



Traccia:

Si vuole realizzare il sistema software WaitBook, per supportare le attività degli addetti di un ristorante.

Il sistema deve consentire ai camerieri di prendere ordinazioni ai tavoli del ristorante. Un'ordinazione comprende tutte le pietanze scelte dagli occupanti del tavolo. Degli occupanti del tavolo è importante ricordare solo il numero complessivo, perché per ciascuna persona bisognerà contare un coperto nel conto finale. Il costo del coperto è prefissato ed uguale per tutti i tavoli del ristorante.

Ogni tavolo è identificato da un numero intero e caratterizzato da un numero massimo di coperti compreso tra 1 e 12, a seconda del tavolo. Il numero massimo di coperti dovrà essere rispettato quando i camerieri effettuano il check-in dei clienti al tavolo. Se il numero del gruppo di clienti non è compatibile con il numero di coperti per il tavolo scelto, il sistema non riserva il tavolo ed il cameriere dovrà trovare un altro tavolo.

Ogni pietanza è composta da ingredienti secondo le regole di preparazione indicate in un ricettario. Il ricettario riporta, per ciascuna pietanza, gli ingredienti necessari e la rispettiva quantità. Quando il sistema prova ad aggiungere una pietanza ad un'ordinazione, esso verifica che tutti gli ingredienti necessari siano disponibili e, in tal caso, li prenota in magazzino (ossia li sottrae dalle scorte disponibili). Se almeno uno degli ingredienti componenti una pietanza non è disponibile, il sistema segnalerà il problema al cameriere, che dovrà proporre al cliente di chiedere un'altra pietanza. Il conto finale di un tavolo è calcolato come somma dei prezzi di tutte le pietanze comprese nell'ordinazione e dei coperti.

Il sistema deve inoltre consentire al cuoco di visualizzare le ordinazioni prese dai camerieri secondo l'ordine di arrivo e prelevare la prossima ordinazione da preparare. Si desidera inoltre che il direttore del ristorante possa visualizzare quali ingredienti riordinare, in quanto le scorte corrispondenti sono esaurite.

Quesiti:

1. Diagramma dei casi d'uso del sistema. [3 pt]
2. System domain model, tenendo conto delle responsabilità delle classi. [10 pt]
3. Sequence diagram di dettaglio per modellare la dinamica della funzionalità che consente di prendere l'ordinazione a un tavolo. [6 pt]
4. Progettare un insieme di casi di test black box (usando la tecnica delle Classi di Equivalenza) per provare la funzionalità di Check-in a un tavolo. Si considerino anche i valori vicini ai confini (boundaries) delle classi di equivalenza [6 pt]
5. Descrivere il pattern pipe-and-filter ed i suoi vantaggi e svantaggi [3 pt]
6. Descrivere il modello Extreme Programming (utilizzando anche un opportuno diagramma per descriverne le fasi) e la tecnica del Pair Programming [4 pt]