PATonTS - Preferences AggregaTor ON Telegram and Signal

Angelo Di Iorio
Roberto Amadini
Leonardo Montecchiari

Progetto

- Il progetto consiste nello sviluppo di un'applicazione stand-alone in Java per raccogliere e gestire le preferenze di diversi utenti
  - Es. docenti che esprimono preferenze sull'orario, amici che organizzano attività di gruppo ecc…
- Il progetto si svolge in gruppi da **2** persone
  - Si può fare in casi particolari in gruppi da 3 persone, ma più complesso e soggetto ad approvazione
  - Non sono ammessi gruppi con >3 persone
  - Non sono ammessi progetti individuali, salvo casi eccezionali giustificati e approvati singolarmente

PATonTS - Preferences AggregaTor ON Telegram and Signal

- PATonTS è un'applicazione Java che permette ad un gruppo di utenti di esprimere preferenze sull'orario di alcune attività da svolgere per lavoro o interesse personale
- Applicazione JavaFX stand-alone: tutti gli utenti accedono alla stessa istanza per esprimere le loro preferenze
  - un'applicazione reale di questo tipo dovrebbe essere distribuita ma qui semplifichiamo
- L'istanza PATonTS memorizza i dati in locale (anche in uno o più file) e li carica/modifica ad ogni esecuzione
- L'utente entra nel sistema e legge o aggiorna le preferenze

Utenti

- **Amministratore**:
  - aggiunge utenti
  - modifica le informazioni degli utenti
  - aggiunge le attività su cui esprimere preferenze
  - associa le attività agli utenti
  - visualizza le preferenze
  - modifica le preferenze
- **Utente semplice**:
  - visualizza le preferenze
  - esprime le proprie preferenze
- Gli utenti esprimono preferenze in vari modi (vedi prossime slide)

Terminologia: Workspace e Attività

- Le preferenze degli utenti sono espresse all'interno di **workspace**, ossia un'area di lavoro o di interesse
  - *Esempi: orario delle lezioni, orario degli allenamenti, prenotazioni di un circolo, etc.*
- Un workspace coinvolge più utenti che possono esprimere le loro preferenze
- Un workspace è composto da **attività**
  - *Esempi: corso di programmazione, sessione di allenamento, serata danzante, etc.*
- Ogni utente può svolgere diverse attività
- Un'attività può essere svolta da più utenti, che quindi possono esprimere preferenze diverse

Workspace

- L'utente entra nel sistema e vede l'elenco dei workspace in cui è coinvolto
- Un workspace è quindi caratterizzato (almeno) da un ID e una descrizione testuale
- L'utente sceglie ed entra in un workspace e può esprimere le proprie preferenze
- L'interfaccia permette di passare da un workspace all'altro
- L'amministratore ha visibilità su tutti i workspace caricati nel sistema e può creare nuovi workspace

Attività e slot

- Un workspace è organizzato in attività, che possono essere ripetute in diversi slot di tempo, ad esempio “lezione il lunedì e il mercoledì”, oppure “sessione di allenamento tutti i martedì e giovedì”
- Un’attività è caratterizza da alcune informazioni obbligatorie:
  - ID e descrizione
  - Luoghi in cui l’attività si svolge e che l’utente potrà scegliere. Possono essere anche più di uno, ad esempio aule diverse delle lezioni oppure campi diversi in un impianto sportivo
- Altre informazioni sono opzionali e ogni gruppo può aggiungerne altre ancora, ad esempio:
  - Data inizio
  - Data fine
  - Durata
  - Organizzazione degli slot

Preferenze

- Per ogni coppia `<utente, attività>` si può esprimere una preferenza

  `<utente, attività>  --->  <preferenza>`

- I dettagli delle preferenze dipendono dal workspace e sono a discrezione di ogni gruppo

- Ad esempio nel caso dell'orario delle lezioni:
  - Durata slot: 2 o 3 ore
  - Tempo diviso in slot orari (una o due ore a scelta)
  - Slot in cui non si vuole svolgere l'attività o si preferisce svolgerla
  - Requisiti sui luoghi

Esempi

Workspace		Orario del Corso di InfoMan	Orario Allenamento Calcio	Orario Sala Prova
Attività		- Lezione di TW - Lezione di LAPI	- Allenamento Pulcini - Allenamento Allievi - ...	- Gruppo A - Grupo B - ...
Preferenze	- slot orari in cui voglio/non voglio fare lezione - sovrapposizione con altre attività - note testuali	- durata min/max attività - giorni in cui voglio/non voglio fare lezione - note testuali	- durata sessione - giorno in cui voglio/non voglio fare allenamento - note testuali	- durata sessione - giorno in cui voglio/non voglio fare le prove - richieste su strumentazione - note testuali

GUI PATonTS

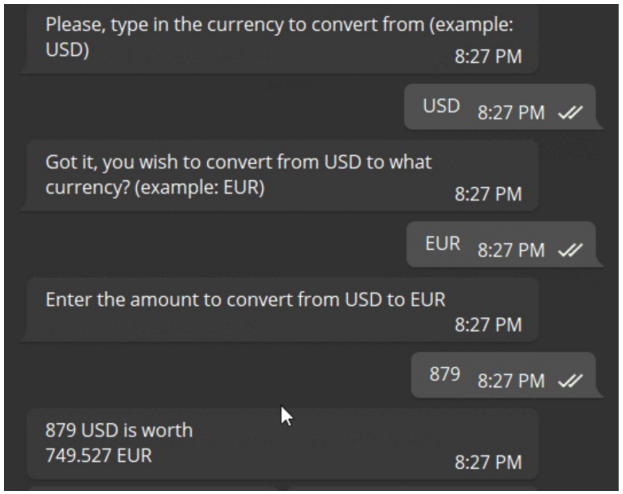
- All'avvio PATonTS mostra una schermata per l'autenticazione
- Importante riconoscere il ruolo dell'utente (amministratore o utente semplice) per mostrare il pannello appropriato
- Include inoltre una o più schermate di *help* e *credits* sul gruppo
- La GUI include anche opzioni per caricare i dati da file ed esportarli, come descritto nella prossima slide

Caricamento da file

- PATonTS permette di caricare un file e da lì legge i dati su :
  - utenti e attività
  - preferenze
- Ogni gruppo sceglie il formato dati da usare e le informazioni che è possibile caricare in questa modalità
- Necessari ovviamente controlli di consistenza sui dati
- In caso di errore è possibile ignorarlo o sovrascrivere solo alcuni dati o tutti
  - Ogni gruppo sceglie la strategia che preferisce purché ben documentata e consistente

Interfaccia Telegram

- L'utente può esprimere preferenze anche interagendo con un bot Telegram
- Nella prossima lezione vedremo come usare l'API di Telegram e scrivere un bot
- Esempio Currency Converter



Interfaccia Telegram

- L'utente interagisce con il bot per esprimere preferenze
- Può aggiungere o sovrascrivere preferenze solo se (1) ha un profilo nel sistema (2) è associato all'attività per cui esprime le preferenze
- Per attivare questa funzionalità è necessario quindi aggiungere al profilo degli utenti anche lo username da usare per Telegram
- Su richiesta dell'utente (base o amministratore), PATonTS si collega aTelegram, legge i messaggi e carica i dati
- Semplificazione: durante la fase di sincronizzazione Telegram le altre funzionalità della GUI sono disabilitate

Requisiti obbligatori

- Queste specifiche sono poco vincolanti di proposito, siete invitati a personalizzare e aggiungere funzionalità
- E' richiesto tuttavia implementare **obbligatoriamente**:
  - GUI - Pannello amministratore
    - aggiunta/modifica utenti
    - aggiunta/modifica workspace
    - aggiunta/modifica attività
      - id, descrizione, luogo
    - modifica delle preferenze espresse dagli utenti
  - GUI - Pannello utente
    - scelta workspace ed elenco attività
    - aggiunta/modifica preferenze
      - *<a scelta da parte del gruppo>*

Requisiti obbligatori

- Gestione file
  - Caricamento dello stato dell'applicazione (utenti, workspace, attività, preferenze, etc.) da file
  - Export dei dati su file (per successivo caricamento)
  - Caricamento di nuove preferenze relative ad una o più attività attraverso upload di un file
- Integrazione Bot
  - Bot che permette di leggere le proprie preferenze e aggiungerne nuove
  - Le altre funzionalità sono a discrezione del gruppo; ed esempio, è possibile elencare i workspace o le attività, dare suggerimenti all'utente, etc.
  - Noi guarderemo Telegram ma si può implementare anche un bot Signal in alternativa

Estensioni

- Qualche idea NON VINCOLANTE per estendere il progetto:
  - Scelta dinamica del Bot e Bot multipli via file di configurazione
  - Notifiche agli utenti sull'interfaccia o via canali esterni (mail, bot, etc.)
  - Controlli sulle preferenze, vincoli e inconsistenze
  - Archivio e statistiche
  - Esportazioni in altri formati

Discussione e consegna

- Il progetto si discute solo dopo aver superato la prova di laboratorio
- Alla discussione del progetto devono essere presenti **tutti** i membri del gruppo, salvo casi eccezionali
  - Es. Borsa di studio o gravi impedimenti
- Il progetto si consegna **prima** della data di discussione (~5 giorni prima) su virtuale.unibo.it
- Il progetto rimane valido per l'a.a. 2020/21
- Discussioni da remoto (Zoom): 10/6, 1/7 e 19/7
  - **Iscrivere una sola persona per gruppo su AlmaEsami**

Consegna

- Il progetto si consegna su Virtuale. Prima di oggi appello saranno pubblicate apposite sezioni per la consegna con le istruzioni
- Cosa consegnare
  - Sorgenti Java
  - JAR eseguibile (provatelo su computer diversi!)
  - **Breve** relazione (max 3 pagine)
  - Alcuni dati precaricati:
    - utente base e amministratore
    - qualche attività e preferenza
    - tutti i dati che ritenete utili per mostrare il funzionamento del software

Valutazione progetto

- Le funzionalità aggiuntive aiutano ad ottenere un voto più alto (ma devono essere ben implementate)
- L'applicazione deve:
  - Soddisfare i requisiti obbligatori
  - Funzionare
  - Essere usabile e con una chiara gestione degli errori
  - Essere ben strutturata e flessibile
  - Funzionare
- Soprattutto: deve essere frutto del vostro lavoro, dovete essere in grado di spiegare/modificare il codice all'esame
- La valutazione non è di gruppo ma singola. Lo stesso progetto può portare a voti diversi in base al contributo di ognuno

Valutazione progetto

VOTO FINALE ≅ VOTO LABORATORIO ± 3

- Il voto di laboratorio è la **base** con cui presentarsi alla discussione del progetto
- Il progetto in sé **non** ha un voto proprio ma è una "*traslazione*" del voto di laboratorio
  - Salvo casi eccezionali, a seconda del progetto **e** della sua discussione orale il voto di laboratorio può essere:
    - **confermato**
    - **decrementato** da **1** a **3** punti
    - **incrementato** da **1** a **3** punti

Domande?