1. Interpretazione della traccia

Viene richiesto di sviluppare un sistema ToDo , progettato per gestire e organizzare le attività personali di ogni utente. Ogni utente deve poter accedere al sistema inserendo le proprie credenziali: nome utente e password univoche.

Strutture e gestione delle attività:

* I ToDo sono organizzati all’interno di bacheche, ciascuna caratterizzata da un titolo e una descrizione.
* L’utente può creare, modificare ed eliminare i ToDo all’interno delle proprie bacheche.
* I ToDo possono essere condivisi con altri utenti e appariranno automaticamente nelle bacheche corrispondenti.

Ogni ToDo contiene:

* Titolo;
* Data di scadenza;
* Link;
* Immagine;
* Descrizione ;
* stato(completato/non completato).

Funzionalità del sistema:

1. Visualizzazione del ToDo in scadenza entro una data specificata;
2. Ricerca avanzata per titolo o nome utente;
3. Gestione delle scadenze:

* I ToDo scaduti verranno evidenziati in rosso;
* Verrà mostrata la data di scadenza per un facile riferimento.

1. Analisi delle classi con relativi attributi , metodi e associazioni

***UTENTI***

Ruolo e motivazioni: La classe Utenti gestisce l’autenticazione degli utenti nel sistema e garantisce che ogni utente abbia un accesso univoco e personalizzato ai propri ToDo e alle bacheche personali.

Attributi:

* Login(String) → Identificativo univoco dell’utente.
* Password(String) → Necessaria per autenticarsi

Metodi:

* CreaUtente() → Registra un nuovo utente nel sistema;
* ModificaLogin() → Permette di cambiare il nome utente;
* ModificaPassword() -> Permette di modificare la password;
* EliminaUtente() -> Cancella un utente da piattaforma;
* InserimentoDaTastirera() → Consente di inserire manualmente i dati dell’utente

Associazioni

Utente → Condivisione con cardinalità 1:N (Un utente può condividere più ToDo)

Non c’è un’associazione diretta fra Utente e Bacheca poiché l’utente interagisce con le bacheche attraverso i ToDo. In più, in questo modello, le bacheche sono generali e organizzano i ToDo senza essere legate a utenti specifici.

***BACHECA***

Ruolo e motivazioni: La classe bacheca organizza i ToDo all’interno di categorie e permette di gestire le attività in modo strutturato, separandole per ambito e facilitando la gestione da parte dell’utente.

Attributi

Titolo(String) → Nome Bacheca

Descrizione(String) → Informazioni sulla bacheca

Metodi:

CreaToDo() → Aggiunge un ToDo alla bacheca

EliminaToDo() → Rimuove un ToDo

ModificaToDo()->Modifica le informazioni di un ToDo

getToDo() → Recupera un ToDo

AggiungiDescrizioneBacheca() → Modifica la descrizione della Bacheca

InserimentoDaTastiera() → Permette di creare una bacheca manualmente.

Associazioni

1:N con ToDo\_Bacheca(Una bacheca contiene più ToDo)

Nel modello utilizzato una bacheca non contiene direttamente i ToDo, ma utilizza ToDo\_Bacheca per gestire il legame tra i due. Questo permette di modificare la posizione di un ToDo dentro una bacheca senza doverlo scollegare direttamente

***ToDo***

Ruolo e motivazioni: La classe ToDo rappresenta un’attività che l’utente deve svolgere e consente di organizzare compiti in bacheche, personalizzandoli con dettagli come descrizione, immagine, URL, colore sfondo e stato.

Attributi

* Titolo(String) → Nome del ToDo
* DataScadenza(LocalDate) → Data di scadenza;
* Url(String) → Link correlato all’attività
* Immagine(string) → Link/percorso associato all’immagine
* Coloresfondo(String) → Colore per la visualizzazione nella GUI
* ListaUtenti(Array di utenti) → Utenti con cui è condivisio
* Stato(String) → Serve a dire se il ToDo è stato completato/Non completato
* LoginCreatore(String) → Utente che ha creato il ToDo

Metodi:

* CambiaColoreSfondo() → Metodo che ha lo scopo di cambiare il colore dello sfondo;
* AggiungiTitolo() → Metodo per aggiungere un titolo al ToDo;
* AggiungiDataDiScadenza() → Metodo per aggiungere la data di scadenza del todo;
* AggiungiUrl() → Metodo per aggiungere il collegamento(link) ad una certa attivita
* AggiungiDescrizione() → Metodo per inserire una descrizione dettagliata del ToDo
* AggiungiImmagine() → Metodo per aggiungere un link/percorso dell’immagine che si desidera inserire;
* AggiungiStato() → Metodo per modificare lo stato del ToDo;
* RimuoviUtente() → Elimina un utente dalla lista di condivisione del Todo, garantendo il controllo della condivisione dell’attività da parte del proprietario;
* InserisciDatastiera() → Consente l’inserimento dei precedenti dati usando la tastiera;

***TODO\_BACHECA***

Ruolo e motivazione: Questa classe gestisce il collegamento tra ToDo e Bacheca, permettendo di spostare i ToDo e mantenere l’organizzazione delle attività. Separare la gestione dei ToDo dalla Bacheca permette di mantenere un modello scalabile e flessibile, evitando dipendenze dirette.

Attributi:

LoginCreatore(String) → Identifica l’utente che ha creato il ToDo, in modo da tenere traccia dell’autore, utile per i permessi di modifica ed eliminazione dei todo.

TitoloToDo(String) → Titolo del Todo;

TipoBacheca → Definisce la bacheca in cui si trova il ToDo, in modo da organizzare le attività in categorie chiare,facilitando la gestione e la ricerca-

Metodi:

Assegna() → Associa il ToDo a una bacheca specifica, consentendo di posizionare correttamente un ToDo nella sua categoria e garantendo un’organizzazione efficiente.

Associazioni

TODO\_BACHECA → ToDo è un’associazione 1:N , poiché un ToDo può essere associato a più bacheche nel tempo.

CONDIVISIONE

Ruolo e motivazione: Questa classe gestisce la condivisione dei todo e permette la collaborazione tra utenti

Attributi

TitoloTodo(String) → Definisce il titolo del ToDo condiviso

LoginUtente(String) → Identifica un utente che ha accesso al ToDo

LoginCreatore(String) → Indica chi ha creato il ToDo

Metodi

Condividi() → Aggiunge un nuovo utente alla condivisione del ToDo

RevocaCondivisione() → Rimuove un utente della condivisione del ToDo

Operation() → Effettua operazioni generali sulla condivisione

Associazioni

CONDIVISIONE → ToDo è un’associazione 1:N poiché un ToDo può essere condiviso con più utenti

CONDIVISIONE → Utente è un’associazione N:1 poiché ogni relazione di condivisione è associata a un solo utente, ma ogni utente può essere parte di più condivisioni.

1. Link Repo: https://github.com/simo-git-crypto/Applicativo.git