



UNIVERSITÀ DI PISA

# RELAZIONE PROGETTO WORTH

**Laboratorio di  
Programmazione di Reti (B)**  
Prof.ssa Federica Paganelli

Alessio Vito D'angelo  
n° 559287

# INDICE:

Descrizione Progetto

Schema dei Thread e Concorrenze

Descrizione delle classi

Istruzioni Worth

## DESCRIZIONE DEL PROGETTO

### SINTESI:

Per questo progetto ho optato per la versione grafica, utilizzando Java swing.

### STRUTTURE:

Il progetto rispetta le seguenti caratteristiche:

- la fase di registrazione viene implementata mediante RMI.
- Il server usa multiplexing
- I progetti, le card e gli utenti sono permanenti mediante file Json
- Il server usa java swing (usa una interfaccia GUI)
- È implementata una struttura dati separata per ciascuna lista di progetto
- La chat di progetto usa UDP Multicast
- L'indirizzo multicast viene stabilito al momento della creazione del progetto.
- Il servizio di notifica degli utenti online/offline è stato implementato tramite RMI callback
- Nella chat vengono notificate anche i messaggi di sistema relativi a uno spostamento di una card.

Ho modificato la classe del progetto in modo tale da inserire anche l'owner, serve solo per tenere traccia di chi ha creato il progetto.

### SERVER:

Il server utilizza multiplexing per inviare/ricevere messaggi, per determinare il successo o l'insuccesso invia i seguenti codici:

LOGIN\_505 = "Login effettuato"

LOGIN\_404 = "Errore psw/name"

PROJECT\_505 = "Progetto creato"

PROJECT\_404 = "Errore creazione progetto"

CARD\_505 = "Card Creata"

CARD\_404 = "Esiste una card con questo nome e descrizione"

MOVCARD\_505 = "Card mossa"

ADDACCESS\_505 = "Accesso utente aggiunto correttamente"

ADDACCESS\_405 = "Accesso già esistente"

ADDACCESS\_404 = "Utente non registrato"

REMOVACC\_505 = "Accesso utente rimosso correttamente"

DELVPRJ\_505 = "Progetto eliminato"

LOGOUT\_505 = "Logout Fatto"

ERROR\_404 = "Errore Server"

## SCHEMA DEI THREADS E CONCORRENZE

### CLIENT:

Il client attiva 2 Thread quando si apre la chat di un progetto, Il primo thread è attivo da un ciclo infinito che aspetta i messaggi dal MultiCast e l'altro viene usato per l'invio dei messaggi e delle notifiche sullo spostamento delle card.

### SERVER:

Il server utilizza multiplexing, quindi non abbiamo problemi di concorrenza in quanto utilizza un solo thread.

## DESCRIZIONE DELLE CLASSI

Il progetto è stato suddiviso in 3 pacchetti (Client, Database e Server) e 2 cartelle (Client.Form e Server.Structure). Ho diviso in questo modo per avere ordine nel progetto in quanto abbiamo 27 classi.

### CLIENT:

Classe che esegue tutte le operazioni del client.

### GUICLIENT:

Classe da compilare utilizzata per l'interfaccia grafica che comunica con la classe Client.

### MEMBERCLIENT:

Questa classe viene utilizzata dopo il login con RMI Callback, per mantenere un Frame dove si visualizzano i membri online.

### CLIENT.FORM:

Tutte le classi in questa cartella sono solo classi per l'interfaccia grafica.

### DATABASE:

Classe usata per la gestione del database.

### INFOPRJ:

Classe usata per la deserializzazione dei file JSON del progetto.

### SERVER:

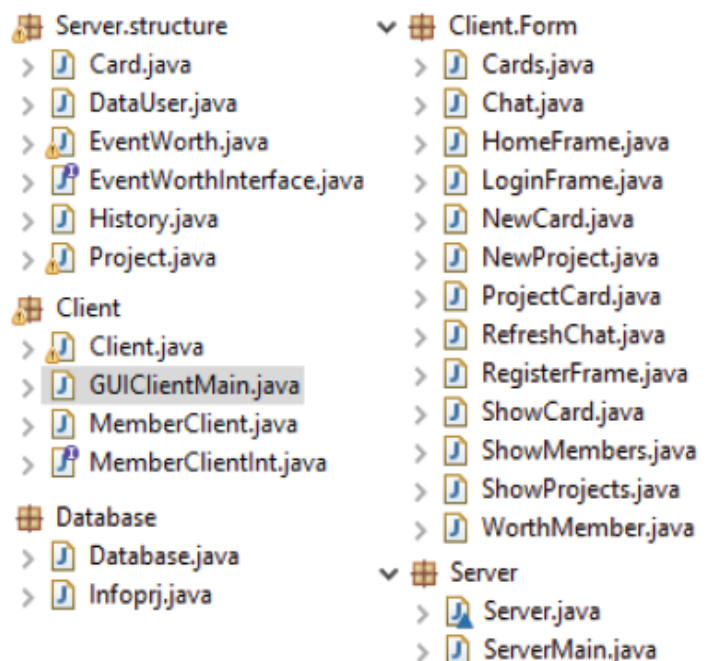
Classe del server vero e proprio, gestisce tutte le richieste.

### SERVERMAIN:

Classe da compilare per l'esecuzione del server.

### SERVER.STRUCTURE:

Tutte le classi della struttura del server.



## ISTRUZIONI WORTH

### LIBRERIE:

Il programma utilizza 3 librerie esterne per serializzare e deserializzare i dati in JSON, ho scelto di usare Jackson 2.9.9 (jackson-annotations-2.9.9, jackson-core-2.9.9, jackson-databind-2.9.9) che è la versione più recente trovata all'inizio della creazione del Progetto. Le librerie per semplicità sono situate all'interno del progetto nella cartella "lib".

### AVVIO:

Per avviare il programma basta compilare ServerMain per il server e GUIClient per l'interfaccia grafica del client. Non richiedono parametri opzionali in quanto ho deciso di utilizzare delle porte prefissate.

### SERVER:

Non appena il server sarà pronto, dopo aver letto il database e inizializzato il RMI e Multiplexing, scriverà :

```
"  
DB ready  
RMI ready  
Multiplexing ready  
"
```

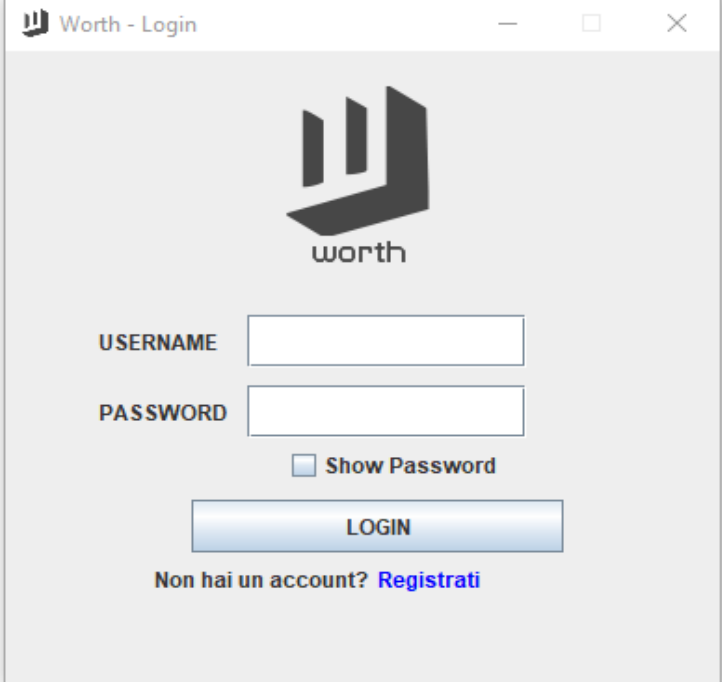
Fatto ciò il server adesso scriverà solo ad ogni connessione in entrata per il multiplexing (per tenere conto di quante connessioni si hanno) e un resoconto dei membri che si registrano (e cancellano la registrazione) per la callback.

## GUICLIENT:

### LOGIN:

Non appena verrà compilato si aprirà la schermata di login:

Il server predispone di 4 account default (dario, alessio, davide, valeria) dove la password è uguale all'username. Questi account sono stati usati per test e debug. Se si volesse creare un nuovo account basta selezionare con il mouse "Registrati" e si aprirà una schermata identica a quella del login ma per la registrazione. Se effettueremo il login, dopo si aprirà la schermata Home.



Worth - Login

worth

USERNAME

PASSWORD

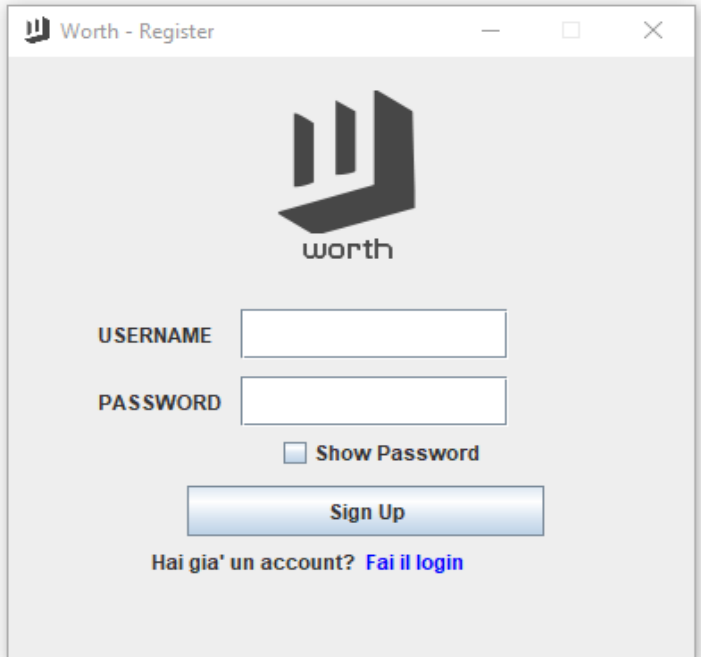
☐ Show Password

LOGIN

Non hai un account? [Registrati](#)

### REGISTER:

In questo form è possibile registrarsi, dopo aver inserito username e password e aver cliccato "Sign Up", se la registrazione è andata a buon fine, si aprirà di nuovo la schermata di Login (In alternativa si può cliccare su "Fai il login"), altrimenti verrà visualizzato un errore per poi re-inserire nuovamente i dati.



Worth - Register

worth

USERNAME

PASSWORD

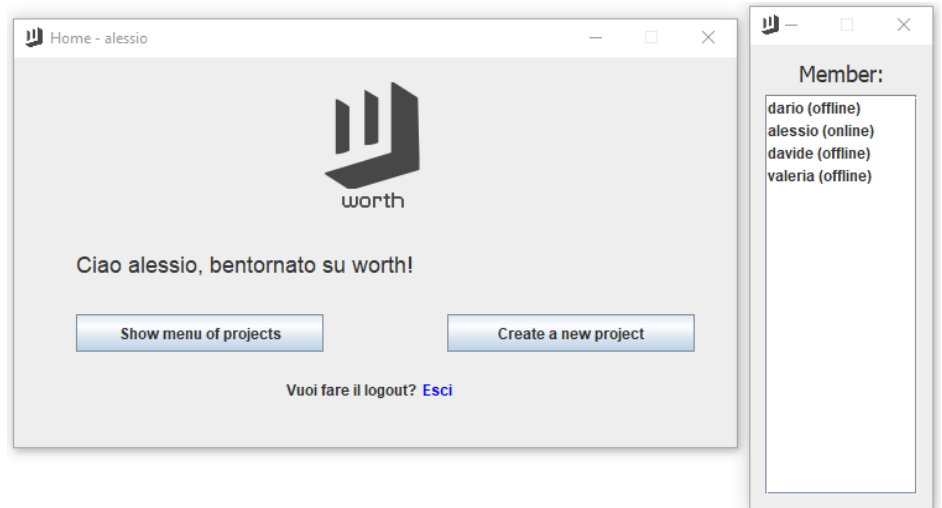
☐ Show Password

Sign Up

Hai gia' un account? [Fai il login](#)

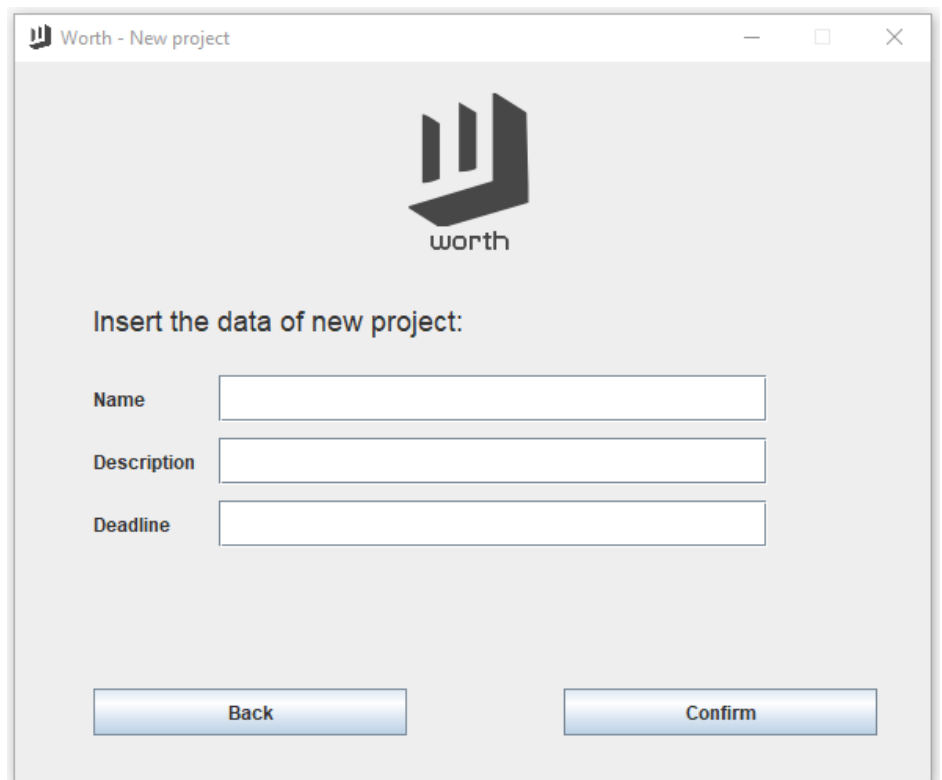
## HOME:

Nella schermata home avremo un messaggio di bentornato e 2 pulsanti, uno per la creazione di un progetto e l'altro per la visualizzazione dei progetti (quelli a cui abbiamo accesso). Avremo anche un'altra schermata che utilizzerà la RMICallback per l'aggiornamento dei membri online/offline, questa schermata sarà presente in tutti gli altri frame dell'interfaccia grafica, scomparirà solo al momento del logout ("premendo il tasto esci").



## NEW PROJECT:

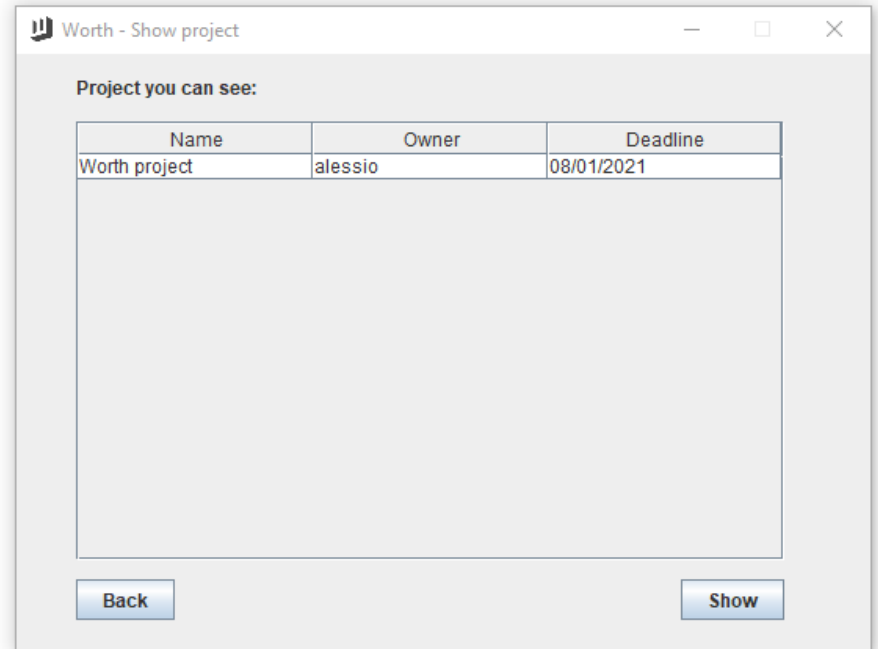
In questa schermata potremmo creare un nuovo progetto, basterà inserire il nome, la descrizione (facoltativa), una scadenza e successivamente premere conferma. Se la creazione andrà a buon fine, ci riporterà alla schermata Home.





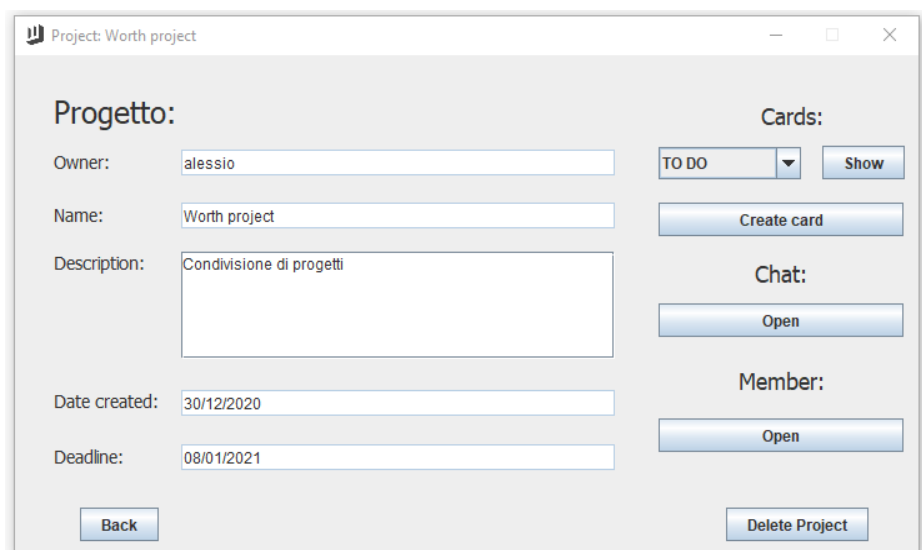
## SHOW PROJECTS:

In questa schermata andremo a visualizzare alcuni dati di tutti i progetti a cui abbiamo accesso. La tabella non è modificabile ma è possibile spostare le celle per cambiare la loro visualizzazione. Per vedere un progetto nel dettaglio basta semplicemente selezionarlo e premere il tasto "Show".



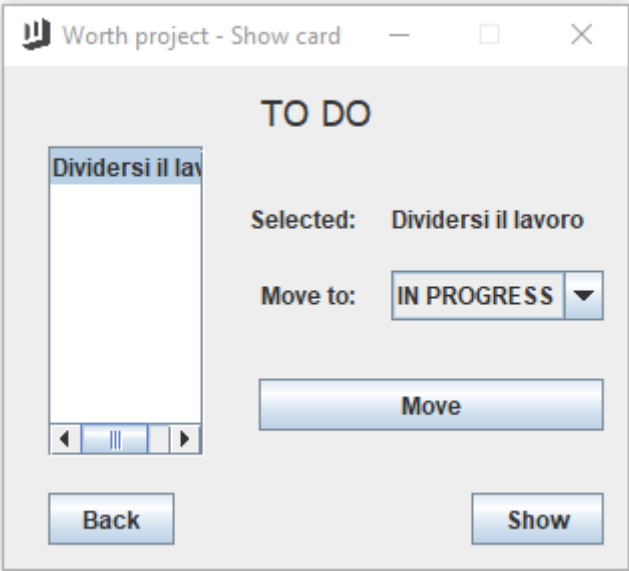
## SHOW PROJECT:

In questa interfaccia andremo a vedere nel dettaglio tutti i dati di un progetto, sarà anche possibile visualizzare i membri, la chat e le card in determinate liste tramite menu a tendina (TO DO, TO BE REVISED etc). Si avrà inoltre la possibilità di creare una nuova card e di eliminare il progetto (ovviamente solo in caso tutte le card siano nella lista DONE).



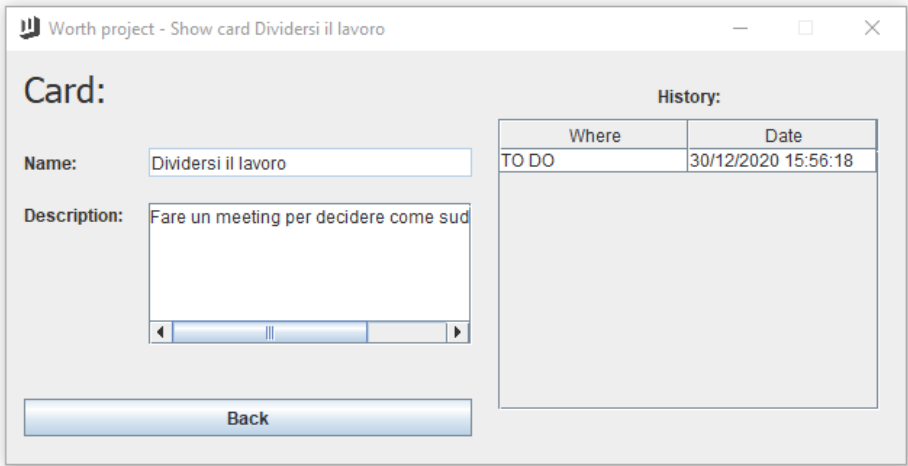
SHOW CARDS:

In questa interfaccia andremo a visualizzare tutte le card in una determinata lista (ad esempio TO DO), sarà possibile visualizzarle nel dettaglio selezionandola e premendo "show" e sarà anche possibile muoverla in un'altra lista (rispettando i vincoli degli spostamenti).



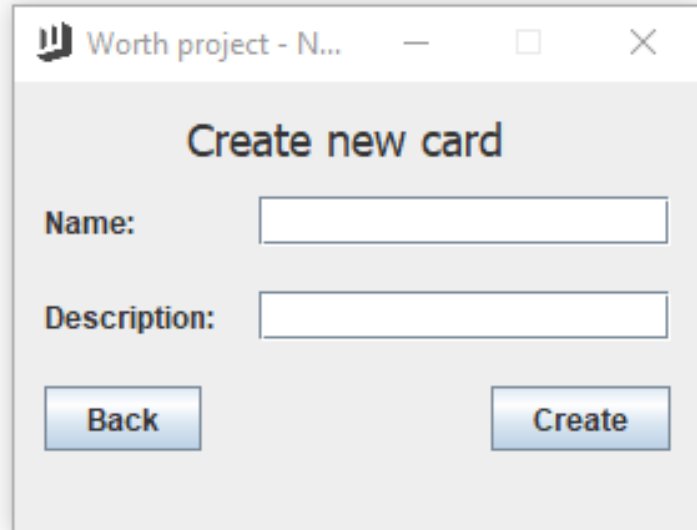
SHOW CARD:

In questa interfaccia visualizzeremo nel dettaglio una determinata card, con il nome, la descrizione e la history (da quando è stata creata al momento in cui apriamo l'interfaccia).



## NEW CARD:

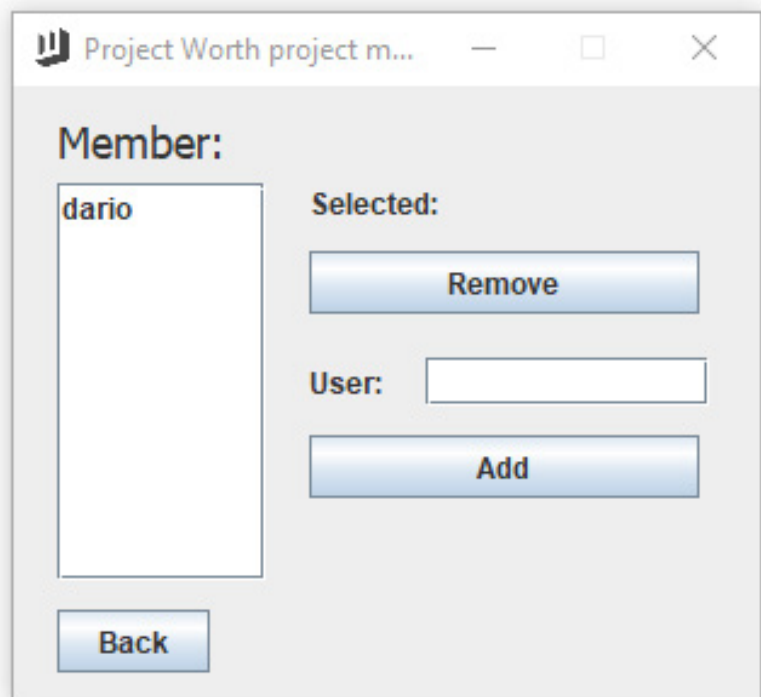
In questa interfaccia avremo l'occasione di creare una nuova card. Verranno richiesti il nome e la descrizione (facoltativa). C'è la possibilità, inoltre, di creare altre card, in quanto non ci riporta nella visualizzazione del progetto in cui siamo. Per ritornare indietro siamo obbligati a premere Back.



The screenshot shows a window titled "Worth project - N...". Inside, the title "Create new card" is centered. Below it, there are two labels: "Name:" and "Description:", each followed by a text input field. At the bottom, there are two buttons: "Back" on the left and "Create" on the right.

## SHOW MEMBER:

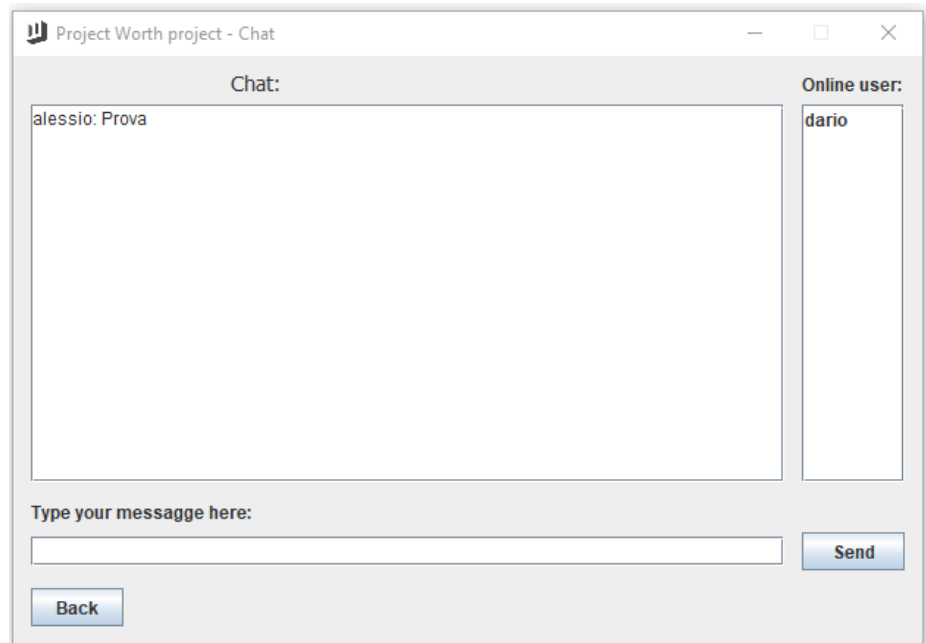
In questa interfaccia visualizzeremo i membri del progetto; potremo aggiungerne di nuovi o rimuovere quelli esistenti.



The screenshot shows a window titled "Project Worth project m...". Inside, the title "Member:" is at the top left. Below it is a list box containing the name "dario". To the right of the list box is the label "Selected:" followed by a "Remove" button. Below the "Remove" button is a "User:" label followed by a text input field, and then an "Add" button. At the bottom left, there is a "Back" button.

## CHAT:

In questa interfaccia avremo la possibilità di chattare con i membri che sono online e che hanno aperto la chat del progetto. Al centro potremo leggere e scrivere messaggi e sulla destra vengono visualizzati i membri online. Arriveranno inoltre delle notifiche di aggiornamento in caso qualcuno spostasse qualche card.



The screenshot shows a web browser window with the title "Project Worth project - Chat". The interface is divided into three main sections:

- Chat:** A large text area on the left containing the message "alessio: Prova".
- Online user:** A vertical list on the right showing the name "dario".
- Input area:** At the bottom, there is a text input field with the placeholder "Type your message here:", a "Send" button to its right, and a "Back" button below the input field.