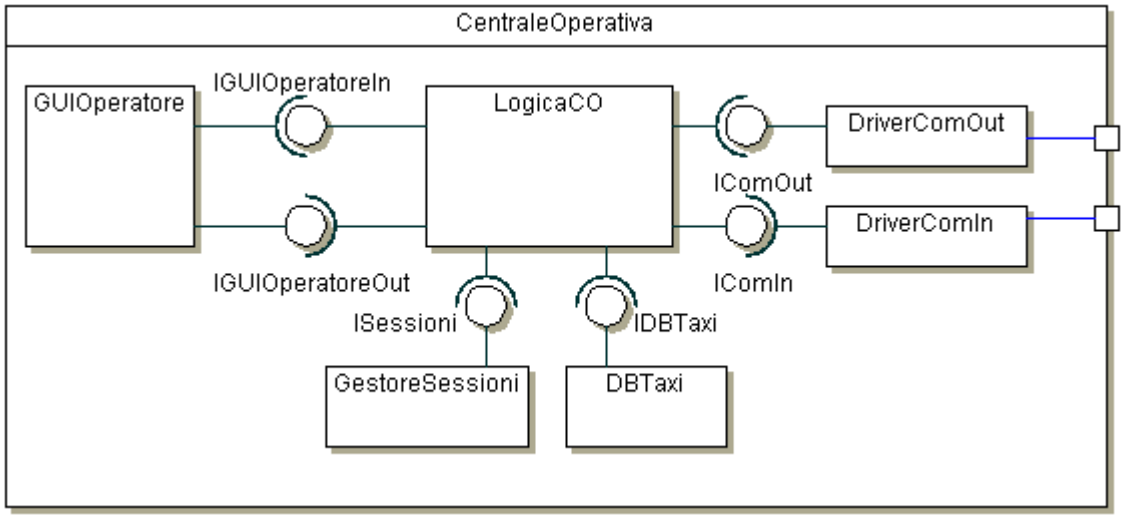


Il componente *CentraleOperativa* ha la responsabilità di tutto ciò che accade dal lato dell'operatore, il componente *Taxi* di ciò che accade dal lato dell'autista. Le connessioni tra i due componenti garantiscono la trasmissione

- da *Taxi* a *CentraleOperativa*: posizione, accettazione di una richiesta, inizio e fine di una corsa, rifiuto di una corsa (per destinazione non ammissibile o servizio speciale non dichiarato preventivamente e non prestabile), richiesta diretta;
- da *CentraleOperativa* a *Taxi*: richieste e loro assegnazioni.

Le connessioni sono realizzate via radio, a livello fisico, con canali virtuali tra la *CentraleOperativa* e le istanze dei *Taxi* interessati a una richiesta, in base all'algoritmo di ricerca dei taxi candidati a soddisfare una richiesta.

La struttura del componente *CentraleOperativa* è data dal seguente diagramma di struttura composita:

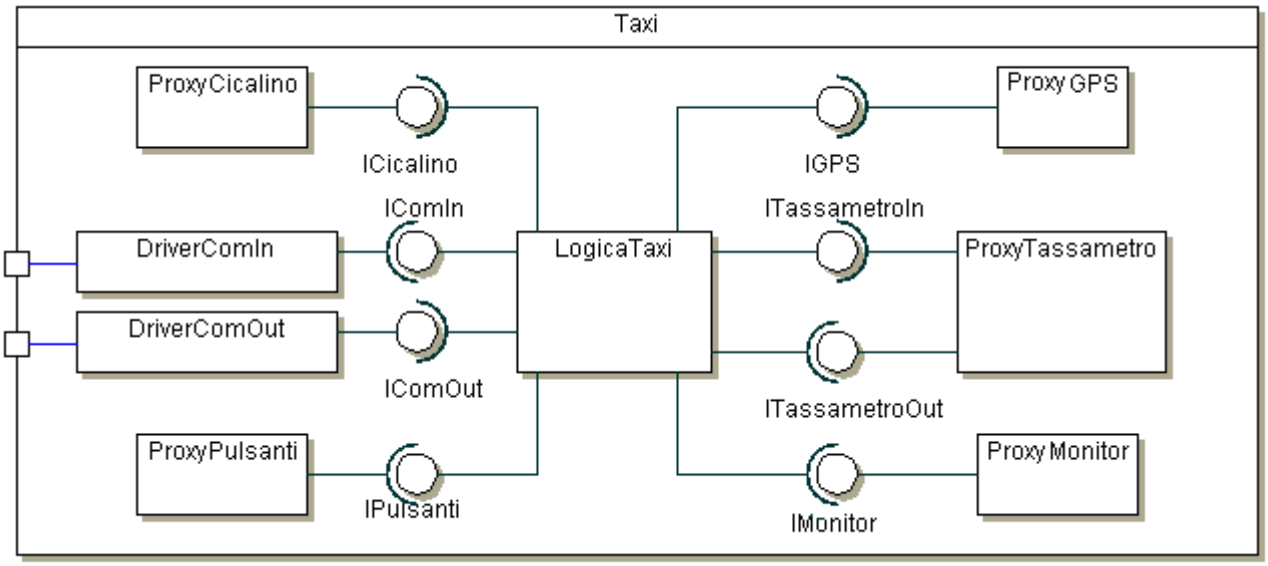


La parte di *LogicaCO*

1. gestisce le comunicazioni da e verso i taxi attraverso le interfacce di comunicazione *IComIn* e *IComOut*;
2. gestisce le comunicazioni da e verso l'operatore attraverso le interfacce della *GUIOperatore*;
3. recupera e aggiorna le informazioni sullo stato e sulla posizione dei taxi tramite l'interfaccia *IDBTaxi* verso il relativo data base;
4. richiede a *GestoreSessioni* la creazione di *Sessioni*, una per ogni richiesta, proveniente dall'operatore (tramite la *GUIOperatore*) o dall'autista (tramite l'interfaccia di comunicazione).

Domanda. (Progettazione di dettaglio) Dare un analogo diagramma di struttura composita per il componente *Taxi*.

Risposta. Un possibile diagramma di struttura composita per il componente *Taxi* è il seguente:



Le responsabilità sono le seguenti:

LogicaTaxi	Gestisce le interazioni tra *Taxi* e *CentraleOperativa*, tramite *DriverComIn e DriverComOut*, inviando alla *CentraleOperativa* le informazioni sulla posizione e lo stato del taxi. Gestisce le interazioni con l'autista, tramite le interfacce *ICicalino*, *IPulsanti* e *IMonitor*.
ProxyCicalino	Controlla il cicalino per richiamare l'attenzione dell'autista, quando arriva una richiesta.
DriverComunicazione	Gestiscono le comunicazioni via radio da (*In*) e per (*Out*) la *CentraleOperativa*.
ProxyPulsanti	Gestisce i segnali generati dall'autista.
ProxyGPS	Gestisce la generazione dei dati sulla posizione sfruttando il GPS.
ProxyTassametro	Gestisce l'interazione con il tassametro, per i segnali di inizio e fine corsa.
ProxyMonitor	Gestisce il monitor che presenta all'autista i dati sulle corse richieste.

La parte di *LogicaCO* interagisce con il *DBTaxi* per soddisfare due responsabilità:
1. mantenere aggiornate le posizioni dei taxi;
2. mantenere aggiornati gli stati dei taxi.

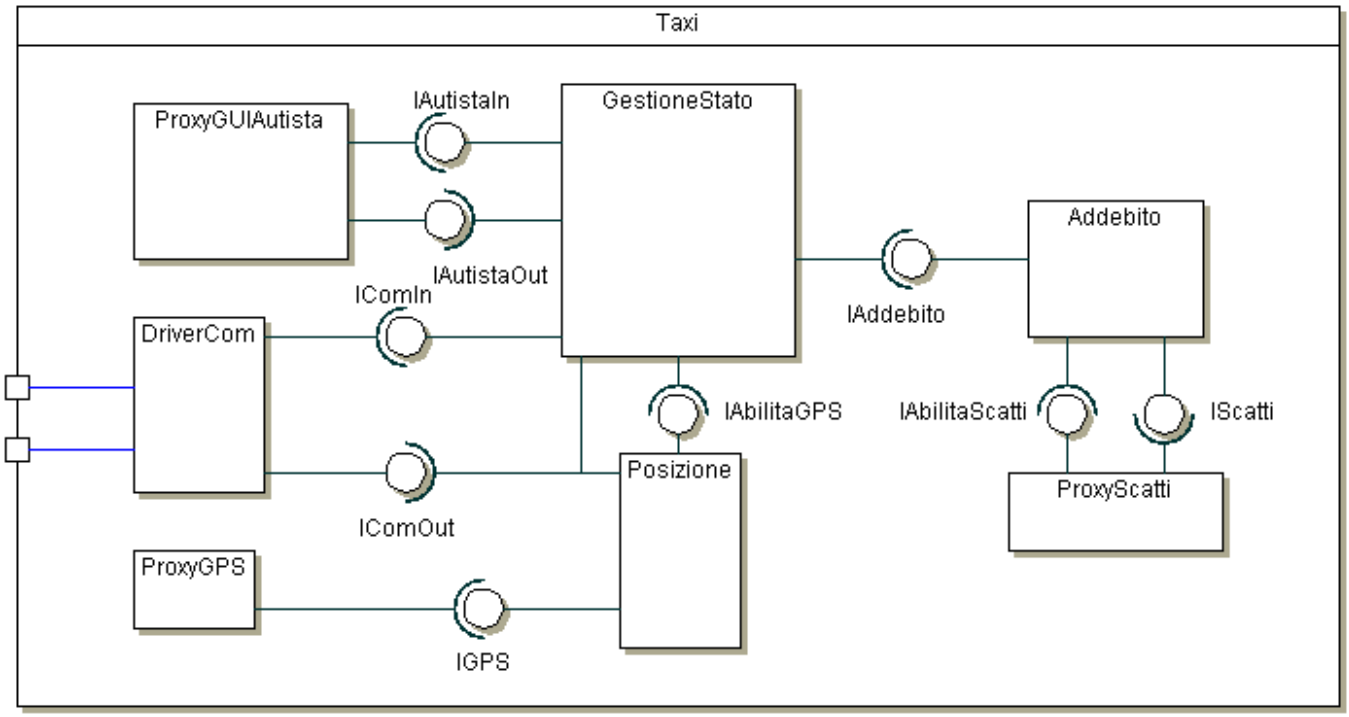
Si vuole riprogettare il componente *Taxi*, per gestire direttamente, anziché con un'apparecchiatura separata, il tassametro. Il nuovo componente deve quindi anche calcolare l'importo della corsa, in base agli scatti generati da un driver dell'odometro (strumento che misura le distanze percorse) e dell'orologio.
Sono state individuate alcune parti, con le seguenti responsabilità:

ProxyGUIAutista	Gestisce tutti i segnali da e per l'autista: richiesta, accettazione prenotazione, cicalino, visualizzazione importo corsa.
DriverCom	Gestisce le comunicazioni da e per la centrale operativa.
ProxyGPS	Gestisce il GPS, generando periodicamente i segnali di posizione del taxi che devono essere inviati alla centrale operativa.
Posizione	Gestisce localmente il trasferimento dei segnali di posizione dal GPS al sistema di comunicazione.
ProxyScatti	Gestisce l'odometro e l'orologio e genera in sequenza gli scatti di addebito.
Addebito	Calcola, in base agli scatti generati da *DriverScatti*, l'importo della corsa. Abilita *DriverScatti*.
GestioneStato	Realizza la logica principale del componente: gestisce le transizioni di stato del taxi, abilita il GPS, richiede il calcolo dell'importo.

Sono state individuate anche le sequenti interfacce: *IAutistaIn*, *IAutistaOut*, *IComIn*, *IComOut*, *IGPS*, *IAbilitaGPS*, *IAbilitaScatti*, *IScatti*, *IAddebito*.

Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita che rappresenti le connessioni tra le parti individuate tramite le interfacce elencate.

Risposta. Un possibile diagramma di struttura composita è il seguente:



Si consideri il seguente metodo Java :

```
public String generaId(Iniziatore i, int ora, int min) {
    String id;
    int codice;
    if (i instanceof Operatore) {
        id = "01"; codice = 1;
    } else if (i instanceof Autista) {
               id = "10"; codice = 10;
           } else { // i instanceof Applicazione
               id = "20"; codice = 20;
           }
    if (ora < 10) {
        id = id + "0" + ora;
    } else id = id + ora;
    if (min < 10) {
        id = id + "0" + min;
    } else id = id + min;
    int controllo = codice + ora + min;
    if (controllo < 10) {
        id = id + "00" + controllo;
    } else if (controllo < 100) {
               id = id + "0" + controllo;
           } else id = id + controllo;
    return id;
}
```

Domanda. (Verifica) Dare un insieme minimo di casi di prova (solo valori di input) che garantisca la copertura totale dei comandi del metodo `generaId`.

Risposta. Uno dei possibili insiemi minimi è il seguente:

Iniziatore (classe)	ora	min
Operatore	1	1
Autista	11	11
Applicazione	23	59

Si consideri il seguente metodo Java :

```
public int calcolaScatti (int distanza, int tariffa) throws Exception {
    if (distanza < 0) {
        throw new Exception("Distanza negativa");
    }
    int tratta = 0;
    switch (tariffa) {
        case 1: tratta = 250; break;
        case 2: tratta = 125; break;
        case 3: tratta = 300; break;
        default: throw new Exception("Tariffa invalida");
    }
    int scatti = 0;
    while (distanza >= tratta) {
        distanza -= tratta;
        if ((tariffa == 3) && (tratta > 100)) {
            tratta -= 10;
        }
        scatti++;
    }
    if (distanza > 0) {
        scatti++;
    }
    return scatti;
}
```

Si consideri il seguente insieme di casi di prova (non completamente definiti):

distanza	tariffa	risultato
200	1	
200	2	
200	3	

Si considerino le seguenti tabelle, relative al numero di scatti corrispondenti alla distanza, calcolati secondo le varie tariffe:

Tariffa 1		
da metri	a metri	scatti
1	250	1
251	500	2
501	750	3
751	1000	4
1001	1250	5

Tariffa 2		
da metri	a metri	scatti
1	125	1
126	250	2
251	375	3
376	500	4
501	625	5

Tariffa 3		
da metri	a metri	scatti
1	300	1
301	590	2
591	870	3
871	1140	4
1141	1400	5

Domanda. (Verifica). Si completi la definizione dei casi di prova, usando le tabelle date come oracolo. Si calcoli il grado di copertura dei comandi (comandi eseguiti/comandi) dell'insieme dei casi di prova, giustificando la **Risposta.**

L'insieme dei casi di prova (completamente definiti) è il seguente:

distanza	tariffa	risultato
200	1	1
200	2	2
200	3	1

Il grado di copertura dei comandi dell'insieme dei casi di prova è pari a 0,75. Il numero di comandi presenti nel metodo è 12; il numero dei comandi eseguiti è 9. Il grado di copertura è, quindi, 0,75 (calcolato come valore di 9/12).

CAP 8. RilevamentoPresenze

Spett. Ditta,

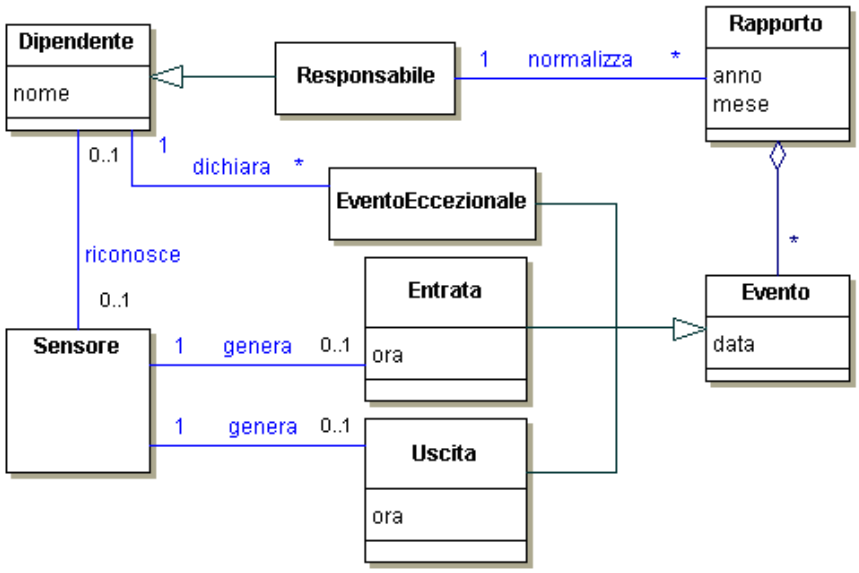
Dopo l'incontro relativo al sistema *RilevamentoPresenze* vi inviamo le informazioni richieste dal vostro analista.

In seguito ai recenti disservizi causati dall'attuale sistema di rilevamento delle presenze, la direzione aziendale ha deciso di sostituirlo con uno più moderno e in grado di gestire le nostre esigenze. In particolare, il sistema dovrà tenere conto della distribuzione geografica delle nostre filiali nel territorio regionale e della mobilità dei nostri **dipendenti**, consentendo loro di registrare l'**entrata** e l'**uscita** mediante uno qualsiasi degli appositi **sensori** del sistema. Il sensore dovrà rilevare la **data** e l'**ora** dell'**evento** e il **nome** del dipendente interessato. Le informazioni relative agli eventi generati dai dipendenti saranno raccolte dal sistema e raggruppate in un **rapporto** periodico (**mese**, **anno**). Il sistema prevede la visualizzazione degli eventi più recenti, su richiesta esplicita del dipendente. Gli **eventi eccezionali** (malattia, ferie, permessi) segnalati dai dipendenti sono registrati da un dipendente della sede centrale, **responsabile** del rilevamento presenze, mediante un'apposita interfaccia utente.

Prima di inviare i rapporti periodici al sistema stipendi, il responsabile del rilevamento presenze li normalizza, rimuovendo eventuali incongruenze (doppie entrate, doppie uscite) anche sulla base delle dichiarazioni presentate dai dipendenti. Può accadere, infatti, che un dipendente registri erroneamente un evento (un'entrata al posto di un'uscita, ad esempio) o che si dimentichi di registrarlo.

Tutte le informazioni relative ai dipendenti, al loro profilo (orari di entrata e di uscita, intervalli per il pranzo, compensazioni e autorizzazione allo svolgimento delle ore straordinarie) sono gestite dal sistema stipendi e inviate al sistema per il rilevamento delle presenze, che le utilizza in sola lettura.

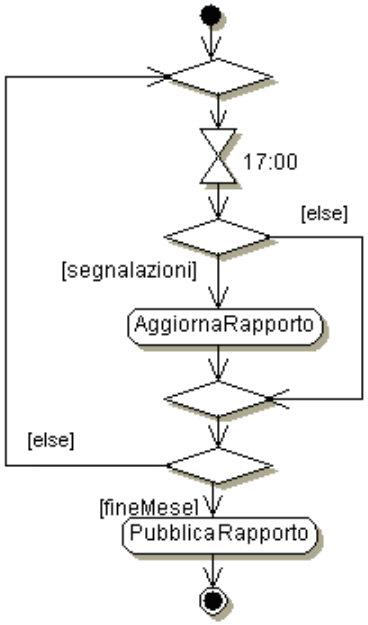
Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.
Risposta. Un possibile diagramma delle classi è il seguente:



La soluzione prevista per la normalizzazione (unico intervento a fine mese) si è dimostrata poco pratica. Si è pertanto scelta una nuova soluzione, in cui ogni giorno:

1. i dipendenti possono, oltre a registrare l'ingresso e l'uscita giornalieri, segnalare correzioni per ovviare a registrazioni dimenticate e segnalare eventi eccezionali,
2. il responsabile, alle 17:00, aggiorna il rapporto con le segnalazioni pervenute.

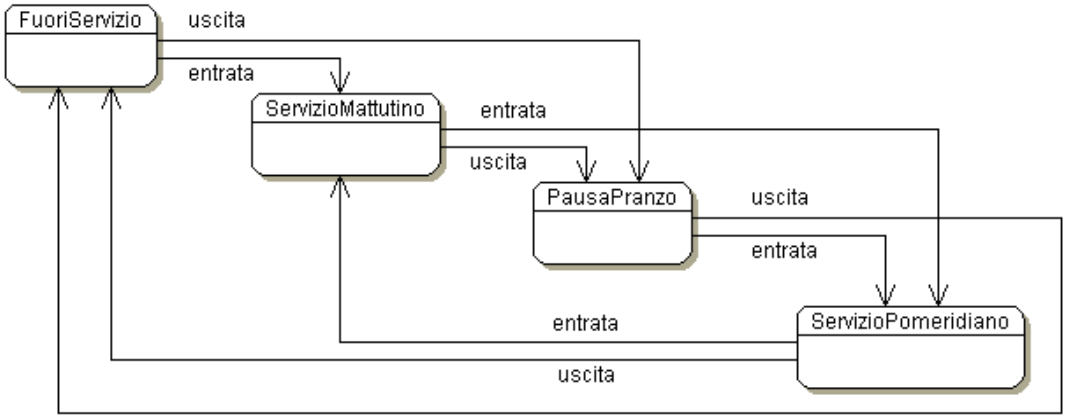
Domanda. (Analisi del dominio) Dare un diagramma d'attività che descriva il comportamento del responsabile durante un mese generico (non dicembre).
Risposta. Un possibile diagramma d'attività è il seguente:



Un dipendente può essere in uno dei seguenti stati: *FuoriServizio*, *ServizioMattutino*, *PausaPranzo* e *ServizioPomeridiano*. L'evoluzione normale del dipendente è descritta dalla sequenza naturale di entrate e uscite. L'evoluzione in caso di mancata registrazione da parte del dipendente è trattata in base alla seguente ipotesi: un evento di uscita (entrata) quando è atteso un evento di entrata (uscita) è interpretato come la dimenticanza della registrazione dell'evento atteso: si transita nello stesso stato in cui si arriverebbe con la sequenza corretta entrata-uscita (uscita-entrata).

Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che modella l'evoluzione degli stati del dipendente.

Risposta. Un possibile diagramma di macchina a stati è il seguente:



Si sono individuati i seguenti caso d'uso:

Registrazione

Breve descrizione	Il dipendente segnala la propria entrata o uscita al sistema.
Attori principali	Dipendente.
Attori secondari	Nessuno.
Precondizioni	Il sistema è nello stato di attesa di informazioni.
Postcondizioni	Il passaggio del dipendente è registrato nel sistema.

VisualizzazioneStoria

Breve descrizione	Il dipendente si identifica e richiede la lista dei propri eventi più recenti; il sistema la visualizza.
Attori principali	Dipendente.
Attori secondari	Nessuno.
Precondizioni	Il sistema è nello stato di attesa di informazioni.

RegistrazioneEventiEccezionali

Breve descrizione	Il responsabile immette nel sistema le informazioni relative a eventi eccezionali (malattia, ferie, permessi, ecc.), per ciascun dipendente interessato.
Attori principali	Responsabile.
Attori secondari	Nessuno

Normalizzazione

Breve descrizione	Il responsabile, basandosi sulle dichiarazioni dei dipendenti, modifica le informazioni contenute nel rapporto, per rimuovere eventuali incongruenze (doppie entrate o doppie uscite).
Attori principali	Responsabile.
Attori secondari	Nessuno

InvioRapporto

Breve descrizione	Il sistema, su richiesta del responsabile, invia il rapporto al *SistemaStipendi*, per il calcolo delle retribuzioni.
Attori principali	Responsabile.
Attori secondari	*SistemaStipendi*
Precondizioni	Scadenza periodica, rapporto normalizzato.

In particolare, il caso d'uso Normalizzazione ha la seguente sequenza principale degli eventi:

Normalizzazione

Sequenza principale degli eventi

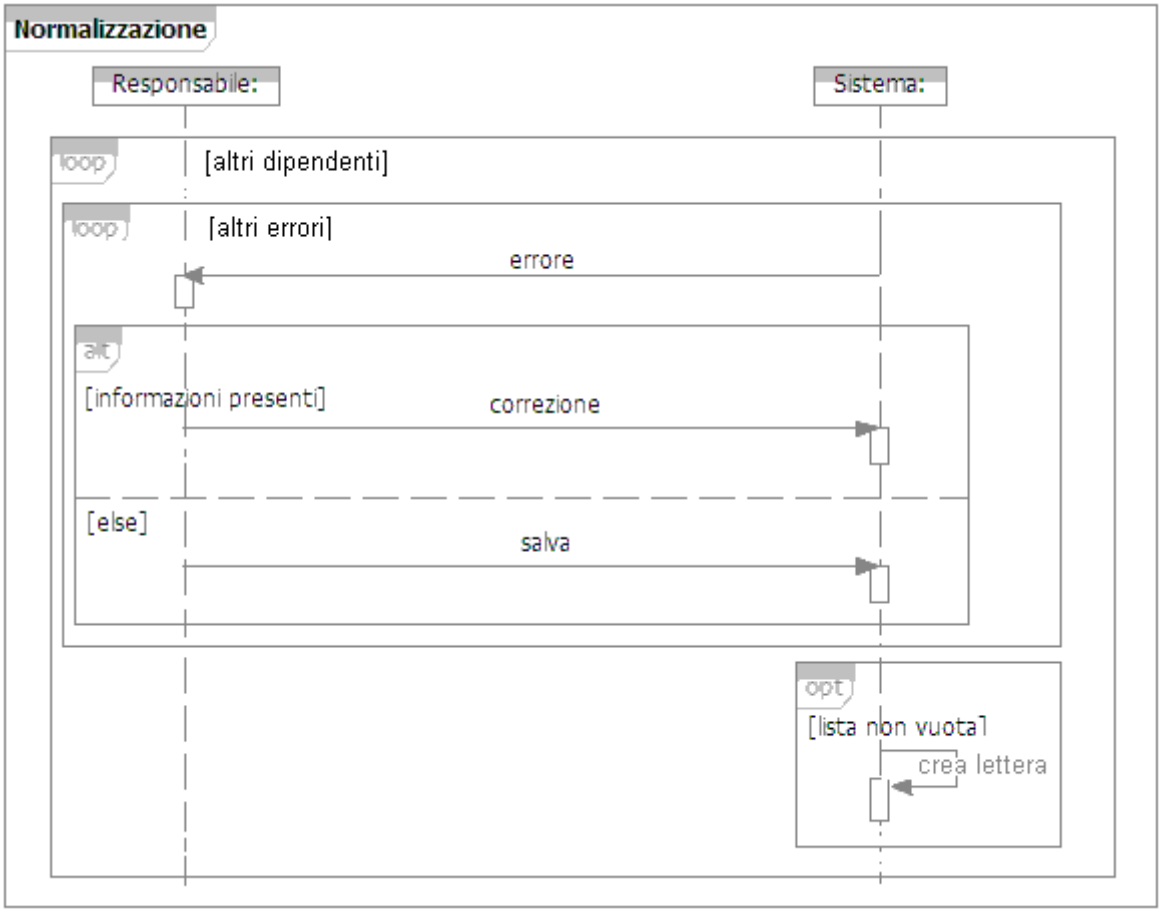
1. Per ogni dipendente:
1.1. per ogni doppia entrata o uscita:
1.1.1. il sistema la presenta al responsabile
1.1.2. se il responsabile ha informazioni al riguardo
comunica l'azione correttiva al sistema

1.1.3. altrimenti
il responsabile segnala al sistema di salvare l'incongruenza
in una lista 'daRisolvere'

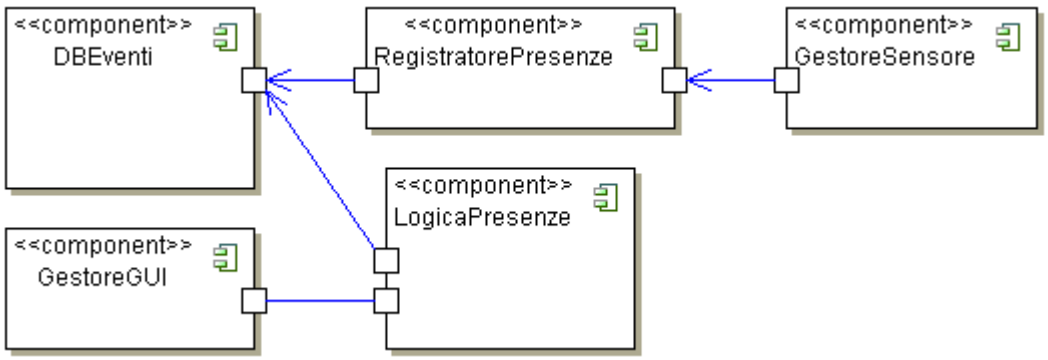
1.2. se la lista 'daRisolvere' non è vuota
il sistema genera una lettera di richiesta di chiarimenti al dipendente.

Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che realizzi il caso *Normalizzazione*.

Risposta. Un possibile diagramma di sequenza è il seguente:



La vista C&C dell'architettura del sistema è data dal seguente diagramma, dove mancano i tipi dei connettori:

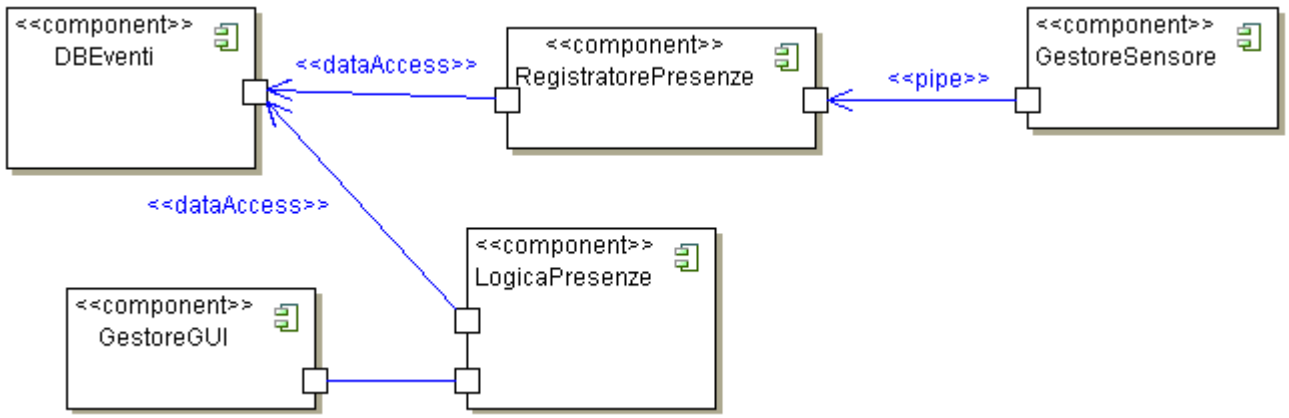


I componenti hanno le seguenti responsabilità:

GestoreSensore	Gestisce un sensore, invia gli eventi rilevati al *RegistratorePresenze*.
RegistratorePresenze	Interfaccia il *GestoreSensore* e il *DBEventi*.
DBEventi	Mantiene gli eventi rilevati dai sensori.
LogicaPresenze	Genera il rapporto periodico e lo invia al *SistemaStipendi*. Per generare il rapporto interroga il *DBEventi* e interagisce con *GestoreGUI*. Riceve dal *SistemaStipendi* le informazioni relative ai dipendenti.
GestoreGui	Permette di visualizzare gli eventi più recenti, di registrare gli eventi eccezionali e di normalizzare i rapporti.

Domanda. (Architettura) Completare il diagramma C&C indicando il tipo dei connettori, ove siano individuabili degli stili architettonici.

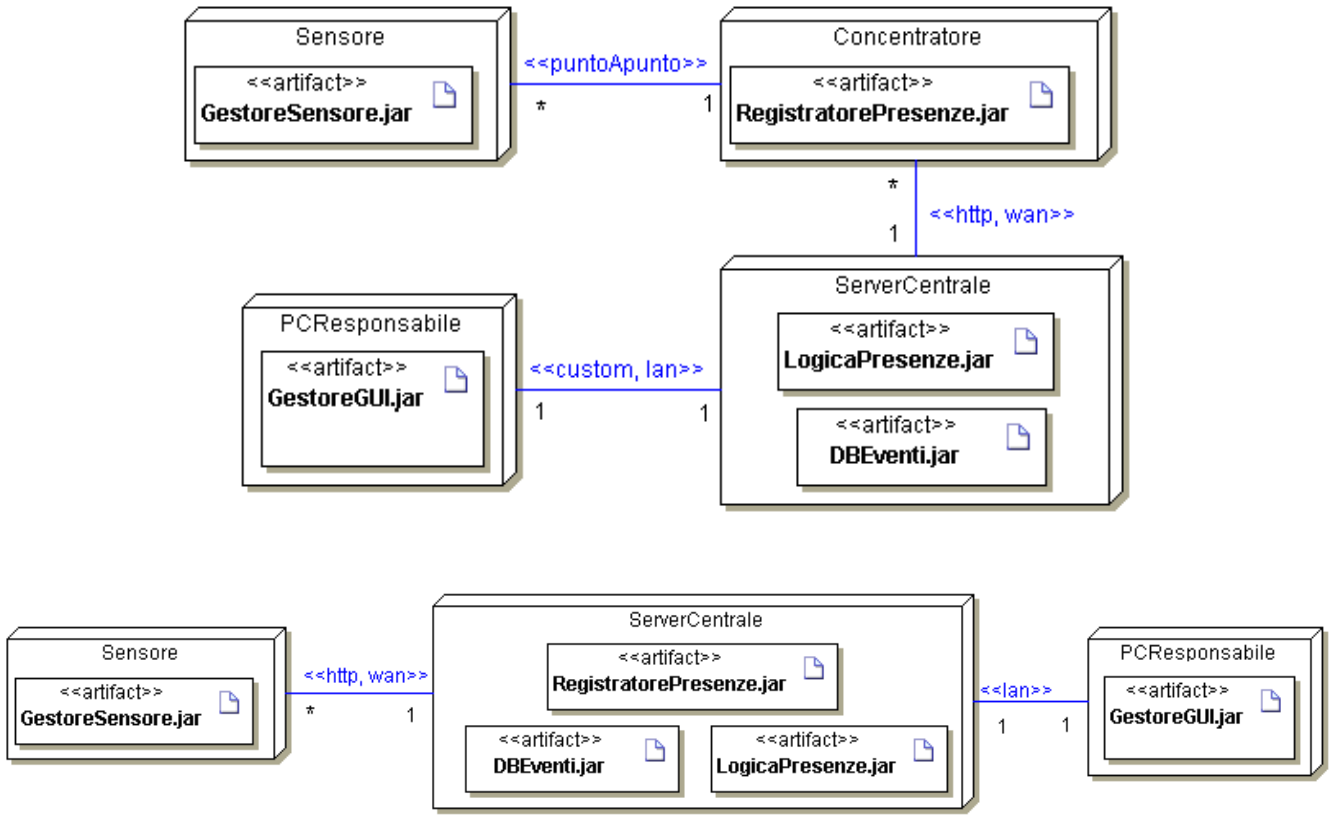
Risposta. Un possibile diagramma C&C è il seguente:



Per la realizzazione del sistema sono state proposte due soluzioni diverse. Entrambe prevedono l'uso di un server centrale e di un PC per il responsabile, presso la sede centrale della società, connesi in rete locale. Le soluzioni differiscono per il tipo di connessione dei sensori. La prima soluzione prevede un concentratore per ogni filiale e una connessione punto a punto tra ogni sensore e il concentratore, a sua volta connesso in rete geografica con il server centrale. La seconda prevede invece un collegamento in rete geografica tra ogni sensore e il server centrale, senza concentratori.

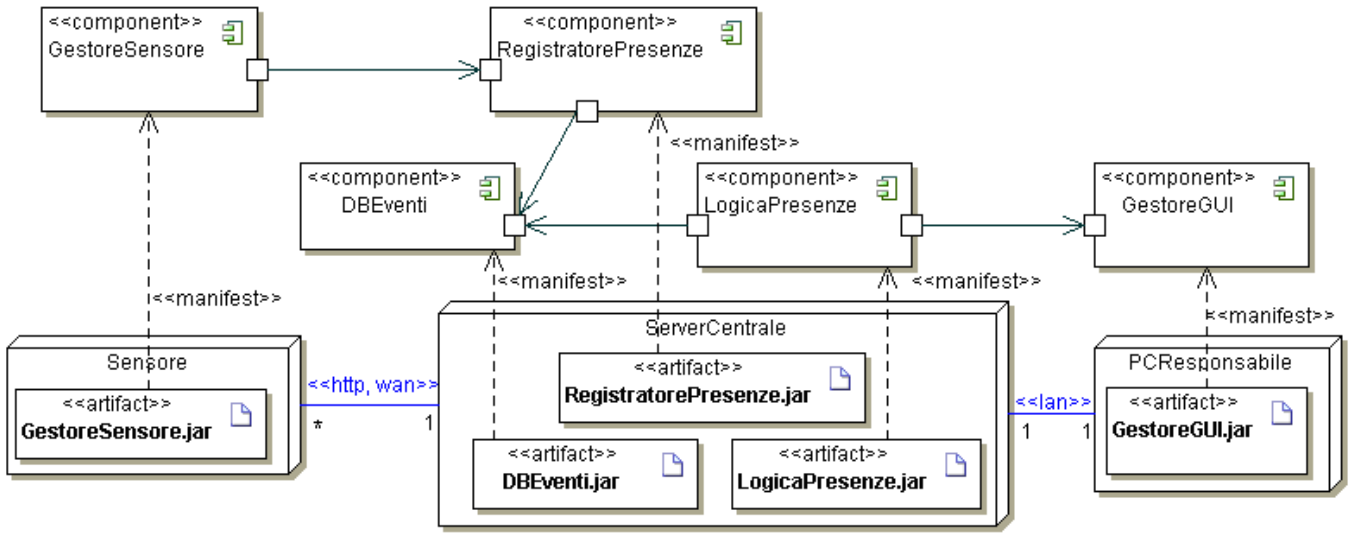
Domanda. (Architettura) Dare i diagrammi di dislocazione dei componenti descritti alla domanda precedente, per ciascuna delle strutture hardware. Si assuma che gli artefatti abbiano lo stesso nome dei componenti che manifestano, e che siano tutti di tipo *jar*.

Risposta. Una possibile soluzione è la seguente:



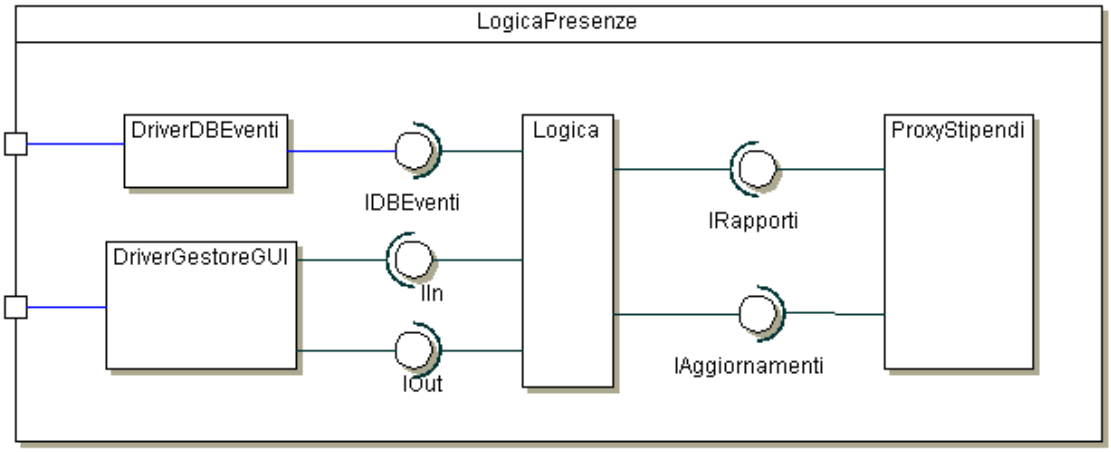
Domanda. (Architettura) Completare il diagramma precedente (relativo al caso di connessione internet tra i sensori e il server centrale), fornendo una vista ibrida di dislocazione con componenti.

Risposta. Un possibile diagramma è il seguente:



Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita del componente *LogicaPresenze* .

Risposta. Un possibile diagramma di struttura composita è il seguente:

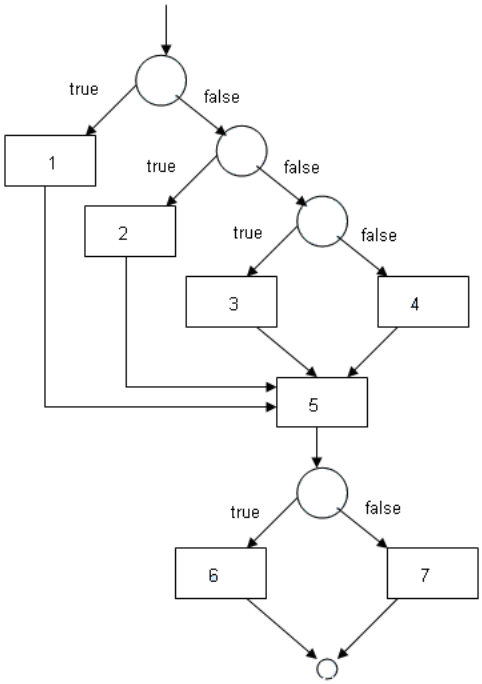


Si consideri la seguente classe Java:

```
public class PuntoNelTempo {
      private int  m; // minuto
      private int  h; // ora
      public  PuntoNelTempo(int ora, int minuto) {
            h = Math.abs(ora) % 24;
            m = Math.abs(minuto) % 60;
      }
      public  int intervallo (PuntoNelTempo p) {
         PuntoNelTempo p1, p2;
         if (this.h > p.h) {
             p1 = p;    p2 = this;
         } else if (this.h < p.h) {
             p1 = this; p2 = p;
         } else if (this.m >= p.m){
             p1 = p;    p2 = this;
         } else {
             p1 = this; p2 = p;
         }
         int int_h = p2.h - p1.h;
         int int_m = p2.m - p1.m;
         if (int_m >= 0) {
             return int_h * 60 +  int_m;
         } else {
             return (int_h - 1) * 60 + (60 + int_m);
         }
      }
}
```

Domanda. (Verifica) Si disegni il grafo di flusso del metodo intervallo. Si fornisca un insieme minimo di valori di input per i casi di prova, che garantiscano la copertura dei cammini del metodo `intervallo`.

Risposta. Il grafo di flusso del metodo `intervallo` è il seguente:



Un insieme di valori di input per minimizzare i casi di prova (con i relativi cammini) è mostrato in tabella:

this	P	ammino
8:40	6:30	(1, 5, 6)
8:40	6:50	(1, 5, 7)
8:40	9:30	(2, 5, 7)
8:40	9:50	(2, 5, 6)
8:40	8:30	(3, 5, 6)
8:40	8:50	(4, 5, 6)

Si noti che i cammini (3, 5, 7) e (4, 5, 7) non sono ammissibili, e che è possibile coprire tutti i cammini dato che il codice non contiene cicli.

Si consideri il seguente metodo Java:

```
int public minuti(int m, int d) {
    while (d >= 60) {d = d - 60;}
    if (m < 60)
       return m + d;
    else return m - d;
}
```

Si consideri la funzione $f(m, d) = |(m + d) \text{ modulo } 60|$

Domanda. (Verifica) Utilizzando la funzione $f(m, d)$ come oracolo si definisca un insieme minimo di casi di prova che copra tutti gli archi del grafo di flusso del metodo `minuti`.

Risposta. Un insieme minimo di casi di prova è mostrato in tabella:

m	d	risultato
10	70	20
70	40	50

Questo insieme di casi di prova permette di scoprire che il codice non rispetta le specifiche definite dall'oracolo.

CAP 9. ControlloChiuse

Chiuse 1

Per aumentare la sicurezza nei canali navigabili si è reso necessario rivedere la gestione del sistema di controllo delle **chiuse**, automatizzando le procedure di apertura e chiusura e introducendo la possibilità di una supervisione remota degli impianti. In particolare, è stata rivista e ridefinita l'interazione tra il sistema di controllo delle chiuse e le **imbarcazioni** in transito.

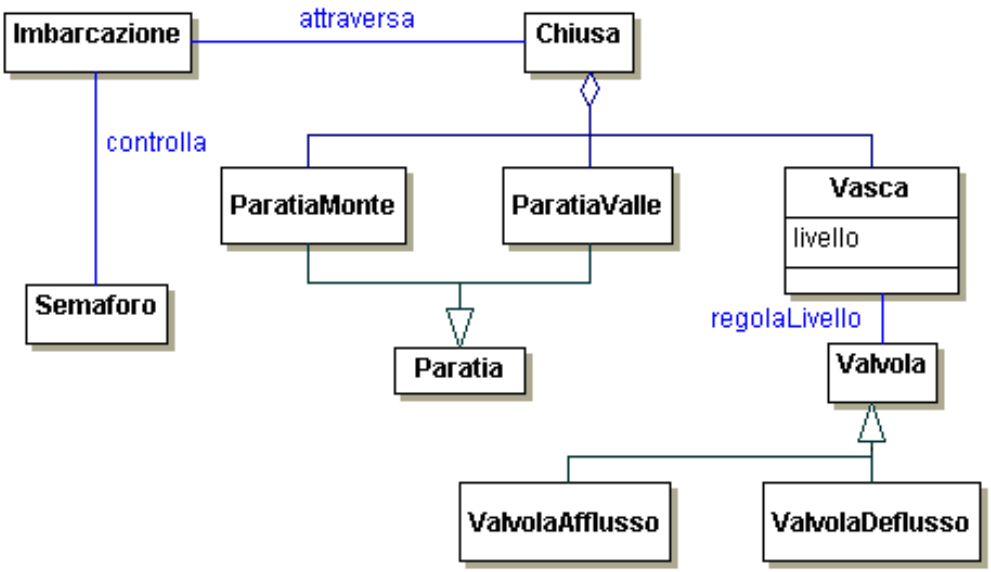
Una chiusa permette alle imbarcazioni in transito in un canale di superare un dislivello. La chiusa è formata da una **vasca** e da due **paratie** (**a monte** e **a valle**). Un'imbarcazione entra nella vasca e, sfruttando il principio dei vasi comunicanti, viene portata al livello desiderato. Il riempimento della vasca, nel caso di spostamento dal lato a valle a quello a monte del canale, avviene mediante l'apertura di una **valvola di afflusso**. Analogamente, lo svuotamento della vasca, nel caso di spostamento dal lato a monte a quello a valle, avviene mediante l'apertura di una **valvola di deflusso**. Si noti che non è necessario l'uso di pompe, in quanto sia l'afflusso sia il deflusso avvengono naturalmente. Prima di entrare nella vasca, il conducente dell'imbarcazione proveniente dal lato a valle deve prenotarne l'uso, mediante la pressione di un pulsante di prenotazione a valle posto prima delle paratie e davanti al lato destro dell'imbarcazione. Se la vasca è libera, il semaforo a valle passa dal rosso al verde. Viene quindi aperta la paratia di valle (se chiusa) per consentire all'imbarcazione di entrare nella vasca. Una volta nella vasca, il conducente dell'imbarcazione deve premere il pulsante di chiusura a valle, che attiverà la chiusura delle paratie, farà passare dal verde al rosso il semaforo a valle, attiverà il riempimento della vasca e l'apertura delle paratie a monte. Dopo essere uscito dalla vasca, il conducente dell'imbarcazione preme il pulsante di uscita a monte, per segnalare al sistema di controllo il transito avvenuto. Analogamente, un'imbarcazione proveniente dal lato a monte incontrerà, nell'ordine, il pulsante di prenotazione a monte, il pulsante di chiusura a monte e il pulsante di uscita a valle, mentre lo stato di prenotazione è rappresentato dal semaforo a monte. Un sensore di livello rileva il **livello** dell'acqua all'interno della vasca. Il sistema di controllo della chiusa gestisce l'apertura e la chiusura delle paratie, l'apertura e la chiusura delle **valvole**, l'accensione delle luci dei **semafori**, il rilevamento della pressione dei pulsanti. Tutte le informazioni relativi agli eventi, ai valori rilevati dai sensori e ai comandi inviati alle apparecchiature di attuazione sono registrate in un log.

Chiuse 2

In seguito a una variazione delle modalità di gestione del sistema delle **chiuse** è stato introdotto il pagamento del **pedaggio**. Ciò ha comportato la revisione della procedura di apertura e chiusura e la modifica degli impianti con l'eliminazione dei due pulsanti di chiusura (a valle e a monte) e l'introduzione di un **apparecchio flottante** (posizionato su un lato della vasca) per la riscossione del pedaggio con pagamento mediante contanti (senza garanzia di resto) o apposita carta prepagata con credito a scalare. La procedura per l'entrata e l'uscita è modificata solo nella parte relativa alla pressione del pulsante di chiusura (sostituita con il pagamento del pedaggio). In caso di mancato pagamento (per insufficiente quantità di contanti o per insufficiente credito sulla carta prepagata) il **conducente** dell'imbarcazione viene informato dell'evento e invitato a uscire dalla vasca premendo, entro due minuti, il pulsante di uscita. Se il pulsante di uscita non è premuto entro due minuti, viene azionata una **sirena** che segnala la situazione anomala e, dopo altri due minuti, il sistema si riporta nello stato di chiusa libera. L'**importo** del pedaggio varia da chiusa a chiusa. La sirena viene anche azionata in caso di pagamento regolare, se il conducente dell'imbarcazione non preme il pulsante di uscita entro due minuti dall'apertura delle paratie, che consente il completamento del passaggio della chiusa.

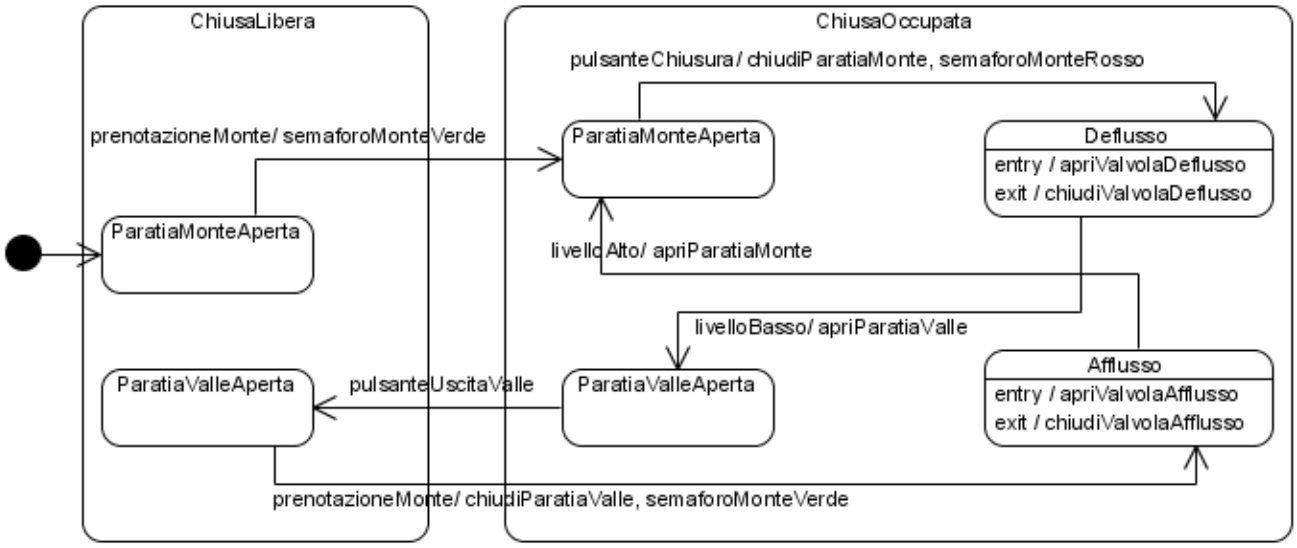
Domanda. (Analisi del dominio, Chiuse 1) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

Risposta. Un possibile diagramma è il seguente:



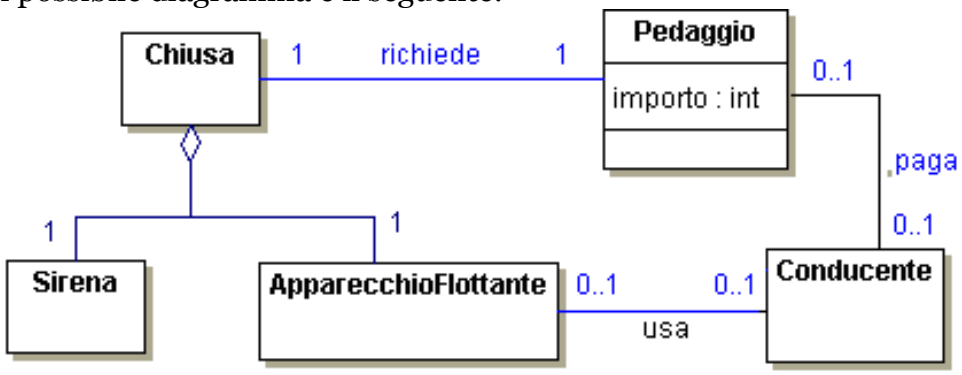
Domanda. (Analisi del dominio, Chiuse 1) Dare un diagramma di macchina a stati che descriva il superamento della chiusa da parte di un'imbarcazione che giunge da monte. Il diagramma deve includere gli stati di *chiusa libera* e *chiusa occupata* e i relativi sottostati. Il diagramma deve poter essere esteso alla descrizione del superamento della chiusa da parte di un'imbarcazione che giunge da valle con la sola aggiunta di alcune transizioni.

Risposta. Una possibile soluzione è la seguente:



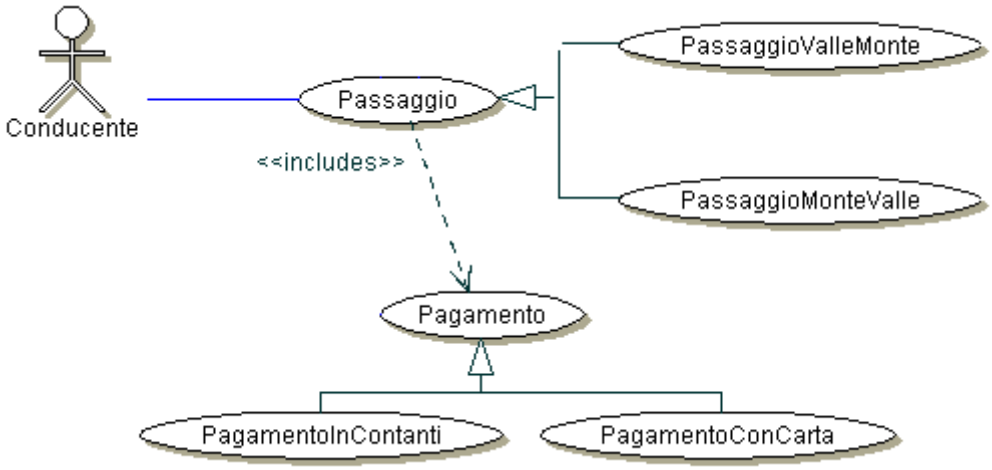
Domanda. (Analisi del dominio, Chiuse 2) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

Risposta. Un possibile diagramma è il seguente:



Domanda. (Analisi dei requisiti, Chiuse 2) Fornire il diagramma dei casi d'uso del sistema *ControlloChiuse*, considerando che il passaggio della chiusa comprende il pagamento e può avvenire da monte a valle o da valle a monte.

Risposta. Una possibile soluzione è la seguente:



Si consideri la narrativa (incompleta) del caso d'uso relativo al passaggio della chiusa da valle a monte:

PassaggioValleMonte

Breve descrizione Passaggio della chiusa dal lato valle al lato monte.
Precondizioni La chiusa è libera, una paratia è aperta.

Domanda. (Analisi dei requisiti) Completare la narrativa del caso d'uso *PassaggioValleMonte*.

Risposta. Una possibile soluzione è la seguente:

PassaggioValleMonte

Breve descrizione Passaggio della chiusa dal lato valle al lato monte.
Attori principali Conducente

Attori secondari Nessuno
Precondizioni La chiusa è libera, una paratia è aperta.
Postcondizioni La chiusa è libera, una paratia è aperta.
Sequenza principale degli eventi

1. Il conducente prenota l'uso della chiusa, premendo il pulsante di prenotazione a valle.
2. Il semaforo a valle passa da rosso a verde.
3. Se è aperta la paratia a monte:
   3.1. viene chiusa tale paratia,
   3.2. si attiva lo svuotamento della vasca,
   3.3. viene aperta la paratia di valle.
4. Dopo che il conducente ha trasferito l'imbarcazione nella vasca, si include *Pagamento*.
5. Viene chiusa la paratia a valle e il semaforo a valle passa da verde a rosso.
6. Si attiva il riempimento della vasca.
7. Viene aperta la paratia a monte.
8. Dopo essere uscito dalla vasca, il conducente preme il pulsante di uscita a monte.

Sequenze alternative degli eventi

1. Non viene correttamente eseguito il pagamento: il conducente viene informato dell'evento e invitato a uscire.
2. Il conducente non preme entro due minuti il pulsante di uscita: la sirena suona e dopo due minuti la chiusa torna nello stato di libera.

Per completezza si fornisce anche la narrativa del casi d'uso *Pagamento* e dei sotto-casi *PagamentoInContanti* e *PagamentoConCarta*.

Pagamento
Breve descrizione Pagamento del pedaggio della chiusa.
Attori principali Conducente.
Attori secondari Nessuno.
Precondizioni La chiusa è occupata, l'imbarcazione è nella vasca.
Postcondizioni La chiusa è occupata, l'imbarcazione è nella vasca, il pedaggio è stato pagato.

PagamentoInContanti
Breve descrizione: Pagamento in contanti del pedaggio della chiusa.
Sequenza principale degli eventi

1. Il conducente inserisce nell'apparecchio flottante un importo in contanti pari o maggiore del pedaggio.
2. Per quanto possibile in base alla disponibilità di contante, l'apparecchio flottante restituisce l'eventuale resto.

Sequenze alternative degli eventi:

1. Il conducente non ha denaro sufficiente: il pagamento fallisce.

PagamentoConCarta
Breve descrizione Pagamento con carta prepagata del pedaggio della chiusa.
Sequenza principale degli eventi

1. Il conducente inserisce la carta nell'apparecchio flottante.
2. L'apparecchio flottante scala l'importo del pedaggio dalla carta.

Sequenze alternative degli eventi

1. La carta ha un credito insufficiente: il pagamento fallisce.

Per realizzare il sistema di controllo conviene introdurre tre componenti. I primi due, da dislocare su una macchina specializzata con elevate caratteristiche di resistenza agli agenti

atmosferici, servono per la gestione dei sensori e degli attuatori, rispettivamente. Il terzo componente, dislocato su un PC standard, è sede del controllo: riceve i segnali dai sensori e trasmette gli ordini opportuni agli attuatori. Le due macchine sono collegate da due collegamenti seriali, uno per ogni verso di comunicazione. Il seguente diagramma fornisce una vista ibrida dell'architettura del sistema: dislocazione in forma istanza e C&C.

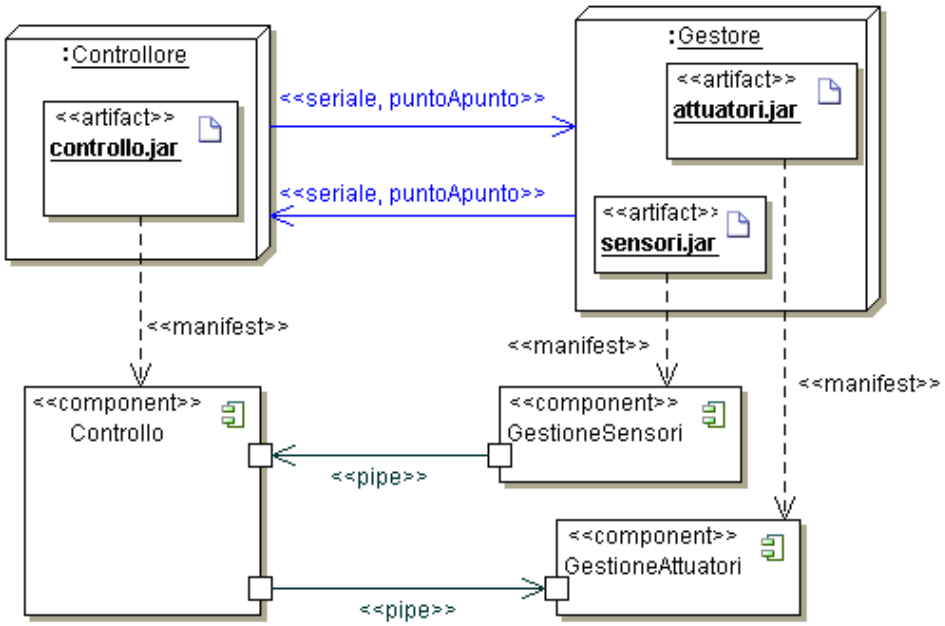
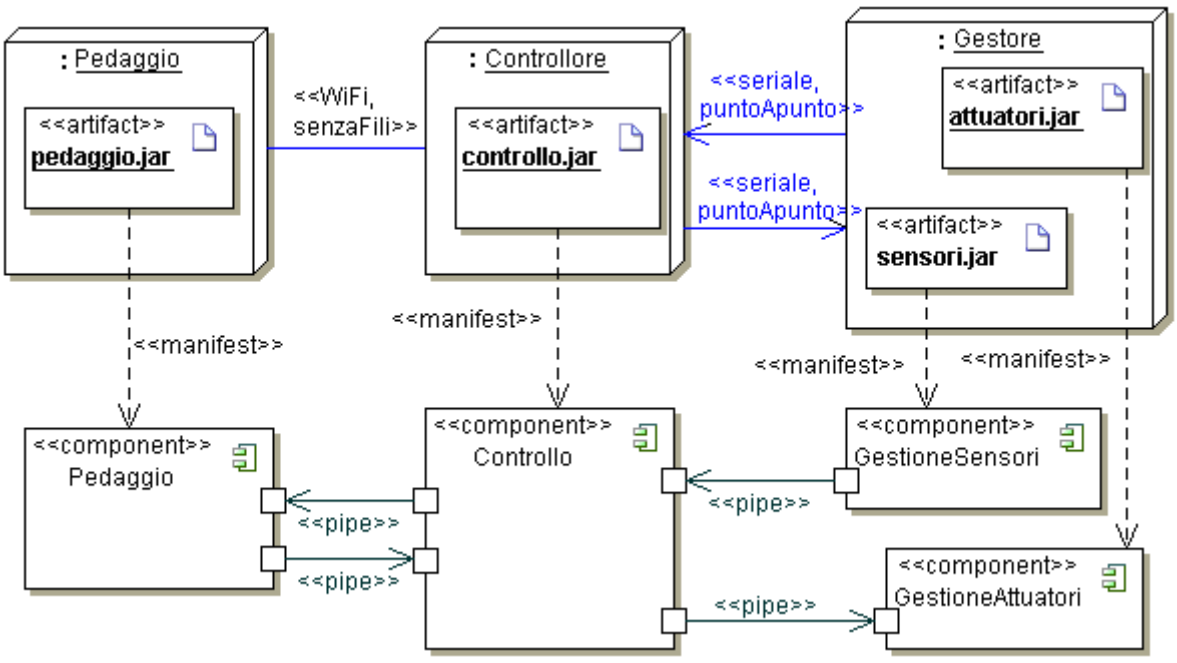


Figura 1: Architettura Chiuse 1

L'introduzione del pedaggio comporta una modifica all'architettura: conviene inserire una scheda nell'apparecchio flottante per la riscossione del pedaggio, su cui dislocare un nuovo componente per la gestione del pagamento. Il collegamento tra il nuovo nodo e il *Controllore* sarà su canale diretto senza fili, con protocollo WiFi.

Domanda. (Architettura, Chiuse 2) Dare una vista ibrida (C&C e dislocazione) della nuova architettura.
Risposta. Una possibile **soluzione** è la seguente:

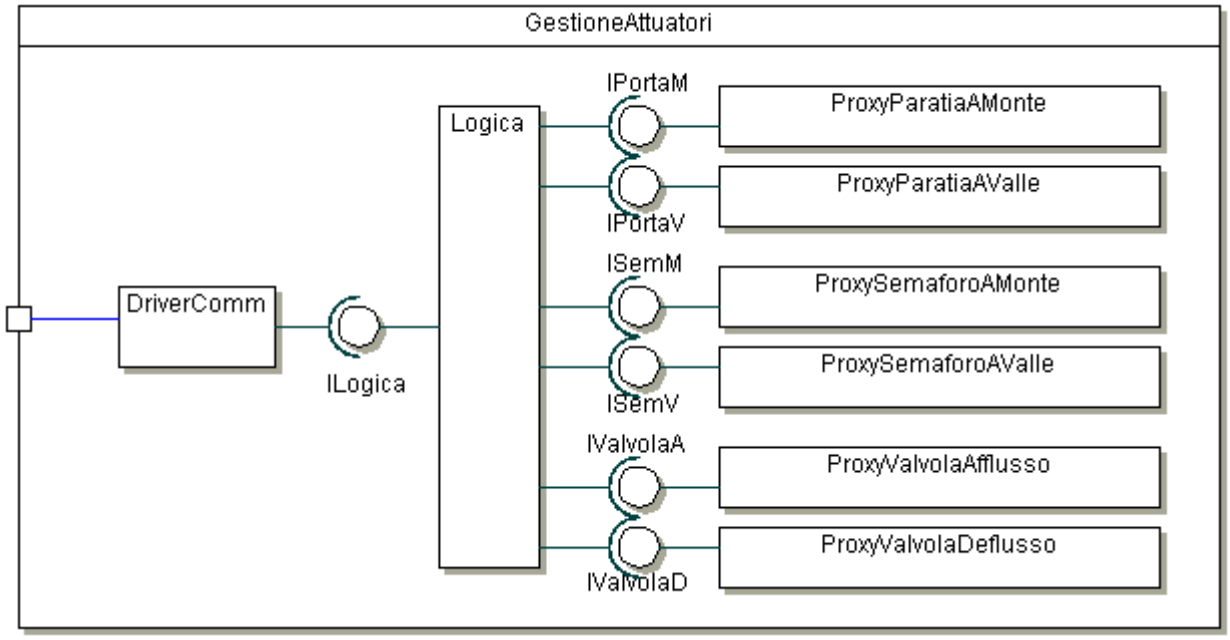


Si consideri la vista ibrida di Figura 1, pag 59. Si considerino, inoltre, le seguenti parti con le relative responsabilità:

1. *DriverComunicazione*: interfaccia il componente con la linea seriale che trasmette i segnali di controllo per gli attuatori.
2. *Logica*: distribuisce i segnali di controllo all'opportuno attuatore.
3. *ProxyX*: gestisce l'interfaccia verso l'attuatore X.

Domanda. (Progettazione di dettaglio, Chiuse 1) Dare un diagramma di struttura composita con il progetto di dettaglio di *GestioneAttuatori*, che tenga conto delle parti *DriverComunicazione*, *Logica*, *ProxyX*, descritte sopra.

Risposta. Un possibile diagramma di struttura composita è il seguente:



Si consideri la seguente classe Java:

```
public class Paratia {
    private static final int
       aperta= 1, chiusa= 2, inApertura= 3, inChiusura = 4, iniziale = 5;
    private int statoCorrente = iniziale;
    public  static final int
        chiudiParatia = 1, apriParatia = 2, fineApertura = 3,
        fineChiusura  = 4, startAperta = 5, startChiusa  = 6;
    public void stato(int evento){
        switch (statoCorrente) {
        case iniziale:
            if (evento == startAperta) statoCorrente = aperta;
            if (evento == startChiusa) statoCorrente = chiusa;
            break;
        case aperta:
            if (evento == chiudiParatia)
                {attivaChiusura();  statoCorrente = inChiusura;} break;
        case inChiusura:
            if (evento == fineChiusura)
               {arrestaChiusura(); statoCorrente = chiusa;    } break;
```

```
        case chiusa:
            if (evento == apriParatia)
               {attivaApertura();  statoCorrente = inApertura;} break;
        case inApertura:
            if (evento == fineApertura)
               {arrestaApertura(); statoCorrente = aperta;    } break;
        }
    }
    public int getStatoCorrente() {return statoCorrente;}
    private void attivaChiusura() {}
    private void arrestaChiusura(){}
    private void attivaApertura() {}
    private void arrestaApertura(){}
    public Paratia() {}
}
```

Si consideri la seguente specifica:
"Se la paratia è aperta, l'evento chiudi ne attiva la chiusura e la mette in chiusura; se la paratia è chiusa, l'evento apri ne attiva l'apertura e la mette in apertura; se la paratia è in chiusura, l'evento fine chiusura ne arresta la chiusura e la mette in chiusa; se la paratia è in apertura, l'evento fine apertura ne arresta l'apertura e la mette in aperta; se la paratia è nello stato iniziale, l'evento start aperta la mette in aperta e l'evento start chiusa la mette in chiusa"

Domanda. (Verifica, Chiuse 1) Usando la specifica data, si definisca un insieme di casi di prova che complessivamente abbiano una copertura totale dei comandi del metodo `stato` della classe `Paratia`.

Risposta. Il seguente insieme di casi di prova è sufficiente a ottenere la copertura richiesta.

Stato corrente	*Evento*	*Stato atteso*
iniziale	startChiusa	Chiusa
iniziale	startAperta	aperta
chiusa	apriParatia	inApertura
inApertura	fineApertura	aperta
aperta	chiudiParatia	inChiusura
inChiusura	fineChiusura	chiusa

Per l'esecuzione dei casi di prova è conveniente definire due batterie di prova, che saranno eseguite dai seguenti driver:

```
public void driver1() {
    Paratia p = new Paratia();
    p.stato(Paratia.startChiusa); System.out.println(p.getStatoCorrente());
    p.stato(Paratia.apriParatia); System.out.println(p.getStatoCorrente());
    p.stato(Paratia.fineApertura); System.out.println(p.getStatoCorrente());
    p.stato(Paratia.chiudiParatia);
    System.out.println(p.getStatoCorrente());
    p.stato(Paratia.fineChiusura); System.out.println(p.getStatoCorrente());
}
```

```
public void driver2() {
    Paratia p = new Paratia();
    p.stato(Paratia.startAperta);
System.out.println(p.getStatoCorrente());
}
```

Si consideri la seguente classe Java:

```
public class Pagamento {
    public Pagamento() {
    }
    public static int numeroMonete(int importo, int valore){
        int resto = importo;
        int valoreMoneta = 100;
        if(valore == valoreMoneta) return resto / valoreMoneta;
        resto = resto % valoreMoneta;
        valoreMoneta = 20;
        if(valore == valoreMoneta) return resto / valoreMoneta;
        resto = resto % valoreMoneta;
        valoreMoneta =  5;
        if(valore == valoreMoneta) return resto / valoreMoneta;
        resto = resto % valoreMoneta;
        valoreMoneta =  1;
        if(valore == valoreMoneta) return resto / valoreMoneta;
        return 0;
    }
}
```

Si considerino i seguenti casi di test:

importo	valore	risultato
654	100	6
654	20	1
654	1	4

Domanda. (Verifica, Chiuse 2) Qual è il risultato dell'esecuzione dei casi di test? Qual è il grado di copertura dei comandi che si ottiene eseguendo i casi di test?

Risposta. Il primo e il terzo caso di test hanno successo, il secondo fallisce (il risultato ottenuto è 2 e non 1). Si noti che, in assenza di specifiche, non è possibile decidere se il codice è difettoso, o se è il secondo caso di test a esserlo. Discutere quale caso è più probabile, inferendo la specifica intesa dal codice.

Il grado di copertura dei comandi che si ottiene eseguendo i casi di test è 84,6% (11 comandi su 13 sono eseguiti).

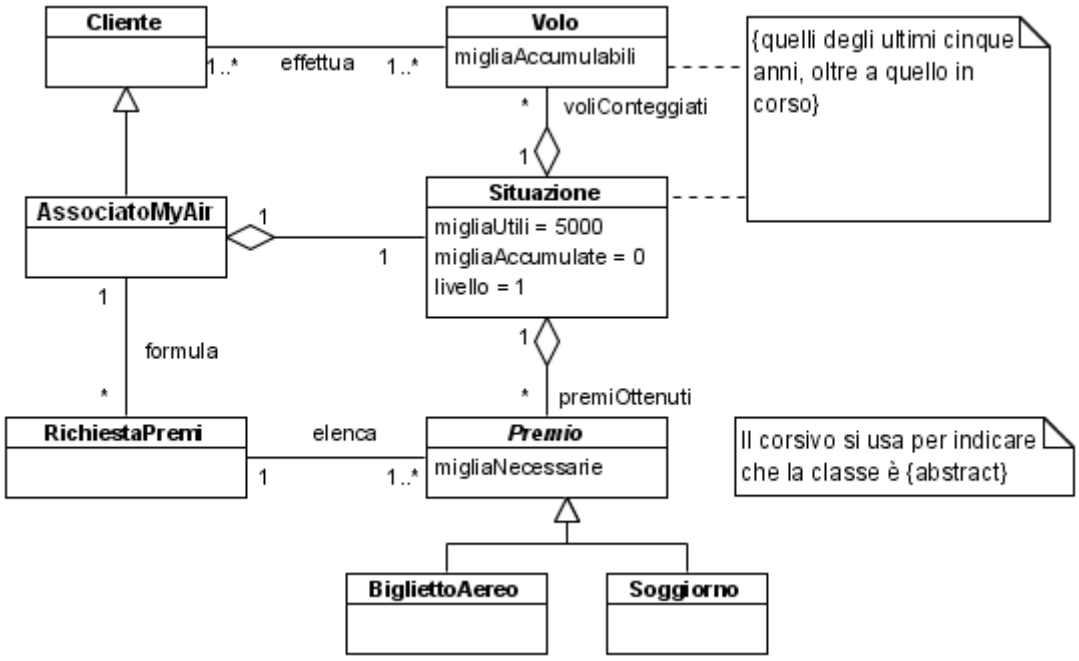
CAP 10. MyAir

Si consideri il seguente messaggio pubblicitario:
Vola con MyAir! Entra a far parte del club MyAir: sarai trattato con l'attenzione che ti meriti ma, soprattutto, potrai ricevere in omaggio **biglietti aereo** o **soggiorni** in località da sogno. Iscriviti al programma e da semplice **cliente** diventerai un **associato MyAir**, guadagnando immediatamente un bonus di 5.000 **miglia utili**. Ogni volta che volerai con MyAir le **miglia accumulabili** del **volo** saranno sommate alle tue miglia utili, permettendoti di raggiungere in poco tempo le **miglia necessarie** per richiedere uno dei nostri favolosi **premi**. Ricordati che la **richiesta premi** deve essere effettuata mediante il portale dedicato ai associati MyAir. Non dovrai compilare moduli o inviare lettere, semplicemente scegli il tuo premio, stampa la ricevuta e inizia a sognare[1]. Ma essere associati MyAir non è solo questo: se accumulerai almeno 15.000 miglia (**miglia accumulate**) sarai promosso dal **livello** standard al livello argento, usufruendo di particolari agevolazioni al momento dell'imbarco. Se invece accumulerai almeno 100.000 miglia entrerai a far parte del ristretto numero di associati del livello oro[2].

---

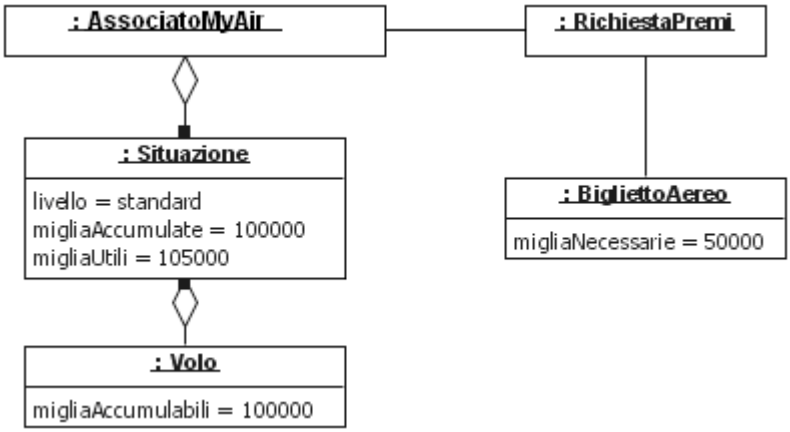
1. Soggetto a disponibilità di miglia utili nella **situazione** dell'associato MyAir e di disponibilità del premio scelto. I premi riscossi danno luogo a una diminuzione immediata delle miglia utili. La situazione è aggiornata il 31 dicembre, mantenendo solo le miglia dei voli effettuati negli ultimi 5 anni.
2. Tutte le condizioni si riferiscono esclusivamente alle miglia accumulate in un anno. Il passaggio da un livello all'altro è effettuato il 31 dicembre. La permanenza nel livello da un anno all'altro è soggetta al rispetto degli stessi requisiti per entrare nel livello. Il bonus iniziale non concorre al raggiungimento delle miglia richieste per cambiare o mantenere un livello.

Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.
Risposta. Un possibile diagramma è il seguente:



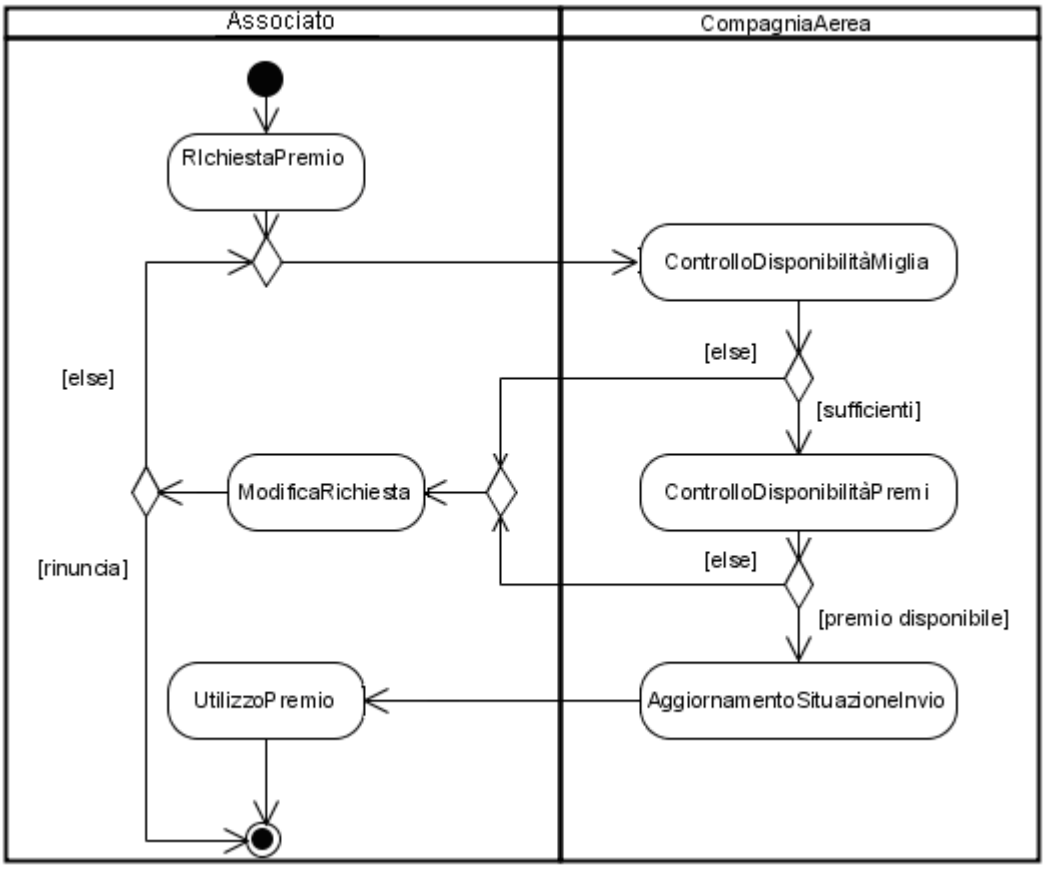
Domanda. (Analisi del dominio) Dare un diagramma degli oggetti che descriva la seguente situazione. Un cliente appena entrato nel club, ha fatto un volo da 100000 miglia, e ha richiesto come premio un biglietto aereo da 50000 miglia.

Risposta.



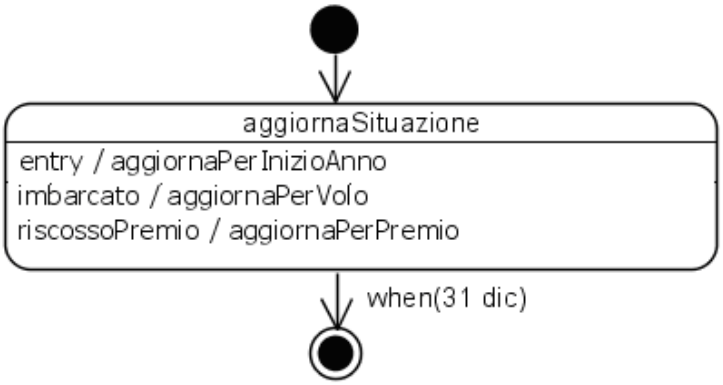
Domanda. (Analisi del dominio) Dare un diagramma di attività che descriva il processo di assegnazione di un premio.

Risposta. Una possibile soluzione è la seguente:

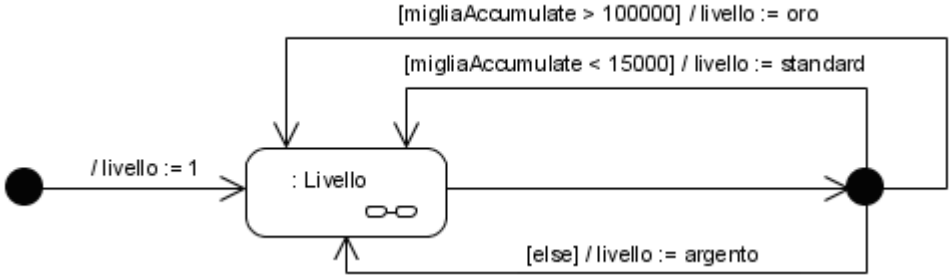


Domanda. (Analisi del dominio) Descrivere, mediante macchine a stati, il comportamento della situazione di un associato, nel corso di un anno e al passaggio da un anno all'altro. Si sappia che le miglia accumulabili di un volo vengono assegnate al momento dell'imbarco.

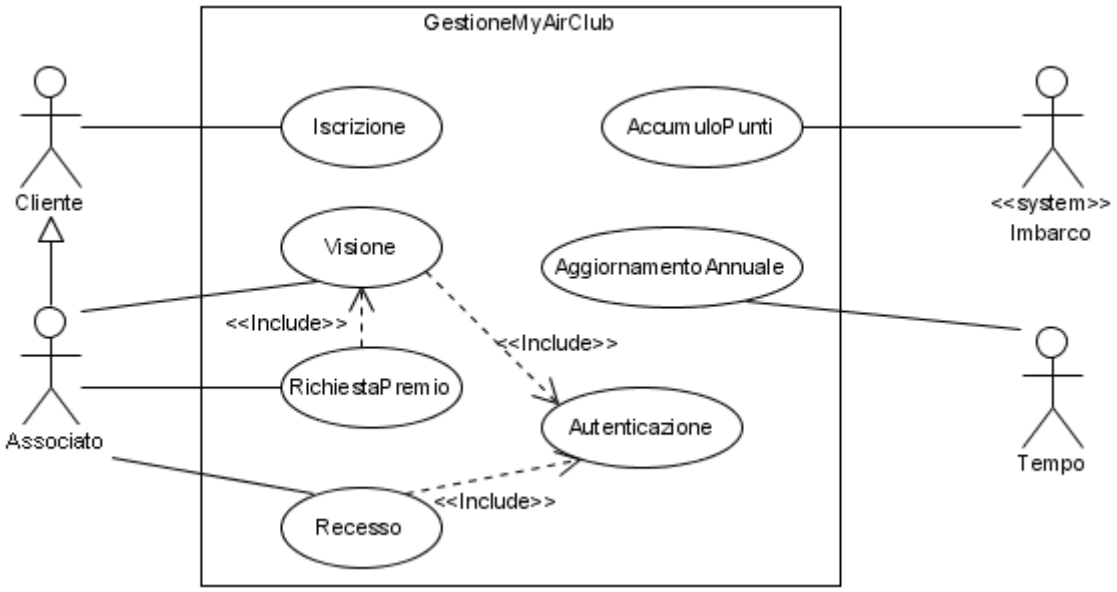
Risposta. Una possibile soluzione è la seguente: data la macchina a stati finiti Livello, descritta dal seguente diagramma, che definisce il comportamento durante l'anno,



il comportamento negli anni della situazione è descritto dalla macchina



L'analisi dei requisiti ha portato al seguente diagramma dei casi d'uso:



Il caso AccumuloPunti ha la seguente breve descrizione: Il sistema riceve la lista dei passeggeri di un volo e la esamina, aggiornando di conseguenza la situazione degli associati del club MyAir.

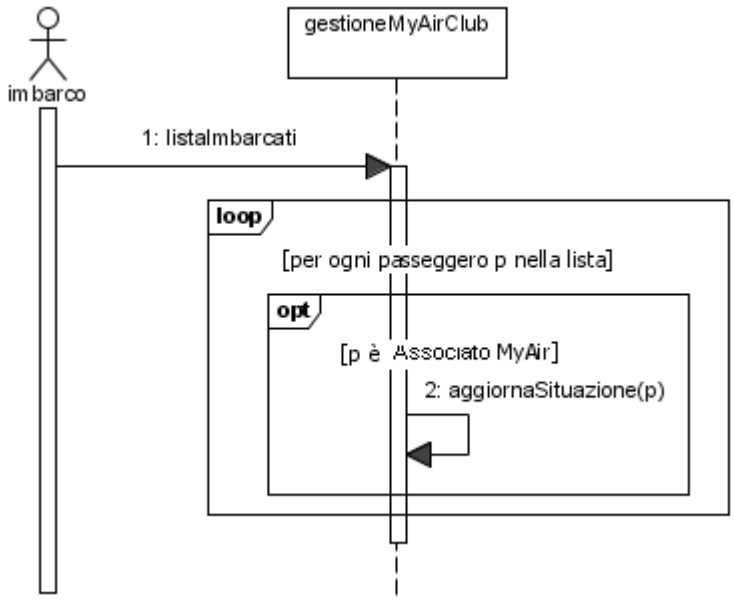
Domanda. (Analisi dei requisiti) Dare la narrativa del caso d'uso AccumulaPunti.

Risposta. Una possibile risposta è la seguente:

Nome del caso d'uso: AccumuloPunti
Breve descrizione: Il sistema riceve la lista dei passeggeri di un volo e la esamina, aggiornando di conseguenza la situazione degli associati del club MyAir.
Attore primario: Imbarco
Attori secondari: Nessuno
Precondizioni: Nessuna
Sequenza degli eventi principale: 1. Il sistema di Imbarco invia la lista con le informazioni sui passeggeri imbarcati al sistema di gestione ClubMyAir. 2. Per ogni passeggero: 2.1. Se il passeggero è associato del Club 2.1.1. La sua situazione viene aggiornata, aggiungendo le miglia accumulabili del volo
Postcondizioni: Volo inserito
Sequenze degli eventi alternative: Nessuna

Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che realizzi il caso d'uso AccumulaPunti, mostrando le interazioni tra il sistema e l'attore coinvolto.

Risposta. Una possibile soluzione è la seguente:



L'analisi architettonica ha portato a individuare le seguenti componenti

Componente	**Responsabilità**
GestoreClientiLatoServer	Realizza la parte server dei casi d'uso iniziati dai Clienti, e il passaggio dall'uno all'altro.
GestoreClientiLatoClienti	Realizza la parte client dei casi d'uso iniziati dai Clienti.

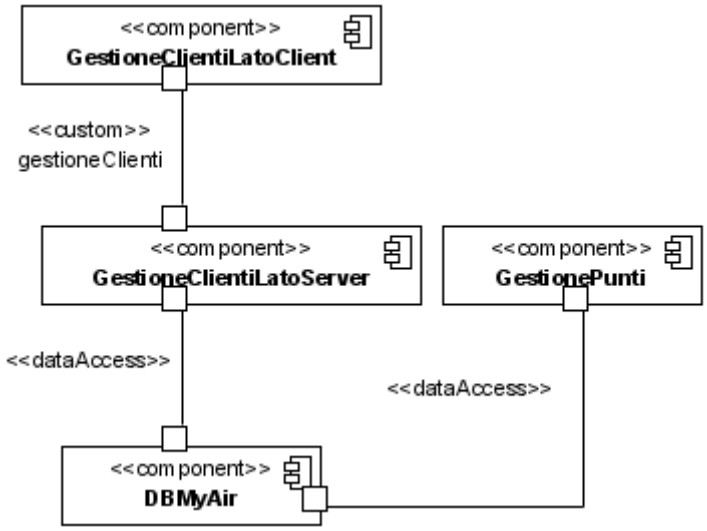
DBMyAir	Gestisce i dati degli associati del Club.
GestionePunti	Realizza i casi d'uso AccumuloPunti e AggiornamentoAnnuale.

e il seguente connettore <<custom>>

Connettore	**Responsabilità**
gestioneClienti	Permette gli scambi d'informazione tra i gestoriClienti.

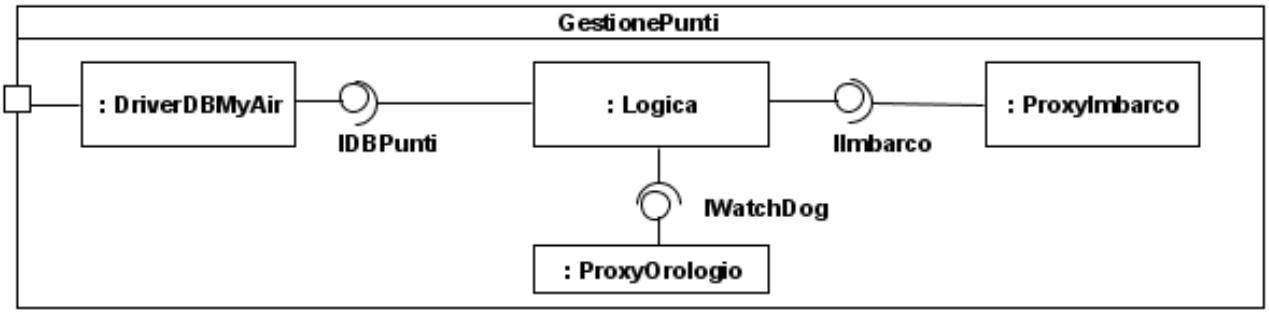
Domanda. (Architettura) Dare una vista C&C sull'architettura del sistema GestioneMyAirClub, utilizzando le componenti suddette, il connettore dato e eventuali connettori standard.

Risposta. Una possibile soluzione è data dalla figura seguente:



Domanda. (Progettazione di dettaglio) Dare, mediante un diagramma di struttura composita, la struttura della componente GestionePunti. Specificare le responsabilità delle parti introdotte, in una tabella.

Risposta. Una possibile soluzione è la seguente:



con le seguenti responsabilità

Parte	Responsabilità
DriverDBMyAir	Realizza l'interfaccia IDBPunti che permette a Logica di accedere tramite il solo porto della componente al DBMyAir.
ProxyImbarco	Realizza la connessione con il sistema Imbarco, passando alla Logica la lista degli imbarcati.
ProxyOrologio	Sveglia la logica alla data e ora richiesta tramite IWatchDog.
Logica	Realizza i casi d'uso, sfruttando le interfacce introdotte.

Si consideri il metodo stimaLivelli, con la seguente specifica: dato un vettore di associati del club MyAir, restituisce una tripla formata, rispettivamente, dal numero degli associati che, in base alla loro situazione: migliorano il proprio livello, non lo modificano, lo peggiorano. Si noti che i livelli standard, argento e oro sono codificati rispettivamente con 1, 2, e 3.

```
public Tripla stimaLivelli(Vector <AssociatoMyAir> associatiMyAir) {
   int inAumento = 0;
   int stabili = 0;
   int inCalo = 0;
   for (AssociatoMyAir m : associatiMyAir) { // itera su associatiMyAir
     int livello = m.getLivello();
     int miglia  = m.getMiglia();
      int differenza;
     if (miglia > 100000)       differenza = 3 - livello;
       else if (miglia > 15000) differenza = 2 - livello;
           else                 differenza = 1 - livello;
     if (differenza > 0)        inAumento++;
       else if (differenza==0)  stabili++;
           else                 inCalo++;
   }
   return new Tripla(inAumento, stabili, inCalo);
}
```

Domanda. (Verifica) Definire un insieme di coppie di attributi (livello, miglia) da attribuire agli elementi del vettore `associatiMyAir` che permetta di raggiungere una copertura del 100% del codice del metodo `stimaLivelli`, secondo il criterio dei comandi.

Risposta. Una soluzione minima è la seguente:

Livello	Miglia
2	120000
3	75000
1	10000

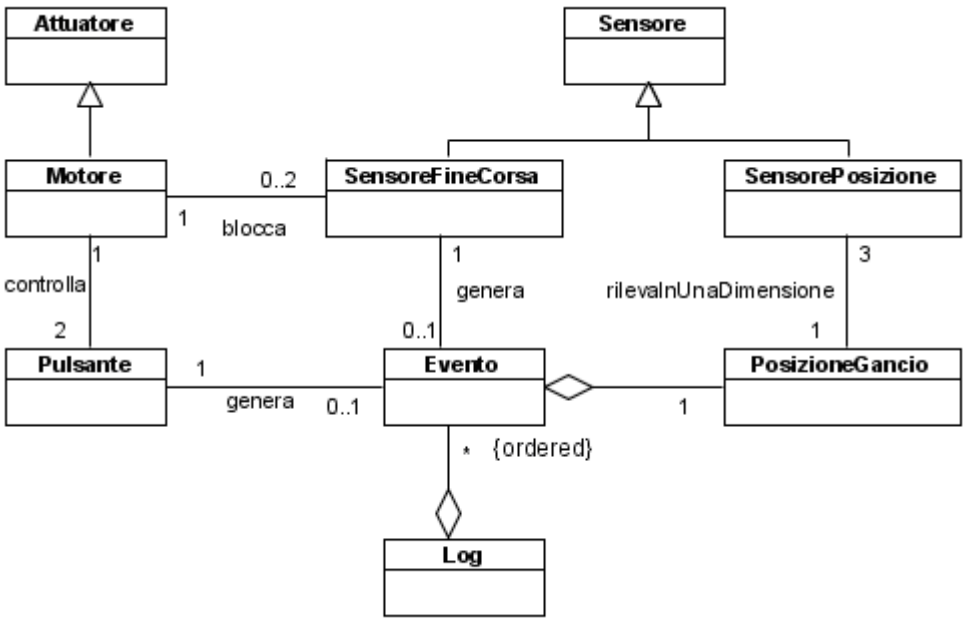
CAP 11. Gru

Spett. Ditta, In seguito alle richieste del vostro analista vi comunichiamo le seguenti informazioni.

Da oltre trent'anni la nostra società produce gru per l'edilizia e l'industria, mantenendo sempre elevati livelli tecnologici, nel rispetto degli standard di sicurezza. L'ultima di generazione di gru, denominate Hilg-6, sono in grado di spostare carichi fino a 25 tonnellate a un'altezza di 70 metri, con uno sbraccio massimo di 80 metri. In accordo agli standard tecnologici, le gru Hilg-6 prevedono tre movimenti, anche concomitanti, del gancio (circolare, orizzontale, verticale). I movimenti sono realizzati da **motori** elettrici da 110 KW. Oltre a questi **attuatori**, la gru è dotata di opportuni **sensori**: i **sensori di posizione** sono in grado di rilevare la **posizione del gancio** nello spazio 3D, i **sensori di fine corsa** segnalano alcuni **eventi** rilevanti ai fini della sicurezza. In particolare, controllano i movimenti in orizzontale e in verticale, e bloccano automaticamente il relativo motore. L'operatore può controllare la gru mediante una pulsantiera dotata di sei **pulsanti** (due per ogni movimento). La pressione e il rilascio di un pulsante generano eventi trasmessi agli attuatori mediante un collegamento wireless. Le norme di sicurezza richiedono la memorizzazione di tutti gli eventi in un **log** che li registra sequenzialmente rispetto al momento in cui accadono, indicando anche la posizione del gancio.
Cordiali saluti, Ing. Urg Crane

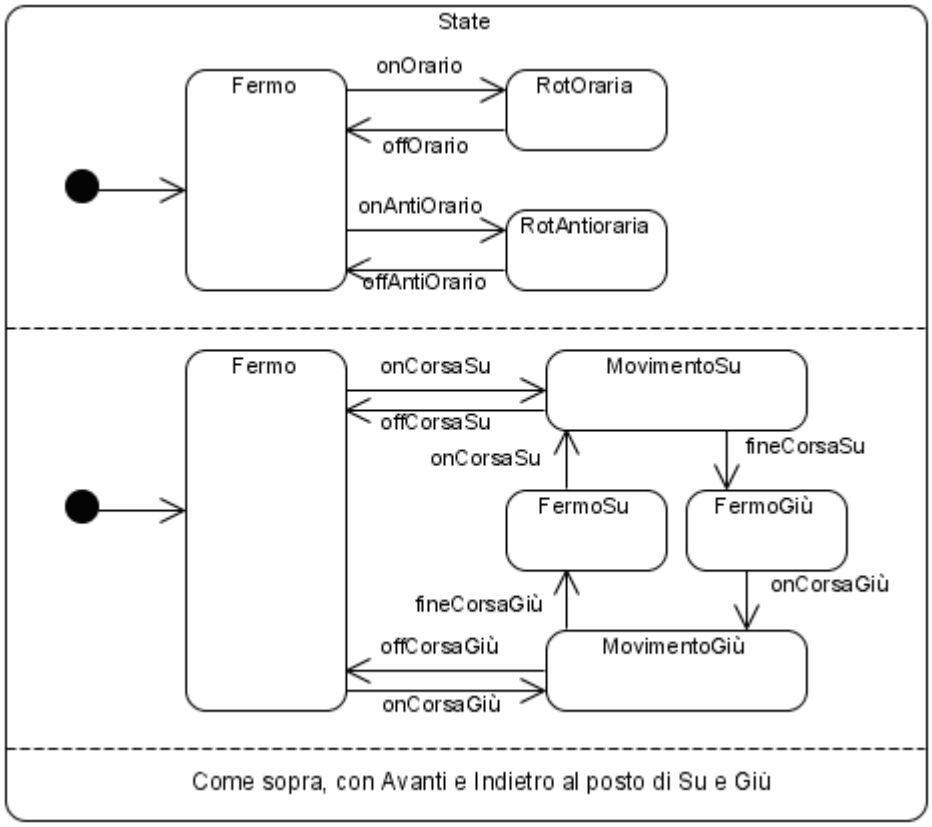
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

Risposta. Un possibile diagramma è il seguente.



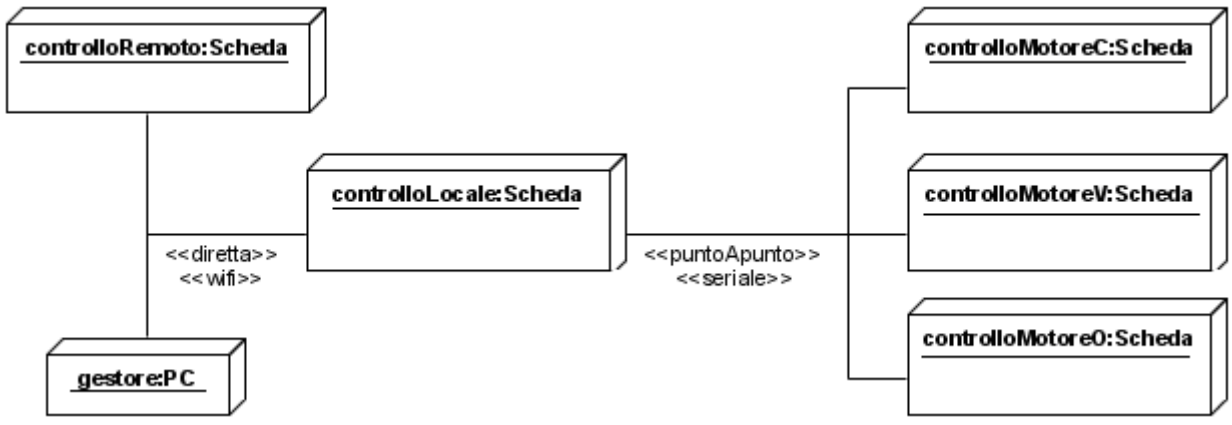
Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva l'evoluzione degli stati in cui si può trovare la gru.

Risposta. Un possibile diagramma è il seguente.



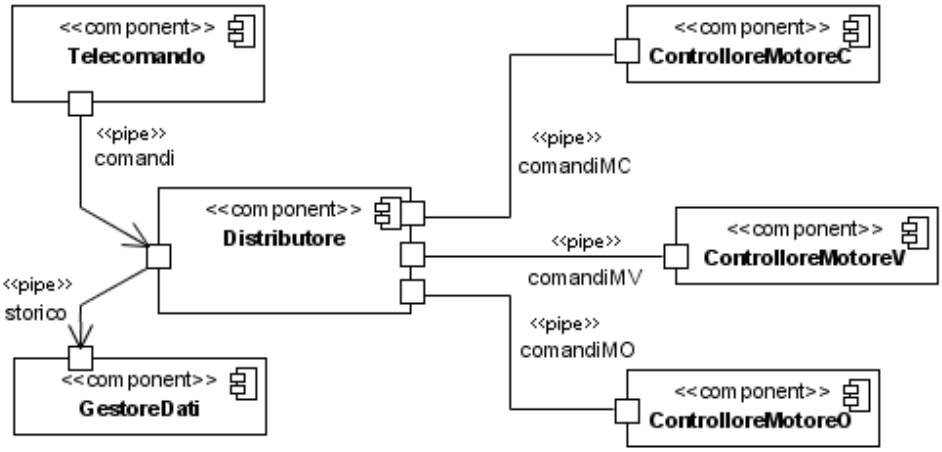
La struttura fisica del sistema è descritto dal seguente diagramma (in forma istanza), dove

1. i nodi *controlloMotoreX* sono Schede per la gestione dei tre motori (azionano, sotto controllo software, gli interruttori che danno e tolgono corrente ai motori e interfacciano i sensori di posizione e i sensori di fine corsa, se presenti),
2. il nodo *controlloRemoto* è una Scheda che interfaccia i pulsanti del telecomando e invia gli eventi che essi generano al
3. nodo *controlloLocale* che interfaccia, sulla gru, il trasduttore per la comunicazione senza fili e i cavi di collegamento alle schede di controllo dei motori, per la distribuzione dei segnali di controllo provenienti da controlloRemoto;
4. il nodo *gestore* è un PC destinato alla raccolta dei dati richiesti dalle norme di sicurezza.



Domanda. (Architettura) Dare una vista C&C sull'architettura del sistema ControlloGru, che possa essere dislocata in modo naturale (una componente per nodo) sui nodi sopra descritti. Si documenti il diagramma con una tabella che elenchi le responsabilità della componente dislocata sul nodo *controlloLocale* e dei connettori ad essa collegati (per questo, oltre a stereotipare opportunamente i connettori, si attribuisca loro anche un nome).

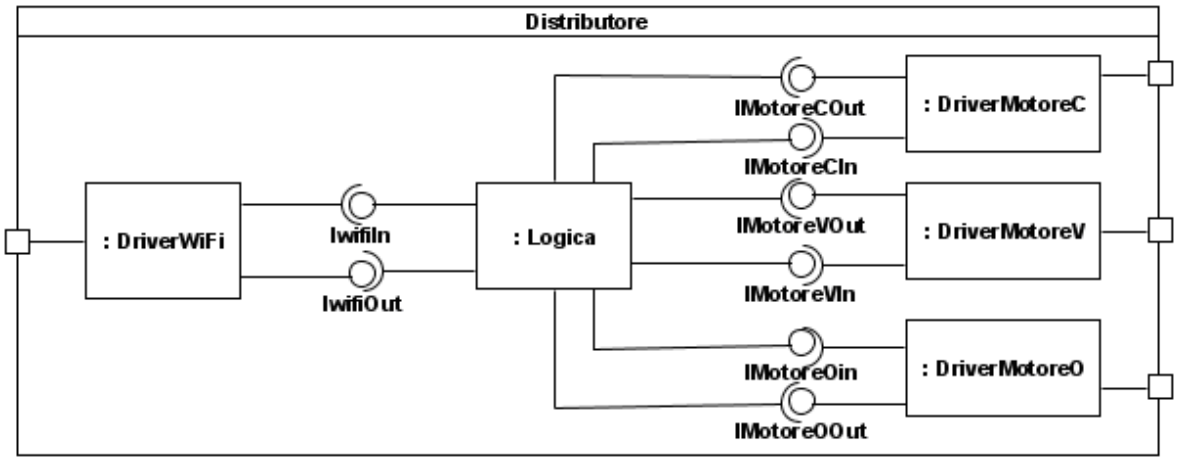
Risposta. Una possibile soluzione è data dalla figura seguente:



Le responsabilità sono:

Componente	**Responsabilità**
Distributore	Riceve i comandi dalla componente Telecomando e li distribuisce alle componenti di controllo motore; Riceve tutti gli eventi rilevanti ai fini della sicurezza, e la posizione del gancio relativa ad ogni evento; Invia a GestoreDati le coppie <evento, posizione>
Connettore	
Comandi	Permette l'invio dei comandi dal Telecomando al Distributore
comandiMX	In un verso permette al distributore l'invio dei comandi, nell'altro l'invio degli eventi di fine corsa e della posizione del gancio, nella direzione X.
Storico	Permette l'invio delle informazioni da registrare nel Log.

Domanda. (Progettazione di dettaglio) Dare, mediante un diagramma di struttura composita, la struttura della componente *Distributore*. Specificare le responsabilità delle parti introdotte.
Risposta. Una possibile soluzione è la seguente:



con le seguenti responsabilità:

Parte	Responsabilità
Logica	Riceve gli eventi dal DriverWiFi e li distribuisce ai DriverMotore, ricevendo la posizione corrente del gancio. Riceve gli eventi di fine corsa con la relativa posizione del gancio. Memorizza gli eventi di interesse per la sicurezza, e li trasmette su richiesta al gestore.
DriverWiFi	Realizza l'interfaccia con il trasduttore per la comunicazione senza fili. Trasforma i messaggi per la Logica in chiamate di metodi, e viceversa.
DriverMotoreX	Trasforma le richieste della Logica in messaggi per il MotoreX, e viceversa.

Il carico massimo che può essere sollevato da una gru dipende dall'estensione in orizzontale del gancio. Il valore del carico massimo, determinato da una tabella di carico specifica per ogni gru, è costante per un certo intervallo e poi diminuisce progressivamente. Nel caso della gru con un braccio da 80 metri, il valore del carico (in tonnellate) è specificato dalla seguente formula che fa riferimento a una tabella di coefficienti da usare in base al valore dell'estensione. Per un'estensione minore o uguale a 15 metri il carico massimo della gru è pari a 25 tonnellate, altrimenti:

$$Carico = estensione \times (C - D \times (estensione - S) / 5)$$

dove i coefficienti sono definiti dalla seguente tabella:

Intervallo	S	C	D
(15 - 20]	15	1,5	0,80
(20 - 30]	20	0,7	0,20
(30 - 60]	30	0,4	0,05
(60 - 80]	60	0,1	0,01

Domanda. (Verifica) Definire alcuni casi di test per un metodo che determini il carico massimo per una data estensione orizzontale. Si giustifichi la scelta dei casi introdotti, e si considerino solo valori ammissibili.

N.B. Non è richiesto che tutti i valori attesi vengano calcolati: ne bastano un paio.

Risposta. Una possibile soluzione, in cui per ogni intervallo si è preso un valore minimo, uno tipico e uno massimo, è la seguente (per completezza sono stati calcolati tutti i valori attesi, in base alla specifica data sopra).

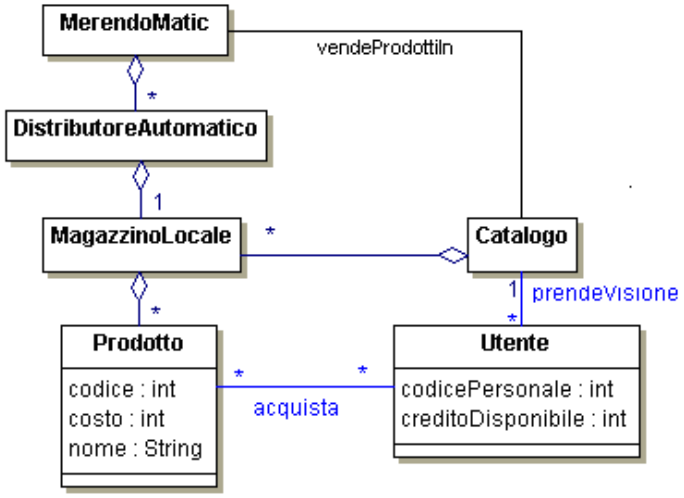
Estensione	Carico massimo
3	25
15,1	22,4084
20,1	13,9896
30,1	12,0099
60,1	5,99798
10	25
17	20,06
25	12,5

Estensione	Carico massimo
45	11,25
70	5,6
15	25
20	14
30	9
60	6
80	4,8

CAP 12. Merendo-matic

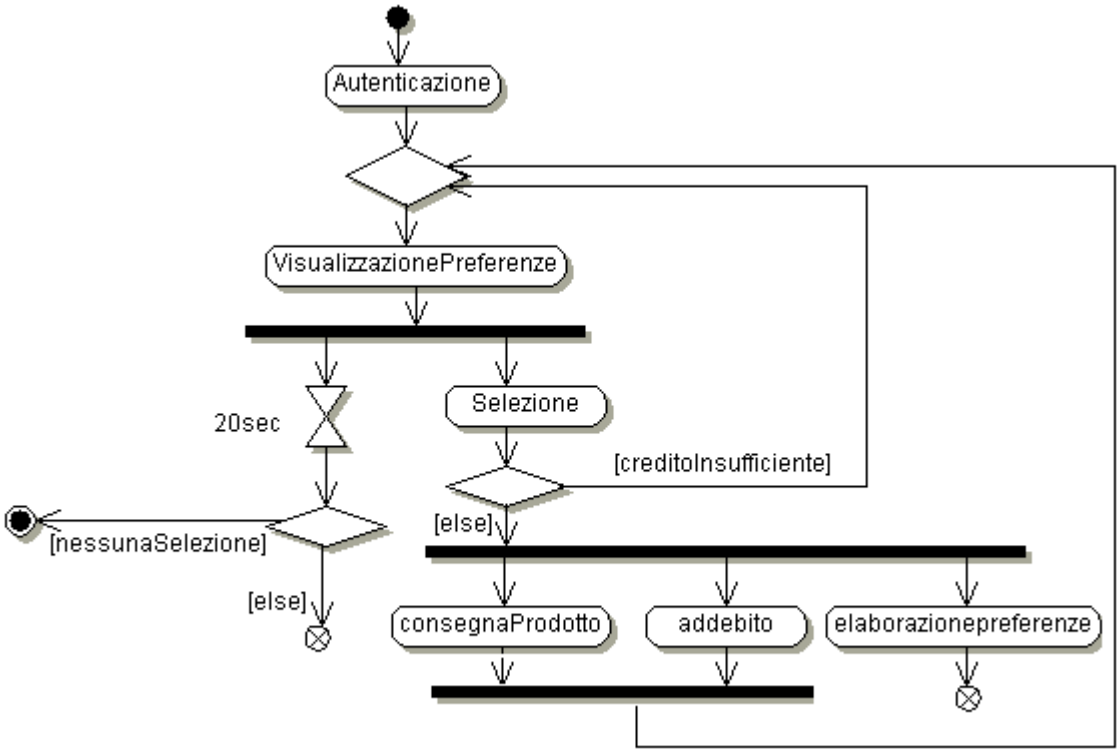
Spett. Ditta, In seguito alle richieste del vostro analista vi comunichiamo le seguenti informazioni.
L'ampia diffusione dei distributori automatici di bibite e merendine ha messo in evidenza alcuni problemi legati alla gestione dell'utenza, alle procedure di costituzione delle scorte, alle informazioni relative alla disponibilità dei prodotti. Il nuovo sistema, provvisoriamente chiamato Merendo-matic, intende risolvere questi problemi offrendo maggiori servizi all'utenza e un miglioramento delle procedure di gestione. **Merendo-matic** è un sistema formato da un gruppo di **distributori automatici**, installati all'interno di un'area, di solito un edificio, collegati tra loro in modo da offrire un servizio integrato. Gli **utenti** di Merendo-matic possono accedere uno qualsiasi dei distributori mediante un **codice personale** che permette loro di usufruire del **credito disponibile** per l'acquisto dei **prodotti**. Le novità del sistema sono costituite dalla elaborazione delle preferenze degli utenti sulla base dei loro acquisti e dalla possibilità che ha l'utente di prendere visione, da un qualsiasi distributore, oltre che dei prodotti disponibili nel **magazzino locale** del distributore, anche di altri prodotti del **catalogo** dei prodotti in vendita su Merendo-matic. In particolare, si visualizzano, usando le informazioni sulle preferenze dell'utente, i primi tre prodotti che non sono disponibili in locale ma sono disponibili su altri distributori.
L'utente interessato a un prodotto non disponibile localmente, perché temporaneamente esaurito, può sapere tempestivamente quale distributore può soddisfare la sua richiesta.
La procedura di acquisto è molto semplice: ogni prodotto è caratterizzato da un **nome** (evocativo del prodotto stesso), da un **codice** (di solito una sequenza di cifre numeriche) e da un **costo**; dopo essersi autenticato mediante il codice personale, l'utente può scegliere uno o più prodotti tra quelli presenti nel magazzino locale e acquistarli sulla base del credito disponibile. Tale credito è automaticamente aggiornato al termine della procedura di acquisto.
Tra le funzioni a disposizione del gestore di Merendo-matic, ricordiamo la possibilità di gestire i magazzini locali ed eventualmente di costituire un magazzino centralizzato per le scorte, e le funzioni standard per aggiungere e rimuovere utenti, prodotti, distributori. Altre funzioni, tra cui citiamo la possibilità di collegarsi a sistemi esterni per la gestione del personale, sono disponibili a richiesta del cliente.

Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.
Risposta. Una possibile soluzione è la seguente:

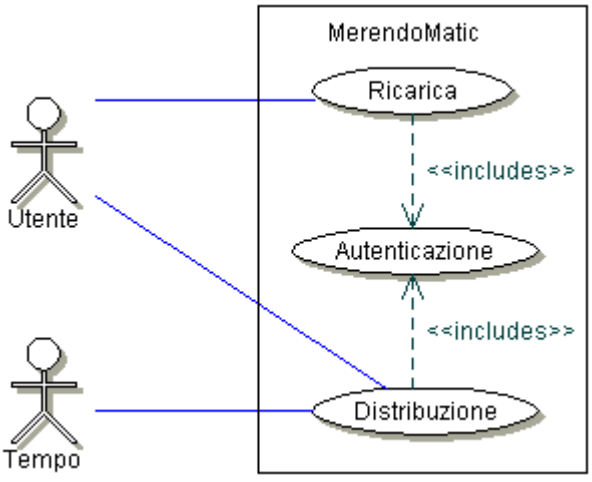


Domanda. (Analisi del dominio) Dare un diagramma di attività che descriva il processo che, a partire dall'autenticazione, porta alla consegna di uno o più prodotti ad un utente. Si richiede di descrivere un time-out: se entro 20 secondi dalla visualizzazione dei prodotti non ne viene selezionato alcuno, l'attività termina. Se possibile, il meccanismo di time-out non deve introdurre ritardi nella consegna.

Risposta. Una possibile soluzione è la seguente:



L'analisi ha portato all'individuazione delle funzionalità minime che il primo rilascio del sistema dovrà realizzare. Queste funzionalità sono documentate dal seguente diagramma, cui seguono le brevi descrizioni dei casi d'uso.



Autenticazione. L'utente viene riconosciuto dal sistema.

Ricarica. L'utente si autentica e ricarica il suo credito.

Distribuzione. L'utente si autentica; vengono visualizzati i prodotti; l'utente acquista uno o più prodotti.

Domanda. (Analisi dei requisiti) Dare la narrativa del caso d'uso Distribuzione.

Risposta.

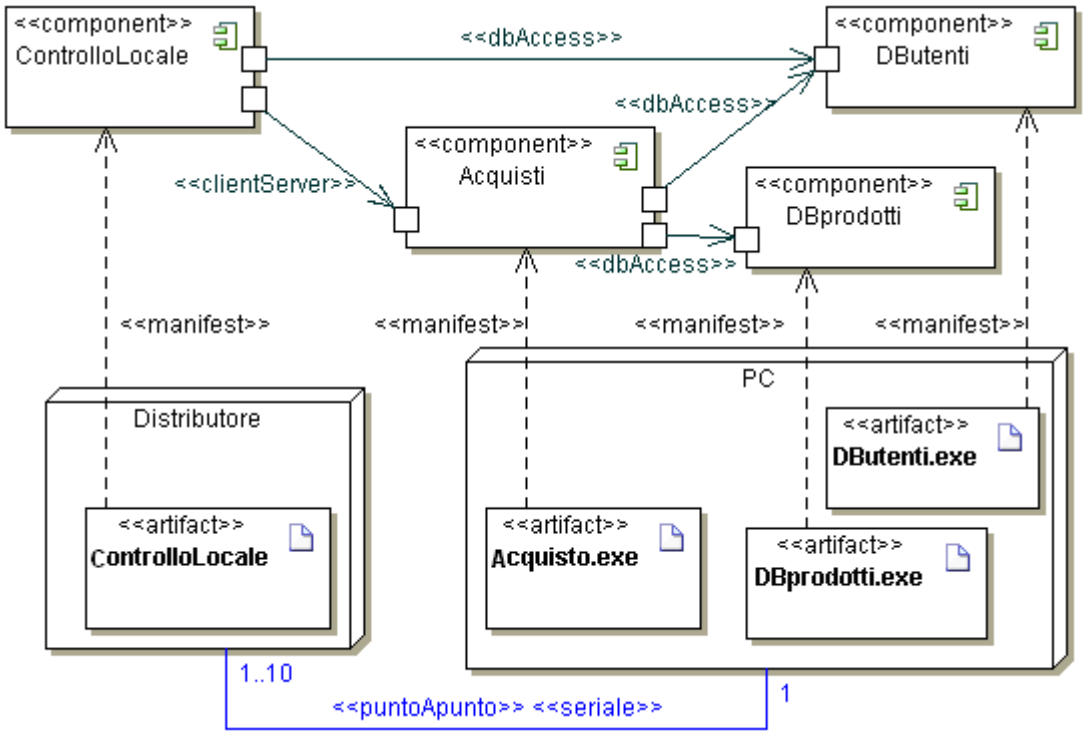
Caso d'uso Distribuzione	
Breve Descrizione	L'utente si autentica e seleziona uno o più prodotti disponibili in locale. Il costo è scalato dal credito.
Attori Primari	Utente.
Attori Secondari	Tempo.
Precondizioni	Nessuna.
Sequenza principale degli eventi	1. Include Autenticazione. 2. for (p in lista preferenze) 2.1. se (prodotti remoti visualizzati < 3 and p disponibile solo in remoto) 2.1.1. visualizza p 3. while (prodotto selezionato prima del time-out) 3.1. aggiorna credito disponibile 3.2. consegna prodotto 3.3. aggiorna catalogo 3.4. elabora preferenze
Postcondizioni	Utente servito. Catalogo, credito disponibile e lista preferenze aggiornati
Sequenza degli eventi alternativa	1. Autenticazione fallita. 2. Credito insufficiente.

La struttura fisica del sistema prevede un nodo centrale, *PCmerendo-matic*, e un numero di *Distributori* inferiore a 10. Per realizzare i casi d'uso considerati, sono state individuate 4 componenti, con le seguenti responsabilità:

Componente	**Responsabilità**
ControlloLocale	Gestisce l'interfaccia con l'utente, e gli attuatori per la consegna dei prodotti. Implementa l'autenticazione e la ricarica interagendo col DButenti. In caso di acquisto si interfaccia con la componente Acquisti.
Acquisti	Interroga il DBprodotti e il DButenti, per restituire i primi tre prodotti preferiti dall'utente e disponibili solo in remoto. In caso di acquisto, informa DBprodotti e DButenti per gli aggiornamenti necessari.
DBprodotti	Mantiene il catalogo dei prodotti, aggiornando il numero di pezzi disponibili in ciascun distributore.
DButenti	Mantiene le informazioni sugli utenti, aggiorna il loro credito disponibile e le preferenze.

Domanda. (Architettura) Dare una vista ibrida di dislocazione e C&C sull'architettura del sistema Merendo-matic.

Risposta. Una possibile soluzione è data dalla figura seguente:



Si consideri la seguente definizione (incompleta) di classe:

```
public class Preferenze {
     // OVERVIEW: Questa classe fornisce una procedura per determinare i
     // primi tre prodotti preferiti da un utente, disponibili in remoto.
 private static boolean disponibileLocale(Integer p) {
     // EFFECTS: restituisce true sse il prodotto di codice p è disponibile
     // nel magazzino locale.
 ... }
 private static boolean disponibileRemota(Integer p) {
     // EFFECTS: restituisce true sse il prodotto di codice p è disponibile
     // in uno dei magazzini remoti.
 ... }
 public static void configura(DistributoreAutomatico d) {
     // EFFECTS: configura la classe in modo che le due procedure precedenti
     // assumano d come distributore locale.
 ... }
 public static Vector<Integer> preferenzaRemota(Vector<Integer>
 prefUtente) {
     // ASSUMES: prefUtente elenca i prodotti preferiti da un utente in
     // ordine decrescente di preferenza.
     // EFFECTS: restituisce un vettore con al più tre prodotti tra quelli
     // in prefUtente, in ordine decrescente di preferenza, non disponibili
     // localmente, ma disponibili in uno dei magazzini remoti.
   Vector<Integer> pref3 = new Vector<Integer>();
   int i = 0;
    while ((i < prefUtente.size()) && (pref3.size() < 3)) {
      Integer prodotto = prefUtente.elementAt(i);
```

```
      if (!disponibileLocale(prodotto) && disponibileRemota(prodotto)) {
            pref3.addElement(prodotto);
      }
      i++;
    }
    return pref3;
 }
}
```

Domanda. (Verifica) Si vuole effettuare una verifica a scatola aperta della procedura `preferenzaRemota`, per raggiungere la copertura del 100% secondo il criterio delle condizioni (condition coverage). Assumendo che ci siano 15 prodotti diversi, codificati da 1 a 15, e che l'ambiente di prova definisca come disponibili

1. nel magazzino locale i prodotti 1,2,3 e 4,
2. nei magazzini remoti i prodotti 11,12,13 e 14,

si diano *due* casi di prova che permettano di raggiungere la copertura richiesta.

Facoltativo. Sarebbe possibile ottenere l'obiettivo con un solo caso?

Risposta. Non è possibile avere un solo caso, perchè per esercitare nel vero e nel falso la seconda clausola nella condizione del ciclo, bisogna mantenere vera la prima, e una volta falsificata la seconda il ciclo termina, e non c'è più modo di falsificare la prima. Conviene quindi falsificare la prima in un caso a parte – tanto vale prendere il caso vuoto -, e trattare tutte le altre possibilità, in un altro caso, con sufficienti elementi in prefUtente. Le clausole della condizione dell'if, infatti, possono essere controllate in una sola iterazione.

Quindi, una possibile soluzione è la seguente:

Input (`prefUtente`)	Output
[]	[]
[1,11,12,5,13,2]	[11,12,13]

Si noti che l'ultimo elemento non viene considerato, ma è necessario per raggiungere la copertura del 100%.

Nota. Alcuni autori danno una definizione del criterio delle condizioni più completo di quello usato qui, che prevede che ciascuna clausola in una condizione sia controllata per i due possibili valori. Ad esempio, Fuggetta et al. (pag. 184) prevede di considerare tutte le possibili combinazioni di valori per le clausole in una condizione. In questo caso, non è possibile ottenere la copertura richiesta con l'ambiente a disposizione (manca il caso di un prodotto disponibile sia localmente che in remoto), e comunque non bastano due casi. Infatti, per esercitare tutte le combinazioni nella guardia del while, occorre coprire i seguenti 4 casi:

	`prefUtente`	`(i < prefUtente.size())`	`(pref3.size() < 3)`
1	Non vuoto	T	T
2	Con più di tre elementi, di cui *almeno* tre disponibili solo in remoto	T	F
3	Con *meno* di tre elementi disponibili solo in remoto	F	T
4	Con *tre* elementi disponibili in remoto, di cui l'ultimo è anche l'ultimo di prefUtente	F	F

Inoltre, per la condizione dell'if, basta considerare: prodotti disponibili solo in locale (caso 5), solo in remoto (caso 6), sia in locale che in remoto (caso 7), e totalmente indisponibili (caso 8).

Modificando l'ambiente, e assumendo:

1. nel magazzino locale i prodotti 1, 2, 3, 4 e 10,
2. nei magazzini remoti i prodotti 10, 11, 12, 13 e 14,

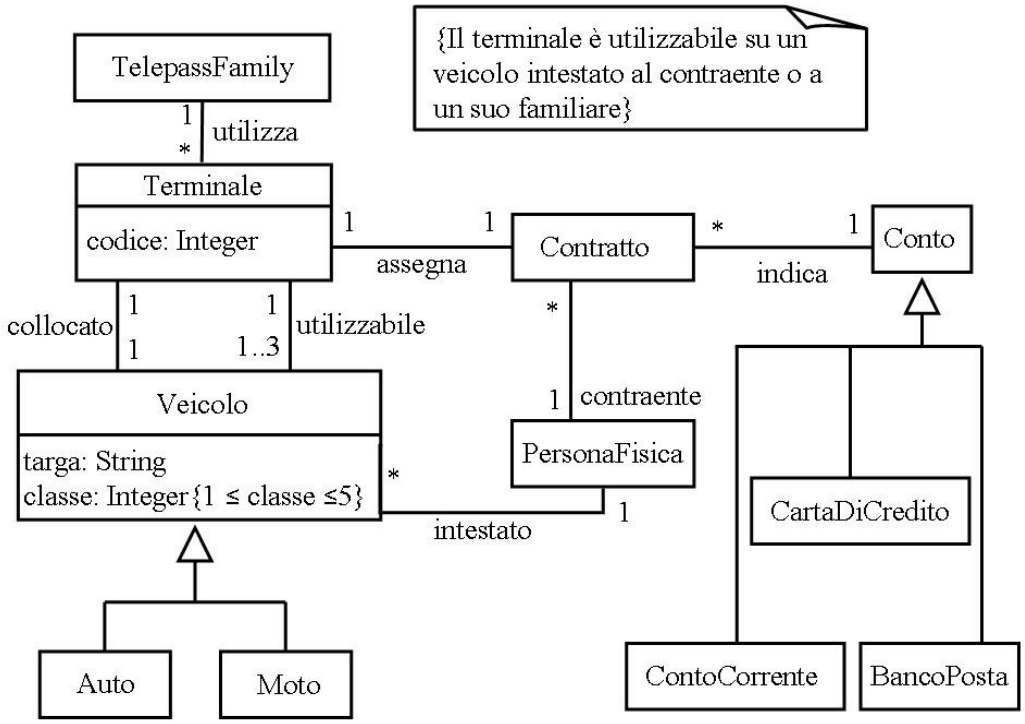
una possibile soluzione è:

Input (`prefUtente`)	Output	Casi
[]	[]	3
[1,11,12,5,13,2]	[11,12,13]	1,2,5,6,8
[10,11,12,13]	[11,12,13]	1,4,7

CAP 13. Telepass

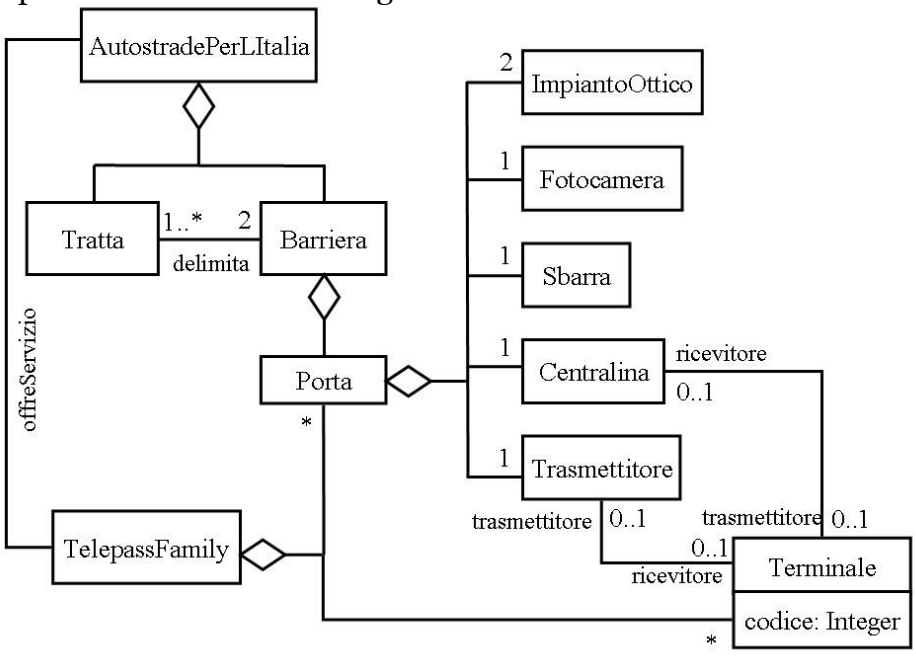
Porta la tua famiglia in giro per l'Italia senza fermarti mai: oggi basta un bip per godere della libertà di viaggiare, senza costi aggiuntivi sul pedaggio.
Telepass Family è un servizio che ti consente, attraverso l'utilizzo di un piccolo **terminale** collocato sul parabrezza della tua **auto** o posizionato sulla tua **moto**, di transitare nelle porte dedicate Telepass di Autostrade per l'Italia e di pagare il pedaggio per la tratta percorsa senza fermarti alla barriera. Il servizio è dedicato alle "**persone fisiche**", e non a ditte. Il terminale Telepass, assegnato a un **contraente** tramite un **contratto** di forma privata che definisce la forma di pagamento, può essere utilizzato su un massimo di tre **veicoli**, intestati al contraente o a suoi familiari, le cui **targhe** devono essere comunicate ad Autostrade per l'Italia. I pedaggi vengono addebitati periodicamente su un **conto**, che può essere un **conto corrente**, un **BancoPosta** o una **carta di credito** abilitata.
Il funzionamento pratico è relativamente semplice: quando un veicolo si avvicina a una porta, un impianto ottico (CTV) riconosce il tipo del veicolo, classificandolo in una delle cinque **classi** tariffarie e attiva l'emissione, da parte della porta, del segnale di richiesta di identificazione. Il terminale risponde al segnale del trasmettitore a terra, ritrasmettendo un **codice** identificativo univoco. Trascorso mezzo secondo, la centralina a terra dà ordine di alzare la sbarra, Se, nel frattempo, ha ricevuto dal terminale un codice corretto, registra il passaggio regolare. Altrimenti, una speciale fotocamera fotografa la targa, per poter risalire all'autore del transito irregolare. In ogni caso, un secondo impianto ottico provvede alla conferma della classificazione del tipo di veicolo.

Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi, attributi o ruoli tutti e soli i termini che nell'enunciato compaiono in grassetto.
Risposta.



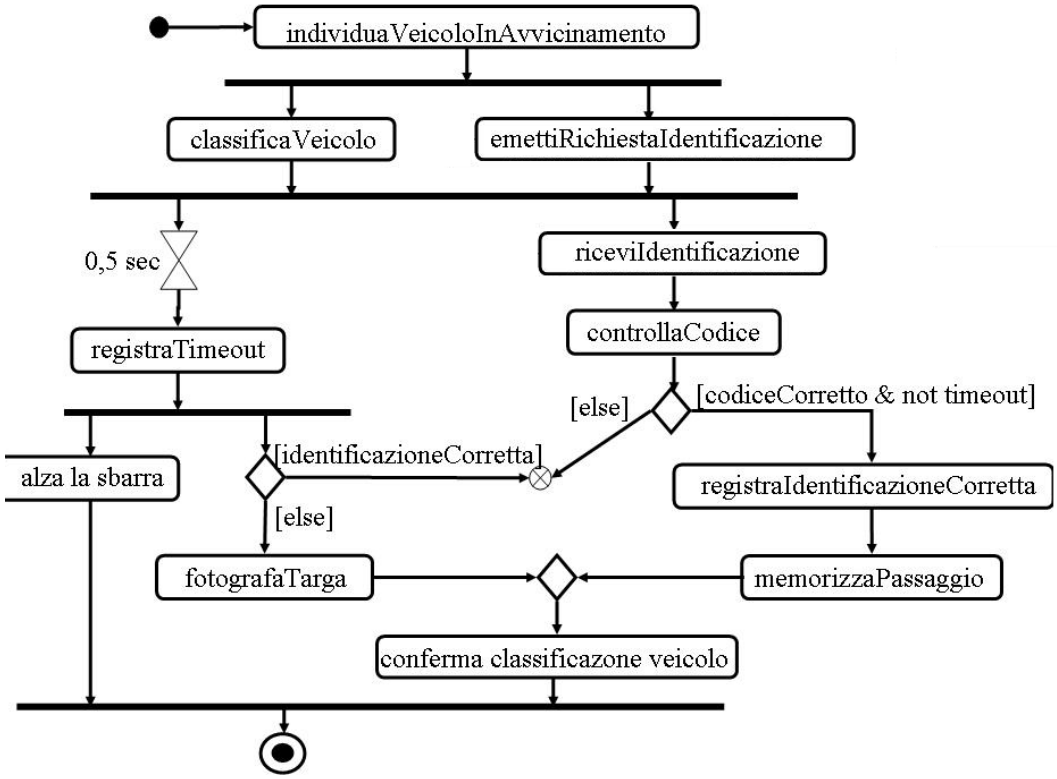
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono sottolineati.

Risposta. Una possibile soluzione è la seguente.



Domanda. (Analisi del dominio) Dare un diagramma di attività che descriva il comportamento della centralina a terra durante il transito di un veicolo a una porta.

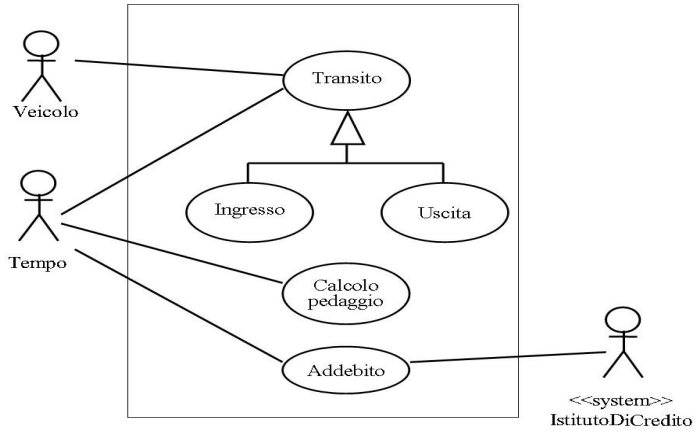
Risposta.



L'analisi ha portato all'individuazione delle funzionalità minime che il primo rilascio del sistema dovrà realizzare: gestione dei transiti in ingresso e in uscita, calcolo del pedaggio e addebito. Si assume che il pedaggio sia calcolato off–line, magari di notte o in momenti di minor traffico. L'addebito avviene bi–mensilmente.

Domanda. (Analisi dei requisiti) Dare un diagramma dei casi d'uso, con una loro breve descrizione.

Risposta.



Transito. Il sistema riconosce il veicolo in transito, e ne registra il passaggio.
Ingresso. Si comporta come Transito, e registra il fatto che il passaggio è in ingresso.
Uscita. Si comporta come Transito, e registra il fatto che il passaggio è in uscita.
CalcoloPedaggio. Calcola il pedaggio.
Addebito. I pedaggi vengono addebitati.

Domanda. (Analisi dei requisiti, Telepass) Dare la narrativa del caso d'uso Transito.

Risposta. Una possibile soluzione è la seguente.

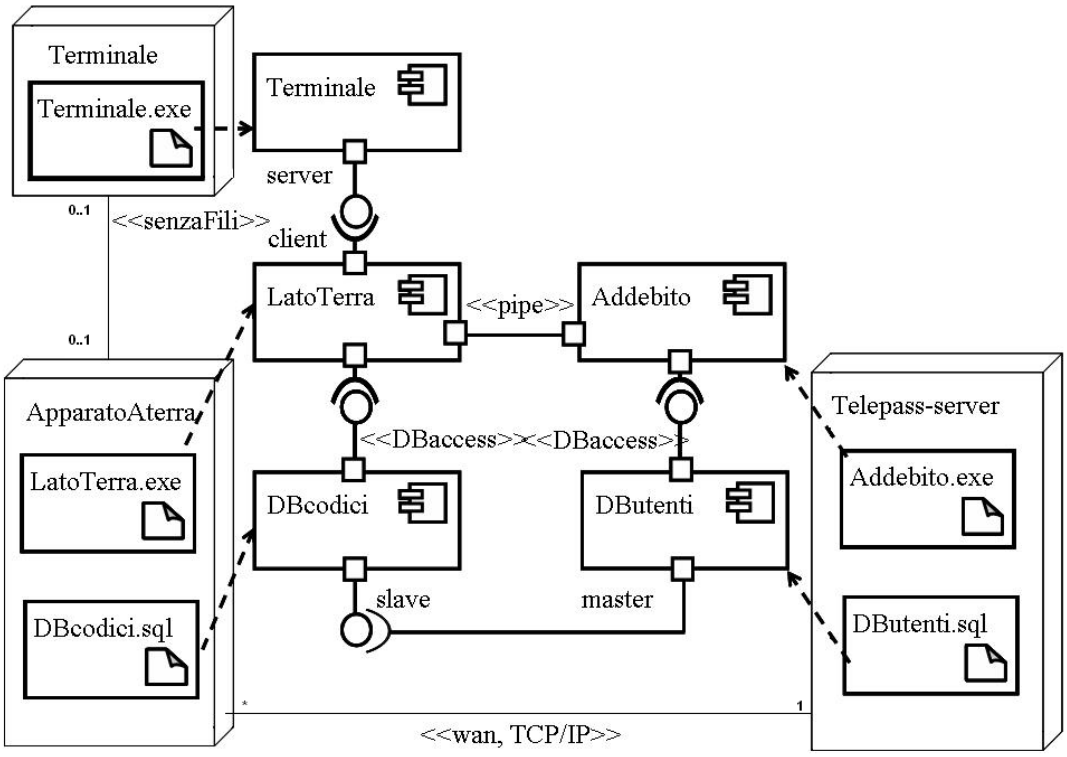
Caso d'uso Transito	
Breve Descrizione	Il sistema riconosce il veicolo in transito, e ne registra il passaggio.
Attori Primari	Veicolo.
Attori Secondari	Tempo.
Precondizioni	Nessuna.
Sequenza principale degli eventi	1. Il caso d'uso inizia quando il veicolo si avvicina alla porta; 2. l'impianto ottico riconosce e classifica il veicolo; 3. la porta emette il segnale; 4. il terminale emette il codice del veicolo 5. il sistema registra il passaggio regolare; 6. **if** (timeout non scaduto) 6.1. aspetta timeout; 7. alza la sbarra; 8. il secondo impianto ottico conferma la classificazione del tipo di veicolo.
Postcondizioni	Transito corretto registrato
Sequenza degli eventi alternativa	1. TransitoConCodiceIllegale 2. TransitoConTerminaleGuasto 3. TransitoDiVeicoloSenzaTelepass

La struttura fisica del sistema prevede un unico nodo centrale, *Telepass-server*, i *Terminali* sui veicoli, e un numero di *Apparati a terra*, uno per porta. Sono state individuate cinque componenti, con le seguenti responsabilità:

Componente	Responsabilità
Terminale	Gestisce il lato veicolo del controllo dei transiti attraverso la porta Telepass.
LatoTerra	Gestisce il lato terra del controllo dei transiti attraverso la porta Telepass.
Addebito	Mantiene informazione sui transiti. Mantiene le informazioni sulla struttura della rete autostradale. Calcola i pedaggi e li memorizza nel DButenti. Gestisce l'addebito.
DButenti	Mantiene le informazioni sugli utenti, tra cui i codici identificativi dei terminali loro assegnati.
DBcodici	Mantiene i codici identificativi dei terminali regolarmente in uso.

Domanda. (Architettura) Dare una vista ibrida di dislocazione e C&C sull'architettura del sistema Telepass Family.

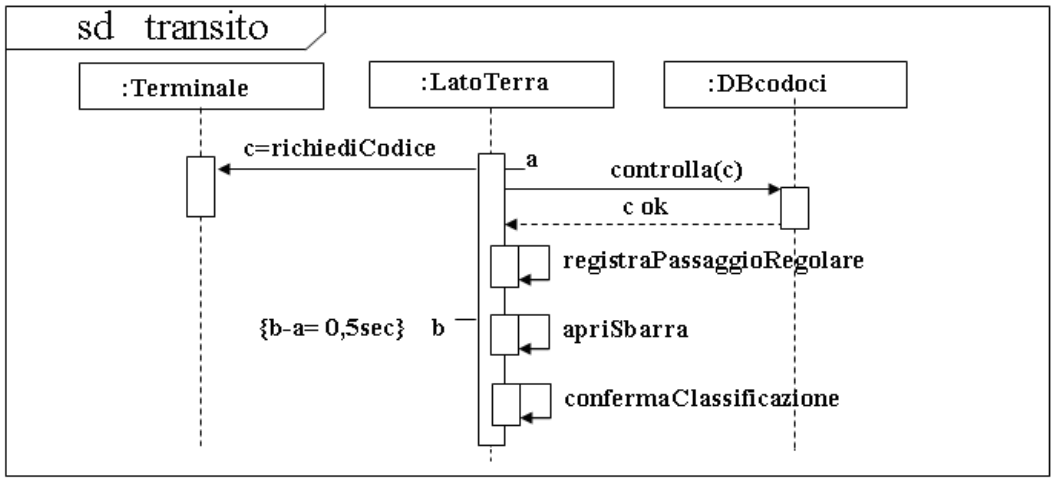
Risposta. Visti i tempi brevi con cui deve essere controllato il codice trasmesso dal terminale, la componente DBcodici deve essere replicata su tutti gli apparati a terra. La soluzione è realizzabile in quanto si può prevedere che le repliche vengano aggiornate solo periodicamente una volta al giorno. Inoltre, richiediamo che i codici identificativi siano contenuti in una tabella del DB utenti, che il DBcodici replichi questa tabella.



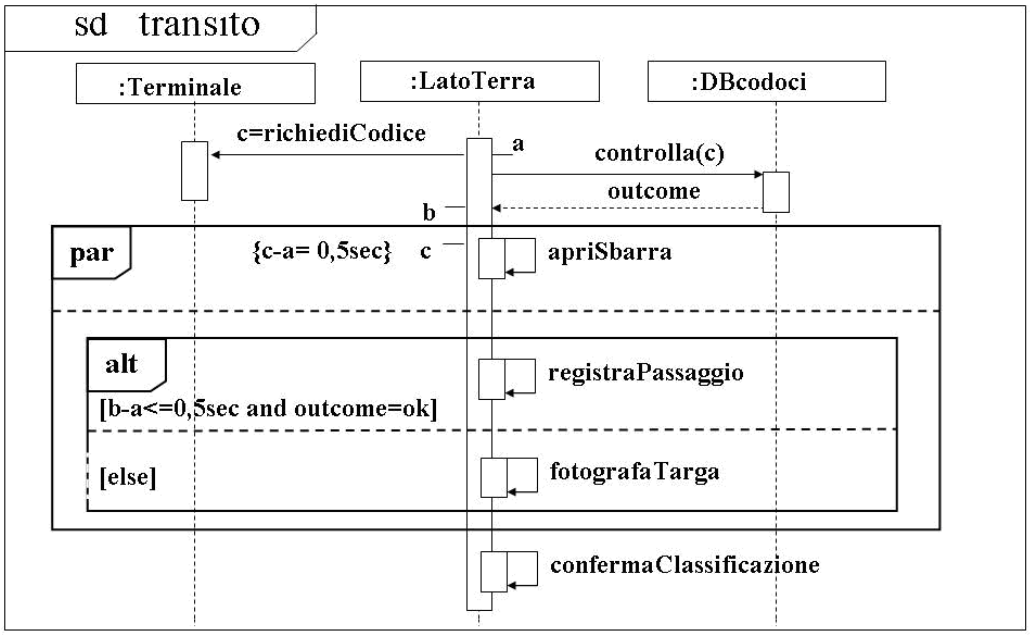
Il master–slave è un noto stile comportamentale in cui un componente (master) ha un controllo unidirezionale su uno o più componenti (slave).
Le dipendenze sono tutte di tipo <<manifest>>.

Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che descriva le interazioni tra le componenti coinvolte nella realizzazione del caso d'uso Transito.

Risposta. Una possibile soluzione è la seguente.



Una soluzione più articolata è la seguente, che tiene conto anche del comportamento relativo alle sequenze alternative:

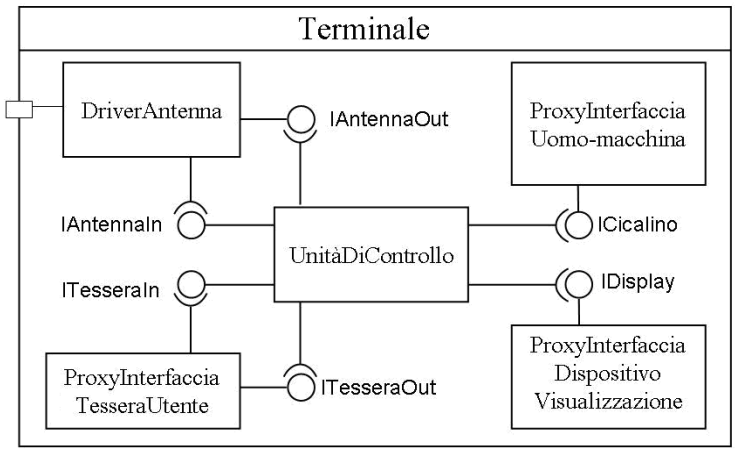


Il componente Terminale può essere scomposto in un insieme di parti con le responsabilità descritte nella tabella seguente.

Unità di controllo	Realizza la logica del terminale per gestire i le unità periferiche.
Proxy interfaccia tessera utente	Gestisce una presa a contatti che consente il colloquio con un'eventuale carta a microchip sia in lettura che in scrittura.
Proxy interfaccia uomo-macchina	Gestisce il dispositivo acustico (cicalino) che informa l'utente sull'andamento della transazione.
Driver antenna	Gestisce l'antenna per le trasmissioni da e per la centralina a terra.
Proxy interfaccia dispositivo di visualizzazione	Gestisce la linea seriale che consente di connettere il terminale di bordo e un display per visualizzare informazioni applicative.

Domanda. (Progettazione di dettaglio, Telepass) Dare, mediante un diagramma di struttura composita, la struttura della componente *Terminale*.

Risposta.



Il pedaggio si calcola considerando: la tariffa unitaria a chilometro, il tipo di veicolo utilizzato (5 classi), le caratteristiche dei tratti autostradali percorsi (di pianura o di montagna).

Si supponga il calcolo sia fatto usando i metodi così specificati:

```
     /*dati i caselli di ingresso e di uscita, restituisce il numero di km
     di pianura e il numero di quelli di montagna*/
int[ ] calcolaChilometri(a String, b String)
     /*dati i caselli di ingresso e di uscita e la classe del veicolo,
     ottiene il numero di Km percorsi e calcola il pedaggio */
double calcolaPedaggio(a String, b String, c Classe)
```

Domanda. (Verifica) Per quale dei due metodi dati sopra potrebbe essere utile creare uno stub, nella verifica del calcolo del pedaggio? Definire un semplice stub, che permetta la ripetibilità dei test, e non sia banale (vari i risultati in funzione degli argomenti).

Risposta. Notando che calcolaPedaggio deve invocare calcolaChilometri, e che la realizzazione di questo metodo richiede l'accesso al DBrete, conviene testare calcolaPedaggio con uno stub per calcolaChilometri. Per permettere la ripetibilità dei test non si può usare un generatore di numeri pseudo-casuali. La soluzione che segue conserva la proprietà commutativa di calcolaChilometri (andando da A a B si fanno gli stessi chilometri che andando da B a A) e può produrre anche risultati estremi (tratto di montagna o di pianura lungo zero).:

```
int[ ] calcolaChilometri(a String, b String){
     int[] coppia = {0,0};
     int mx = max(a.length(),b.length());
     int mn = min(a.length(),b.length());
     coppia[0]= mx – mn ;
     coppia[1]= (2*mn >= mx ? 2*mn–mx : mn) ;
     return coppia;}
```

Si supponga ora che il calcolo del pedaggio sia fatto usando il metodo così specificato:

```
/*dati i il numero di km di pianura p, il numero di km di montagna m e
la classe del veicolo c, calcola il pedaggio */


double calcolaPedaggio(p int, m int, c int) {
double pedaggio = 0.0;
     if (m <= SOGLIA_MONTAGNA) {
           pedaggio = (p + m) * c * COSTO_UNITARIO;
      } else {
           pedaggio = (p + 2*m) * c * COSTO_UNITARIO;
      }
      if (c<5) {
           pedaggio = pedaggio * (1 + ALIQUOTA_IVA);
        }
       return pedaggio;
}
```

Domanda. (Verifica)

a) Definire i valori di input di un insieme di casi di prova per la convalida del metodo *calcolaPedaggio*, applicando un criterio di progettazione delle prove a *scatola aperta*.
L'ambiente di prova prevede i seguenti valori per le costanti usate dal metodo:

```
private final double ALIQUOTA_IVA      = 0.20;
private final double COSTO_UNITARIO    = 0.20;
private final double SOGLIA_MONTAGNA   = 50;
```

b) perchè, in base alle informazioni fornite nel testo, non è possibile completare i casi di prova?

Risposta.

a) Una possibile selezione di casi di prova, relativamente ai valori di input, che permette una copertura del 100% secondo il criterio delle decisioni è data dalla seguente tabella:

b) Input		
c) p	d) M	e) c
f) 10	g) 40	h) 1
i) 10	j) 60	k) 5

b) Il completamento richiederebbe la specifica dei risultati attesi in output, ma le informazioni nel testo non permettono di costruire un oracolo.

CAP 14. ZTL

Caro Assessore,

in relazione alla sua richiesta circa l'uso dei **terminali** Telepass per l'accesso alla Zona a Traffico Limitato (ZTL) del suo comune, le invio la seguente analisi preliminare.

Sono previsti due tipi di **utenti**: i **residenti**, che possono accedere alla ZTL in qualsiasi momento della giornata, da qualsiasi **varco**, e sostare a tempo indeterminato, e gli utenti **carico-scarico** (ad es. commercianti e genitori di bimbi delle scuole del centro) che possono entrare solo in uno o due determinati **intervalli**, da un unico varco e devono lasciare la ZTL entro un'ora dall'ingresso.

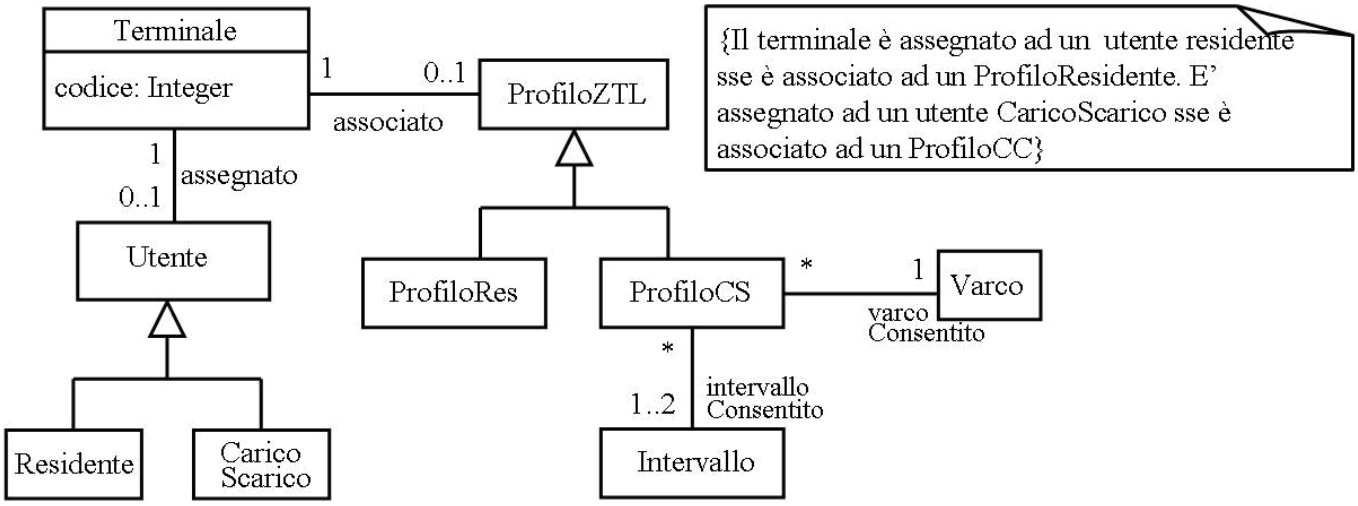
Un terminale telepass è dotato di **codice** univoco. L'abilitazione di un terminale alla ZTL prevede l'acquisizione del codice da parte del sistema e la sua associazione con un **profilo**: **profiloRes** o **profiloCS** Quest'ultimo indica gli intervalli e il varco consentiti per l'ingresso.

Il funzionamento pratico è relativamente semplice: quando un veicolo si avvicina a un varco di ingresso, un impianto ottico (CTV) attiva l'emissione, da parte della centralina, del segnale di richiesta di identificazione. Il terminale risponde ritrasmettendo il codice identificativo. Se non viene inviato dal terminale un codice che permetta l'accesso, una speciale fotocamera fotografa la targa, per poter risalire all'autore del transito irregolare. Nel caso di utenti carico-scarico, la centralina registra il codice di identificazione e l'orario di ingresso. Il funzionamento dei varchi di uscita è più semplice. All'avvicinarsi di un veicolo richiedono l'identificazione, come i varchi di ingresso. Poi però ignorano la maggior parte dei veicoli: sia quelli che non inviano un codice corretto, sia quelli dei residenti. Nel caso di utente carico-scarico, la centralina del varco trasmette il codice e l'orario di uscita al varco di ingresso consentito per quell'utente, che provvede a verificare se l'uscita e' avvenuta nel rispetto dei tempi stabiliti.

Ogni notte, le informazioni sui transiti irregolari vengono trasmesse dai varchi ad un server centrale; inoltre, eventuali modifiche alle utenze vengono distribuite a tutti i varchi.

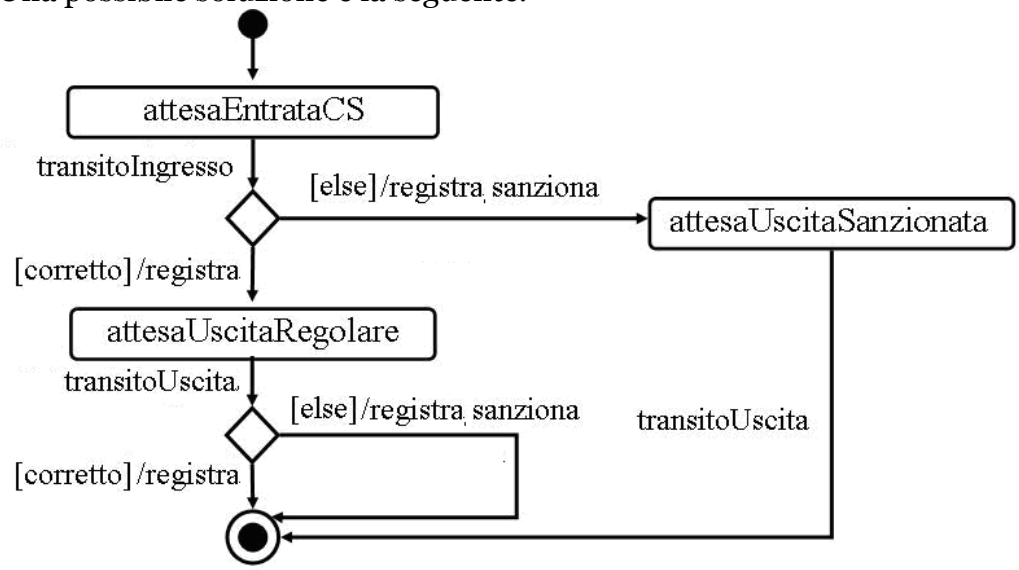
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

Risposta. Una possibile soluzione è la seguente.



Domanda. (Analisi del dominio) Dare una macchina a stati che descriva il comportamento richiesto al sistema in relazione a un transito nella ZTL del veicolo di un utente carico–scarico.

Risposta. Una possibile soluzione è la seguente.



L'analisi ha portato all'individuazione delle funzionalità che il primo rilascio del sistema dovrà realizzare: gestione utenti, gestione dei transiti in ingresso e in uscita, emissione sanzioni, invio sanzioni. Il caso d'uso Emissione sanzioni ha la seguente narrativa.

Domanda. (Analisi dei requisiti) Dare una narrativa del caso d'uso Emissione sanzioni.

Risposta.

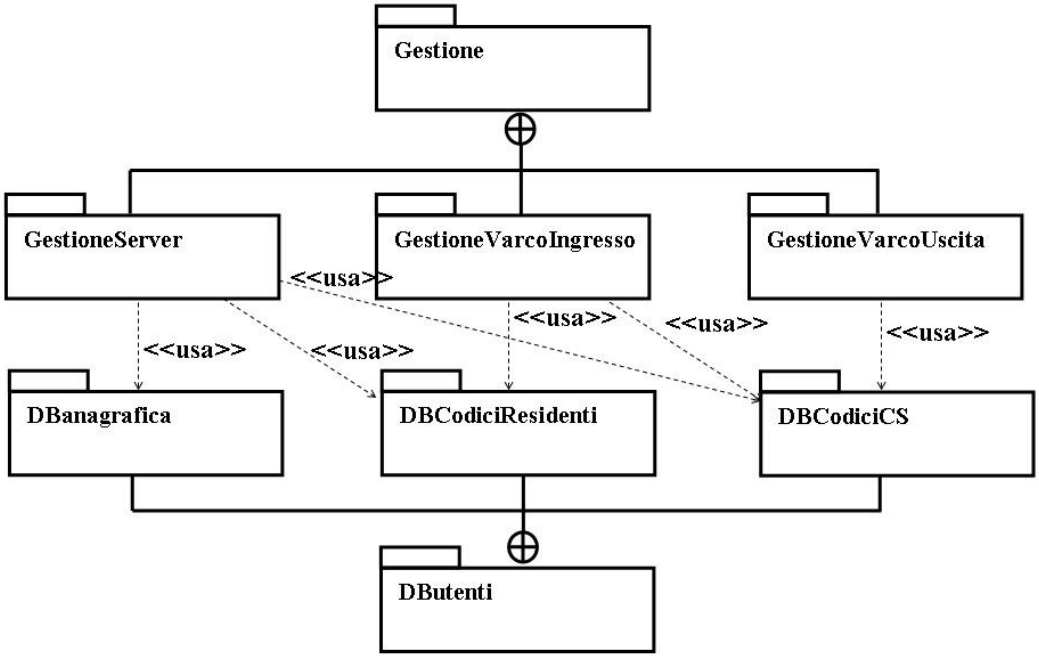
Caso d'uso Emissione sanzioni	
Breve Descrizione	Il sistema individua i proprietari dei veicoli che hanno commesso un illecito, e stampa i relativi verbali di sanzione.
Attori Primari	Tempo.
Attori Secondari	Pubblico Registro Automobilistico (PRA)
Precondizioni	Un varco di ingresso ha segnalato un illecito inviando un numero di targa o un codice identificativo.
Sequenza principale degli eventi	1. Il caso d'uso inizia alle ore 24.00; 2. **for** (ogni illecito segnalato) 2.1. **if** (utente carico-scarico) 2.1.1. individua proprietario; 2.2. **altrimenti** 2.2.1. richiedi proprietario al PRA; 2.3. calcola ammontare sanzione; 2.4. invia alla stampante il verbale di sanzione.
Postcondizioni	Sanzione stampata
Sequenza degli eventi alternativa	1. TransitoTargaStraniera 2. Autorizzazione temporanea

Tra i moduli software che devono essere realizzati si considerino:

DBanagrafica	mantiene l'anagrafica utenti
DBcodiciResidenti	mantiene i codici telepass e i profiliZTL dei residenti
DBcodiciCS	mantiene codici telepass e i profiliZTL degli utenti carico–scarico

GestioneServer	modulo che realizza le funzionalità per la gestione utenti e l'emissione delle sanzioni
GestioneVarcoIngresso	modulo che realizza le funzionalità del varco in ingresso
GestioneVarcoUscita	modulo che realizza le funzionalità del varco in uscita

Domanda. (Architettura) Fornire una vista ibrida strutturale sull'architettura con una vista di decomposizione, che indichi come raggruppare moduli omogenei per funzionalità, e una vista d'uso.
Risposta. Una possibile soluzione è la seguente



Si ricorda che il tempo di permanenza massimo consentito, per gli utenti carico-scarico, è di 60 minuti. Si assuma che il calcolo della liceità della permanenza sia definito dal seguente metodo:

```
      /* restituisce false sse la permanenza supera un'ora */
public boolean valutaPermanenza( ) {
      boolean lecita ;
      if (this.minutiUscita < this.minutiIngresso)
            if  (this.oreIngresso == 0)
                  this.oreIngresso = 23;
            else
                  this.oreIngresso-- ;
      lecita =  (this.oreIngresso == this.oreUscita);
      return lecita;
}
```

Il seguente caso di test evidenzia un malfunzionamento.

Input				Output
oraUscita	**minutiUscita**	**oraIngresso**	**minutiIngresso**	
14	30	13	30	true

Domanda. (Verifica, Telepass e ZTL) Dire quali criteri strutturali e/o funzionali garantiscono di definire un caso di test che permette di evidenziare il malfunzionamento.

Risposta. L'unico criterio di definizione dei casi di test che garantisce di individuare il malfunzionamento è il criterio che considera le condizioni di confine. Questo criterio si applica sia nel caso scatola aperta (caso `this.minutiUscita = this.minutiIngresso)` che a scatola chiusa (permanenza di un'ora esatta).
Si noti che anche il caso di permanenza di zero minuti, anche se improbabile, dovrebbe essere verificato.

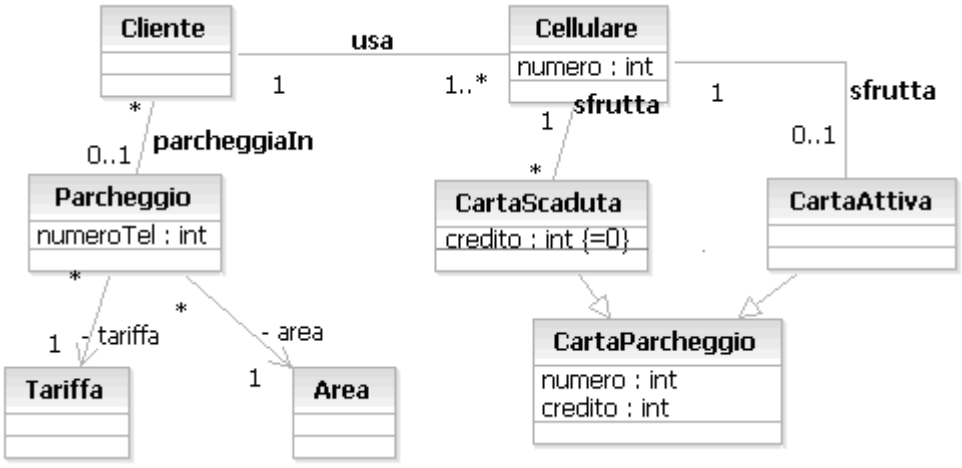
CAP 15. Easy Park©

Si vuole realizzare una innovativa soluzione per il pagamento del **parcheggio** via telefono **cellulare**. Il **cliente** telefonando (non vi è risposta e quindi spesa) viene identificato e attiva la sosta scalando il pagamento dalla sua "**Carta Parcheggio**" prepagata. Seguono le principali funzionalità richieste:

1. Attivazione
   Il cliente acquista una Carta Parcheggio prepagata e l'attiva indicando il **numero** di telefono cellulare da abbinare. D'ora in poi potrà pagare la sosta con il proprio cellulare. Durante l'attivazione, il sistema trasferisce sulla nuova carta l'eventuale **credito** residuo su una carta già associata al numero di telefono indicato.
2. Utilizzo del servizio
   Il cliente parcheggia ed espone sul cruscotto la Carta Parcheggio. Nel cartellone del Parcheggio verifica qual è il **numero di telefono** che identifica l'**area** e la **tariffa**. Il cliente telefona a questo numero, il cliente è identificato attraverso il proprio numero di telefono cellulare e il sistema attiva il pagamento della sosta.
3. Verifica da parte del controllore
   Il Controllore del Parcheggio controlla l'effettivo pagamento della sosta inserendo il **numero della Carta Parcheggio** in un applicativo fruibile tramite Pocket PC connesso a internet o Telefono Cellulare.
4. Disattivazione
   Disattivazione della sosta con chiamata via cellulare: l'utente chiama il numero associato al parcheggio, il sistema riconosce l'utente e disattiva il pagamento. Inoltre il sistema comunica vis SMS la disattivazione, la somma pagata, la durata della sosta e il residuo presente sulla Carta Parcheggio.

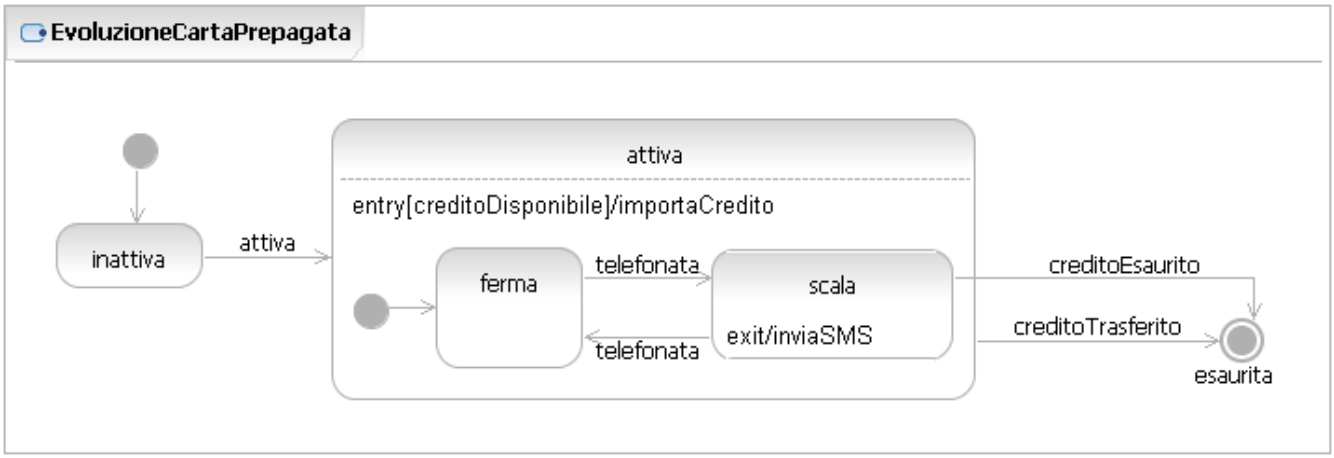
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva il dominio, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto. Si aggiungano due sottoclassi di CartaParcheggio, per esprimere il vincolo che a un cellulare può essere associata al più una carta con credito positivo.

Risposta. Una possibile soluzione è la seguente.



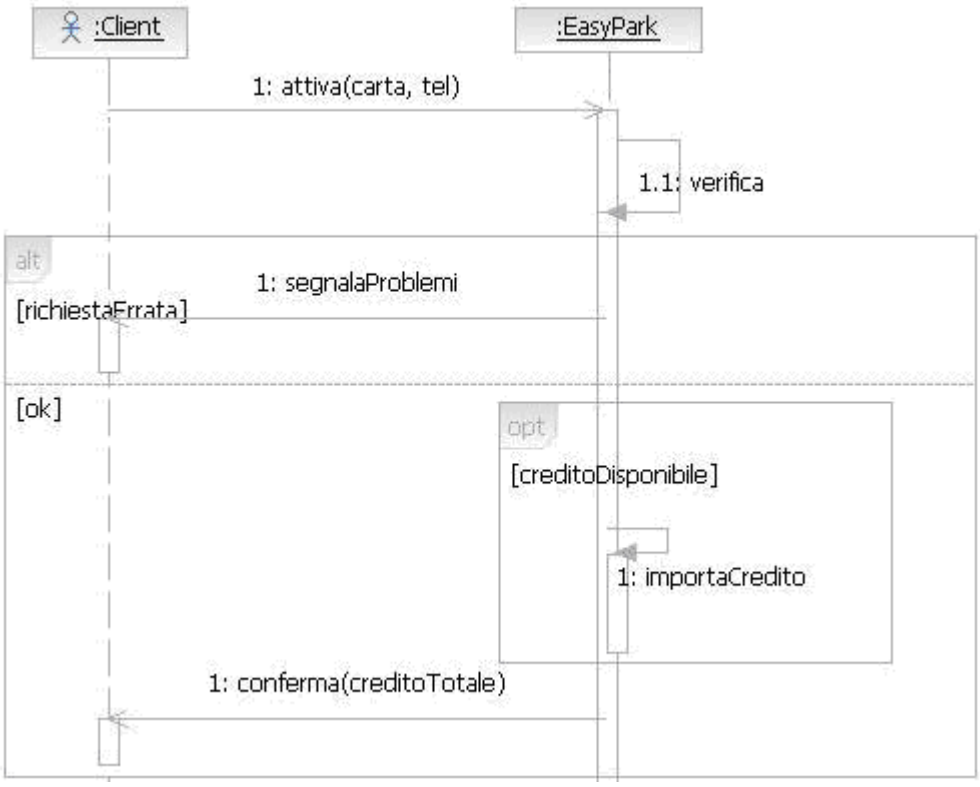
Domanda. (Analisi del dominio) Dare una macchina a stati che descriva l'evoluzione di una carta prepagata, dall'attivazione all'esaurimento del credito o al suo trasferimento su un'altra carta.

Risposta. Una possibile soluzione è la seguente.



Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che descriva le interazioni tra Cliente e Easy Park per il caso d'uso Attivazione.

Risposta. Una possibile soluzione è la seguente.



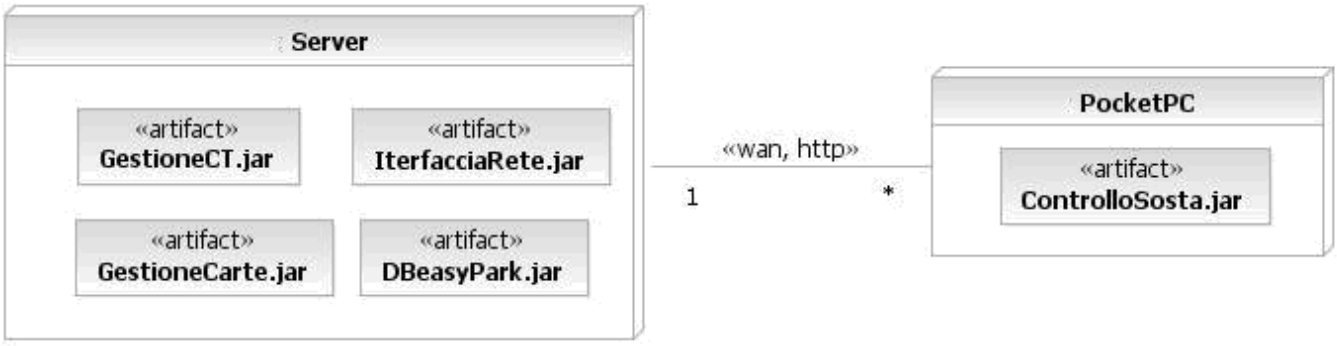
Nota. Abbiamo usato un diagramma di sequenza per descrivere la narrativa di un caso d'uso. Sebbene questi diagrammi siano più frequentemente usati in fase di progettazione di dettaglio, sono anche spesso usati in fase di analisi dei requisiti.

La progettazione architettonica ha portato a individuare le seguenti componenti:

DBeasyPark	Memorizza tutte le informazioni sullo stato delle carte prepagate.
GestioneCarte	Realizza la logica dell'applicazione, interpretando i messaggi in arrivo, aggiornando lo stato delle Carte, e restituendo le debite risposte.
ControlloSosta	Invia a GestioneCarte i dati immessi dal controllore nel PocketPC, e visualizza quelli ricevuti come risposta**.**
GestoreCT	Realizza l'interfaccia con le linee telefoniche, e trasforma chiamate telefoniche e sms del cliente o del controllore in dati che passa a GestioneCarte e, viceversa, trasforma i dati passati da GestioneCarte in sms per il cliente.
InterfacciaRete	Trasforma i messaggi ricevuti dal controllore via rete nello stesso formato utilizzato da GestoreCT per passare i dati a GestioneCarte, e viceversa.

Domanda. (Architettura) Assumendo che ogni componente si manifesti in un artefatto jar omonimo, dare un diagramma di dislocazione del sistema Easy Park.

Risposta. Una possibile soluzione è la seguente:



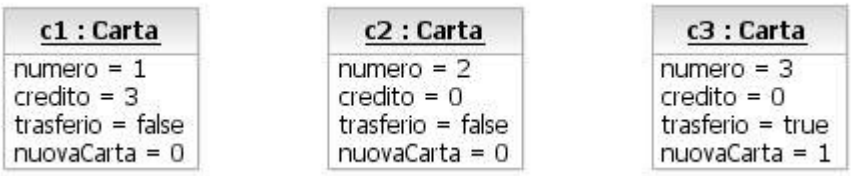
Si consideri il seguente metodo per controllare che ci sia credito per pagare il parcheggio (la classe Carta è definita a destra):

```
public boolean controlloSosta(int numeroCarta){
  Carta c = DBeasyPark.get(numeroCarta);
  int credito = c.credito;
  if (credito > 0) return true;
  else { boolean trasferito = c.trasferito;
         if (!trasferito) return false;
         Carta nc = DBeasyPark.get(c.nuovaCarta);
         return (nc.credito > 0);
}}
```

Il metodo DBeasyPark.get, invocato con n, restituisce un oggetto di tipo Carta, il cui attributo *numero* vale n.

Domanda. (Verifica) Si vuole testare il metodo, soddisfacendo il criterio “tutti i comandi”, utilizzando uno stub per il metodo statico DBeasyPark.get. Dare un diagramma degli oggetti che mostri gli oggetti di tipo Carta che lo stub deve restituire per raggiungere l’obiettivo posto.

Risposta.

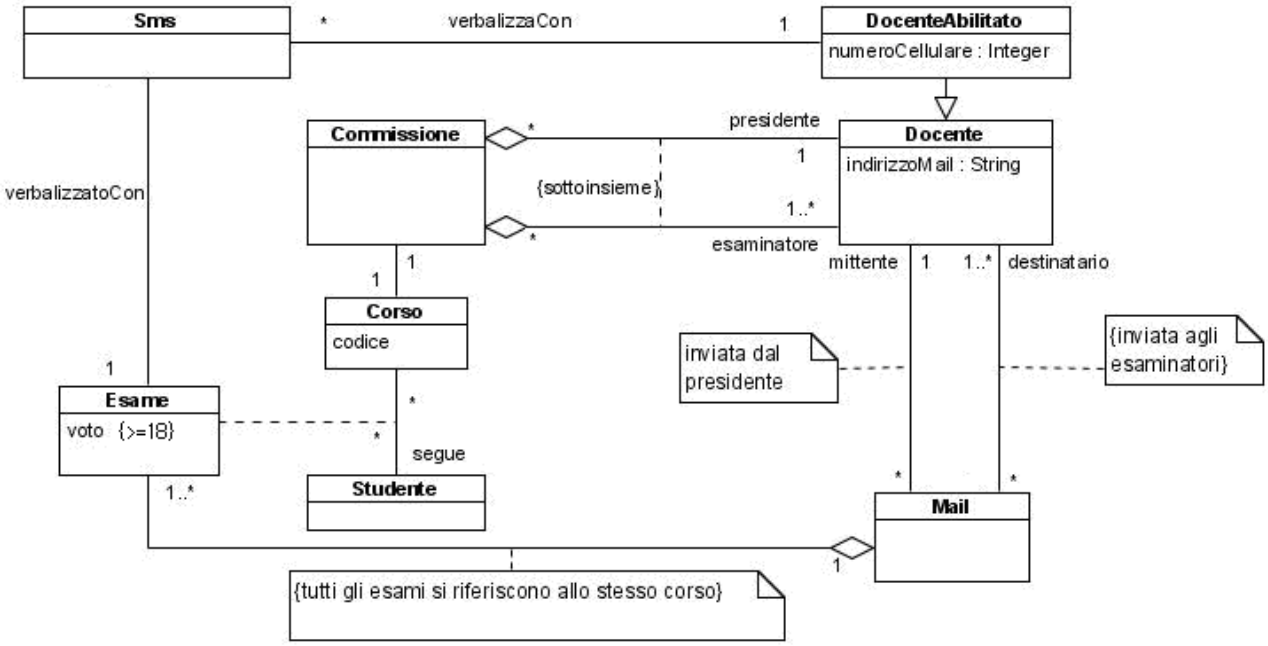


CAP 16. Registrazione Esami

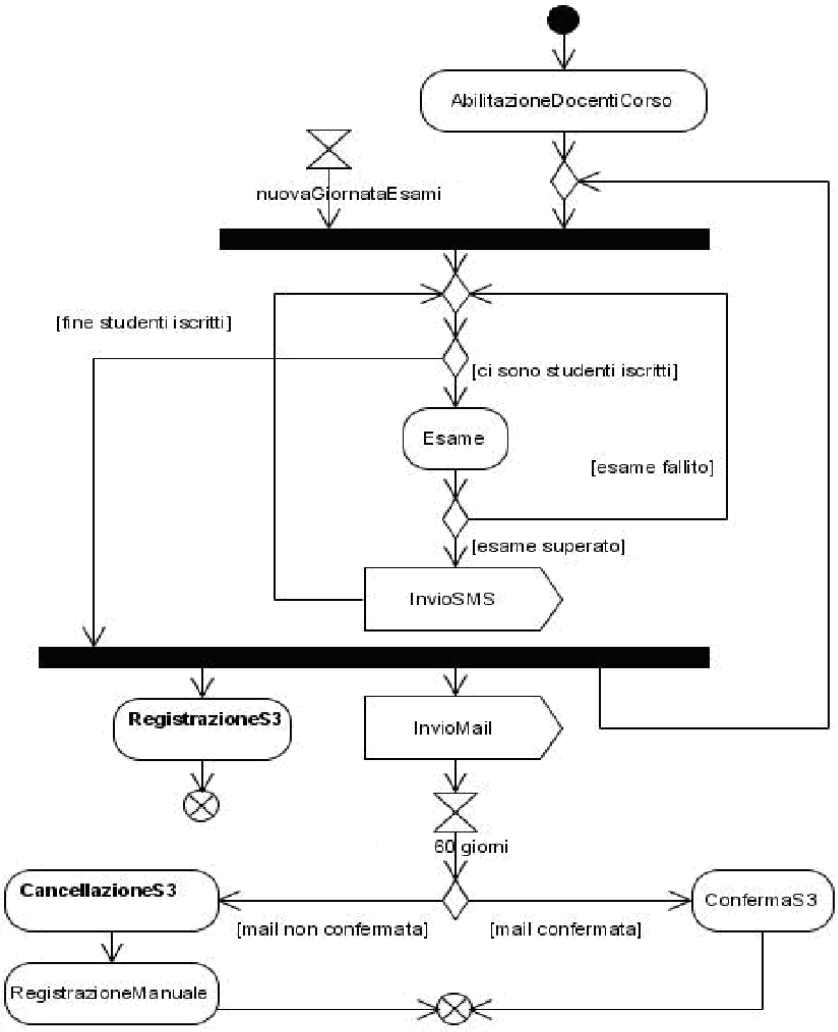
Il processo di dematerializzazione dei documenti cartacei e di innovazione delle procedure amministrative del nostro ateneo può fare un significativo balzo in avanti con la sostituzione degli attuali statini cartacei con un nuovo sistema di registrazione degli esami basato sull'uso dei telefoni cellulari da parte dei **docenti**. Per ogni **corso**, caratterizzato da un **codice**, è nominata una **commissione**, formata dai docenti che possono svolgere il ruolo di esaminatore. Tra di essi, è designato il presidente di commissione. Il sistema, denominato *CellEx*, prevede che i **docenti abilitati** all'uso di *CellEx* che fanno parte della commissione, utilizzino il cellulare per verbalizzare gli **esami**, mediante l'invio di un **SMS** al numero telefonico del servizio. Come misura di sicurezza e autenticazione, i docenti si abilitano al servizio registrando (una volta sola) su *CellEx* il **numero** del loro telefono cellulare. Ogni docente ha un **indirizzo mail**. Giornalmente, *CellEx* registra gli esami della giornata nel sistema S3 (che è in uso da anni nell'Ateneo) e invia agli esaminatori di ogni corso una **mail** contenente un'indicazione di tutti gli esami del loro corso registrati in giornata. In seguito alla ricezione della mail, il presidente della commissione provvede a rispondere per confermare. *CellEx* attende 60 giorni l'invio della mail di conferma. Gli esami confermati sono definitivamente archiviati in S3, quelli non confermati sono cancellati da S3 e devono essere registrati manualmente dal presidente della commissione, secondo una procedura che prevede l'uso di uno statino in formato cartaceo. Utilizzando l'elenco degli **studenti** iscritti a un appello, organizzato in sottoelenchi per ogni giornata di esami, il docente procede a effettuare l'esame che può avere un esito positivo (**voto** maggiore o uguale a 18 trentesimi) o negativo. Solo gli esami con esito positivo sono registrati mediante l'invio di un SMS.

Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

Risposta. Un possibile diagramma delle classi è il seguente:

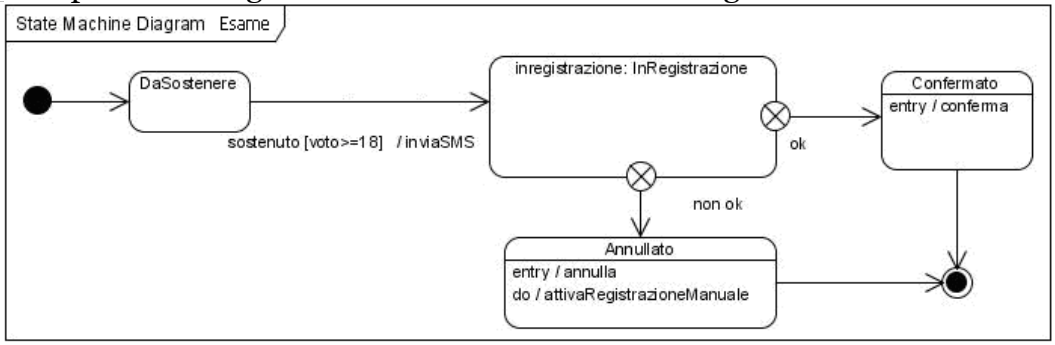


Domanda. (Analisi del dominio) Dare un diagramma di attività della procedura descritta nell'enunciato, relativamente ad un corso, iniziando dall'abilitazione, e iterando sulle giornate di un appello.
Risposta. Un possibile diagramma di attività è il seguente:

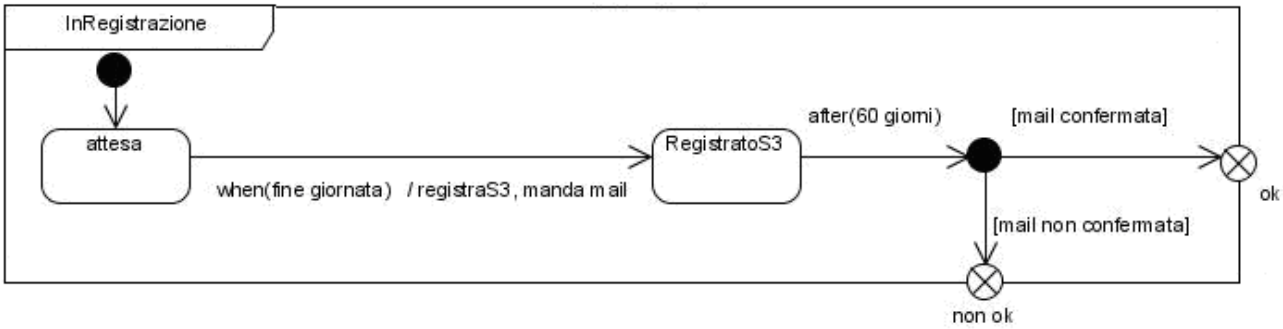


Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva l'evoluzione dello stato di un esame.

Risposta. Un possibile diagramma di macchina a stati è il seguente:



dove



Per registrare un esame, occorre inviare al numero 3401234567 un SMS nel seguente formato:

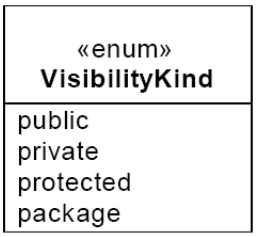
```
CodiceEsame CodiceMembro Matricola Voto
```

Il voto si indica in cifre ("31" per trenta e lode) oppure per esteso se un giudizio ("sufficiente", "discreto", "buono", "ottimo"). Ad esempio:

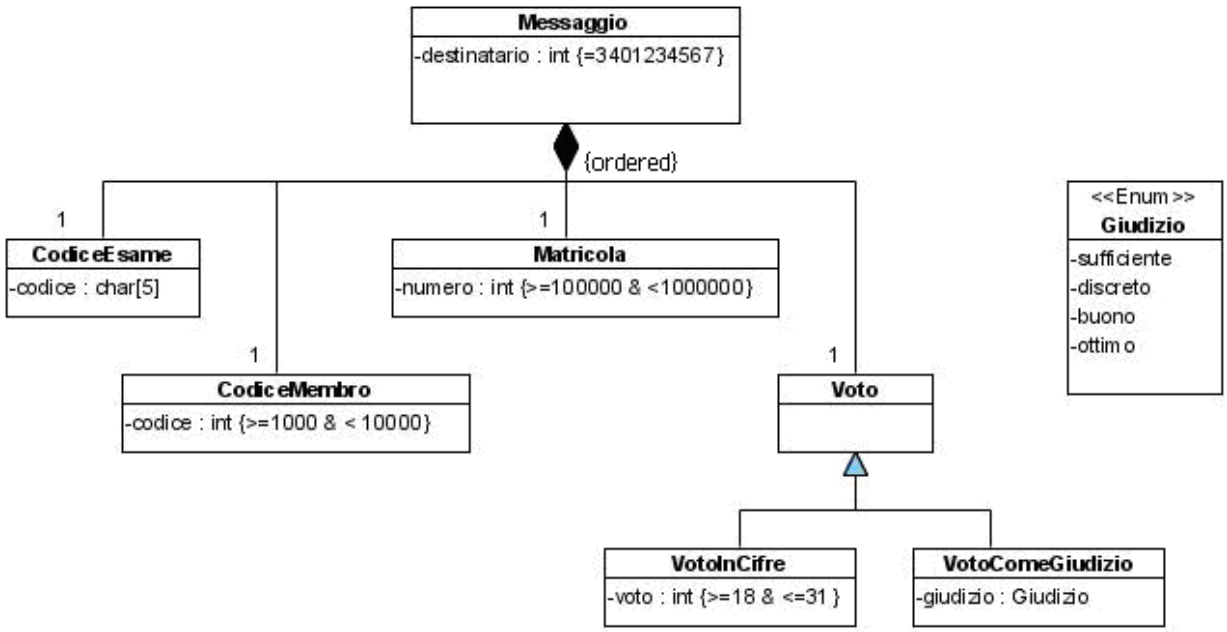
```
AA017 4567 123456 25     e     AA017 4567 123456 ottimo
```

dove 4567 è il codice di un membro della commissione. Nel caso di cultore della materia (membro di commissione non accademico), indicare CM. Ad esempio: AA017 CM 123456 28

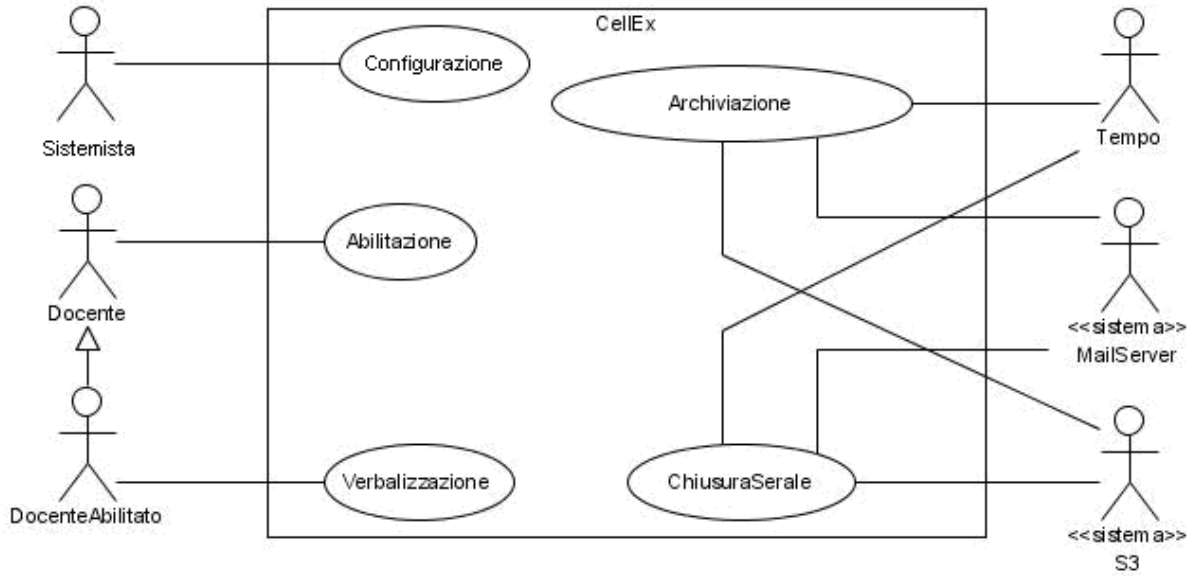
Domanda. (Analisi del dominio) Fornire un diagramma delle classi che descriva i messaggi SMS di CellEx, usando composizione e generalizzazione. Per specificare le enumerazioni si usi uno stereotipo, come nell'esempio.



Risposta.



L'analisi dei requisiti ha portato al seguente diagramma dei casi d'uso:



con le seguenti narrative incomplete:

Configurazione: Configurazione iniziale del servizio.
Abilitazione: Un docente richiede l'abilitazione all'uso del servizio.
Verbalizzazione: Permette di verbalizzare un esame usando il telefono cellulare.
ChiusuraSerale: La sera vengono registrati in S3 gli esami verbalizzati in giornata e viene mandato un e-mail ai docenti.
Archiviazione: La sera si archiviano in S3 gli esami verbalizzati 60 giorni prima e confermati; si cancellano da S3 quelli non confermati.

Domanda. (Analisi dei requisiti) Dare la narrativa di **Verbalizzazione**, **ChiusuraSerale e Archiviazione.**
Risposta. Possibili narrative sono:

Verbalizzazione

Breve descrizione: *Permette di verbalizzare un esame usando il telefono cellulare*
Attore principale: *DocenteAbilitato*
Attore secondario: *Nessuno*
PreCondizioni: *Esame superato*
PostCondizioni: *Esame verbalizzato*
Sequenza principale degli eventi:
1. *il docente invia un SMS con i dati dell'esame*
2. *il sistema registra l'esame*

Sequenza alternativa degli eventi:
1. *assenza di campo*

ChiusuraSerale

Breve descrizione: *La sera vengono registrati in S3 gli esami verbalizzati in giornata e viene mandato un e-mail ai docenti.*
Attore principale: *Tempo*
Attori secondari: *S3, MailServer*
PreCondizioni: *Nessuna*
PostCondizioni: *Gli esami verbalizzati sono registati, la mail di richiesta conferma è inviata.*

Sequenza principale degli eventi:
1. *Tempo invia il segnale che è sera*
2. *per ogni corso con almeno un esame verbalizzato in giornata*
   - 2.1. *per ogni esame del corso*
     - 2.1.1. *il sistema registra l'esame in S3*
     - 2.1.2. *il sistema memorizza l'esame come "registrato"*
   - 2.2. *il sistema invia un e-mail indirizzato ai docenti del corso*

Sequenza alternativa degli eventi: *Nessuna*

Caso d'uso: Archiviazione

Breve descrizione: *La sera si archiviano in S3 gli esami verbalizzati 60 giorni prima e confermati, si cancellano da S3 quelli non confermati.*
Attore principale: *Tempo*
Attori secondari: *S3, MailServer*
PreCondizioni: *Nessuna*
PostCondizioni: *Gli esami registati e confermati sono archiviati, quelli non confermati sono cancellati.*
Sequenza principale degli eventi:
1. *Tempo invia il segnale che è sera*
2. *per ogni esame registrato 60 giorni prima*
   - 2.1. *se (è confermato)*
     - 2.1.1. *il sistema archivia l'esame in S3*
   - 2.2. *altrimenti*
     - 2.2.1. *il sistema cancella l'esame da S3.*
     - 2.2.2. *il sistema manda un mail al presidente della commissione.*

Sequenza alternativa degli eventi: *Nessuna*

Si supponga di disporre di almeno tre tipi di nodi hardware:
1. *ServerCellEx* che funge da server per l'applicazione *Cellex* (un solo nodo di questo tipo);
2. *ServerMail* dove è dislocato il *MailServer* (un solo nodo di questo tipo);
3. *PC* usato dai docenti quando vogliono abilitarsi o leggere/scrivere messaggi di posta elettronica (un numero imprecisato di nodi di questo tipo).

Il canale tra il nodo *ServerCellEx* e i nodi *PC* supporta HTTP
I canali con il Server Mail supportano SMTP

La progettazione architettonica ha individuato le seguenti componenti, alcune da realizzare, altre che già esistono. Si noti che il MailServer, anche se di fatto esterno al sistema Cellex –al pari di S3 e del sistema di ricezione SMS dell'operatore telefonico utilizzato– viene esplicitamente descritto nell'architettura per completezza. Si noti anche che l'utente interagisce con la componente *Abilitazione* attraverso un browser.

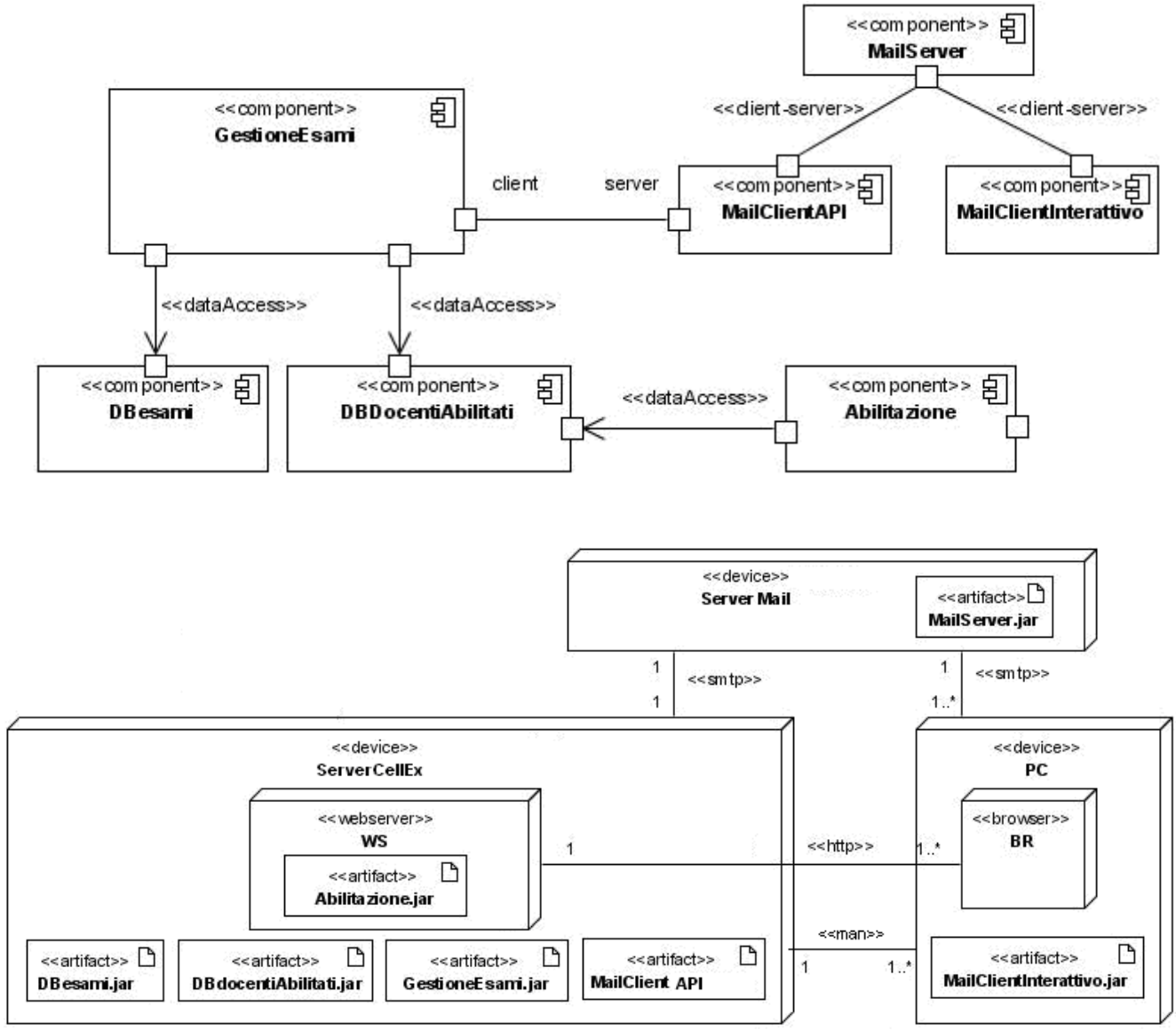
Componente	**Disponibile**	**Responsabilità**
DBdocentiAbilitati		Mantiene i nomi e i numeri di telefono dei docenti abilitati al servizio.
DBesami		Mantiene i dati degli esami verbalizzati e di quelli registrati.
Abilitazione		Permette ai docenti di abilitarsi.
GestioneEsami		Realizza i casi d'uso Verbalizzazione, ChiusuraSerale e Archiviazione.
MailServer	X	Server di posta.
MailClientAPI	X	Client di posta con interfaccia verso altra componente.

MailClientInterattivo	X	Client di posta con interfaccia utente.

Si assuma che ogni componente si manifesti in un artefatto omonimo, di tipo jar

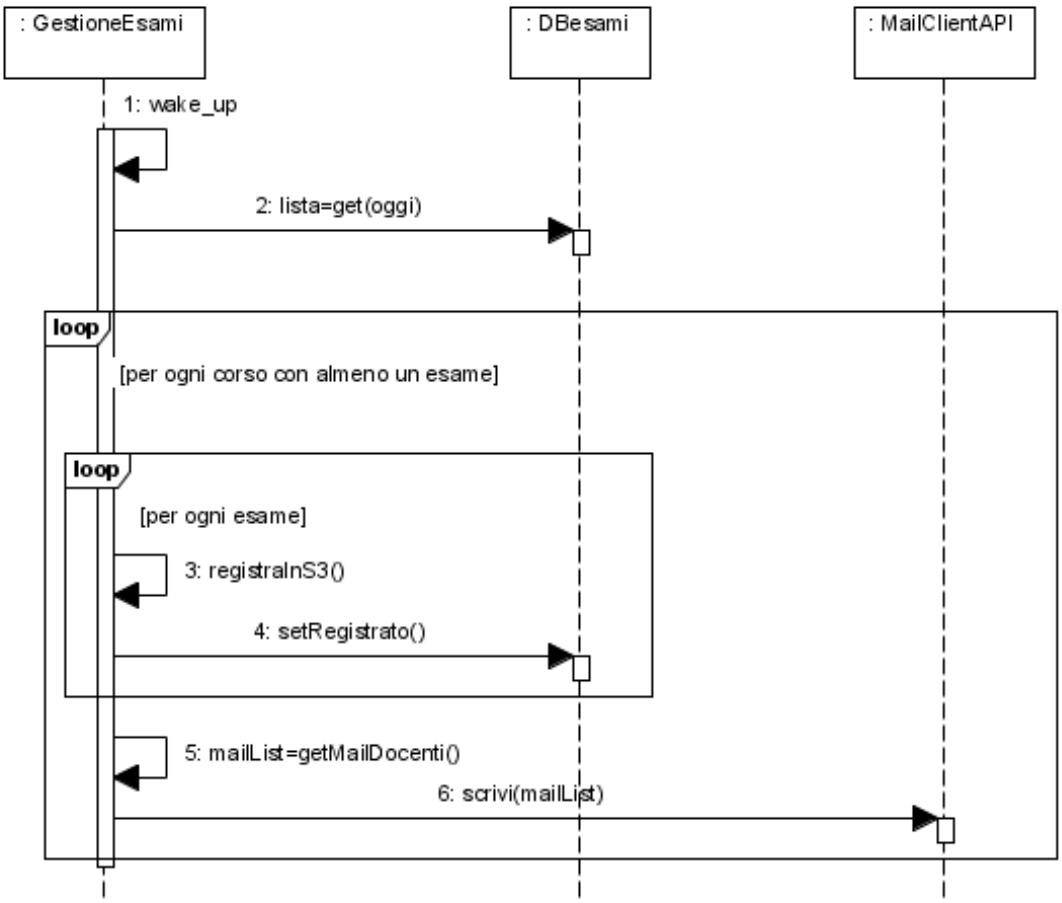
Domanda. (Architettura)
a. Fornire una vista C&C
b. Fornire una vista logistica di dislocazione.

Risposta.



Domanda. (Realizzazione dei casi d'uso) Fornire un diagramma di sequenza che illustri le componenti individuate (si veda la figura precedente) interagiscono per realizzare il caso d'uso ChiusuraSerale. Si assuma che le informazioni sulle commissioni d'esame e sugli indirizzi mail dei docenti siano mantenute da S3, cui GestioneEsami accede direttamente.

Risposta.

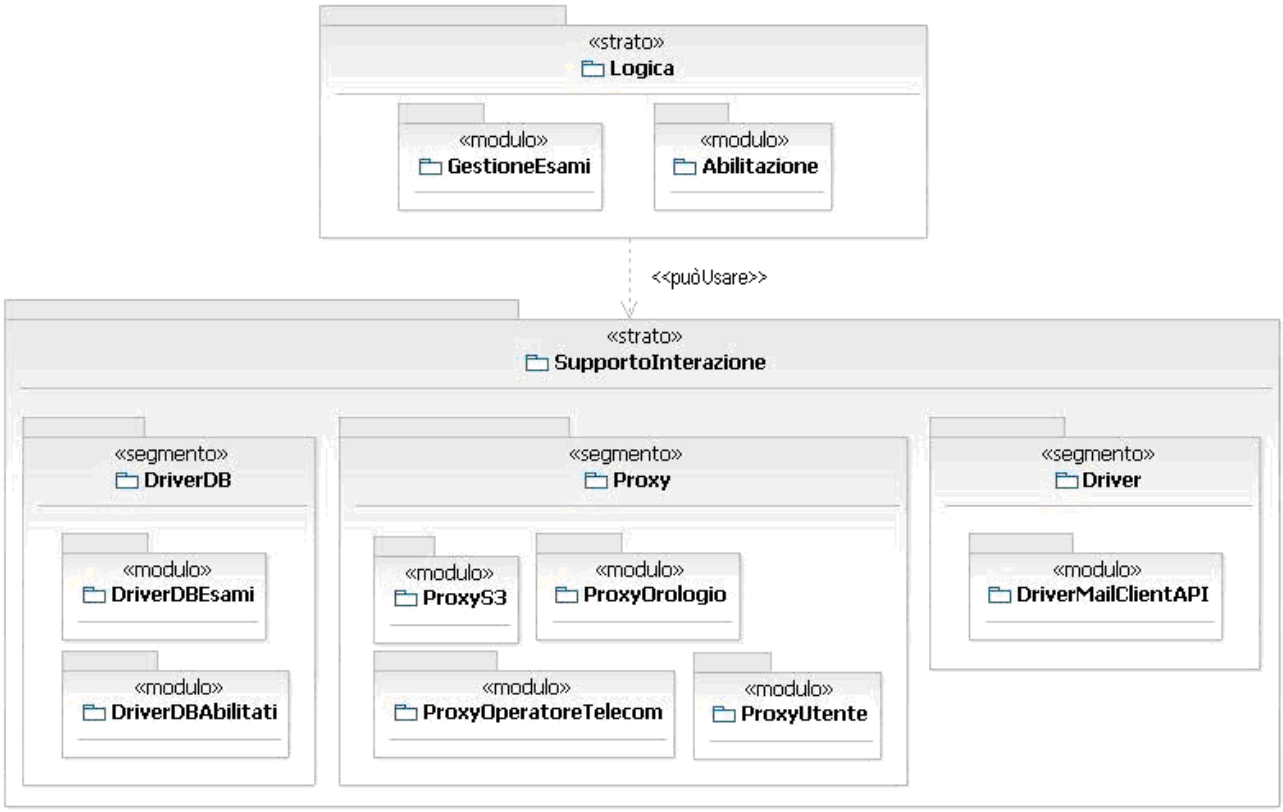


L'architetto software di CellEx ha previsto una struttura a strati del software del sistema, in cui lo strato superiore raggruppa i moduli che realizzano le parti di logica delle componenti, e lo strato inferiore raggruppa i moduli relativi alle altre parti. Questo strato è inoltre organizzato a segmenti, per distinguere driver per l'accesso a data base, altri driver e proxy.

Domanda. (Architettura) Fornire una vista strutturale del sistema CellEx, limitatamente ai moduli relativi alle parti delle componenti indicate sotto. Si utilizzino gli stereotipi *strato*, *segmento*, *modulo*, e si usi il nome della componente per il modulo che ne realizza la logica.

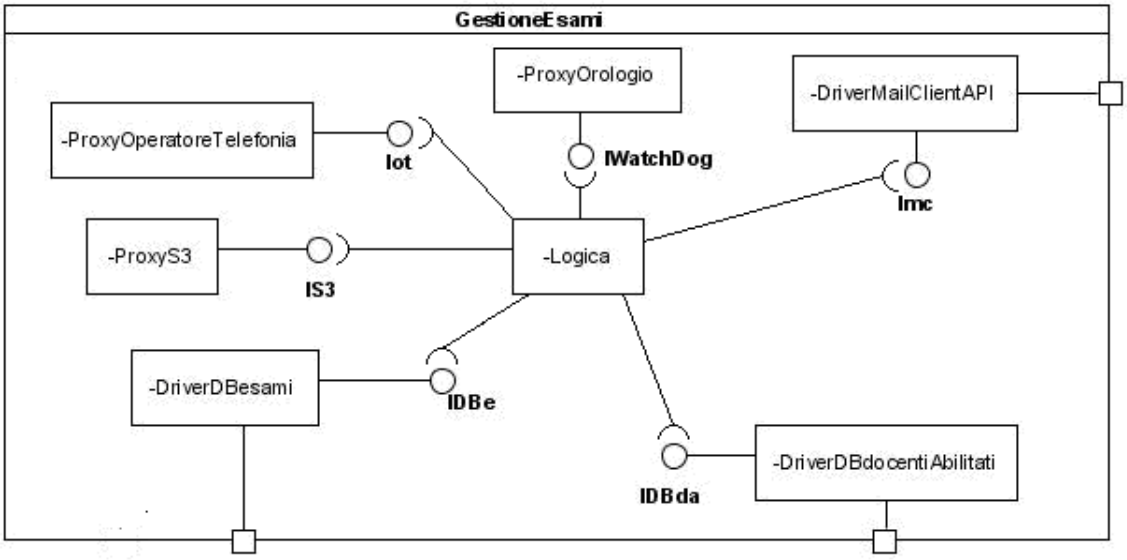
Componente: GestioneEsami	
Parte	**Responsabilità**
Logica	Realizza la logica della componente
DriverMailClientAPI	Permette alla logica di accedere al MailClientAPI
DriverDBAbilitati	Permette alla logica di interrogare il DBdocentiAbilitati
DriverDBesami	Permette alla logica di accedere al DBesami
ProxyOrologio	Comunica alla logica l'ora offrendo l'interfaccia IWatchDog.
ProxyOperatoreTelefonia	Riceve i messaggi relativi agli SMS
ProxyS3	Permette di comunicare col sistema S3
Componente: Abilitazione	
Parte	**Responsabilità**
Logica	Realizza la logica della componente
DriverDBAbilitati	Permette alla logica di interrogare il DBdocentiAbilitati
ProxyUtente	Gestisce le interazioni con l'utente.

Risposta.



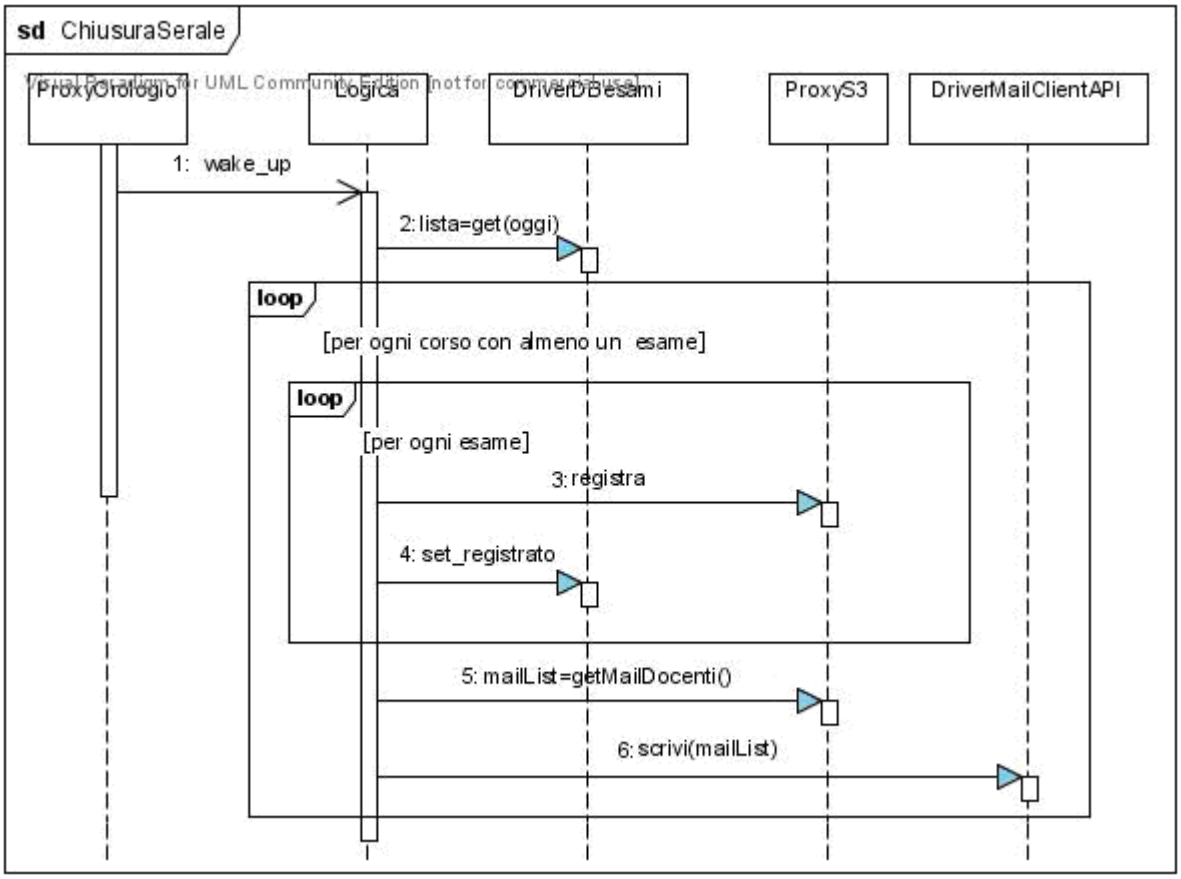
Domanda. (Progettazione di dettaglio) Fornire un diagramma di struttura composita per la componente GestioneEsami, considerando le parti descritte nel precedente esercizio.

Risposta.



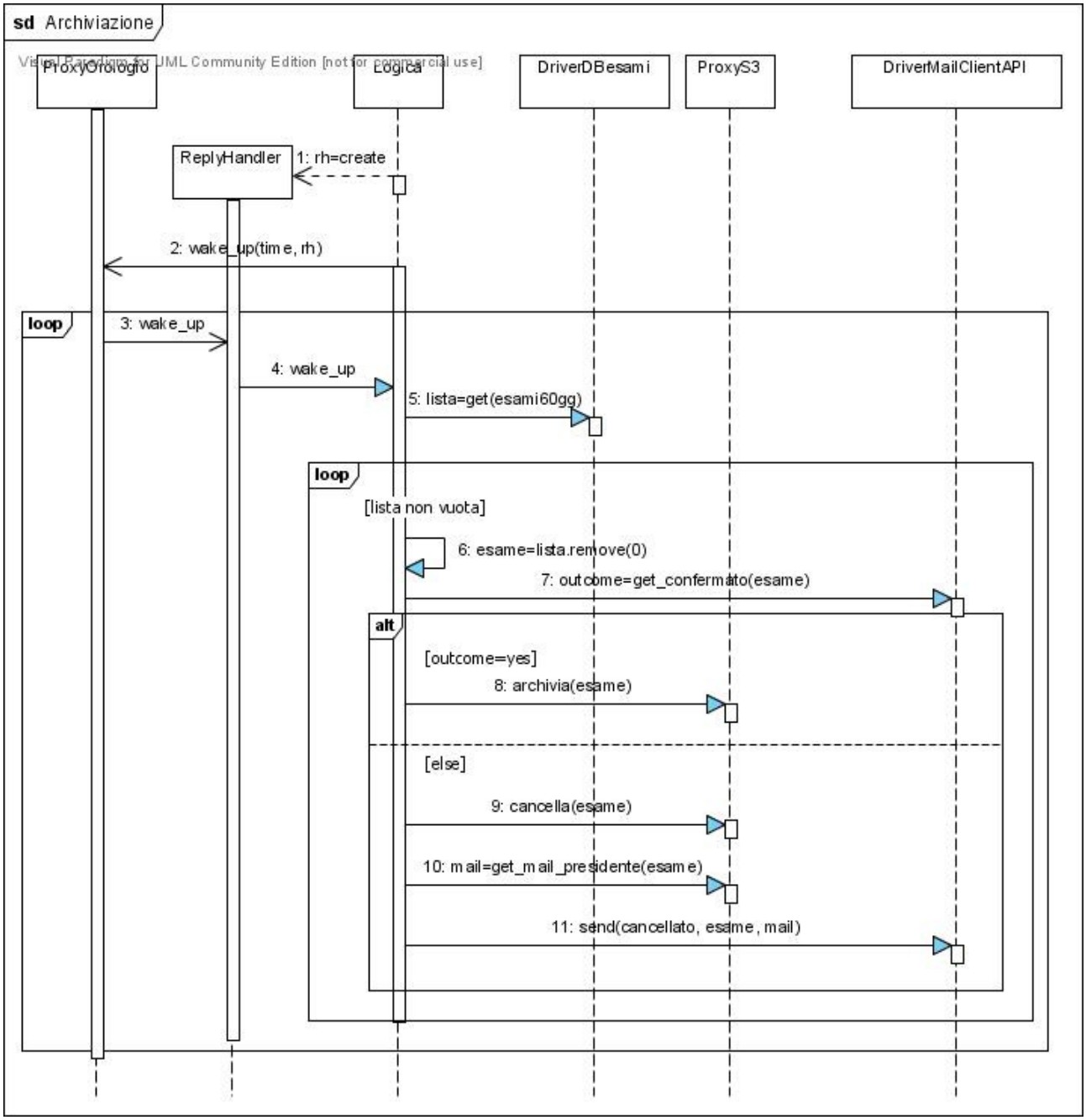
Domanda. (Realizzazione dei casi d'uso) Fornire un diagramma di sequenza che illustri come le parti della componente GestioneEsami (si veda la figura precedente) interagiscono per realizzare il caso d'uso ChiusuraSerale. Si assuma che le informazioni sulle commissioni d'esame e sugli indirizzi mail dei docenti siano mantenute da S3.

Risposta.



Domanda. (Realizzazione dei casi d'uso) Dare un diagramma di sequenza che illustri le interazioni tra le parti di GestioneEsami coinvolte nel caso d'uso Archiviazione. Si modelli anche l'ultima fase del caso d'uso Configurazione, in cui la componente Archiviazione chiede di essere svegliata ogni sera. Per le interazioni con l'interfaccia a WatchDog, si utilizzi (una semplificazione) del modello *Callback* per le invocazioni asincrone di metodi: il metodo viene invocato passando il riferimento ad un oggetto di tipo Reply Handler precedentemente creato dal cliente. A questo punto il cliente recupera il controllo. La risposta all'invocazione del metodo viene passata al Reply Handler, che la passa al cliente.

Risposta.



Nota: questo esercizio non è in alcun modo collegato al problema della registrazione degli esami. Si consideri la seguente funzione definita per casi:

$$f(x, y) = \begin{cases} (x + y) / 2 & \text{se } x \geq y \\ (2y - x) / 2 & \text{se } 0 < x < y \\ (2y + x) / 2 & \text{se } x < y \text{ and } x \leq 0 \end{cases}$$

Si consideri il seguente metodo Java scritto per realizzare la funzione f:

```
integer f (x: integer, y: integer) {
   if (x < y) {
```

```
        y = 2 * y;
        while ((y > 0) & (x > 0)) {
           x--; y--;
        }
    }
    return (x + y) / 2
}
```

Domanda. (Verifica) Dare un insieme di casi di prova che garantisca la copertura totale dei comandi del metodo Java.

Risposta. Un insieme minimo è per esempio il seguente:

Input	Output atteso
(2,6)	5

Una delle funzioni ausiliarie del sistema CellEx è di fornire informazioni statistiche sull'andamento degli esami. In particolare, il sistema deve fornire informazioni sul numero d'esami mediamente sostenuti ogni giorno, per corso di laurea, facoltà, e per tutta l'università. A tale scopo, si prevede l'utilizzo di una funzione `numeroMedioEsami` che, dato un vettore di numeri d'esame, ne restituisce la media, arrotondata all'intero superiore. Per verificare la funzione si prevede un test a scatola nera.

Domanda. (Verifica) Fornire cinque casi di prova per la funzione `numeroMedioEsami`, giustificando per ciascuno la ragion d'essere.
Risposta.

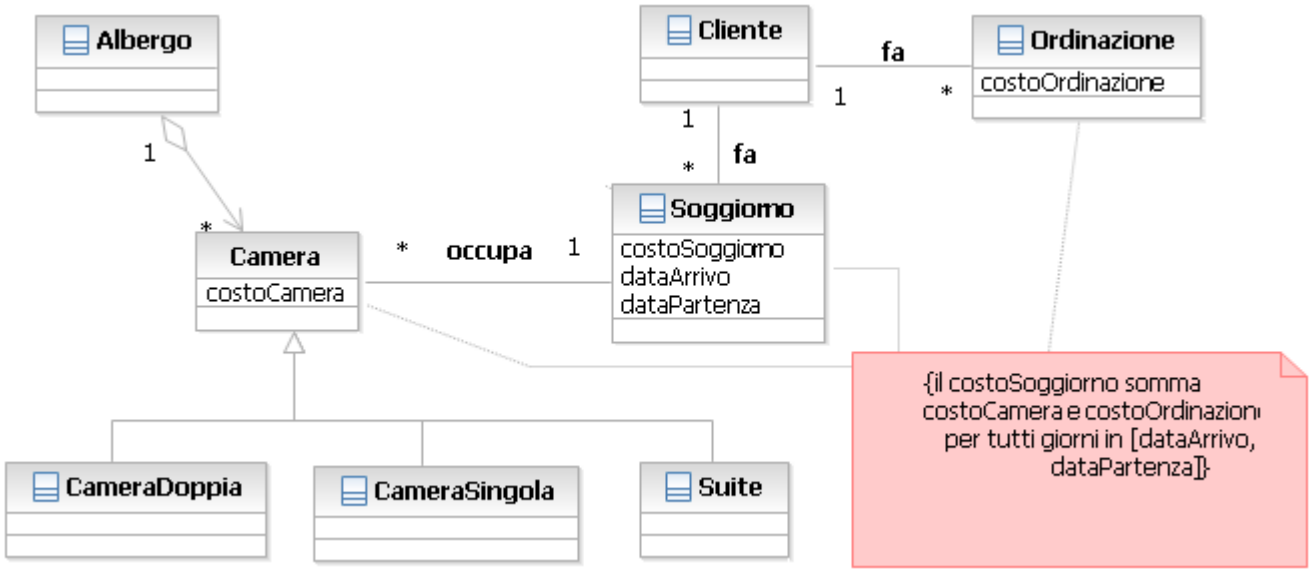
Casi di prova		Giustificazione
Input: valori	**Output: media**	
[ ]	0	Caso limite: vettore vuoto
[1]	1	Caso limite: un solo elemento
[1,1]	1	Caso speciale: tutti uguali
[1,2]	2	Verifica arrotondamento
[4,1,2]	3	Caso generico

CAP 17. Albergo dei Fiori

L'*Albergo dei Fiori* intende adottare un sistema di gestione che gestisca prenotazione e pagamento delle camere, attività di pulizia, bar e ristorante. I **clienti** possono prenotare una o più camere tramite vari mezzi (call center, web, agenzie), indicando la **data di arrivo**, la **data di partenza**, i tipi di **camera** (**camera singola**, **camera doppia**, **suite**).
All'arrivo presso l'**albergo**, il cliente si registra all'accettazione, fornendo al portiere un documento di identificazione. Dopo la registrazione gli vengono assegnate una o più camere del tipo richiesto e fornite le chiavi di accesso. Durante il soggiorno, il cliente può effettuare **ordinazioni** al bar o al ristorante dell'albergo, sia direttamente sul posto, sia chiedendo il servizio in camera (in questo caso, l'ordinazione viene effettuata al telefono con l'accettazione o attraverso un TV-terminale di cui sono dotate le camere, fornito di un telecomando che consente di navigare una serie di menu tramite i quali effettuare l'ordinazione). Nel caso di servizio in camera, il cameriere porta direttamente in camera quanto ordinato.
Il servizio di pulizia viene effettuato nelle camere occupate tutte le mattine, fra le 10:00 e le 12:00. Se un cliente espone il cartello "Non disturbare", la pulizia della camera non è effettuata e si effettua un secondo tentativo fra le 15:00 e le 17:00. Se anche in questo caso è presente il cartello, non è effettuata la pulizia per quel giorno. Nelle camere libere non viene invece effettuato alcun servizio di pulizia e se una camera rimane libera per molto tempo, si effettua una pulizia straordinaria.
Quando il cliente termina il suo **soggiorno**, si presenta all'accettazione per il pagamento del **costo del soggiorno**, calcolato in base al **costo delle camere** occupate (che dipende dal loro tipo) e del **costo delle ordinazioni** effettuate al bar e al ristorante, sia direttamente, sia in camera.
L'accettazione, il bar, il ristorante sono dotati di un terminale fisso, mentre i camerieri e il personale delle pulizie sono dotati di dispositivi palmari collegati con la rete wireless dell'albergo. Inoltre, come già detto, in ogni camera è presente un TV-terminale a disposizione del cliente. Tutti i terminali sono collegati al sistema centrale che gestisce prenotazioni, pulizie e ordinazioni; il sistema è anche collegato tramite un servizio web ai mezzi esterni di prenotazione.
Il personale dell'accettazione usa il sistema per verificare le prenotazioni, registrare gli arrivi e le partenze e per conoscere il saldo dovuto da ogni cliente. Il personale addetto al bar e al ristorante riceve le ordinazioni dei clienti (di persona, dalla TV o via telefono) e segnala al momento della consegna che l'ordinazione è stata soddisfatta, se l'ordinazione è stata richiesta di persona. Lo stesso personale, inoltre usa il sistema per calcolare il costo dell'ordinazione e segnalarlo all'accettazione per il successivo addebito (il listino costi è immesso e gestito dal personale addetto al bar e al ristorante). I camerieri accedono alla lista delle ordinazioni fatte dai clienti via TV o telefono e segnalano l'avvenuta consegna in camera. Il personale delle pulizie accede alla lista delle camere da pulire e, per ogni camera, segnala sia se la pulizia è stata fatta che se è stata rinviata.
I terminali fissi sono dotati di un'apposita interfaccia grafica, i palmari usano un'interfaccia web.

Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi o attributi tutti e soli i termini che nell'enunciato compaiono in grassetto.

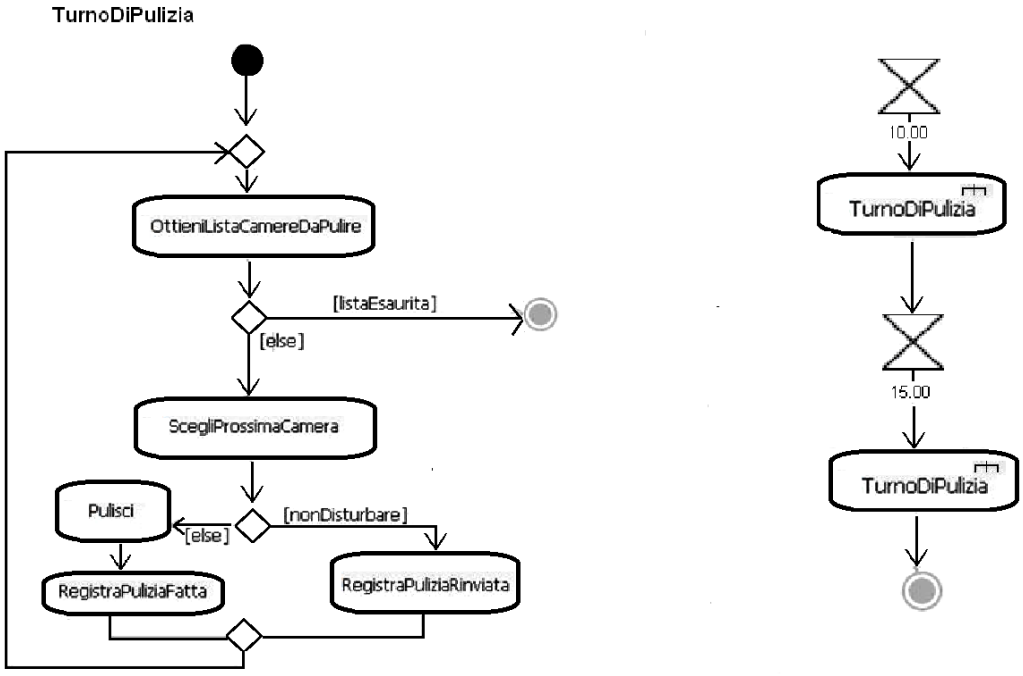
Risposta. Una possibile soluzione è la seguente:



Domanda. (Analisi del dominio). Si dia un diagramma di attività che descriva l'attività di pulizia delle camere, una volta realizzato il sistema.

Risposta. Nel seguito, il diagramma di destra mostra il comportamento di *TurnoDiPulizia*, utilizzato nel diagramma di sinistra, che descrive l'attività giornaliera di ciascun addetto.

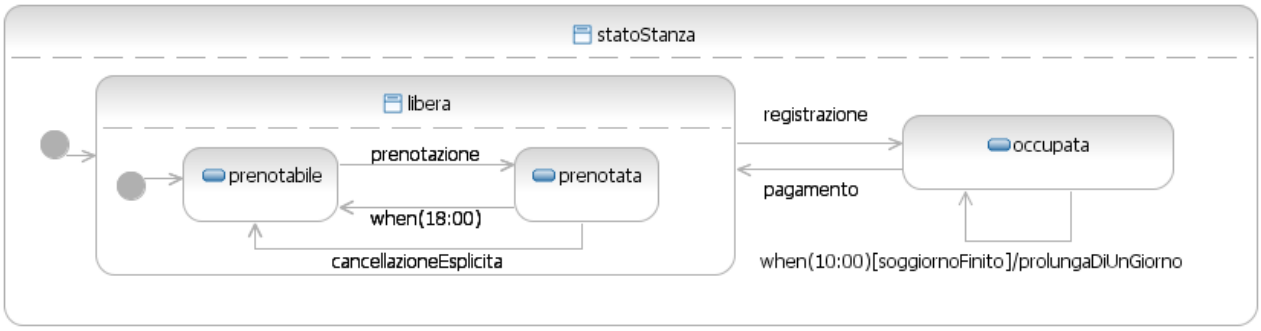
Si noti che *OttieniListaCamereDaPulire* si effettua all'interno del ciclo, perché la lista può essere aggiornata da altri addetti, che operano in parallelo.
Si noti anche che non ci sono controlli sulla fine dei turni, in base all'assunzione che se il personale non riesce a svolgere il lavoro nel tempo previsto, effettui degli straordinari, i cui compensi siano calcolabili in base alle informazioni registrate dal sistema.



Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descrive l'evoluzione dello stato di una stanza assumendo che sia inizialmente libera e prenotabile, che

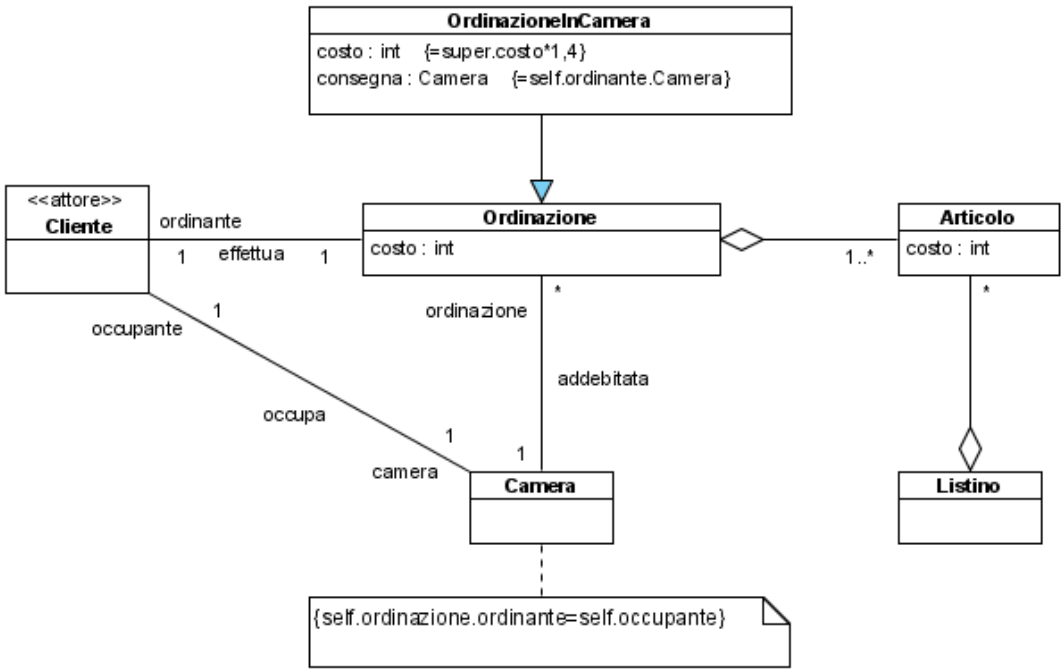
debba essere liberata entro le 10, pena il pagamento di un'altra notte, e che, in caso si arrivi tardi, le prenotazioni debbano essere confermate entro le 18, pena la cancellazione. (Non devono essere considerate le pulizie)

Risposta. Una possibile soluzione è la seguente:



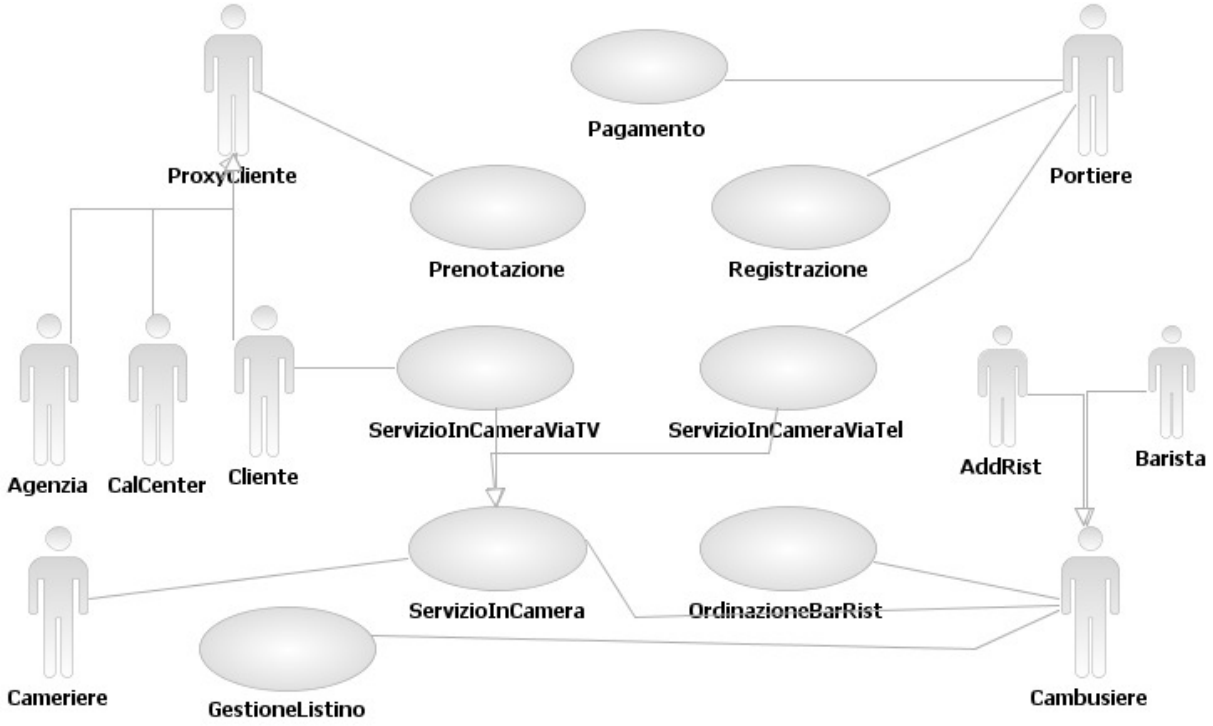
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva la parte del dominio relativa alle ordinazioni. Devono essere considerati come classi o attributi quelli del seguente elenco, più eventuali classi o attributi ausiliari, necessari per esprimere i vincoli: **cliente, ordinazione, costo, listino, camera**. Si esprimano i seguenti vincoli: il costo di un'ordinazione è addebitato alla camera occupata dal cliente che ha richiesto l'ordinazione; il servizio in camera è previsto solo per la camera di chi ha fatto la richiesta; il servizio in camera prevede una maggiorazione del costo dell'ordinazione pari al 40%.

Risposta. Una possibile soluzione è la seguente:



Domanda. (Analisi dei requisiti) Dare un diagramma dei casi d'uso del sistema, con relative brevi descrizioni, trascurando gli aspetti relativi alle pulizie. Si richiede, per le ordinazioni al bar e ristorante, di definire una gerarchia, per tener conto delle varie forme di servizio in camera. Si

considerino l'attore Cambusiere, quale generalizzazione di Barista e di Addetto al Ristorante, e l'attore ProxyCliente quale generalizzazione di Cliente, CallCenter, Agenzia.
Risposta. Una possibile soluzione è la seguente:



Domanda. (Analisi dei requisiti) Dare la narrativa del caso d'uso *ServizioInCameraViaTV*. Per semplicità si consideri il caso d'uso come se fosse a sé stante e non come specializzazione di un altro caso d'uso.
Risposta. Una possibile narrativa è la seguente:
ServizioInCameraViaTV
Breve descrizione: *Permette di gestire le ordinazioni da servire in camera*
Attore principale: *Cliente*
Attore secondario: *Cambusiere*, *Cameriere*
PreCondizioni: *Nessuna*
PostCondizioni: *Ordinazione consegnata e addebitata*
Sequenza principale degli eventi:

1. *Il Cliente inserisce l'ordine;*
2. *Il Cambusiere, preparato il vassoio, immette la richiesta di consegna;*
3. *Il Cameriere accetta di consegnare;*
4. *Il Sistema avverte gli altri camerieri;*
5. *Il Cameriere, consegnato il vassoio, immette l'evasione dell'ordine;*
6. *Il sistema addebita l'ordinazione al conto del Cliente*

Sequenza alternativa degli eventi:

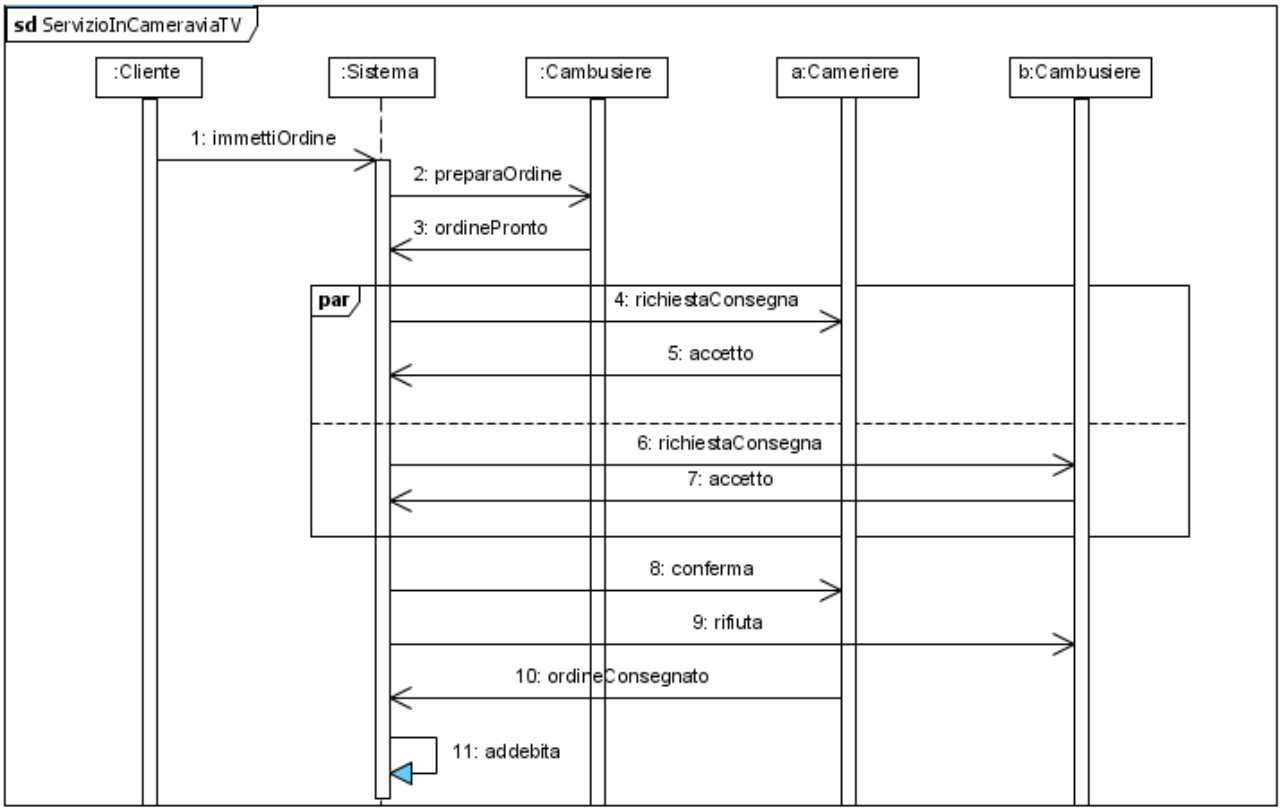
1. Il Cameriere inciampa e rovescia il vassoio

L'analisi dei requisiti ha portato all'individuazione, tra l'altro, del caso d'uso *PuliziaCamera*, in cui un *AddettoPulizia* richiede la lista delle camere da pulire, ne sceglie una, e quindi registra l'avvenuta pulizia oppure il rinvio, a causa del cartello "Non disturbare".
Domanda. (Analisi dei requisiti). Dare un diagramma di sequenza del caso d'uso *PuliziaCamera*.

Risposta. Una possibile soluzione è la seguente:

Domanda. (Realizzazione del caso d'uso) Dare un diagramma di sequenza che realizzi il caso d'uso ServizioInCameraViaTV, nel caso in cui due camerieri accettino di portare il vassoio (e solo uno sia selezionato).

Risposta. Una possibile soluzione è la seguente.



La progettazione architettonica ha portato a individuare, tra l'altro, un sotto-sistema di gestione delle pulizie, con le componenti descritte nella seguente tabella:

Sotto-sistema di gestione delle pulizie	
Componente	Responsabilità
AdminPuliziaCamere	Ogni mattina carica il *DBPuliziaCamere* con la situazione di occupazione delle stanze. Realizza funzioni di gestione del

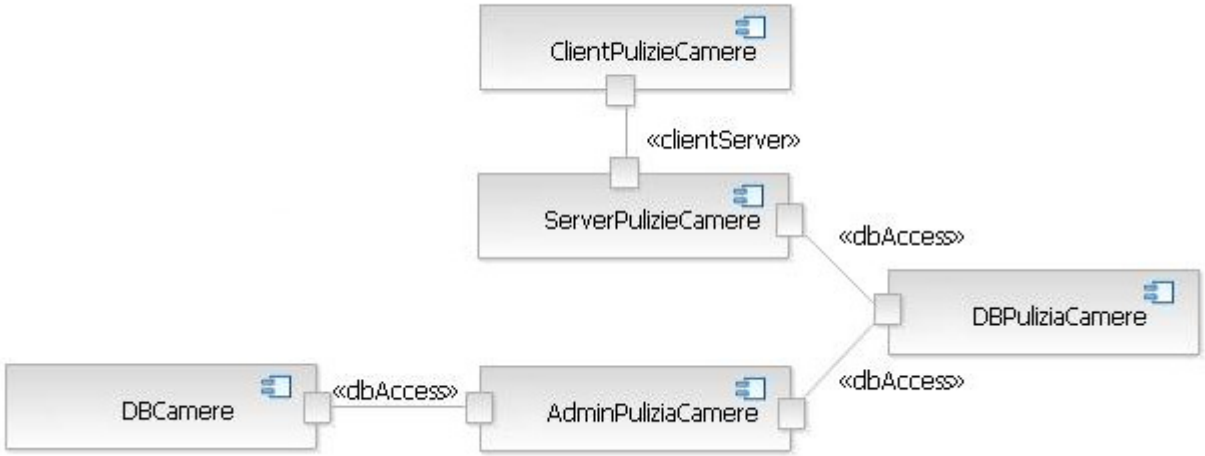
	personale di pulizia.
DBPuliziaCamere	Tiene conto dello stato di pulizia corrente delle camere, e di informazioni correlate, a fini gestionali.
ServerPulizieCamere	Interfaccia i palmari degli addetti al *DBPuliziaCamere*.
ClientPulizieCamere	Gestisce le interazioni degli addetti alla pulizia con il sotto-sistema di gestione delle pulizie.

Si tenga conto inoltre che il sistema completo deve contenere una componente che memorizza tutte le informazioni relative alle camere, come indicato dalla seguente porzione di tabella relativa al sistema.

Sistema di gestione Hotel dei Fiori	
Componente	Responsabilità
DBCamere	Tiene conto delle informazioni relative alle camere, ivi compreso il loro stato di occupazione.
…	…

Domanda. (Architettura). Dare una vista C&C relativa alle componenti elencate sopra.

Risposta. Una possibile soluzione è la seguente:



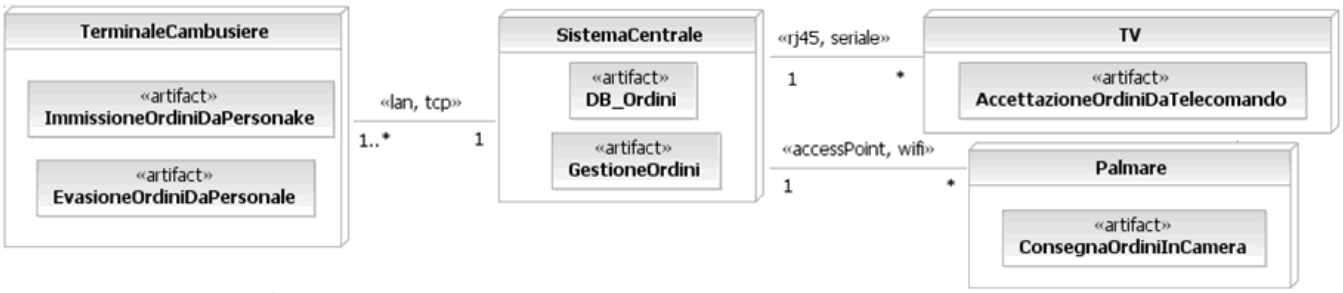
La progettazione architettonica ha portato a individuare, tra l'altro, un sotto-sistema di gestione del servizio in camera, con le componenti descritte nella seguente tabella:

Sotto-sistema OrdiniInCamera	
Componente	Responsabilità
AccettazioneOrdini DaTelecomando	Raccoglie l'ordine dato dal cliente tramite telecomando TV, e lo trasmette alla componente di GestioneOrdini.
ImmissioneOrdiniDa Personale	Raccoglie l'ordine dato dal cliente a un membro del personale, e lo trasmette alla componente di GestioneOrdini.
ConsegnaOrdiniIn	Avvisa il cameriere che usa il palmare che ci sono ordini in

Camera	camera da servire, gli permette di offrirsi di servirlo, e in seguito di dichiarare evaso l'ordine.
EvasioneOrdiniDa Personale	Fornisce al personale la lista degli ordini da servire e permette di dichiararli evasi (singolarmente).
DB_Ordini	Memorizza gli ordini e il loro stato
GestioneOrdini	Coordina le altre componenti, in modo che ogni ordine venga soddisfatto e quindi addebitato al cliente.

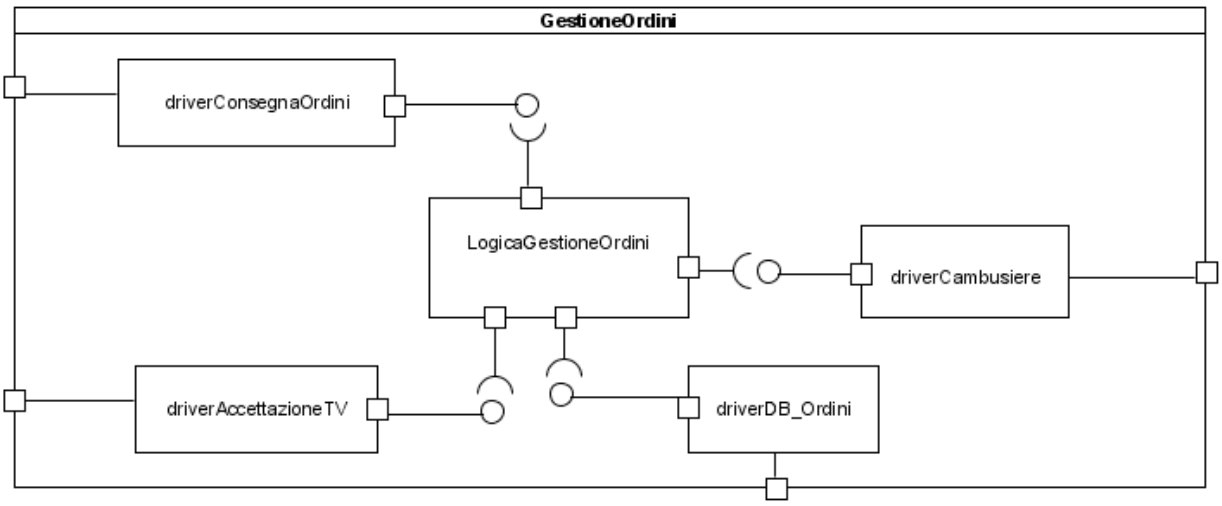
Domanda. (Architettura). Dare una vista di dislocazione del sotto-sistema OrdiniInCamera, assumendo che ogni componente sia manifestata da un omonimo artefatto.

Risposta. Una possibile soluzione è data dal seguente diagramma di dislocazione:



Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita che descriva la struttura del componente GestioneOrdini.

Risposta. Una possibile soluzione è la seguente.



L'albergo ha il seguente listino prezzi:

Camera singola	50,00 €
Camera doppia	80,00 €
Suite	100,00 €
Ordinazione al bar	5,00 €
Ordinazione al ristorante	10,00 €

Ordinazione in camera	15,00 €

Il metodo `CostoSoggiorno` calcola il costo del soggiorno a partire dal tipo e dal numero delle camere occupate, dalla durata del soggiorno, dal numero di ordinazioni al bar, al ristorante, in camera.

Domanda. (Verifica) Fornire cinque casi di prova per il metodo .

Risposta. Una possibile soluzione è la seguente:

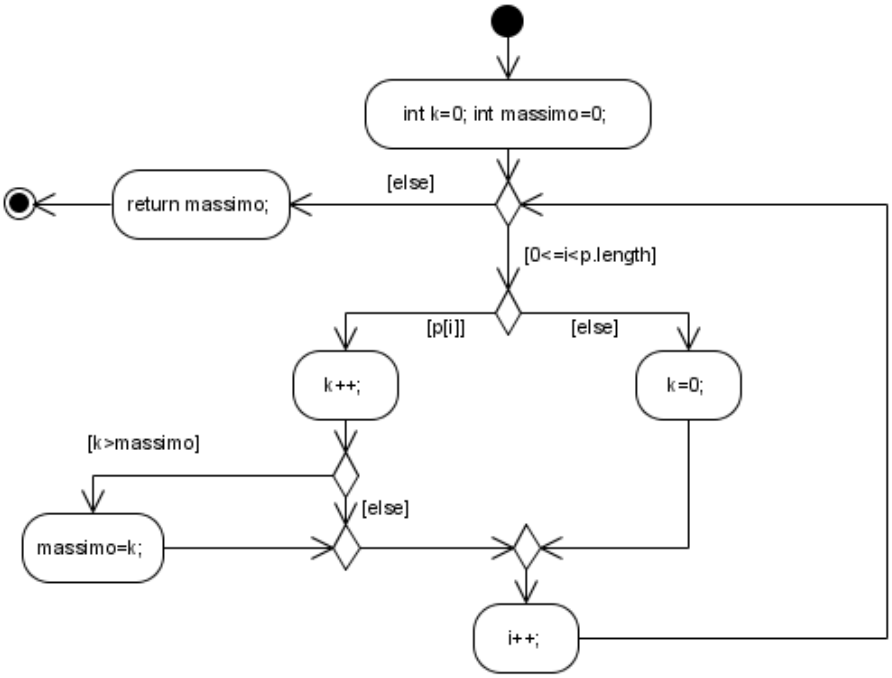
Durata	N. singole	N. doppie	N. suite	N. ord. bar	N. ord. rist.	N. ord. cam.	Costo totale
1	1	0	0	0	0	0	50
2	1	0	0	1	0	0	105
2	0	1	0	0	1	0	170
2	0	0	1	0	0	1	215
2	1	1	1	1	1	1	490

Il seguente metodo determina la durata del più lungo periodo di occupazione di una stanza in una settimana.

```
public int massimoPeriodo (boolean [] p) {
    int k = 0;
    int massimo = 0;
    for (int i = 0; i < p.length; i++) {
     if (p[i]) {
         k++;
         if (k > massimo) {
          massimo = k;
         }
     } else {
         k = 0;
     }
    }
    return massimo;
}
```

Domanda. (Verifica).
a) Disegnare il grafo di flusso (o grafo di controllo) del metodo.
b) Dare un insieme di cardinalità minima di casi di prova per la copertura delle decisioni.

Risposta. Una possibile soluzione è la seguente:



Un insieme minimale di casi di prova che soddisfa la copertura richiesta è il seguente:

input	output
T F T	1

CAP 18. Snel-Incheck-Doorgang.

Meno code al check-in, in futuro per chi parte da Amsterdam. L'aeroporto di Schiphol ha infatti commissionato un nuovo sistema, Snel-Incheck-Doorgang, che permette ai **passeggeri** di fare il check-in, oltre che dai tradizionali desk, anche nei box self-service o via Internet, e di fare il check-in di un eventuale **bagaglio** attraverso un nuovo macchinario.

La prima fase prevede il check-in dei passeggeri. I box self-service sono videoterminali touch-screen, via Internet viene offerta un'interfaccia web. In ogni caso il passeggero deve avere a disposizione il **codice** di **prenotazione** oppure il numero di passaporto e il numero del **volo**. A questo punto sceglie il suo **posto** e controlla dove è situato sull'**aereo;** infine stampa la sua carta d'imbarco, con la quale si reca al controllo sicurezza da effettuarsi primadell'imbarco. L'addetto alla sicurezza segnala attraverso un terminale che il controllo è stato effettuato.

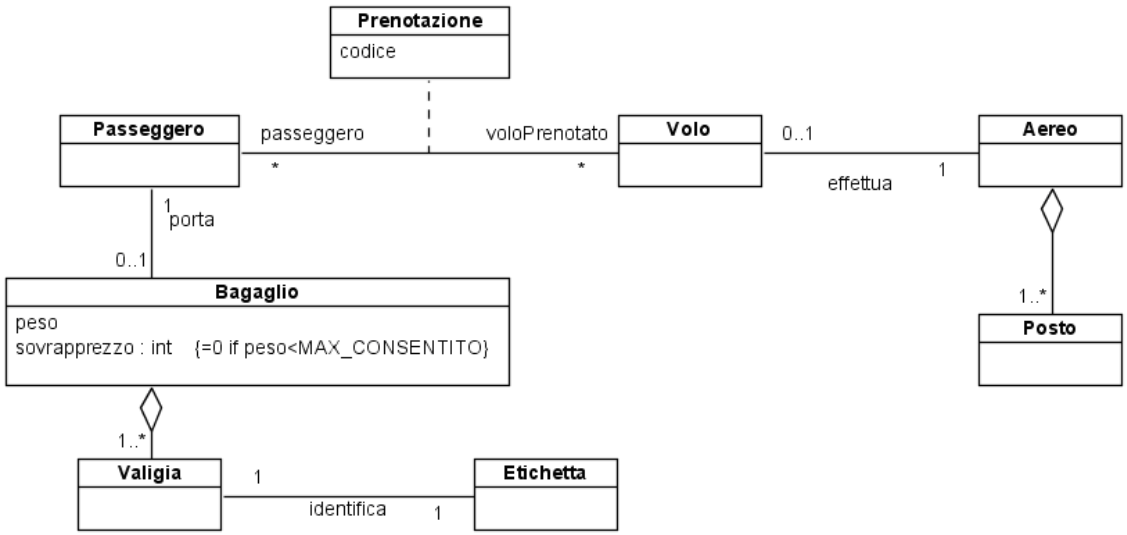
L'interfaccia è disponibile in olandese, inglese, francese, tedesco, italiano e spagnolo. E' permesso fare il check-in al più presto 24 ore prima del volo. Se il ritorno è previsto nelle successive 24 ore, è possibile fare il check-in anche per tale volo.

Per quanto riguarda il check-in dei bagagli, i passeggeri pongono i propri bagagli in un ampio macchinario bianco, inseriscono la propria carta di imbarco e rispondono alle abituali domande sul contenuto attraverso un touch-screen. La macchina pesa il bagaglio e stampa un'**etichetta** per ogni **valigia**. Il viaggiatore attacca le etichette al bagaglio. Il bagaglio viene inoltrato e sarà soggetto alle stesse misure di sicurezza adottate nei tradizionali desk.

Nel caso di problemi (ad esempio **peso** dei bagagli superiore al massimo consentito, nel quel caso al bagaglio è imposto un **sovrapprezzo**) sarà un impiegato ad un desk tradizionale a effettuare il check-in da un desk tradizionale.

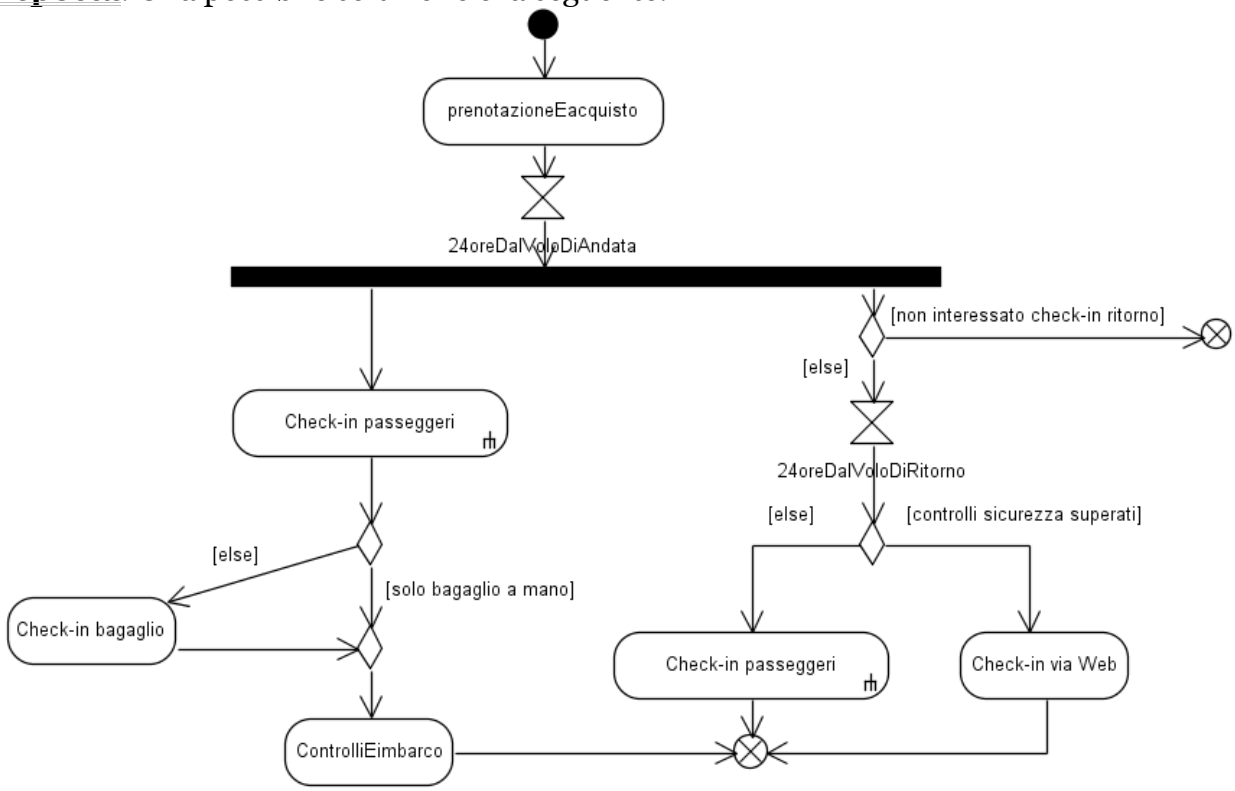
Domanda. (Analisi del dominio) Dare un diagramma delle classi che descriva parte del dominio. Devono essere considerati come classi o attributi quelli in neretto nel testo. Si richiede di esprimere la relazione tra Passeggero e Volo come *classe associazione* e di esprimere il vincolo che il sovrapprezzzo del bagaglio è 0 se non supera il peso massimo consentito.

Risposta. Una possibile soluzione è la seguente:

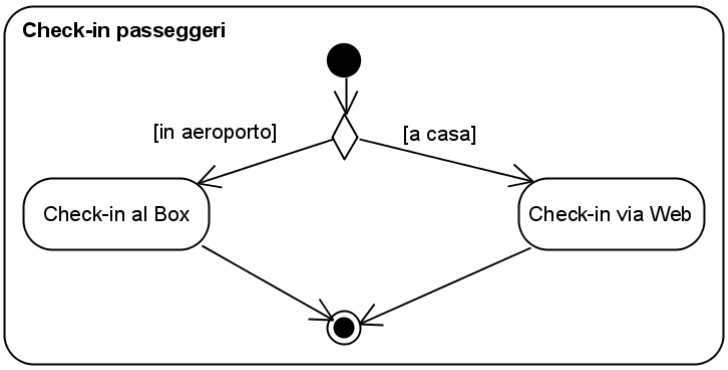


Domanda. (Analisi del dominio) Dare un diagramma di attività che descriva il comportamento di un passeggero dall'acquisto del biglietto ai controlli e imbarco.
Per descrivere il check-in dei passeggeri usare un'azione che chiama una sotto-attività (rastrello). Usare una fork per la descrizione di: "E' permesso fare il check-in al più presto 24 ore prima del volo. Se previsto nelle successive 24 ore, è possibile fare il check in anche per il volo di ritorno."

Risposta. Una possibile soluzione è la seguente:

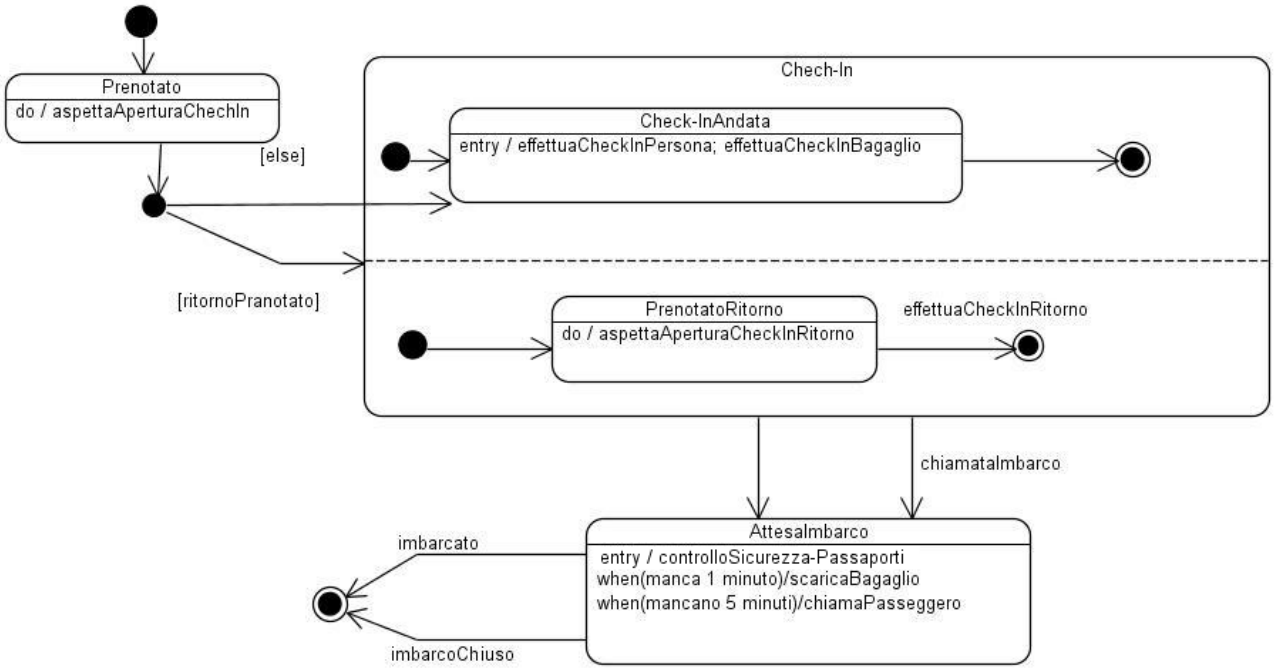


dove



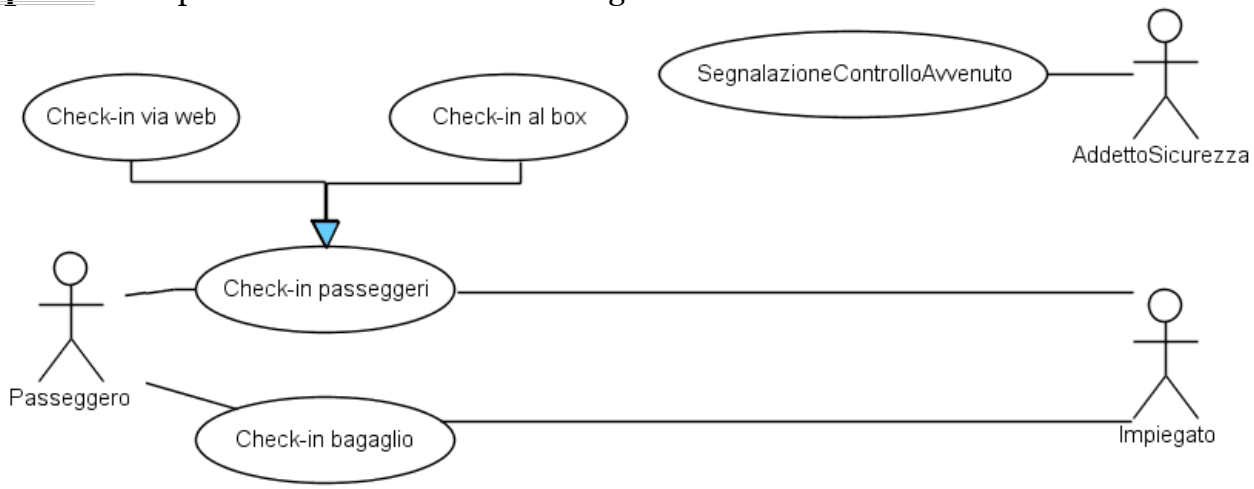
Domanda. (Analisi del dominio) Dare un diagramma di macchina a stati che descriva un passeggero tenendo conto, oltre a quanto già descritto, che a 5 minuti dalla partenza del volo i passeggeri che abbiano fatto check-in e non siano imbarcati vengono chiamati. A 1 minuto dalla partenza, se non ancora imbarcati, viene scaricato l'eventuale bagaglio . Quando viene chiamato l'imbarco non è più possibile fare il check in per l'eventuale volo di ritorno.

Risposta. Una possibile soluzione è la seguente:



Domanda. (Analisi dei requisiti) Dare un diagramma dei casi d'uso del sistema *Sistema Snel-Incheck-Doorgang*. Fornire le brevi descrizioni

Risposta. Una possibile soluzione è data di seguito.



Un successivo raffinamento dei requisiti ha portato a distinguere tra il check-in self-service e il check-in al desk del bagaglio. Si consideri la seguente narrativa:

Check-in bagagli self-service

Breve descrizione: *Un passeggero registra i propri bagagli per l'imbarco.*
Attore principale: *Passeggero*
Attori secondari: *Nessuno*
PreCondizioni: *E' già stato effettuato il check-in del passeggero.*
PostCondizioni: *Il bagaglio è registrato per l'imbarco.*

Sequenza principale degli eventi:

1. *il passeggero inserisce la propria carta di imbarco;*
2. *il passeggero risponde alle domande sul contenuto del bagaglio;*
3. *per ogni valigia*

3.1. il passeggero posiziona il proprio bagaglio nella macchina;

3.2. la macchina pesa il bagaglio

3.3. la macchina stampa un'etichetta, che il passeggero attacca alla valigia.

4. *Il passeggero ritira la carta di imbarco*

Sequenza alternativa degli eventi:

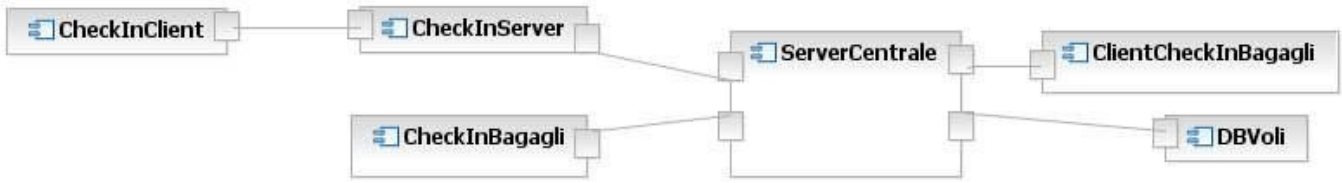
peso dei bagagli superiore al massimo consentito.

L'architettura del sistema di check-in ha le componenti descritte nella seguente tabella:

Nome	Descrizione
CheckInClient	Scaricata nel browser di un utente remoto, gli permette di fare check-in per un volo prenotato, a partire da 24 ore prima del volo. Se previsto nelle successive 24 ore, permette anche il check-in per il volo di ritorno.
CheckInServer	Gestisce i CheckInClient, interagendo indirettamente con il DBVoli via il ServerCentrale.
DBVoli	Mantiene le informazioni sui voli, comprese quelle dei bagagli.
ServerCentrale	Smista le informazioni tra le componenti, filtrando quelle rilevanti anche per gli altri sottosistemi del sistema di gestione dell'aeroporto.
CheckInBagagli	Controlla il check-in automatico dei bagagli, gestendo il lettore delle carte d'imbarco, il touch-screen per le domande sul contenuto, la bilancia e la stampante delle etichette. Inoltre, interagisce con il ServerCentrale per trasmettere le informazioni sui bagagli presi in carico.
ClientCheckInBagagli	Permette a un impiegato di fare il check-in del bagaglio, in caso di problemi con il sistema automatico.

Domanda. (Architettura) Dare una vista C&C dell'architettura del sottosistema di check-in descritta dalla tabella precedente.

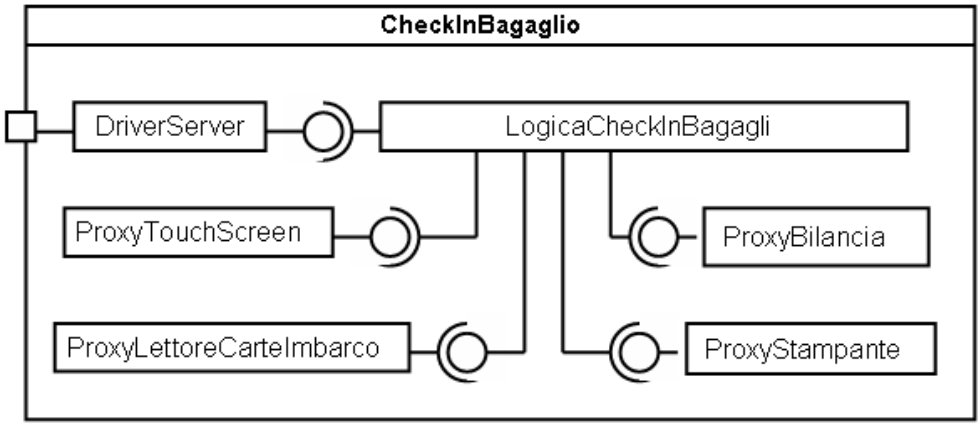
Risposta. La soluzione è data dal seguente diagramma delle componenti, dove tutti i connettori sono di tipo client server.



La componente CheckInBagagli, oltre a interagiRe col server centrale, gestisce alcune periferiche: il lettore delle carte d'imbarco, il touch-screen per le domande sul contenuto, la bilancia e la stampante delle etichette.

Domanda. (Progettazione di dettaglio) Dare un diagramma di struttura composita che descriva la struttura del componente CheckInBagaglio.

Risposta. Una possibile soluzione è la seguente.



Nel rispondere alla domanda seguente, si tenga conto che nei diagrammi di attività UML è possibile utilizzare anche archi uscenti da un nodo attività, con lo stereotipo <<exception>>, per indicare una terminazione anomala dell'attività.

Domanda. (Progettazione di dettaglio) Dare un diagramma di attività che descriva il comportamento del componente CheckInBagaglio.

Risposta. Una possibile soluzione è la seguente.

Un array P di booleani, con R righe e C colonne può rappresentare lo stato di occupazione di un aereo: dato un volo su un aereo con R file di posti, ciascuna con C poltrone, un posto in posizione [i,j] è libero se e solo se P[i,j] vale true.
La funzione `prenota`, dato un array P di booleani, due interi positivi R e C (numero di righe, numero di colonne), un intero K compreso tra 1 e 3, cerca in P una riga in cui sono liberi K posti adiacenti. In caso affermativo restituisce la posizione del primo posto trovato, altrimenti restituisce `null`.

Domanda. (Verifica) Si fornisca un insieme di almeno cinque casi di prova per la funzione `prenota`, assumendo K = 2. Si trascuri l'ambiente, e si giustifichino le scelte fatte.

Risposta. Un possibile insieme di casi di prova è dato dalla seguente tabella, assumendo gli indici di riga e colonna in [0,R) e [0,C) rispettivamente:

Input (fissati R=10, C=6, K=2)	Output	Motivazione
tutti gli elementi hanno valore true	[0,0]	Caso estremo globale favorevole
tutti gli elementi hanno valore false	`null`	Caso estremo globale sfavorevole
tutti gli elementi, tranne uno, hanno valore false	`null`	Caso estremo globale sfavorevole
tutti gli elementi, tranne i primi due della prima riga, hanno valore true	[0,2]	Caso tipico favorevole
tutti gli elementi, tranne i primi cinque della prima riga, hanno valore true	[1,0]	Caso estremo locale (sulla riga) favorevole
Ogni coppia di elementi adiacenti ha un valore true e uno false	`null`	Caso estremo sfavorevole (massimo numero di posti liberi) (sussume il caso tipico sfavorevole)

La funzione costoBagali ha tre argomenti: un peso, una tabella con i valori che definiscono le soglie di soprappeso, la tariffa massima. La funzione restituisce il costo del bagaglio secondo il seguente schema: se il peso è uguale o inferiore al primo elemento della tabella il costo è zero, se il peso è superiore all'ultimo elemento della tabella il costo è pari alla tariffa massima, negli intervalli di peso intermedi si considera come peso l'estremo inferiore dell'intervallo e il costo è proporzionale alla tariffa massima.

```
public int costoBagaglio (int peso, int [] tabella, int maxTariffa) {
    int i = 0;
    while (i < tabella.length && peso > tabella[i])
       i++;
return (maxTariffa / tabella.length) * i;  }
```

Domanda. (Verifica)
a) Fornire un insieme di casi di prova basati su criteri black box.
b) Fornire inoltre dei casi di prova basati su criteri white box, con copertura completa dei cicli.
Risposta. Una possibile soluzione è la seguente:
Il seguente insieme di casi di prova soddisfa un criterio black box che considera input validi e valori non limite. Inoltre permette la copertura completa dei cicli.

peso	Tabella	tariffa massima	risultato atteso
5	{10, 20, 30}	30	0
15	{10, 20, 30}	30	10
25	{10, 20, 30}	30	20
35	{10, 20, 30}	30	30

Il seguente insieme di casi di prova soddisfa un criterio black box che considera input non validi e valori limite.

peso	Tabella	tariffa massima	risultato atteso
5	{}	30	Eccezione: divisione per zero
10	{10, 20, 30}	30	0
11	{10, 20, 30}	30	10
20	{10, 20, 30}	30	10
21	{10, 20, 30}	30	20
30	{10, 20, 30}	30	20
31	{10, 20, 30}	30	30

CAP 19. Tirocini formativi

1 Introduzione

L'obiettivo del progetto è automatizzare le procedure per l'assegnamento dei tirocini agli studenti del corso di laurea in informatica, rendendo completamente automatico l'assegnamento. Lo scopo è di permettere agli studenti di esprimere le loro preferenze in modo accurato, di soddisfarle al meglio e di assegnare ai proponenti (enti, aziende) tirocinanti che abbiano le competenze richieste.

1.2 Riferimenti

I regolamenti e la procedura per l'assegnamento dei tirocini e le FAQ sono consultabili alla pagina: http://compass2.di.unipi.it/didattica/inf/tirocini/tirocini_testo.asp .

2 Descrizione del dominio e dei requisiti

L'ordinamento didattico del corso di laurea in informatica prevede lo svolgimento di un tirocinio, in dipartimento o presso un'azienda o un ente esterno. I tirocini proposti si differenziano per il carico di lavoro, espresso in crediti formativi - CFU (12 o 18), e per il numero di studenti richiesti: uno o due (tirocinio in coppia).
Quando ha raggiunto il numero di CFU necessario (144 o 150 rispettivamente per quello da 18 o da 12), lo studente può fare richiesta di tirocinio durante una delle tre sessioni disponibili in ogni anno accademico.
Ogni sessione di tirocinio prevede cinque fasi:

1. la proposta di tirocinio da parte di docenti o esterni;
2. la valutazione, da parte della commissione, delle proposte di progetto, che possono essere accettate o rifiutate;
3. la pubblicazione dell'albo delle proposte di progetto assegnabili;
4. la presentazione delle domande da parte degli studenti;
5. l'assegnamento dei tirocini da parte della commissione.

Nella prima fase, i proponenti presentano la propria offerta alla commissione. L'offerta viene inserita via web, previa autenticazione. Ogni azienda può proporre uno o più tirocini e, per ognuno di essi, deve indicare:

- titolo;
- descrizione;
- tutore;
- esperienza formativa prevista;
- durata del progetto (12 o 18 crediti);
- numero di studenti richiesti (al più due);
- eventuali aree di interesse;
- eventuali esami vincolanti;
- sede del tirocinio ed eventuale rimborso delle spese;
- scadenza del tirocinio.

Gli esami vincolanti sono esami della laurea triennale, che il proponente può richiedere come prerequisito al tirocinante. Poiché in molti casi la nomenclatura degli esami è cambiata nel corso degli anni, l'azienda può scegliere da un elenco di nomi sui quali la commissione ha mappato i codici di tutti gli esami. Il numero massimo di esami vincolanti viene deciso dalla Commissione. La seconda fase inizia allo scadere del termine di presentazione delle offerte di tirocinio. In questa fase la commissione filtra le offerte e produce la lista dei tirocini assegnabili. La commissione aggiunge a questa lista anche tirocini delle sessioni precedenti, non assegnati ne scaduti
Terminata la valutazione dei tirocini, ne viene reso pubblico agli studenti l'albo, attraverso interfaccia web. Gli studenti, previa autenticazione, visionano la lista e compilano la domanda indicando:

- eventuali esami già sostenuti, ma non ancora registrati in segreteria;
- le preferenze per i tirocini;
- i crediti del tirocinio (da 12 e/o da 18);
- la matricola dell'eventuale compagno di tirocinio, unico per tutti i progetti.

2.1 Le preferenze

Una volta completata l'immissione degli esami non registrati, allo studente viene presentato l'albo, completo di eventuali messaggi di warning per i progetti sui quali non risulta idoneo perché i crediti non sono sufficienti o a causa degli esami vincolanti, ed esprime le preferenze, in due passi. Nel primo passo, obbligatorio, lo studente assegna ad ogni tirocinio una preferenza bassa (valore di default), media o alta, oppure lo classifica come inaccettabile: il numero dei tirocini che uno studente può classificare come inaccettabili non è limitato. Nel secondo passo, facoltativo, allo studente viene presentata la lista dei tirocini che ha classificato con preferenza alta. Per ognuno di essi lo studente può impostare un voto di preferenza compreso tra uno e dieci, ed è inizialmente impostato a 5.

Alla scadenza della presentazione delle domande, il sistema determina gli assegnamenti, utilizzando l'algoritmo dei matrimoni stabili, in base alle preferenze degli studenti e alle richieste dei proponenti i tirocini. In particolare, i tirocini di ogni studente sono ordinait in base alle preferenze, e, in caso di parità, in base all'interesse totale degli altri studenti. L'interesse totale di un tirocinio è calcolato come somma dei gradi di preferenza di tutti gli altri studenti che l'hanno richiesto. Fra i tirocini allo stesso livello, lo studente otterrà quello con interesse totale più basso, in modo da massimizzare il numero di progetti assegnati.

D'altra parte, per ogni tirocinio, gli studenti sono ordinati in base al numero delle aree di interesse coperte. Per coprire un'area lo studente deve aver conseguito un totale di almeno 6 crediti nell'area. In caso di pari numero di aree, si ordina in base alla somma dei punteggi (numero di crediti * voto degli esami) ottenuti nelle varie aree, e in caso di ulteriore parità si considera la media complessiva dello studente.

Due studenti in coppia sono tenuti ad esprimere la stessa preferenza per un tirocinio in coppia. Se ciò non avviene, il sistema interviene impostando per la coppia la preferenza massima fra i due studenti. Per coordinare la scelta, allo studente della coppia che compila per ultimo la domanda, vengono impostati, come valori di default per la preferenza dei progetti a coppie, quelli che ha espresso precedentemente il compagno. Per uno studente che non ha indicato un compagno, i tirocini a coppia vengono considerati come inaccettabili dal sistema. Infatti la politica della commissione prevede che i tirocini a coppia possano essere svolti solo da coppie "già formate", per evitare di unire studenti in modo arbitrario. Gli assegnamenti dei progetti in coppia hanno precedenza su quelli singoli, però solo all'interno di una determinata classe di preferenza (alta, media e bassa).

Gli accoppiamenti risultanti vengono memorizzati nella base di dati delle segreterie. Deve essere possibile visualizzarli, come pure gli studenti e i tirocini che sono rimasti liberi al termine dell'asseganzione.

3 Intervento manuale sugli assegnamenti

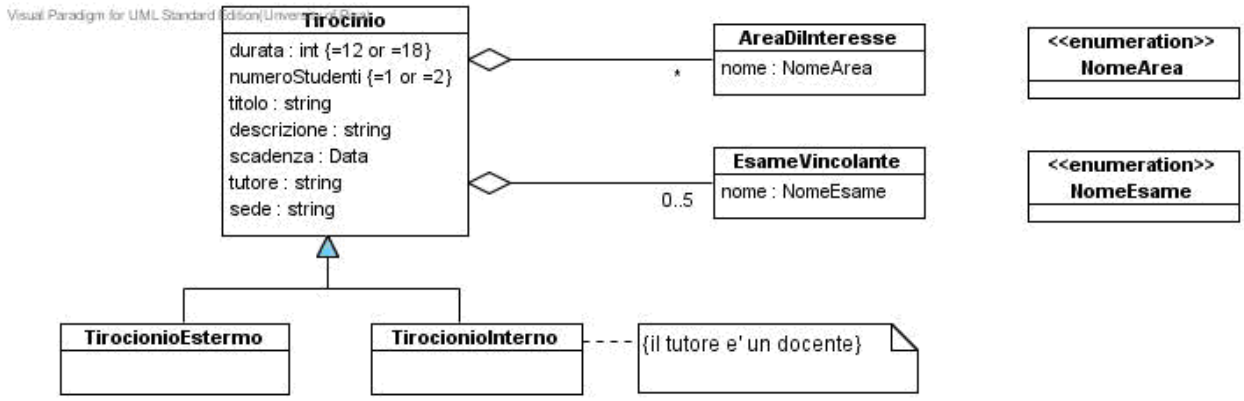
L'applicativo deve consentire alla commissione di poter intervenire sui dati iniziali e sull'output. Si, danno, infatti, scenari che non possono essere risolti in maniera automatica. Un esempio è dato dai tirocini procacciati: se uno studente procura un contatto fra un esterno e l'università, acquisisce come "premio" la priorità nell'assegnamento del tirocinio derivante dal contatto. In conclusione, le operazioni straordinarie sono l'assegnazione e la revoca manuale di un tirocinio.

4 Vincoli sulla piattaforma

L'applicazione sarà installata sul server Web compass2 (piattaforma Windows Server) e sarà utilizzabile tramite interfaccia Web. L'applicazione dovrà collegarsi al database Studenti delle segreterie (piattaforma Microsoft SQL Server). L'autenticazione avverrà interfacciandosi con il sistema di autenticazione del cli.

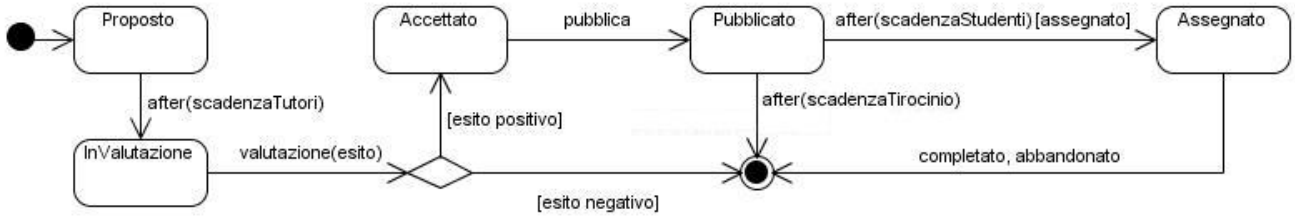
Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi, attributi tutti e soli i termini seguenti: tirocinio, tirocinio interno (al dipartimento), tirocinio esterno (azienda o ente), durata (in crediti), numero studenti, titolo, descrizione, tutore, area di interesse, nome, esame vincolante, sede, scadenza. Esprimere eventuali vincoli imposti dal testo. Si assuma inoltre che la commissione abbia posto un limite massimo di 5 agli esami vincolanti. Si assuma siano state definite le enumerazioni per nome esame e nome area.

Risposta.



Domanda. (Dominio) Dare un diagramma di macchina a stati che descrive l'evoluzione di un oggetto di tipo Tirocinio.

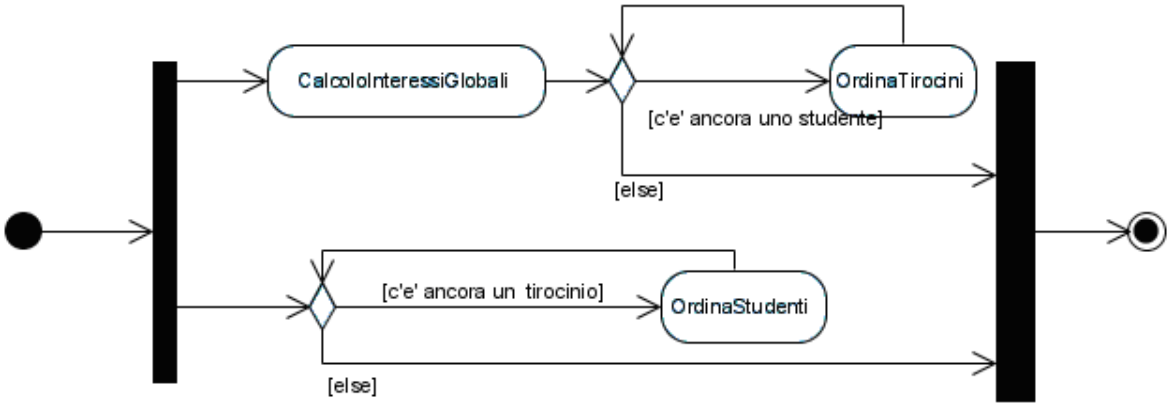
Risposta.



Si consideri la determinazione degli assegnamenti. Si ignori il problema dei tirocini a coppie.

Domanda. (Analisi del dominio). Fornire un diagramma delle attività che descriva il processo (non gli algoritmi) di ordinamento di studenti e tirocini, necessario per l'assegnazione.

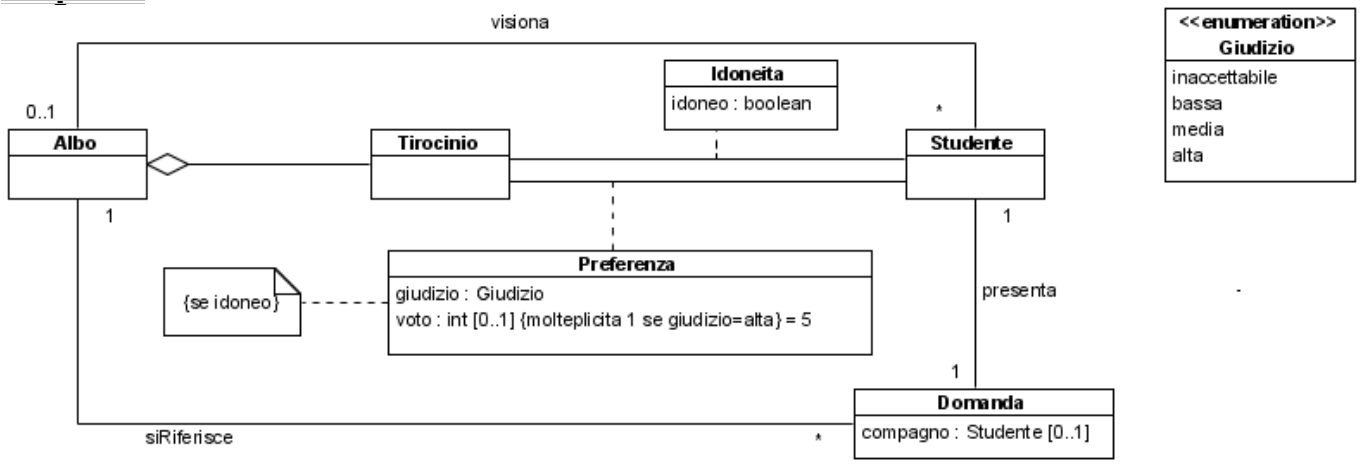
Risposta. Una possibile soluzione è la seguente.



Si consideri il primo paragrafo della sezione 2.1. Si ignori quanto detto successivamente per vincolare gli studenti in coppia a preferenze uguali.
Si considerino sinonimi Tirocinio e Progetto. Per chiarire una ambiguità: una Preferenza è caratterizzata da un giudizio (bassa…..) e eventualmente da un voto.

Domanda. (Analisi del dominio) Dare un diagramma delle classi, considerando come classi, attributi tutti e soli i termini seguenti: tirocinio, studente, idoneità, idoneo, preferenza, giudizio, inaccettabile, bassa, media, alta, voto, domanda, compagno, albo. Esprimere eventuali vincoli imposti dal testo. Per il giudizio, si definisca una classe enumerazione.

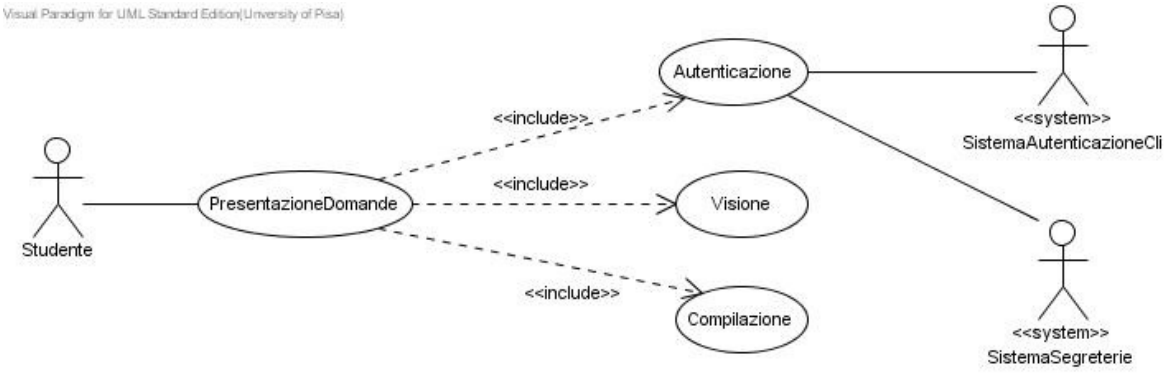
Risposta.



Il caso d'uso P*resentazioneDomande* prevede tre fasi: *Autenticazione* (comprende l'immissione degli esami non registrati), V*isione* (della lista progetti) e *Compilazione* (della domanda).

Domanda. (Analisi dei requisiti). a. Dare il diagramma dei casi d'uso del sistema limitatamente alla presentazione delle domande. b. Dare la narrativa di *Compilazione.*

Risposta. a.

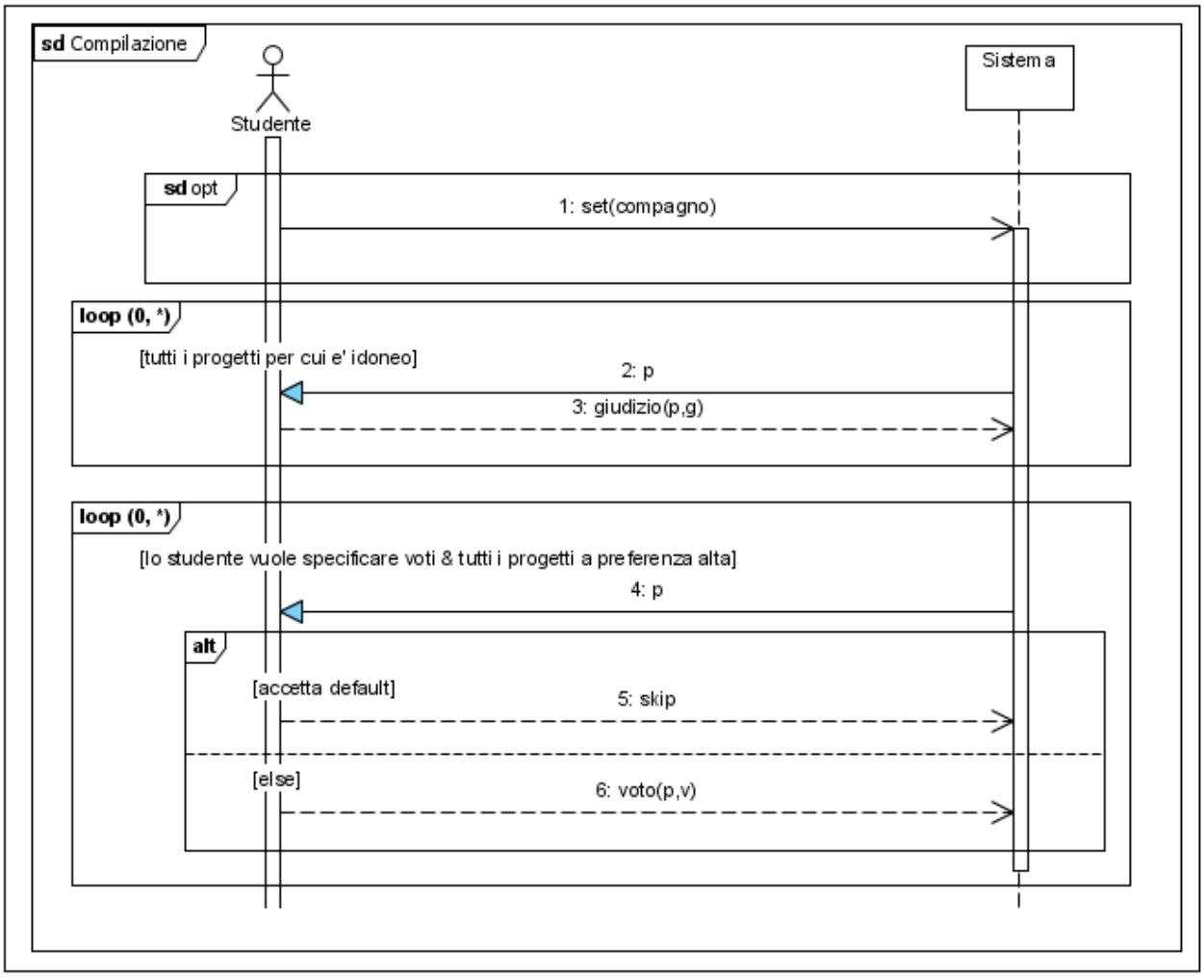


Risposta b.
Caso d'uso: **Compilazione**
Breve descrizione: *Permette di compilare la domanda di tirocinio.*
Attore principale: *Studente*
Attore secondario: *Nessuno*
PreCondizioni: *Studente autenticato, esami completi, lista progetti visualizzata.*
PostCondizioni: *Domanda compilata.*

Sequenza principale degli eventi:

1 **se** *(lo studente vuole)*
1.1 *lo Studente indica la matricola di un collega*
2 **per** *(ogni progetto)*
2.1 *lo Studente esprime un giudizio tra: inaccettabile, bassa, media, alta*
2.2 ***se*** *(giudizio alto)*
2.2.1 *il sistema assegna il voto 5*
3 **se** *(lo studente vuole)*
3.1 **per** *(ogni progetto con preferenza alta)*
3.1.1 **se** *(lo studente vuole)*
3.1.1.1 *lo Studente esprime un voto tra 1 e 10*

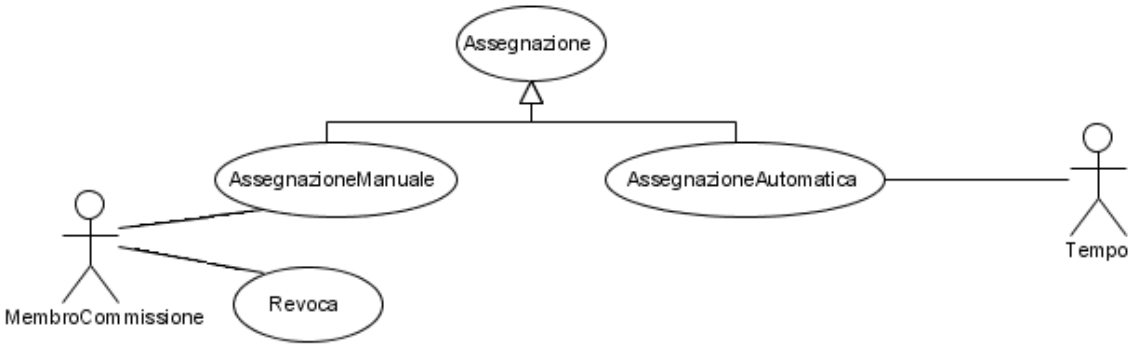
Domanda. (Analisi dei requisiti) Esprimere la narrativa di *Compilazione* con un diagramma di sequenza.
Risposta.



Si consideri la determinazione degli assegnamenti. Si ignori il problema dei tirocini a coppie.

Domanda. (Analisi dei requisiti).
a. Dare un diagramma dei casi d'uso parziale che comprenda l'assegnazione dei tirocini, sia manuale sia automatica, e la revoca.
b. Dare una possibile narrativa per il caso d'uso revoca.

Risposta. a.



b.

Revoca

Breve descrizione: *Viene disfatta una assegnazione tra uno studente e un tirocinio*
Attore principale: *MembroCommissione*
Attori secondari: *Nessuno*
PreCondizioni: *Tirocinio assegnato allo studente*
PostCondizioni: *Tirocinio non assegnato, studente non assegnato.*
Sequenza principale degli eventi:

1. *Il MembroCommissione richiama il tirocinio*
2. *Il sistema lo visualizza*
3. *Il MembroCommissione annulla l'assegnazione*
4. *Il sistema chiede conferma*
5. *Il MembroCommissione conferma*
6. *Il sistema modifica il tirocinio*

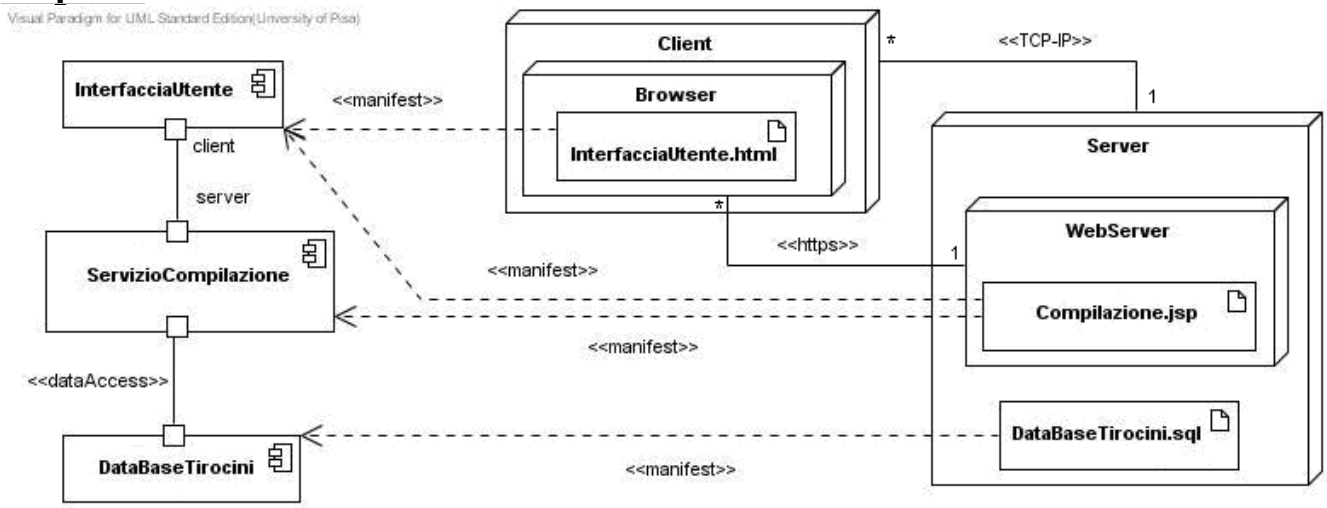
Sequenza alternativa degli eventi:

1. *Il MembroCommissione non conferma*
2. *Il MembroCommissione termina la sessione dopo la visione*

Per realizzare il caso d'uso Compilazione sono necessarie tre componenti connesse in stile multi-tier : *InterfacciaUtente*, *ServizioCompilazione* e *DataBaseTirocini.*

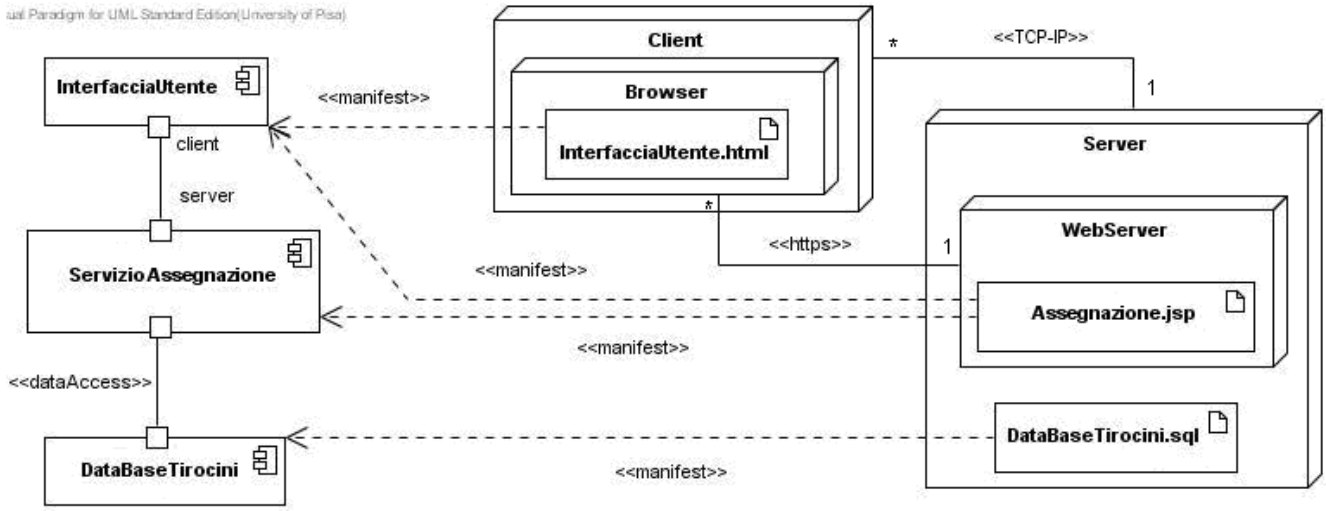
Domanda. (Architettura) Fornire una vista ibrida (C&C e dislocazione) dell'architettura del sotto-sistema di Compilazione, assumendo che gli artefatti che manifestano le componenti citate siano: *Compilazione.html*, visualizzato da un browser di una macchina client e *Compilazione.jsp* (dislocata su un web server) e *DataBaseTirocini.sql*, mantenute su una macchina server.

Risposta.



Si consideri la determinazione degli assegnamenti. Si ignori il problema dei tirocini a coppie. Per realizzare il caso d'uso Assegnazione automatica sono necessarie tre componenti connesse in stile multi-tier : *InterfacciaUtente*, *ServizioAssegnazione* e *DataBaseTirocini*.

Domanda. (Architettura) Fornire una vista ibrida (C&C e dislocazione) dell'architettura del sotto-sistema di Assegnazione automatica, assumendo che gli artefatti che manifestano le componenti citate siano: *Assegnazione.html*, dislocato su un browser di una macchina client e *Assegnazione.jsp* (dislocata su un web server) e *DataBaseTirocini.sql*, mantenute su una macchina server.
Risposta.



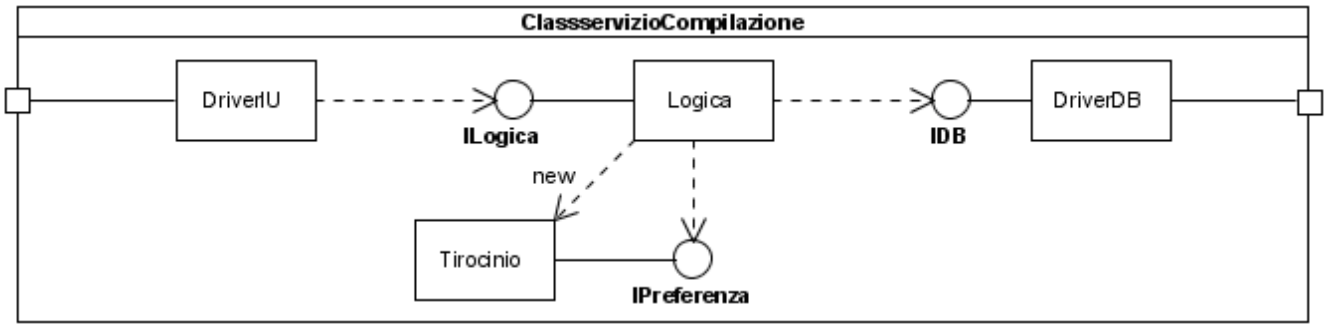
Per realizzare il caso d'uso Compilazione sono state progettate tre componenti connesse in stile multi-tier: *InterfacciaUtente*, *ServizioCompilazione* e *DataBaseTirocini*.
Il *ServizioCompilazione* riceve la richiesta di compilazione dall'interfaccia utente e richiede al DataBase i tirocini per cui lo studente è idoneo. Per ogni tirocinio:

- crea un oggetto di tipo Tirocinio;
- lo passa all'interfaccia utente;
- aspetta un giudizio di preferenza;
- aggiorna la base di dati con il giudizio: se la preferenza è alta indica 5 come voto, altrimenti indica 0 (serve per uniformare le tabelle e non ha valore in fase di assegnamento)
- distrugge l'oggetto Tirocinio

A questo punto, se lo studente lo desidera, il *ServizioCompilazione* richiede al DataBase i tirocini con preferenza alta. Per ogni tirocinio:

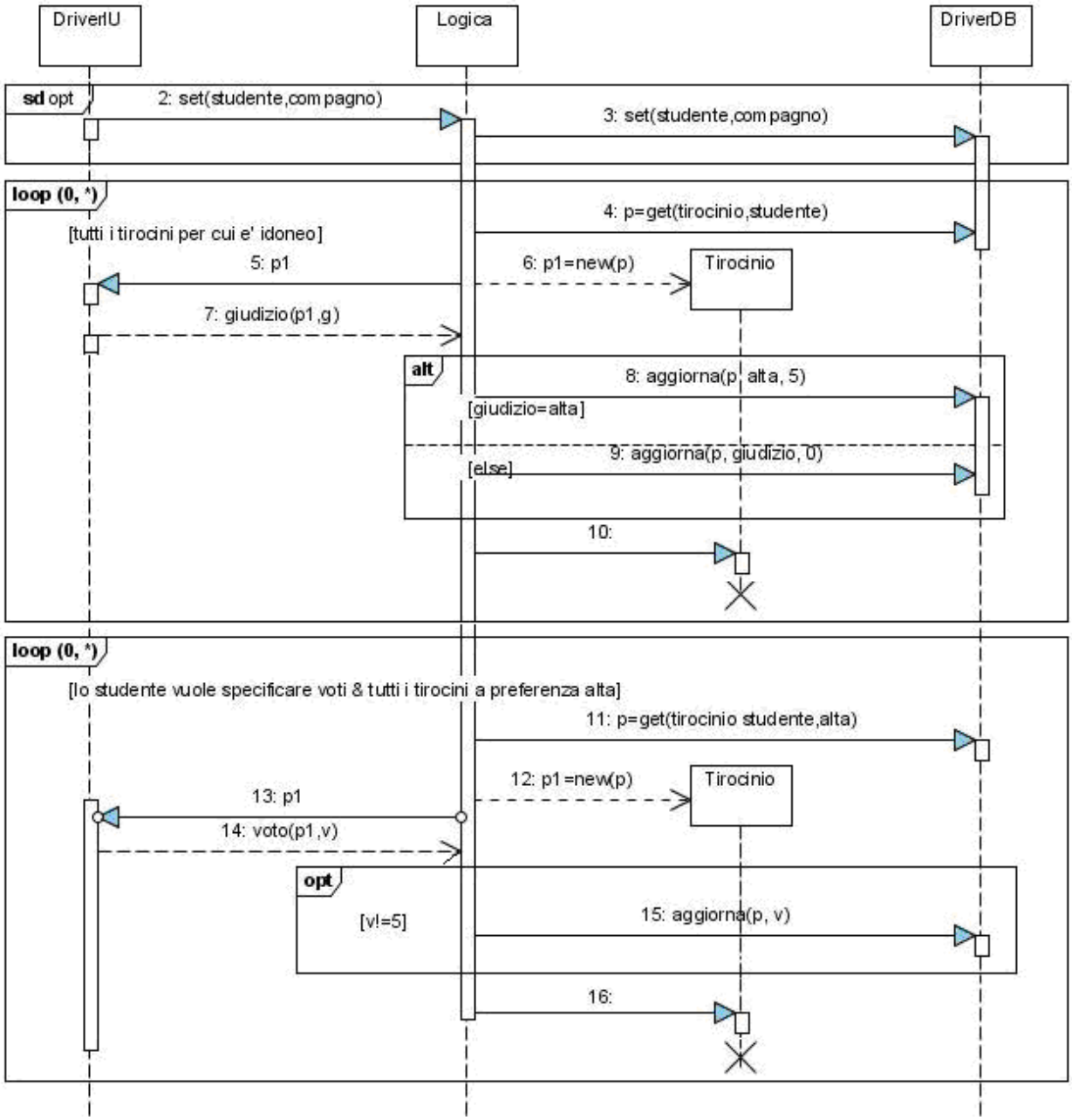
- crea un oggetto di tipo Tirocinio;
- lo passa all'interfaccia utente;
- aspetta un voto o un ok per il voto di default;
- aggiorna la base di dati se iol voto è diverso da 5;
- distrugge l'oggetto Tirocinio

Domanda. (Progettazione di dettaglio) Fornire un diagramma di struttura composita per la componente *ServizioCompilazione*.
Risposta.



Domanda. (Realizzazione dei casi d'uso) Fornire un diagramma di sequenza che mostra come la componente ServizioCompilazione, come dettagliata nel precedente esercizio, realizza il caso d'uso *Compilazione*.

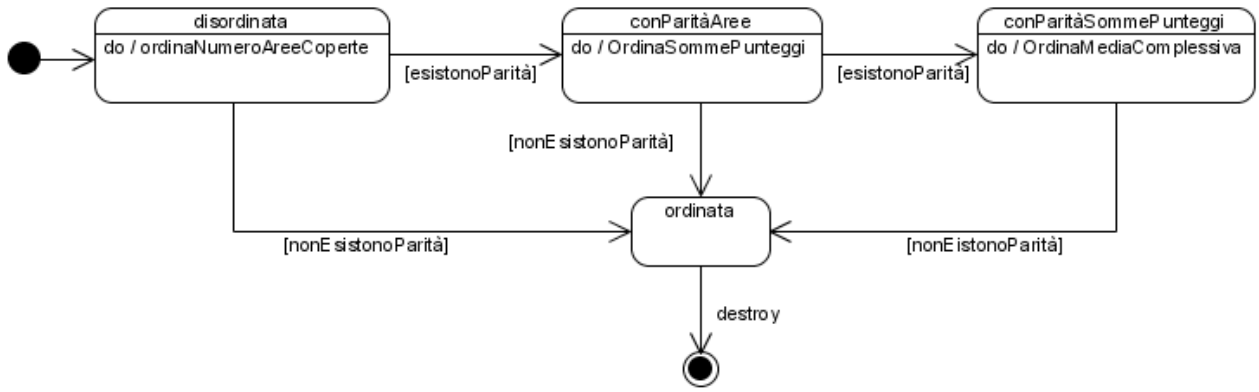
Risposta.



Si consideri la determinazione degli assegnamenti. Si ignori il problema dei tirocini a coppie.

Domanda. (Progettazione di dettaglio) Dare un diagramma di macchina a stati che descriva l'evoluzione della lista usata per ordinare gli studenti.

Risposta:



L'algoritmo dei matrimoni stabili che permette di automatizzare l'assegnamento fra studenti e tirocini, è stato introdotto da David Gale e Lloyd Shapley. L'algoritmo accoppia uno ad una gli elementi di un insieme di uomini ed un insieme di donne, in matrimoni stabili.

Informalmente, un matrimonio stabile è un accoppiamento uno ad una fra gli elementi di un insieme di uomini ed un insieme di donne, tale che non esistono motivi per lasciare il partner assegnato e sposarne un altro. Viceversa, un matrimonio è instabile se esistono un uomo e una donna che non sono stati accoppiati fra loro e che si preferiscono reciprocamente rispetto al partner che è stato loro assegnato.

L'algoritmo riceve in input, per ogni uomo, un ordinamento delle donne, e per ogni donna un ordinamento degli uomini. Nel problema classico gli insiemi composti da uomini e donne hanno la stessa cardinalità. Ogni persona ha associata una *lista di preferenze* strettamente ordinata, contenente tutti i membri del sesso opposto.

Possiamo descrivere l'algoritmo come una serie di proposte degli uomini alle donne. In ogni momento dell'esecuzione, ogni persona può essere fidanzata o libera; ogni uomo può alternare stati in cui è libero a stati in cui è fidanzato, ma se una donna diventa fidanzata, non sarà mai più libera, mentre l'identità del suo ``promesso'' può cambiare. Un uomo che è fidanzato più di una volta ottiene ogni volta fidanzate meno desiderabili per lui, mentre, a ogni nuovo fidanzamento, la donna ottiene sempre un partner migliore. Quando una donna libera riceve una proposta, la accetta subito, diventando così fidanzata del pretendente. Quando una donna fidanzata riceve una proposta, confronta il nuovo pretendente col suo fidanzato e sceglie il migliore, abbandonando l'altro. Se cambia fidanzato, il vecchio fidanzato diventa libero. Ogni uomo fa proposte alle donne nella sua lista di preferenze nell'ordine in cui appaiono, finchè non diventa fidanzato. Quando il fidanzamento è rotto da una donna, l'uomo ridiventa libero e continua a fare le sue proposte, iniziando dalla prossima donna nella lista. L'algoritmo termina quando tutti sono fidanzati.

L'algoritmo base di Gale-Shapley, orientato agli uomini, può essere riassunto in questo modo:

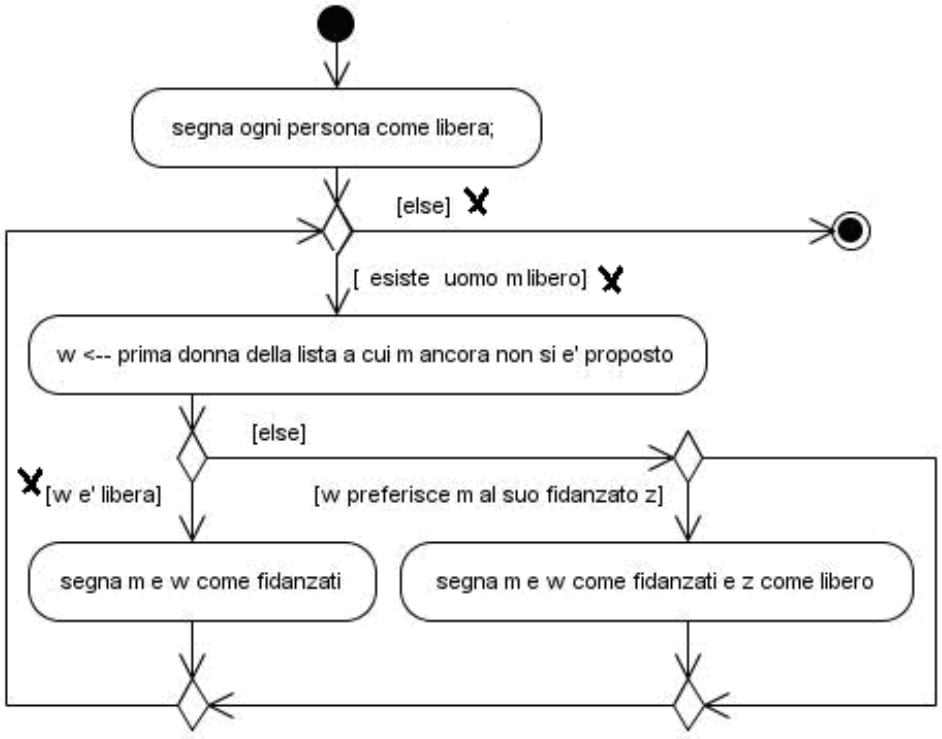
```
segna ogni persona come libera;
while (esiste un uomo m libero) do
      w <-- prima donna della lista a cui m ancora non si e' proposto;
      if (w e' libera) then
            segna m e w come fidanzati;
      else
            if (w preferisce m al suo fidanzato z) then
                  segna m e w come fidanzati e z come libero;
            end
      end
end
```

Domanda. (Verifica)
a) Dare il diagramma di flusso dell'algoritmo dato sopra.
b) Si considerino i seguenti dati di input, relativi agli uomini A, B, C e alle donne a, b, c:

Uomini			Donne		
Pref A	Pref B	Pref C	Pref a	Pref b	Pref c
A	b	c	A	B	C
B	c	a	B	C	A
C	a	b	C	B	A

b) Assumendo che il test del while consideri gli uomini in ordine alfabetico, dare la percentuale di copertura del codice ottenuta con questo caso di prova, secondo il criterio delle *decisioni*. Motivare la risposta, indicando con una croce le decisioni percorse.

Risposta. a) Il seguente diagramma risponde alla domanda, trascurando le croci che invece motivano la risposta successiva. b) La copertura è del 50% (tre decisioni prese, su sei nel grafo).



Domanda. (Verifica) Si consideri l'algoritmo dei matrimoni stabili e il caso in cui ci sono due uomini (1 e 2) e due donne (A e B). Definire, usando criteri funzionali, e tabelle tipo la seguente, un insieme di almeno 4 casi di test. Esiste un caso di test per cui sono possibili entrambi gli accoppiamenti?

	Lista di preferenze		Output accettati
1			
2			
A			
B			

Risposta

	Lista di preferenze		Output accettati
1	AB		(1, A), (2, B)
2	BA		
A	12		
B	21		

	Lista di preferenze		Output accettati
1	AB		(1, A), (2, B)
2	AB		
A	12		
B	12		

	Lista di preferenze		Output accettati
1	AB		(1, A), (2, B)
2	BA		
A	12		
B	12		

	Lista di preferenze		Output accettati
1	AB		(1, A), (2, B) (1, B), (2, A)
2	BA		
A	21		
B	12		

La componente *Assegnazione* utilizza algoritmi di ordinamento. Si assuma che i requisiti prevedano un numero di studenti mediamente pari a 1000, con punte di 1500 per sessione e un numero di tirocini assegnabili di circa 2000, ma che in realtà i numeri effettivi siano molto variabili e poco prevedibili.
Domanda. (Verifica) Indicare almeno due caratteristiche di qualità del sistema che è naturale controllare e quali test di sistema sono necessari per verificarle. Dare una breve descrizione dei temini utilizzati.
Risposta L'*efficienza* può essere valutta con *performance test* e *storage use test*, l' *affidabilità* con *volume test* e *stress test*. Si veda la dispensa per le definizioni.