VirtualLabs

Progetto Applicazioni Internet a.a. 20/21

Indicazioni generali

- Sviluppo di un applicazione web basata su
  - Java Spring
  - Angular 2+
  - PostgreSQL/MongoDB/MariaDB
- Il server deve esporre le funzionalità offerte al client attraverso una interfaccia REST utilizzabile anche da applicazioni terze (es. dall'applicazione mobile).
- Gruppi composti da massimo 4 persone
  - I gruppi sono formati in modo autonomo dagli studenti
- La consegna
  - deve prevedere l'utilizzo di docker per istanziare server, db ed (eventualmente) applicazione client tramite server web
  - deve essere corredata da un breve testo con le istruzioni per compilazione, deploy e utilizzo (es. username e password dei vari utenti già presenti)
- Consegna e revisione su appuntamento preferibilmente in sessione esami
  - Necessaria comunicazione ai docenti con preavviso di almeno una settimana tramite email con oggetto: "Applicazioni Internet - Appuntamento x revisione progetto"
  - Upload del codice sorgente nell'area elaborati del corso prima della richiesta di appuntamento (con istruzioni per compilazione, esecuzione e utilizzo)
  - Presenza attiva di tutti i membri del gruppo durante la revisione (presumibilmente telematica)
  - Gruppo di discussione: gruppo slack ai-polito-2020.slack.com, channel #progetto

Descrizione

Questo progetto è volto a realizzare una applicazione web per favorire la gestione di laboratori virtuali in cui gli studenti di un corso universitario lavorano a gruppi su macchine virtuali che il docente può monitorare e a cui lo stesso docente può partecipare per dare consigli, verificare problematiche o collaborare nello svolgimento di esercizi.

Le macchine virtuali possono essere utilizzate e gestite dagli studenti dopo essersi organizzati in gruppi. Gruppi che devono essere formati da un numero di studenti compreso nei limiti minimi e massimi decisi dal docente del corso.

La composizione dei gruppi avviene "tipo un'asta" dove ogni studente può lanciare una offerta dichiarando il nome di un gruppo, univoco, e la composizione che vorrebbe "nominando" gli studenti da cui vorrebbe fosse composta, oltre ovviamente a se stesso. Ciascuno studente può

aderire ad una e una sola offerta per volta. Se almeno uno degli studenti presenti nell'offerta per un gruppo segnala di non essere disponibile a partecipare al gruppo, l'offerta si scioglie.

Modello dati

Utente: Studente

Lo studente dell'università è caratterizzato dall'email @studenti.polito.it, dal nome, cognome, matricola e foto. Lo studente, può essere associato ad uno e un solo *gruppo*, può creare/cancellare/eseguire/spegnere istanze di macchine virtuali di quel gruppo. Lo studente può essere iscritto a zero, uno o più corsi.

Gruppo

Il gruppo corrisponde ad un gruppo di studenti, ha un nome, un identificativo ed è associato ad uno e uno solo *modello* di macchina virtuale e a uno e uno solo corso. Non ci possono essere due gruppi con lo stesso nome all'interno dello stesso corso.

Per ciascun gruppo il docente imposta un limite di risorse utilizzabili in termini di numero di vcpu, spazio disco e ram, numero di istanze attive contemporaneamente e numero massimo di istanze disponibili (cioè somma di quelle attive e spente). Il limite è per gruppo così gli studenti interni al gruppo si coordinano tra di loro, non è per corso altrimenti un gruppo potrebbe esaurire le risorse a danno degli altri gruppi.

Corso

Il corso universitario è caratterizzato da un nome, un acronimo, e ad esso sono associati gli studenti iscritti a quel corso. Il corso può essere attivo o spento (se è spento non si possono utilizzare le corrispondenti macchine virtuali). Su ogni corso è impostata la dimensione minima e massima di studenti che possono comporre un gruppo, per quel corso.

Utente: Docente

Il docente dell'università è caratterizzato dall'email, da un nome, cognome, matricola e foto. Il docente può essere associato a zero, uno o più corsi. E' il docente stesso che crea i corsi e li popola con gli studenti iscritti. Il docente non crea o gestisce macchine virtuali, ma può connettersi a quelle degli studenti/gruppi dei corsi che gestisce. Attenzione: ci possono essere più docenti gestori dello stesso corso.

Modello VM

A ciascun ~~gruppo~~ **corso** è collegato un modello di VM (pensatelo come una configurazione docker) definito ~~dagli studenti del gruppo~~ **dal docente** e usato per creare le istanze delle macchine virtuali.

VM

Le istanze delle macchine virtuali vengono create a partire da un Modello di VM e, all'atto della creazione lo studente può impostarne alcuni parametri di configurazione: numero di vcpu, spazio disco e ram. Lo studente che crea la VM ne è l'owner, ma può scegliere di condividere questa qualifica con tutti gli appartenenti del gruppo. Chi ha la qualifica di owner può attivare, spegnere, cancellare la VM.

Il docente può accedere in qualsiasi momento a tutte le vm degli studenti/gruppi dei suoi corsi, gli studenti solo a quelle del proprio gruppo.

NOTA BENE. Al momento non si ritiene attuabile l'avvio e il collegamento ad una VM reale, si utilizzi quindi uno screenshot, cioè una immagine, al posto di una VM vera e propria.

Consegna

La consegna consiste nel testo di una esercitazione da svolgere utilizzando la VM del corso. La consegna è creata dal docente, è associata ad un corso, ha una data di rilascio, una scadenza e un contenuto. Per semplicità si assuma che il contenuto sia una immagine in cui è rappresentato il testo della consegna.

Elaborato

L'elaborato è la soluzione svolta dallo studente per una data consegna. Anche questo elemento può essere rappresentato semplicemente da una immagine in cui è visualizzato lo svolgimento della consegna.

Stato della elaborato (IMPORTANTE)

Si vuole implementare un meccanismo di gestione dello stato dell'elaborato, dove a ciascuno stato corrisponde una versione dell'elaborato (cioè una immagine diversa). Descrizione dello stato:

1. Inizialmente l'elaborato deve riflettere il fatto che lo studente abbia letto o no la consegna, quindi impropriamente il suo stato sarà NULL fintanto che lo studente non avrà aperto/letto la consegna, dopodiché lo stato diventerà LETTO.
2. In questa fase l'elaborato sarà in svolgimento fintanto che non verrà consegnato o attivamente perché è lo studente stesso che lo consegna o passivamente perché arriva la scadenza definita nella consegna. Quando l'elaborato viene consegnato il suo stato diventa CONSEGNATO.
3. A questo punto è compito del docente rivedere/correggere l'elaborato. Quando il docente l'avrà corretto allora lo stato dell'elaborato diventerà RIVISTO.

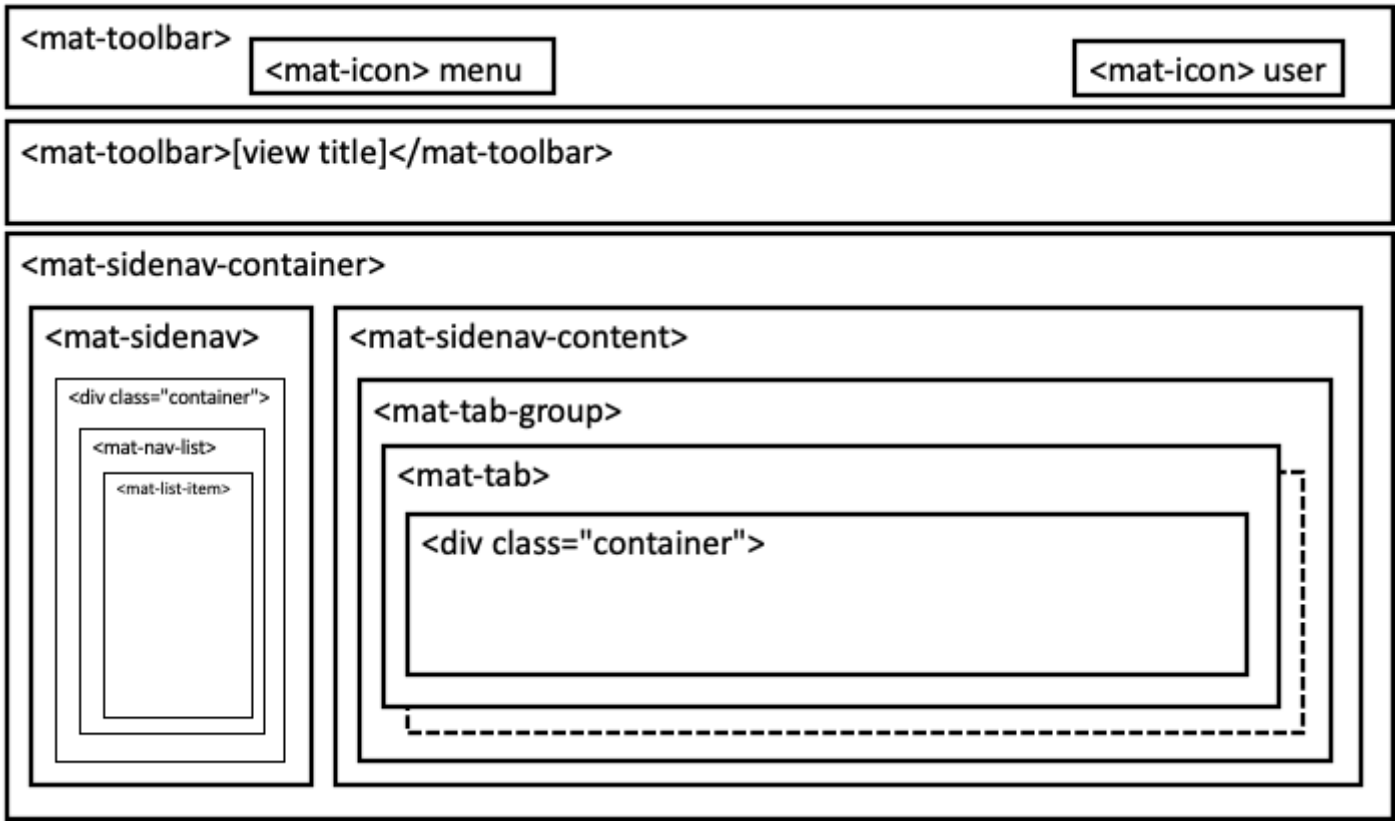
A questo punto si vuole gestire il fatto che l'elaborato può essere letto, consegnato e rivisto un numero arbitrario di volte, fintanto che il docente ne dà possibilità allo studente (settando un apposito flag sull'elaborato). N.B. le varie immagini corrispondenti alle differenti fasi dell'elaborato, così come i timestamp in cui vengono caricate, non devono essere scartati o sovrascritti di volta in volta, ma occorre mantenerne memoria per poter rivedere anche successivamente l'evoluzione della prova.

Qualora l’elaborato consegnato sia da ritenersi definitivo e così la correzione, il docente potrà assegnare un voto/giudizio allo stesso.

Viste dell’applicazione

L’applicazione deve avere un layout indicativamente simile al seguente (laddove non insorgano limiti particolari), come sviluppato nei laboratori. Dove:

- La toolbar contiene una icona per aprire/chiudere la sidenav e una icona per accedere al profilo utente, login/logout, e similia.
- La sidenav contiene i link per navigare tra i vari corsi a cui è iscritto
- Il contenuto della vista è renderizzato nel sidenav content
- Il sidenav content è diviso in varie tab per “passare” a contesti specifici (es. studenti del corso, gruppi e macchine virtuali, consegne ed elaborati)



Utente: login, register

La registrazione di un utente nel sistema è possibile solo agli utenti del dominio universitario. Si definiscano quindi dei domini abilitati per gli indirizzi email (es. @polito.it, @studenti.polito.it). Alla richiesta di registrazione da parte di un utente si richieda nome, cognome, matricola, password e indirizzo email. Ad avvenuta registrazione all’utente viene inviata una mail con un link di attivazione per verificare che l’email indicata sia corretta. E’ anche necessario definire una interfaccia per il login/logout dell’utente, tramite la password registrata.

Docente: gestione corsi

Il docente ha a disposizione una vista per gestire gli studenti iscritti al corso.

Si permetta all'utente di creare / cancellare / modificare i corsi e definirne il nome.

Una tabella in elenca gli studenti iscritti al corso con matricola, nome, cognome, nome gruppo.

Gli studenti vengono aggiunti o uno a uno utilizzando un campo di input testuale in cui, tramite la funzionalità autocomplete, è possibile ricercare e selezionare da un db lo studente da aggiungere, oppure a partire da un file csv.

Gli studenti sono rimossi utilizzando delle checkbox nella tabella per selezionarli e un bottone per eliminarli.

La tabella preveda l'ordinamento e la paginazione. Attenzione: è richiesto (come fa gmail) che la master checkbox della tabella abbia due modalità di utilizzo, di default seleziona solo le righe della pagina corrente, ma chiede poi e permette di scegliere se la selezione deve essere applicata a tutte le righe della tabella.

Docente: gestione VM

Il docente ha a disposizione una vista per visualizzare le VM degli studenti, anche qui accessibile attraverso una tab della "pagina" del corso.

In questa vista vengono elencati i gruppi presenti nel corso e per ogni gruppo le VM disponibili con l'informazione dello studente che l'ha creata, dello stato attuale (attiva, spenta, ecc.) e con un link per effettuare il collegamento alla stessa.

Da questa pagina il docente può configurare per ogni gruppo i limiti di utilizzo di risorse (calcolati sull'insieme delle vm del gruppo) in termini di vcpu, ram e disco totali, ma anche numero massimo di VM attive contemporaneamente e numero massimo di VM totale (cioè ottenuto sommando le attive e le inattive) Si predisponga un popup per editare queste informazioni, uno specchietto riassuntivo. All'atto dell'editing da parte del docente, lo si avverta se l'eventuale modifica delle impostazioni è minore delle risorse già utilizzate dal gruppo perché in questo caso i limiti non possono essere applicati). Opzionale: fornire anche una indicazione in tempo reale delle risorse attualmente utilizzate dal gruppo.

Studente: gestione gruppi

Lo studente ha a disposizione una vista per visualizzare ed effettuare le richieste di adesione a gruppi per ciascun corso, anche qui in una tab del corso.

Nella vista, se l'utente è iscritto ad un gruppo, viene indicato il nome e gli studenti che lo compongono.

-- Altrimenti --

Nella parte superiore potrà creare la richiesta di composizione gruppo: da una tabella con l'elenco degli studenti iscritti al corso potrà scegliere coloro ai quali inviare la richiesta e, prima dell'invio,

definire il nome del gruppo e la durata della proposta (timeout). Possono essere selezionati solo gli studenti che non sono già parte di un gruppo completato (già accettato cioè da tutti i componenti proposti). N.B. chi invia la richiesta viene automaticamente considerato come aderente al gruppo. Attenzione: si verifichi che la richiesta rispetti il range min max di studenti per gruppo imposto dal docente.

Nella parte inferiore vedrà invece le varie richieste, per ognuna il proponente (chi l'ha creata), il nome scelto per il gruppo, la possibile composizione con matricola, nome, cognome degli studenti, dove, per ciascuno studente è indicato lo stato di adesione.

Lo studente può scegliere se accettare o rifiutare la proposta. Appena uno studente rifiuta una proposta, la proposta viene disabilitata e non è più possibile per nessuno aderirvi. La proposta si disabilita anche allo scadere del timeout definito in fase di creazione. Si definisca un modo per eliminare dalla vista le proposte "vecchie" per fare pulizia della schermata (es. le cancella l'utente, come si fa per le email, oppure si cancellano in automatico passato un tot di tempo).

Studente: gestione VM

Lo studente ha a disposizione una vista per visualizzare le VM a lui accessibili, anche qui in una tab del corso.

La vista è simile a quella della gestione VM del docente, ma in questo caso, per quelle di cui è owner, può provvedere all'accensione, spegnimento, cancellazione. L'owner può anche modificare le risorse associate alla VM, ma soltanto se è spenta e se non superano i limiti imposti al gruppo.

Nella vista è disponibile anche un icona/bottone per creare una nuova istanza di VM, in questo caso l'istanza deve essere configurata con il numero di vcpu, ram e disco e occorre verificare, prima della creazione, che tali risorse siano disponibili, altrimenti segnalare l'errore.

Docente: consegne ed elaborati

Il docente ha a disposizione una vista (cioè una tab all'interno di ciascun corso) per visualizzare le consegne affidate agli studenti e gli elaborati in cui sono state svolte.

Si raggruppi la vista per consegne in ordine temporale e per ogni consegna si visualizzi la lista degli elaborati degli studenti con il loro stato. Qui gli elaborati devono poter essere filtrati in base al loro stato (es. visualizzare tutti quelli NULL, cioè non ancora letti dallo studente, oppure tutti quelli consegnati e da rivedere) e ordinati in ordine temporale. Per ogni elaborato occorre poter vedere il nome, cognome, matricola dello studente a cui corrisponde, lo stato e il timestamp in cui quello stato è stato assegnato.

Cliccando poi sull'elaborato lo il docente può visualizzarne lo "storico", rivedere l'elaborato "caricato" in ciascuna delle fasi passate, e caricare la sua correzione indicando se attenderà o no da parte dello studente una revisione o se questo è da considerarsi l'elaborato definitivo. In quest'ultimo caso il docente assegnerà anche un voto o un giudizio all'elaborato.

Studente: consegne ed elaborati

Si realizzi per lo studente una vista analoga a quella disponibile per il docente. Qui lo studente prenderà visione della consegna, sottometterà poi gli elaborati o in seguito alla consegna o in seguito alle correzioni (richieste di revisione) del docente.

Valutazione

La valutazione terrà conto dei seguenti aspetti:

- Soddisfacimento delle specifiche e correttezza del funzionamento del sistema e gestione delle varie funzionalità
- Usabilità e funzionalità dell'interfaccia utente
- Ingegnerizzazione e scalabilità della implementazione proposta;
- Leggibilità del codice e documentazione tramite commenti
- Originalità ed efficacia nel raggiungimento degli obiettivi del progetto