

Informatica – A.A. 2024/2025

Documentazione Object-Oriented

Studenti

Gennaro Schiazzano: N86005318

Pasquale Punziano: N86005283

Progettazione e sviluppo di un sistema informativo in grado di tenere traccia delle proprie attività personali (ToDo).

1. Descrizione e Analisi del Problema
   1. Descrizione

Traccia 1: Gestore ToDo

Si sviluppi un sistema informativo, composto da una base di dati relazionale e da un

applicativo Java dotato di GUI (Swing) che consente di tenere traccia delle attività personali

da svolgere (nel seguito chiamate semplicemente “ToDo”), ispirato al software Trello.

Il software deve consentire all'utente di organizzare e gestire le proprie attività personali in

modo efficiente, utilizzando un'interfaccia intuitiva e flessibile simile a quella di Trello

(<https://trello.com/>).

* 1. Analisi del Problema

Il progetto Gestore ToDo si propone di realizzare un sistema informativo per la gestione delle attività personali, ispirato al funzionamento di Trello. Il sistema deve permettere agli utenti di autenticarsi tramite login e password, organizzare le proprie attività in bacheche tematiche (Università, Lavoro, Tempo Libero), e gestire liberamente la creazione, modifica, condivisione e spostamento dei propri ToDo. Inoltre, l’interfaccia grafica (realizzata con Java Swing) deve risultare intuitiva e reattiva, permettendo una visualizzazione ordinata e flessibile dei dati.

La traccia specifica che “*il software deve consentire all’utente di organizzare e gestire le proprie attività personali in modo efficiente, utilizzando un’interfaccia intuitiva e flessibile simile a quella di Trello*”. Da questa indicazione si deduce che l’applicativo dovrà presentare un’interfaccia grafica user-friendly e funzionalità dinamiche, come lo spostamento e l’ordinamento dei ToDo, operazioni che implicano l’utilizzo di strutture dati ordinabili e modificabili.

Gli utenti accedono al sistema mediante “*login e password, che devono essere univoche*”. È dunque necessario implementare un meccanismo di autenticazione sicura e garantire l’unicità delle credenziali, vincolo che ha guidato l’inclusione dell’attributo username nella classe Utente con un vincolo di univocità.

Le attività vengono organizzate in “*tre bacheche (ogni bacheca ha un titolo e una descrizione)*”, i cui titoli sono vincolati a un insieme chiuso: “*Università, Lavoro e Tempo Libero*”. Questo ha portato all’adozione di un’enumerazione (enum) per rappresentare i titoli delle bacheche, in modo da assicurare coerenza e prevenire errori di input.

Ogni ToDo è caratterizzato da numerose proprietà opzionali: “*titolo, data di scadenza, un link a una URL correlata all’attività, una descrizione dettagliata e un’immagine*”. Inoltre, “*tutti questi elementi sono opzionali e possono essere modificati in qualsiasi momento*”. Pertanto, nella progettazione delle classi è stato considerato un modello dati capace di gestire attributi nulli o mancanti, e in futuro sarà necessario implementare metodi di modifica (setter) e costruttori flessibili che permettano di inizializzare o aggiornare dinamicamente i valori di questi attributi.

Un aspetto rilevante è la possibilità di condividere i ToDo: “*ogni ToDo può contenere inoltre una lista di altri utenti che condividono quel ToDo*”. Questo implica una relazione molti-a-molti tra ToDo e Utente, che è stata implementata come una lista di utenti nella classe ToDo. Inoltre, la condivisione è gestita esclusivamente dall’autore: “l*’autore del ToDo può aggiungere o eliminare condivisioni*”, il che suggerisce che un ToDo debba mantenere un riferimento al suo creatore.

Nel modello è presente una classe astratta Operazione, collegata sia ai ToDo sia alle Bacheche, la cui finalità non è il tracciamento delle azioni, ma una questione di modularità. In particolare, la classe rappresenta un concetto generico di operazione che può essere eseguita su entità diverse. Questo approccio generalizzato consente di riutilizzare le stesse classi derivate (come Create, Modificate, Delete) sia per le Bacheche che per i ToDo, migliorando la coerenza e la manutenibilità del codice, in un’ottica di programmazione orientata agli oggetti.

L’utilizzo delle aggregazioni è stato scelto per rappresentare relazioni di tipo “parte di” senza vincoli di vita condivisa. Ad esempio, una Bacheca contiene una lista di ToDo, ma questi possono esistere anche indipendentemente dalla bacheca stessa (soprattutto nei casi di spostamento o condivisione tra più utenti).

Infine, il sistema deve fornire strumenti di ricerca e filtraggio: “*l’elenco di ToDo in scadenza nella giornata odierna, oppure quelli in scadenza entro un certo giorno specificato dall’utente*”, e anche la “*ricerca per nome o per titolo dei ToDo*”. Queste funzionalità impongono l’utilizzo di filtri e algoritmi di ricerca all’interno delle strutture dati, possibilmente tramite query sulla base di dati relazionale o tramite metodi di filtro in memoria.

1. Progettazione concettuale

2.2. Class Diagram del dominio del problema

Immagine che contiene diagramma, testo, Piano, Disegno tecnico

Il contenuto generato dall'IA potrebbe non essere corretto.