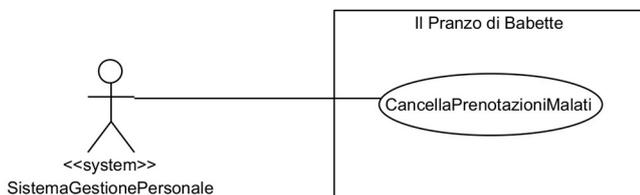


Il pranzo di Babette-Una possibile soluzione

Si consideri il seguente nuovo requisito: Il sistema (esterno) di gestione del personale invia ogni mattina alle ore 8:20 la lista dei dipendenti assenti per malattia e, per ognuno, il numero di giorni di malattia indicati dal medico curante. Il sistema provvede a cancellare eventuali prenotazioni dei dipendenti in malattia, ri-accreditando il costo dei pasti sul portafoglio virtuale dei dipendenti.

Domanda 1 Fornire il diagramma dei casi d'uso con la relativa narrativa **ESCLUSIVAMENTE** in riferimento al nuovo requisito.

Soluzione



Caso D'uso: CancellaPrenotazioniPerMalattia

Breve descrizione: Il sistema di gestione del personale invia la lista dei dipendenti assenti per malattia e, per ognuno, il numero di giorni di malattia indicati dal medico curante. Il sistema provvede a cancellare eventuali prenotazioni dei dipendenti in malattia, ri-accreditando il costo dei pasti sul portafoglio virtuale dei dipendenti.

Attore Principale: Sistema di gestione del personale

Attori Secondari: nessuno

Precondizione: nessuna

Postcondizione: tutte le prenotazioni corrispondenti a giorni di malattia sono state cancellate e i soldi accreditati

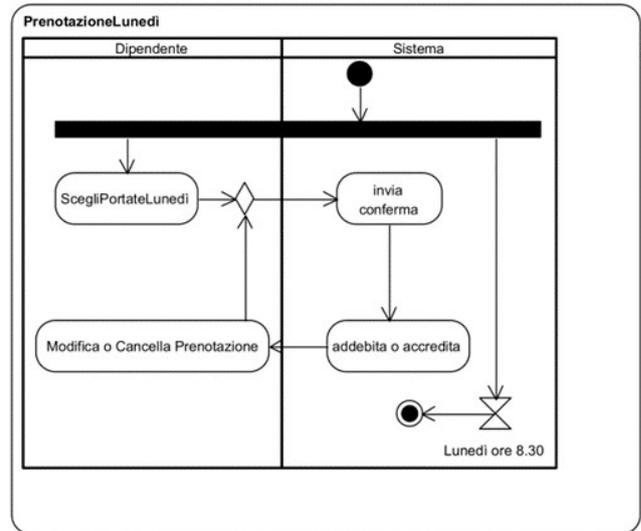
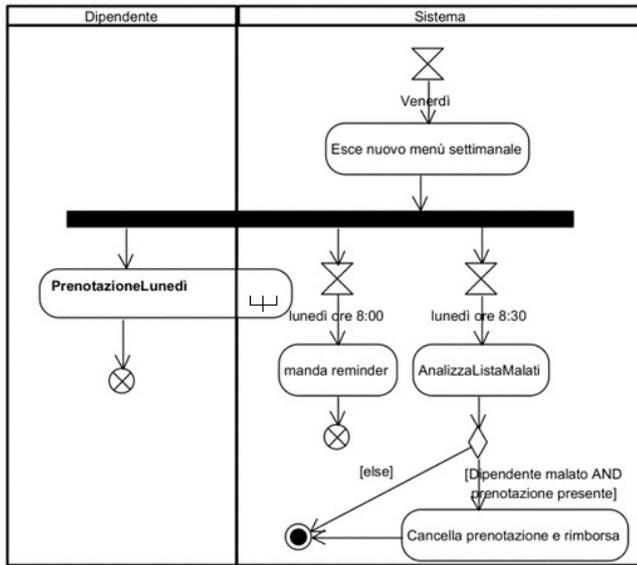
Sequenza principale degli eventi:

1. Il sistema di gestione del personale invia la lista dei dipendenti malati
2. Per ogni dipendente nella lista
 - 2.1 Se (il dipendente ha prenotazioni aperte)
 - i. Cancella prenotazioni
 - ii. Ri-accredita il costo dei pasti sul portafoglio virtuale del dipendente

Sequenza alternativa degli eventi: nessuna

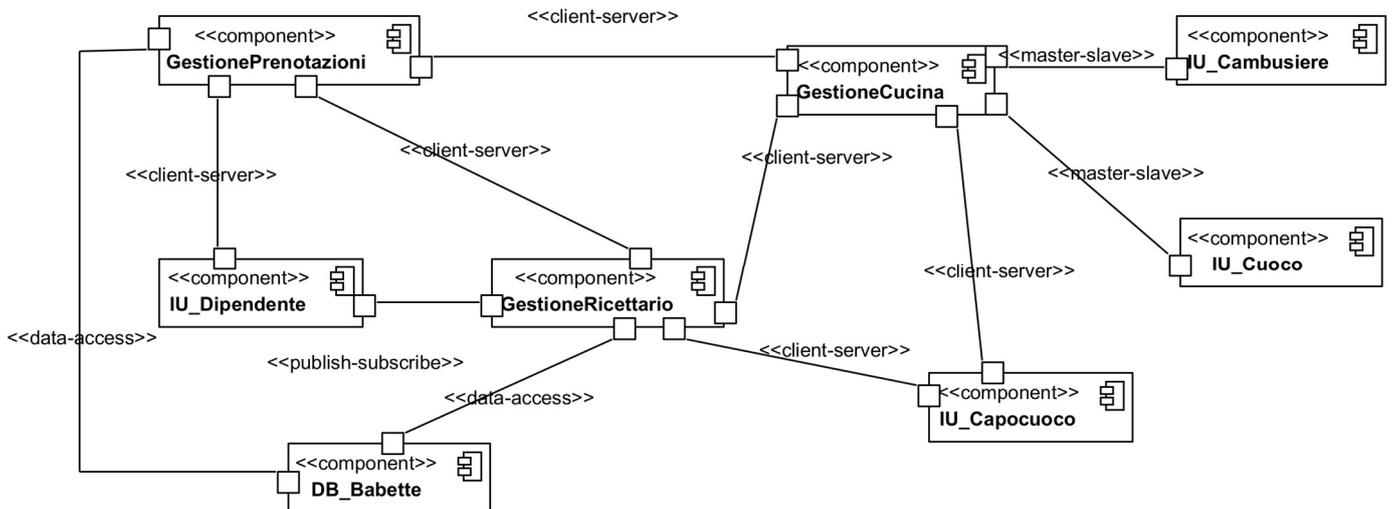
Domanda 2 Dare un diagramma di attività che descriva il processo di prenotazione/modifica/cancellazione di un pasto. Considerare **ANCHE** il nuovo requisito.

Una possibile soluzione è la seguente, esemplificata sul pasto di un lunedì

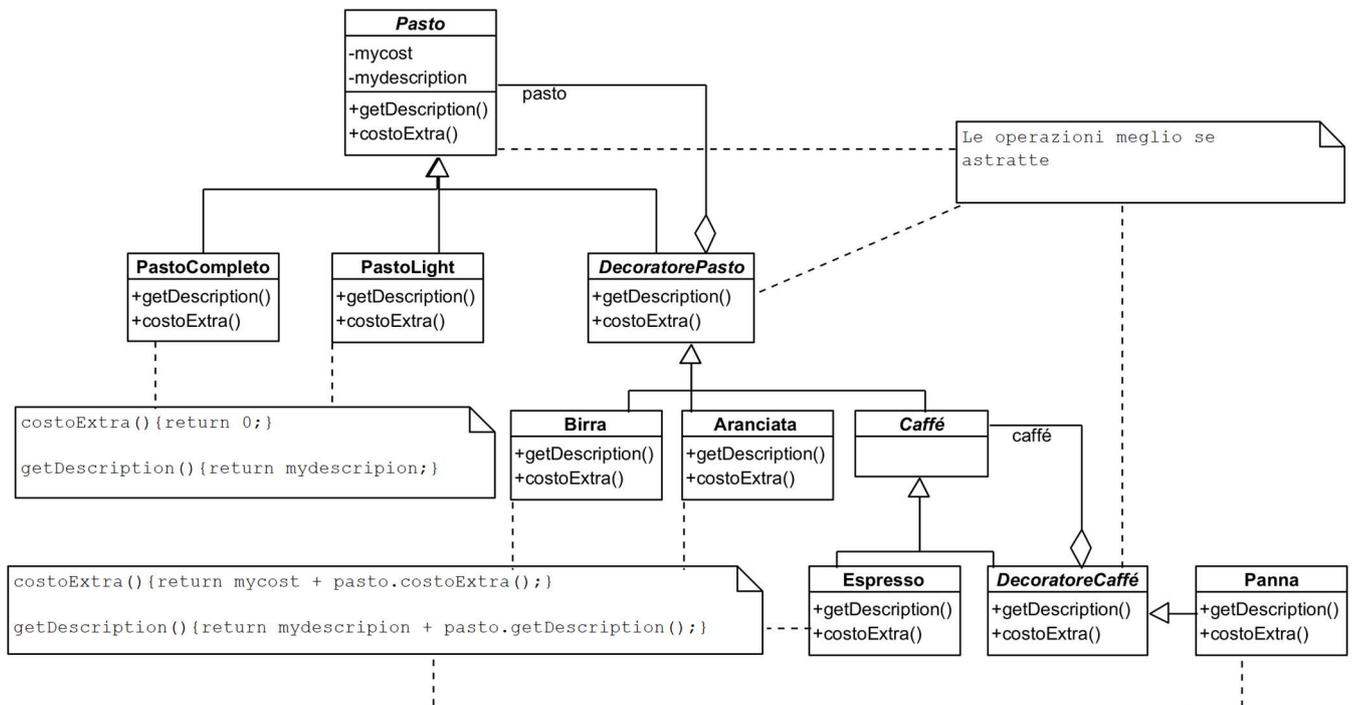


Domanda 3 Dare un diagramma componenti e connettori che descriva l'architettura del sistema. Considerare ANCHE il nuovo requisito.

Una soluzione possibile è la seguente. Ci sono altre scelte legittime: una tra tutte avere un'unica componente di gestione, si perde però in coesione. (Gestione personale è un sistema esterno, non una componente del sistema che deve essere realizzato. Accettavo anche soluzioni in cui compariva come componente, non avendone parlato dettagliatamente a lezione.)



Domanda 4. Al fine di ridurre l'utilizzo della plastica, è disponibile in mensa un dispositivo di purificazione dell'acqua. I dipendenti hanno la possibilità di riempire gratuitamente qualsiasi quantità di bicchieri desiderino. Le bevande in bottiglia, invece, sono soggette a un costo, così come il caffè. Al momento del passaggio in cassa, il cassiere richiama il pasto prenotato dal dipendente e lo arricchisce con eventuali bevande e caffè (è prassi che a turno si offra il caffè ai colleghi con cui si pranza). Si richiede di illustrare con un diagramma delle classi un'istanza del pattern decorator per la definizione di oggetti di tipo Pranzo, che implementino i metodi getDescription e costoExtra (costo bevande e caffè). Si prevedano 2, max 3, tipi di bevande. **Opzionale:** modellare il fatto che un caffè possa essere arricchito a sua volta con una o più cucchiate di panna montata.



Domanda 5 Il metodo `checkPasto` controlla che ogni pasto soddisfi i requisiti di offerta per vegani e vegetariani.

```

public static boolean checkPasto(MenuOption[] primo, MenuOption[] secondo,
MenuOption[] contorno, MenuOption[] dolce) {
    return haOpzioneVegana(primo) & haOpzioneVegetariana(secondo) &
        haOpzioneVegana(contorno) & haOpzioneVegana(dolce);
}
  
```

5.1 Dare una proof obligation per avere copertura delle decisioni;

Soluzione

Testare con 4 array che contengono almeno un piatto vegano (atteso true) e poi testare con 4 array di cui almeno 1 contiene sono piatti a base carne o pesce (atteso false)

ATTENZIONE: qui in realtà che andava bene anche richiedere solo un caso, essendo una restituzione di valore, la soluzione data

5.2 Dare una proof obligation per avere copertura delle condizioni semplici (basic condition coverage);

Soluzione

Testare con 4 array che contengono almeno un piatto vegano/vegetariano (tutte le condizioni atomiche true) e poi con 4 array che contengono sono piatti a base carne o pesce (tutte le condizioni atomiche false)

5.3 Quanti casi di test occorrono per avere copertura delle condizioni multiple (Multiple condition coverage)?

Soluzione

$$2^4 = 16$$