

Diario di lavoro

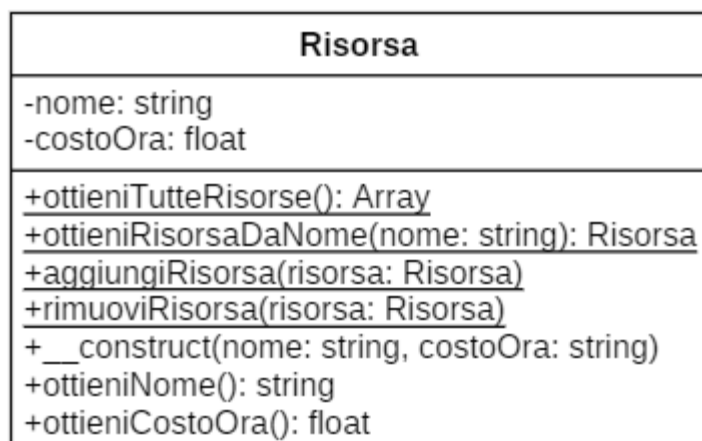
Titolo progetto	Gestione ore
Luogo	Trevano-Canobbio
Data	19 settembre 2019

Lavori svolti

Oggi ho cominciato la progettazione delle classi presenti nell'applicazione grazie ai seguenti diagrammi UML:



Questo diagramma UML mostra la classe Lavoro, di tipo Model, usata per interfacciarsi alla tabella lavoro del database. Contiene un campo per ogni colonna della tabella, ognuna con il proprio metodo getter, e due metodi statici utili a ottenere uno o più lavori dal database, più uno per aggiungere e uno per rimuovere un lavoro dal database.



La classe mostrata in quest'immagine è un Model per la tabella risorsa, anch'essa con un campo per ogni sua colonna, con tanto di metodo getter, e altri due metodi per ottenere i dati dal database, più uno per aggiungere e uno per rimuovere una risorsa dal database.

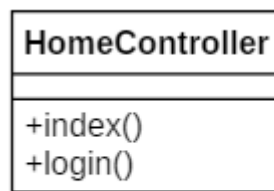
Assegnazione
-lavoro: Lavoro -risorsa: Risorsa
<u>+ottieniTutteAssegnazioni(): Array</u> <u>+eAssegnato(lavoro: Lavoro, risorsa: Risorsa): bool</u> <u>+aggiungiAssegnazione(lavoro: Lavoro, risorsa: Risorsa)</u> <u>+eliminaAssegnazione(lavoro: Lavoro, risorsa: Risorsa)</u> +__construct(lavoro: Lavoro, risorsa: Risorsa) +ottieniLavoro(): Lavoro +ottieniRisorsa(ritorno: Risorsa)

Questo schema mostra la classe Assegnazione, adattata dalla tabella assegna della banca dati, sempre con un campo per ogni colonna, sempre ognuno con il suo metodo getter, sempre con dei metodi utili per ottenere dati dal database, più uno per aggiungere e uno per rimuovere un'assegnazione dal database. È importante notare come la funzione "eAssegnato" non ritorna un'assegnazione, come al solito, bensì controlla se una risorsa è stata assegnata a un lavoro interrogando il database.

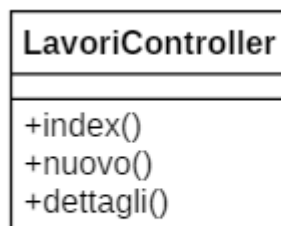
OreLavoro
-lavoro: Lavoro -risorsa: Risorsa -data: DateTime -numeroOre: int
<u>+ottieniTutteOreLavoro(): Array</u> <u>+haLavoratoInData(lavoro: Lavoro, risorsa: Risorsa, data: DateTime): int</u> <u>+ottieniOreLavoro(lavoro: Lavoro, risorsa: Risorsa): Array</u> <u>+aggiungiOreLavoro(oreLavoro: OreLavoro)</u> <u>+eliminaOreLavoro(oreLavoro: OreLavoro)</u> +__construct(lavoro: Lavoro, risorsa: Risorsa, data: DateTime, numeroOre: int) +ottieniLavoro(): Lavoro +ottieniRisorsa(): Risorsa +ottieniData(): DateTime +ottieniNumeroOre(): int

Questo diagramma UML mostra la classe OreLavoro, l'ultimo Model progettato, di nuovo con un campo per ogni colonna della tabella ore_lavoro, di nuovo ognuno ha il suo metodo getter e, stavolta, tre metodi statici: uno ottiene tutte le ore di lavoro per tutti i progetti, uno il numero di ore in cui una risorsa ha lavorato a un lavoro e l'ultimo il numero di ore lavorate in totale da una risorsa su un lavoro. Ci sono poi una funzione statica per aggiungere e una per rimuovere delle ore di lavoro dal database.

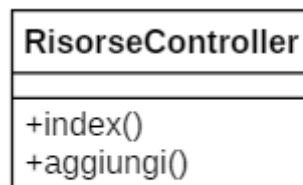
Poi ho terminato progettando anche le classi controller, mostrate in seguito:



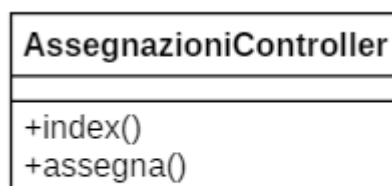
Questo controller è il primo che si visita e contiene due metodi: il primo per mostrare la pagina di benvenuto e il secondo la pagina di accesso e per controllare se le informazioni fornite sono valide.



Il prossimo controller è quello che gestisce i lavori e contiene tre metodi: “index” mostra l’elenco di tutti i lavori, “nuovo” permette di aggiungerne uno e “dettagli” mostra la pagina con i dettagli di un singolo lavoro.



“RisorseController” si occupa di gestire tutto quello che riguarda le risorse, quindi la funzione “index” mostra il form per aggiungerne una, mentre “aggiungi” ne registra una nuova nel database basandosi sui dati forniti dall’utente nel form.



Come per “RisorseController”, “AssegnazioniController” mostra il form per assegnare una risorsa a un lavoro grazie alla funzione “index” e salva la nuova assegnazione nella banca dati con “assegna”.



Infine, “OreLavoroController” fa esattamente lo stesso: con “index” mostra il form per registrare le ore di lavoro di una risorsa, mentre con “registra” salva le ore lavorate nel database.

Problemi riscontrati e soluzioni adottate

Mi sono accorto di non aver finito la progettazione delle pagine web, perché non vi avevo inserito la pagina di aggiunta di un lavoro. Perciò ho aggiunto un bottone al mockup della pagina contenente la lista dei lavori e ne ho creato uno nuovo che mostri il form di inserimento delle informazioni relative al nuovo lavoro.

A mockup of a web form titled "NUOVO LAVORO". The form contains five input fields, each with a label and a red asterisk indicating a required field:

- Nome*
- Data inizio*
- Data consegna*
- Ore preventivate*
- Note

The input fields are represented by rectangular boxes. The "Note" field is a larger, square-shaped box.

Devo quindi modificare i diagrammi di flusso per riflettere questa nuova aggiunta.

Punto della situazione rispetto alla pianificazione

Sono in orario rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Aggiornare i diagrammi di flusso per riflettere l'aggiunta della pagina di inserimento di un lavoro e portare avanti la documentazione della pianificazione.