

Diario di lavoro

Titolo progetto	Gestione ore
Luogo	Trevano-Canobbio
Data	11 ottobre 2019

Lavori svolti

Oggi, prima di cominciare, abbiamo assistito a una presentazione di uno dei docenti responsabili che ci ha spiegato come funzionano e come vengono valutati i lavori d'esame che verranno svolti a giugno. Questa presentazione è durata per la prima ora e mezza della giornata, quindi ho potuto lavorare solo per un'altra ora e mezza.

Dopo la traduzione di alcune parti dell'applicazione dall'Italiano in Inglese (vedi sezione Problemi riscontrati e soluzioni adottate), ho potuto dedicarmi all'implementazione delle classi Model che si interfacciano al database ed eseguono le query su di esso.

Più di preciso ho portato a termine la funzione "aggiungiLavoro", rinominata in "addActivity". Essa si occupa di inserire una nuova riga all'interno della tabella 'lavoro' del database. Il suo codice è il seguente:

```
/**
 * Inserts a new row into table 'lavoro' of database with data passed as parameter.
 * @param Activity $activity The data to add to the database.
 * @return bool true if the insert operation is successful, false otherwise
 */
public function addActivity(Activity $activity): bool
{
    //Write the query that will write into the database.
    $query = "INSERT INTO lavoro(nome, note, data_inizio, data_consegna, ore_preven
tivate) VALUES (:name, :notes, :startDate, :deliveryDate, :hours)";
    //Prepare the query.
    $statement = $this->database->prepare($query);
    //Replace placeholder ':name' with activity name taken from field 'name' of 'ac
tivity' parameter.
    $statement->bindParam(":name", $activity->name);
    //Replace placeholder ':notes' with activity notes taken from field 'notes' of
'activity' parameter.
    $statement->bindParam(":notes", $activity->notes);
    //Replace placeholder ':startDate' with activity starting
//date taken from field 'startDate' of 'activity' parameter.
    $statement->bindParam(":startDate", $activity->startDate);
    //Replace placeholder ':deliveryDate' with activity delivery
//date taken from field 'deliveryDate' of 'activity' parameter.
    $statement->bindParam(":deliveryDate", $activity->deliveryDate);
    //Replace placeholder ':hours' with activity estimated hours
//taken from field 'estimatedHours' of 'activity' parameter.
    $statement->bindParam(":hours", $activity->estimatedHours);
    //Return the result of query execution
    return $statement->execute();
}
```

Oltre a questa funzione ho modificato la progettazione della classe per aggiungere il nuovo tipo di ritorno delle funzioni “aggiungiLavoro” e “eliminaLavoro”:

+aggiungiLavoro(lavoro: Lavoro): bool

+eliminaLavoro(lavoro: Lavoro): bool

Lo stesso cambiamento è stato effettuato per tutte le altre sottoclassi di Model.

Problemi riscontrati e soluzioni adottate

Ho scoperto che bisogna mantenere una singola lingua per tutti i file creati, quindi ho deciso di usare l'Inglese. Questo implica un'operazione di cambiamento di tutti i nomi delle classi e di tutti i commenti scritti finora, che mi ha preso una buona metà delle due ore rimaste dopo.

Più tardi ho notato che avevo sbagliato la progettazione della classe model 'Assegnazione' e il tipo di ritorno della funzione “ottieniRisorsa” era impostato come parametro normale. Ora la funzione ha questo aspetto:

+ottieniRisorsa(): Risorsa

Punto della situazione rispetto alla pianificazione

Sono in anticipo rispetto alla pianificazione.

Programma di massima per la prossima giornata di lavoro

Portare avanti la documentazione della pianificazione e la realizzazione delle classi Model che si interfacciano al database.