

Progetto di Programmazione Internet NABOO

Corso di Informatica per il Management

“NaBOo”

Laura Specchiulli — 988074 — laura.specchiulli@studio.unibo.it

Filippo Baroncini — 987720 — filippo.baroncini2@studio.unibo.it

Luca Ferri — 998090 — lucaferri8@studio.unibo.it

Istruzioni per l'esecuzione

Su Virtuale, oltre a tale relazione, vengono forniti due file zip.

Il primo, “*naboo_project_programmazione_jar*”, contiene il Jar del progetto realizzato, eseguibile tramite il comando da terminale:

```
java -jar --module-path ".\javafx-sdk-20.0.1\lib" --add-modules  
javafx.controls,javafx.fxml naboo-project-programmazione.jar
```

Purtroppo tale Jar funziona solo in lettura, dati dei problemi nel creare un Jar che riuscisse a tener conto della cartella del database anche in scrittura.

Dunque l'effettivo funzionamento del progetto lo si può visualizzare tramite il secondo file zip, “*naboo-project-programmazione*”, il quale per essere testato totalmente **necessita dell'IDE IntelliJ**.

N.B: bisogna assicurarsi che alla prima esecuzione venga settata la SDK di JavaFX alla versione 20 ed inoltre impostare il path corretto della SDK.

Questo lo si può fare così: Run → Edit Configurations → Modify options → Add VM options → aggiungi come VM options la seguente stringa:

```
--module-path <yourPathTo>/javafx-sdk-20.0.1/lib --add-  
modules=javafx.controls,javafx.fxml
```

Come gruppo ci teniamo a fornire anche un video riassuntivo sul funzionamento del progetto ed un file README-database che riassume i metodi messi a disposizione dal database, utilizzati nell'interfaccia grafica amministrativa e dal bot.

FUNZIONAMENTO DELLA PARTE AMMINISTRATIVA

Naboo è un'applicazione JavaFX alla quale si può accedere tramite un profilo Admin con:

Username: **SuperUser**

Password: **Naboo**

Una volta entrato correttamente, all'admin verrà mostrata una nuova finestra "Home", la quale può essere gestita solo dai profili amministratori.

L'interfaccia Home, tramite una serie di bottoni, permette agli admin di entrare nello specifico dell'applicazione, e dunque:

1) **Gestire i FEED RSS**

Pannello che permette di:

- aggiungere al database un Feed Rss tramite Url;
- importare i Feed RSS dalla base di dati per visualizzarli tramite scorrimento array ed eventualmente rimuoverli;
- Approvare i Feed che hanno una condizione di stato "non approvato", perché inseriti da bot.

2) **AGGIORNARE la BASE DI DATI**

Tramite il pulsante "news collector" permette di aggiornare tutte le notizie degli Rss presenti nel database.

3) **Gestire le NOTIZIE**

Pannello che permette di:

- scorrere le notizie presenti nel sistema, con i relativi attributi, ed eventualmente rimuoverle.

4) **Gestire i profili degli UTENTI**

Pannello che permette di:

- aggiungere al database un nuovo utente tramite telegram ID, nickname e ruolo;
- modificare il ruolo di un utente da Admin a User, e viceversa ;
- approvare profili utente che si sono registrati tramite Bot
- eliminare interi profili utente.

5) **MODERARE I COMMENTI degli utenti**

- eliminando eventualmente un intero commento, relativo ad una notizia
- settando da true a false un commento "Marked", che è stato segnalato, eliminando dunque la segnalazione.

6) **Visualizzare i RATINGS** dati dagli utenti per determinate notizie

In generale tutti i pannelli appariranno inizialmente con campi Texfield vuoti, è dunque necessario premere i pulsanti Back o Next nel momento di apertura, in modo tale da visualizzare e scorrere gli elementi relativi a Feed Rss, News, Utenti, Commenti e Ratings presenti nel database.



FUNZIONAMENTO DEL BOT “NaBOot”

NaBOot è un bot di Telegram che permette all'utente di interrogare l'applicazione e ricevere prima notizia del giorno in corso (la notizia del giorno), le notizie di un certo feed (navigabili in sequenza dalla più recente alla più datata), le notizie di una determinata data (navigabili anche queste dalla più recente alla più datata), suggerire un nuovo feed ed infine modificare le proprie opzioni.

Il bot permette a un utente di registrarsi in autonomia.

Per ogni notizia l'utente ha la possibilità di votarla, commentarla, segnalare il Feed di origine della notizia come inopportuno. L'utente dalle opzioni può decidere di cancellare la propria utenza.

NaBOot mantiene in cache lo stato della conversazione che intrattiene con ogni utente in modo tale da poter sostenere più chat in contemporanea e rispondere correttamente a ciascun utente.

Il bot è integrato in un thread parallelo al thread della UI di JavaFX per far sì che il pannello amministrativo e il bot possano essere indipendenti e contemporanei.

Implementazione

Il progetto è stato sviluppato inizialmente su *JEdit*, per gestire la parte relativa al Bot e al database, ed inseguito ultimato su *IntelliJ IDEA* come progetto JavaFX per gestire tutti i Controller per il funzionamento dei pannelli. Gli elementi grafici invece sono stati creati e modificati inizialmente tramite *SceneBuilder*, per poi essere trasferiti e richiamati su IntelliJ tramite file *.fxml*.

È stato realizzato un merge su IntelliJ per creare il Jar.

Suddivisione del lavoro

In generale il lavoro si è svolto tramite una serie di incontri per definire idee e opinioni sull'architettura e sul funzionamento dell'app.

I vari compiti inizialmente sono stati suddivisi in modo netto, per poi finire a collaborare in maniera blended nel corso del tempo, in modo tale da testare al meglio l'efficienza dell'intero progetto.

Suddivisione del lavoro:

Filippo:

- Struttura e implementazione del bot;

Luca:

- Creazione e gestione della base di dati
- Importazione FeedRss tramite ROME

Laura:

- Creazione elementi grafici interfaccia Admin
- Implementazione Controller in JavaFX

