

Gestore ToDo

Scelte implementative

Per la creazione di un **class diagram** di dominio del problema ho scelto un'implementazione concettuale, per questo motivo ho deciso di non scrivere i tipi delle variabili all'interno delle classi. Durante la creazione del **class diagram** ho preso spunto dalla possibile soluzione, ritenendo che sia una delle soluzioni più complete e semplici. Ho apportato qualche modifica seguendo sempre uno stile basico e cercando di mantenere una coerenza visiva.

Bacheca

Partendo dalla classe **Bacheca** ho inserito 3 attributi di cui: - titolo - descrizione - tipo le prime due variabili sono autoesplicative, mentre l'ultima è una variabile che ha come tipo un'enumerazione. Questo perchè le bacheche possono essere solo di 3 tipi: 1. Università 2. Lavoro 3. Tempo Libero

Utente

Nella tabella **Utente** ho inserito anche qui 3 attributi: - id - login - password
l'attributo **id** è stato inserito poichè preferisco gestire ogni utente tramite un id univoco e permettere così di distinguerli in modo efficiente.

ToDo

La tabella **ToDo** è quella con più attributi: - titolo - scadenza - link - descrizione - immagine - stato - posizione - sfondo

sono tutti autoesplicativi, ma il tipo dell'attributo **stato** e dell'attributo **sfondo** sono due enumerazioni. Questo poichè lo stato di un'attività è vincolato tramite due attributi: 1. completato 2. non completato

Per quanto riguarda l'attributo **sfondo** ho deciso di rendere vincolata la scelta dei colori del **ToDo** tramite 3 attributi: 1. rosso 2. giallo 3. verde

Questo per mantenere una coerenza visiva e una facilità di utilizzo migliore per gli utenti, poichè credo che rendere la scelta del colore libera rende l'utilizzo del programma più complesso e meno coerente. Non escludo un'implementazione di altri colori in futuro.

Associazioni e cardinalità

Partendo dalla classe **Bacheca** abbiamo come unica associazione un'enumerazione **TipoBacheca**. La classe **Utente** ha un'associazione con la classe **Bacheca** che indica che un singolo utente visualizza tre bacheche (Università, Lavoro e Tempo Libero). L'altra associazione della classe è con **ToDo**, poichè un numero non definito di utenti può avere un numero non definito di ToDo (associazione molti a molti). Analizziamo ora la classe **ToDo** che ha le restanti 4 associazioni 2 delle quali sono delle enumerazioni **StatoToDo** e **ColoreSfondo**. La terza è una composizione con **Bacheca** poichè secondo il mio ragionamento

(prendendo spunto anche dalla possibile soluzione) non potrebbero esistere ToDo senza delle bacheche.

Considerazioni finali

Con questo ho indicato ogni mia scelta, come ho già detto all'inizio ho deciso di non entrare troppo nel dettaglio perchè di alcune scelte (vedi la condivisione dei **ToDo**), non sono ancora del tutto sicuro di come implementarla. Considerando che ho la possibilità di modificarle in futuro sono sicuro che queste mie scelte possono essere cambiate e migliorate.

Inoltre voglio specificare che usando un sistema linux, **visual paradigm** non funziona nel migliore dei modi. Quindi per il momento ho deciso di usare **starUml** poichè tra tutti reputo che sia quello più affidabile.