Classe: Interfaccia GUI

**IN GENERALE**: serve un metodo che interpreta i valori degli attributi del metodo GET/POST e decide come chiamare e con quali parametri i vari metodi che riporto qui sotto.

Dopo aver fatto il login è necessario salvare in una variabile Globale il nicname dell’utente —> la chiamo varNic.

Dove c’è Void mettere un bool per sapere se la procedura è andata a buon fine

Void displayDashboard( <enum>TypeDashB , i )

**TypeDashB** indica il tipo di dashboard (MyDashboard, ImPlayer, Esplora), a

seconda di questo TypeDashB vado a impostare diversi valori per: var che indica **il file da modificare**, variabile che indica il **segnaposto per l’ <ul>** e una variabile che indica il **segnaposto della navigazione a freccette** della data dashboard. Grazie a queste variabili potremmo scrivere il **metodo generico** che va a creare le giuste card nelle giuste Dashboard senza ricopiare codice.

**i** indica il numero della pagina da visualizzare della data dashboard

**segue i seguenti passi**:

* 1. Salva il valore **i** in una variabile globale es. currentPageMyReport se TypeDashB= MyReport.
  2. Rimuove tutte le vecchie card della dashboard
  3. Aggiunge le nuove cards, facendo attenzione che tipi di dashboard differenti hanno bisogno di card con differenti elementi (vedi es. html)

Per fare questo prima invoca il metodo che ricerca gli ID: **{IDs\_repo}+NumP getIDMyRepos(TypeDashB, varNic , i ).**

Se TypeDashB=Esplora allora metto NULL al posto di varNic.

Se MaxP=0 => al posto del segna posto metto un messaggio tipo: “Non hai ancora aggiunto nessun report” (se è dashB=MyReport)

Poi a partire da questi ID richiedo l’array dei dati con:

**Array<DatiRCorti> getRepoDatiCards( {IDs\_repo} )**, e infine **creo le card** tenendo, delle informazioni passate, solo ciò ce mi serve.

1. Aggiorna la **navigazione a freccette** infondo alla dashboard, quindi: Disabilita freccetta sinistra se **i** = 1 (quella di destra se **i**=**MaxP**) e in ogni caso abilita le freccette utilizzabili, potrebbero essere state cambiate. **IMPORTANTE**: se MaxP= **0** oppure 1 => va a disabilitare con **display=none** l’intera navigazione a freccette.

Devo rispettare l’ **ASSUNZIONE** riportata nel paragrafo: **Void espandi( isMyrepo, isSuccessivo)**

**ATTENZIONE**: o per ciascuna card si mette in fase di scrittura all’attributo **value** dei propri <button> il valore = ID\_report oppure è necessario salvare in una variabile globale le corrispondeze: card1->ID\_repo1 , card2->ID\_repoo2 …

Area Personale

Void displayAreaPersonale( varNic )

Chiamata alla prima creazione della pagina AreaUtente

* 1. Display delle info utente utilizzando: **DatiAnagrUtente getUtente(Nicname)**
  2. Display delle cards dei personaggi: il segnaPosto è <ul> , usando: **DatiPers getPersonaggioDati( ID\_pers )**
  3. Chiamare: **Void displayDashboard( MyRepo , 1 )**
  4. Chiamare: **Void displayDashboard( ImPlayer , 1 )**

Void changeEsplorabile( ID\_repo )

Chiamata quando viene pigiato il pulsante “Publica in/Rimuovi da Esplora” di una certa card in MyRepo\_dashboard.

Come detto prima (in: **ATTENZIONE**) l’ID\_repo è ricavato andando a ricercare la corrispondenza #card->ID\_repo o più semplicemente la form mi ritorna (alla pressione del <button> cardN\_setEsplora) un name=“changeEplorabile” value=“ID\_repoN”.

Grazie a questo poi invoco : **Void setEsplorabile( ID\_repo)** e successivamente mi **ricordo di modificare l’HTML** col pulsante modificato.

Void espandi( <enum>TypeDashB, isSuccessivo)

**isSucessivo** indica se è stata cliccata la freccetta di destra.

**ASSUNZIONE**: Assumo che le cose siano state fatte bene, cioè che con la creazione della pagina con: **Void displayAreaPersonale( varNic )** io abbia messo **display=none** alla dashboard se questa ha solo una pagina, e se ce ne ha altre, siccome sono solo alla prima pagina, ho **disabilitato** il pulsante che mi permette di andare a sinistra.

Sotto questa assunzione si capisce che il metodo viene invocato se la dashboard ha più di una pagina, quindi, **Espandi**

Il metodo semplicemente chiama per esempio:

**Void displayDashboard( MyRepo , CurrentPageMyRepo +1 )** , se TypeDashB=MyRepo e isSuccessivo=true

Modifica Profilo

**MODIFICA al HTML** necessaria per renderci le cose più semplici: metter due form con due relativi input\_submit, una per le info utente e l’altra per la modifica password.

Void displayModificaUtente( DatiAnagrUtente )

Void setUtente( DatiAnagrUtente + PswdCorrente + NuovaPswd + ConfermaPswd)

1. Invocare: **Void setDatiAnagrUtente (DatiAnagrUtente)**
2. Verificare se si ha avuto intenzione di modificare la password (se uno degli ultimi 3 campi ha qualcosa)

Se si vuole cambiare password allora:

* 1. Verifico passwordVecchia: => invoco **Bool verificaPassword(VarNic, pswdProva)** con pswdProva=**PswdCorrente.**
  2. Verifico che NuovaPaswd = ConfermaPswd
  3. Se tutte le verifiche sono andate correttamente => invoco **Void setPassword(nuovaPswd)**
  4. Se c’è stato un errore devo mandargli la stessa pagina con **Void displayModificaUtente( datiSalvati )**, + I vari messagi di errore.

1. Ritornare una pagina HTML (nuova) in cui si è data conferma dell’avvenuta corretta delle modifiche + un link per tonare all’area personale.