

Laboratorio B
Progetto SeatIn
Simple E-learning at Insubria
Anno accademico 2017/2018

Relazione del progetto



P. Farina (727825) , D. Muraro (716298), A. Roshka (730662), C. Stroppolo (724781)

1.Introduzione	4
1.1 Descrizione del problema	4
1.2 Motivazione	4
2. Panoramica	5
3. Requisiti	6
3.1 Avvio del server	6
3.2 Autenticazione	6
3.3 Registrazione corsi ed utenti	6
3.4 Attivazione profilo utente	6
3.5 Reset password	6
3.6 Gestione utenti	6
3.7 Accesso ai contenuti	6
3.8 Contenuti del corso	7
3.9 Comunicazione	7
3.10 Monitoraggio	7
4. Architettura scelta	8
4.1 Modello Struttura	8
4.1.1 Server Based	8
4.1.1.1 Sistema client/server	8
4.1.1.2 Connection Pool	8
4.2 Metodo Trasmissione	9
4.2.1 TCP/IP	9
4.2.1.1 Differenza con RMI	9
4.3 Gestione Progetti Software	10
4.3.1 Maven	10
5. Requirements UML Analysis	11
5.1 Use case diagram	11
5.2 Package diagram (Struttura server)	12
5.3 seatInServer Class diagram	13
5.4 seatInAdmin Class diagram	13
5.5 seatInUser Class diagram	14
5.6 seatInBeans Class diagram	15
6. Organizzazione	16
6.1 Divisione del lavoro	16
7. Informazioni sull'installazione	18
7.1 Prerequisiti	18
7.2 Installazione database	18
7.2.1 Installazione usando il client PGAdmin	18
7.2.2 Installazione usando la console interattiva	18
7.3 Installazione applicativo	19

7.4 Avvio applicazione	20
7.5 Documentazione Java	20

1.Introduzione

L'E-learning è un portale per l'apprendimento online di nozioni che possono essere utilizzate da tutti o, comunque, dagli utenti abilitati. Questo portale migliora la qualità dell'apprendimento, rappresentando un compromesso efficace tra la formazione tradizionale e quella basata sulla tecnologia.

Poiché basato sulle tecnologie qualsiasi utente può riuscire ad accedere alle informazioni presenti in qualsiasi momento e in qualsiasi luogo, sempre se abbia una connessione alla rete internet attiva.

Diversi sono i vantaggi nell'utilizzo dell'E-learning, ma vediamo i tre principali.

Il primo è la sua *dinamicità*. A differenza dei libri infatti l'e-learning può essere costantemente aggiornato con nuovi oggetti e si possono correggere eventuali sbagli in tempo pressoché nullo.

Il secondo è il contesto di *immediatezza*. Difatti non essendoci intermediari il materiale viene distribuito direttamente agli utenti che ne necessitano.

Il terzo, infine, è la sua *modularità*. Infatti un E-Learning può essere organizzato attraverso i propri contenuti, permettendo così all'utente finale di avere minori perdite di tempo nel caso dei ricerca di oggetti.

Il portale di e-learning presentato nel documento è frutto di un tema d'esame del corso di Laboratorio B per l'anno accademico 2017/2018, il cui fine è quello di presentare un e-learning funzionante.

Nel documento si troveranno analisi e diagrammi creati in fase di progettazione dell'applicativo SeatIn (Simple e-learning at Insubria).

1.1 Descrizione del problema

Il problema che deve risolvere il nostro applicativo è la gestione dei dati delle persone e dei corsi registrati nell'università.

1.2 Motivazione

Il motivo principale per la creazione del sistema SeatIn è la gestione di dati che vengono messi a disposizione agli utenti regolarmente iscritti all'università, con i relativi permessi, che siano amministratori, docenti e studenti. Questo avviene tramite un applicativo Client/Server con gestione manuale delle iscrizioni ad e-learning ed attivazione corsi.

2. Panoramica

Il sistema SeatIn è composto dai moduli: client e server.

Il modulo client può essere di tipo Student, Lecture o Admin mentre il modulo ha invece il compito di gestire gli account con: inizializzazione del primo accesso al server, attivazione dell'utente, generazione password temporanea, invio e-mail e recupera gli account bloccati.

Ognuno ha poteri maggiori all'interno dell'applicazione, partendo dal modulo Student che è quello avente minori possibilità di iterazione, fino ad arrivare all'Admin che gestisce il tutto.

La registrazione degli utenti è gestita dall'admin, che creano l'account con i relativi permessi ad ogni persona che si iscrive all'università, con una password temporanea che al momento dell'accesso verrà sostituita.

La creazione di corsi è gestita dall'admin, che crea ed assegna ai propri corsi i docenti, i quali caricheranno a propria discrezione argomenti delle proprie lezioni.

3. Requisiti

Per poter iniziare ad operare il sistema deve garantire:

3.1 Avvio del server

- Credenziali per accensione al database di supporto
- Host del database
 - Se nessun profilo è contenuto procede alla registrazione

3.2 Autenticazione

- Diversi login se admin e studente o docente deve fare l'accesso
- Fallimento: dopo 10 tentativi viene bloccato il profilo
 - Recupero: amministratore

3.3 Registrazione corsi ed utenti

- Creazione da parte dell'admin su indicazione delle segreterie didattiche
- Assegnazione
 - corsi a docenti
 - per gli utenti una password temporanea

3.4 Attivazione profilo utente

- In seguito al primo accesso da parte dell'utente, seatIn fornisce la possibilità di attivare l'account, che consiste nella verifica dell'identità dell'utente, ed in seguito nella sostituzione della password temporanea assegnata da seatIn con la password inserita dall'utente stesso.

3.5 Reset password

- L'utente richiede il recupero della password dimenticata. SeatIn sostituisce la password dimenticata con una generata automaticamente, e la manda via mail all'utente stesso.

3.6 Gestione utenti

- Amministratore modifica i dati registrati
- Studente o Docente visualizzano le informazioni profilo

3.7 Accesso ai contenuti

- Studente: Visualizzazione corsi a cui è iscritto
- Docente: Corsi gestiti

- Gestione dati dei corsi
- Iscrizione ad un corso notificata tramite email

3.8 Contenuti del corso

- Inserimento sezioni a corso, sottosezioni, file
- Modifica titolo o descrizione, visibilità
- Cancellazione di una sezione e propagazione
- Visualizzazione contenuti

3.9 Comunicazione

- SeatIn offre la possibilità di comunicazione tramite Newsletter che può essere usata da professori e Admin, per comunicare gli aggiornamenti agli studenti di un determinato corso, o ad una selezione di studenti.
- Gli studenti possono, a sua volta, mandare le mail ai docenti.

3.10 Monitoraggio

- Visualizzazione complessiva utenti connessi che visualizzano o effettuano il download di contenuti
- Derivazione tempo medio connessione

4. Architettura scelta

Per la realizzazione dell'e-learning si sono dovute effettuare delle scelte per ottenere la migliore soluzione.

4.1 Modello Struttura

4.1.1 Server Based

L'applicativo di E-Learning realizzato è basato su una tecnologia di tipo server based.

Questo significa che le operazioni avvengono su server esterni, invece che sul personal computer, consentendo di estendere le risorse, semplificare la diffusione delle informazioni e gestire le applicazioni.

Grazie a questa innovazione al momento di aggiornare un elemento su tutte le periferiche collegate, basta eseguirlo un'unica volta sul server, che tutte si troveranno l'elemento modificato.

Una tecnologia server based ha tre vantaggi principali.

Un *sistema operativo multi-utente*, che consente l'avvio simultaneo di applicazioni a più utenti contemporaneamente. Nonostante questi siano collocati in qualsiasi luogo al mondo.

Una *tecnologia di elaborazione efficiente*, che fa in modo che le operazioni effettuate siano indipendenti dall'ampiezza della banda disponibile in quanto la parte logica dell'applicazione è separata dall'interfaccia utente.

Una *gestione centralizzata* dell'applicazione e del client, che elimina i problemi che le applicazioni possono avere, come i problemi di gestione, accesso, rendimento e sicurezza.

4.1.1.1 Sistema client/server

I server di e-learning comunicano con un sistema chiamato client-server. Questo significa che l'e-learning usa un tipo di rete che permette al client di usufruire ai servizi offerti dai server, tramite dei protocolli di comunicazione che gestiscono le connessioni e permettono di comunicare chiaramente.

4.1.1.2 Connection Pool

Con il termine connection pool si definisce la memorizzazione in una struttura dati delle connessioni ad un database. Utilizzando questo metodo si permette all'utente di fare gli accessi maggiormente veloci in futuro e senza rischiare di sovraccaricare il server, che riesce così a gestire un maggior numero di visitatori.

4.2 Metodo Trasmissione

4.2.1 TCP/IP

Il Transmission Control Protocol/Internet Protocol è l'insieme dei protocolli di trasmissione usati per l'interscambio di dati su internet che si occupano di rendere affidabile, tramite dei controlli di trasmissione, la comunicazione in rete fra mittente e destinatario.

Il TCP organizza la frammentazione in pacchetti dei dati da inviare e l'IP invece monitora e verifica il corretto instradamento dei pacchetti.

E' caratterizzato da quattro livelli principali:

- 1- Applicazione: Protocolli di alto livello e dialogo con l'utente. Sono:
 - a. FTP (File Transfer Protocol) – si occupa del trasferimento dei file
 - b. DNS (Domain Name System) – traduce i nomi in indirizzi
 - c. NFS (Network File System) - permette l'accesso a file memorizzati su un dispositivo remoto
 - d. SMTP (Simple Mail Transfer Protocol) – amministra le trasmissioni di e-mail in rete
 - e. Telnet (Terminal Emulation) – possibilità di accedere in remoto ad un altro computer
- 2- Trasporto: Crea una connessione logica tra mittente e destinatario, indipendentemente dalla rete utilizzata, assemblando e segmentando i dati che riceve dal livello di applicazione e inviando al destinatario un segmento alla volta. Può assegnare anche un numero che permette di verificare la consegna a destinazione
- 3- Internet: Seleziona il miglior percorso per recapitare il messaggio, interconnettendo più reti in modo da renderlo funzionante anche in caso di guasti
- 4- Rete: Funzionamento dipende da host e host, includendo le tecnologie LAN e WAN.

4.2.1.1 Differenza con RMI

A differenza di TCP il Remote Method Invocation rende possibile la comunicazione tra oggetti remoti con l'invocazione di metodi tra gli oggetti.

E' stato utilizzato TCP al posto di RMI in quanto offre maggiori vantaggi. Tre sono quelli che, in particolare, ci hanno portato ad utilizzare TCP:

- 1 – risposta mancante da parte del server se il client inserisce dati errati;
- 2 - il server richiede che in qualsiasi postazione in cui viene installato il client sia presente java;
- 3 - minore efficienza di RMI rispetto ad una tecnologia socket, utilizzato invece in TCP.

4.3 Gestione Progetti Software

4.3.1 Maven

Per la creazione di E-Learning è stato utilizzato Apache Maven, uno strumento di build automation per la gestione di progetti Java.

Il vantaggio dell'uso di questa tool è l'automatizzazione del processo, difatti Maven effettua automaticamente il download di librerie Java e plugin. Questo permette di recuperare in modo uniforme i vari JAR e di poter spostare il progetto da un ambiente ad un altro avendo la sicurezza di avere le stesse versioni delle librerie.

5. Requirements UML Analysis

5.1 Use case diagram

Il diagramma in Fig.1 è stato creato per schematizzare l'elenco di tutti i servizi offerti ed evidenziare i servizi di cui ogni tipo di utente: Student, Lecturer (Teacher) e Admin può usufruire utilizzando la piattaforma.

L'elemento non standardizzato presente nel diagramma (il cerchio) si riferisce ai servizi comuni tra tutti i tipi di utente, ed è stata usata per evitare l'eccesso sfruttamento di frecce, semplificando, dunque, lo schema.

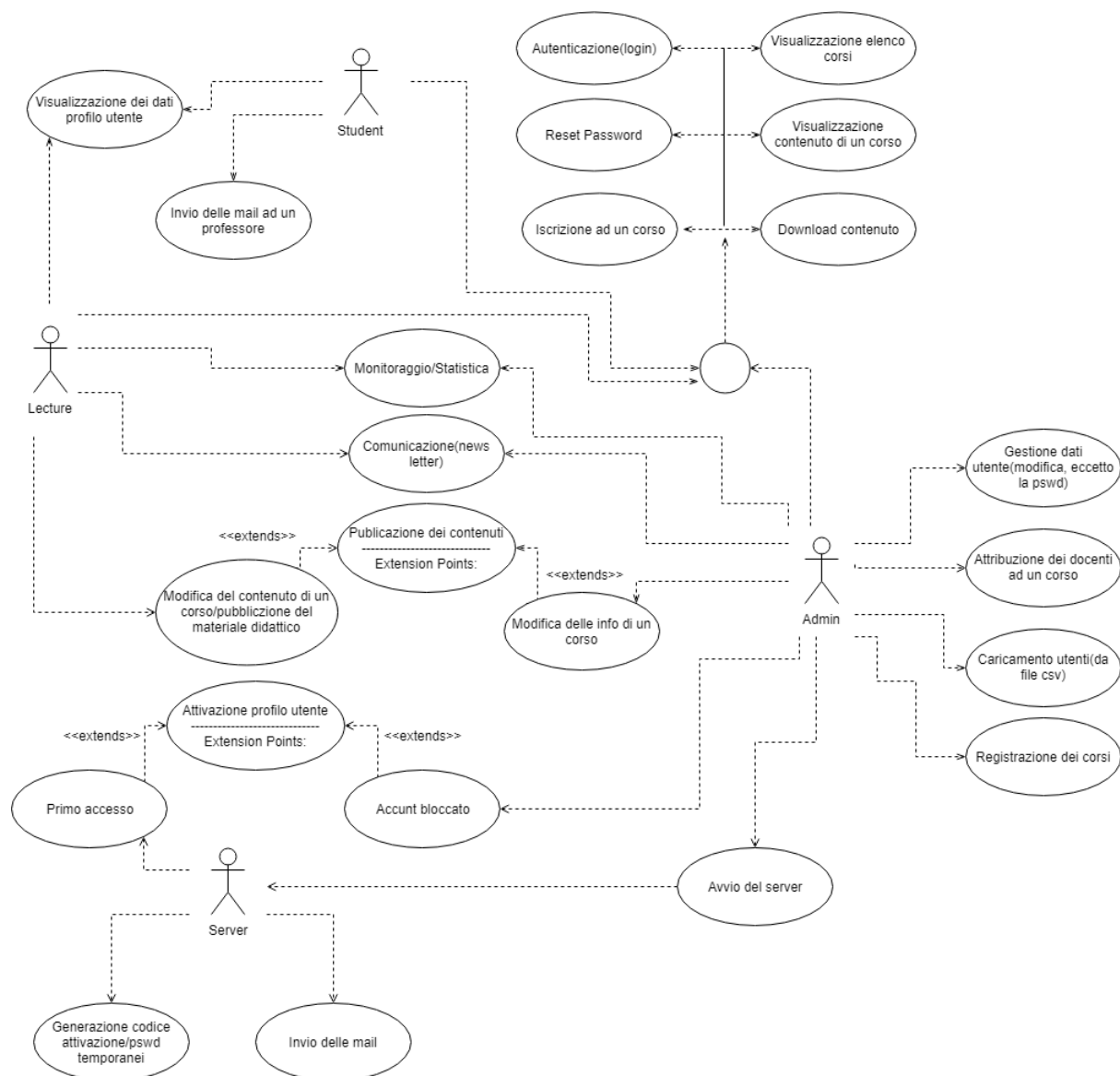


Fig. 1 - Use case diagram

https://drive.google.com/file/d/1U1BHSVgVfJoGCpb3ON08eSJ54_6PoZus/view?usp=sharing

5.2 Package diagram (Struttura server)

Nel diagramma in Fig 2. si vuole evidenziare la struttura generale del server, schematizzandone i package con le loro relazioni.

Nel diagramma è presente solo la relazione tra alcune classi, in quanto necessarie per completare l'idea della struttura.

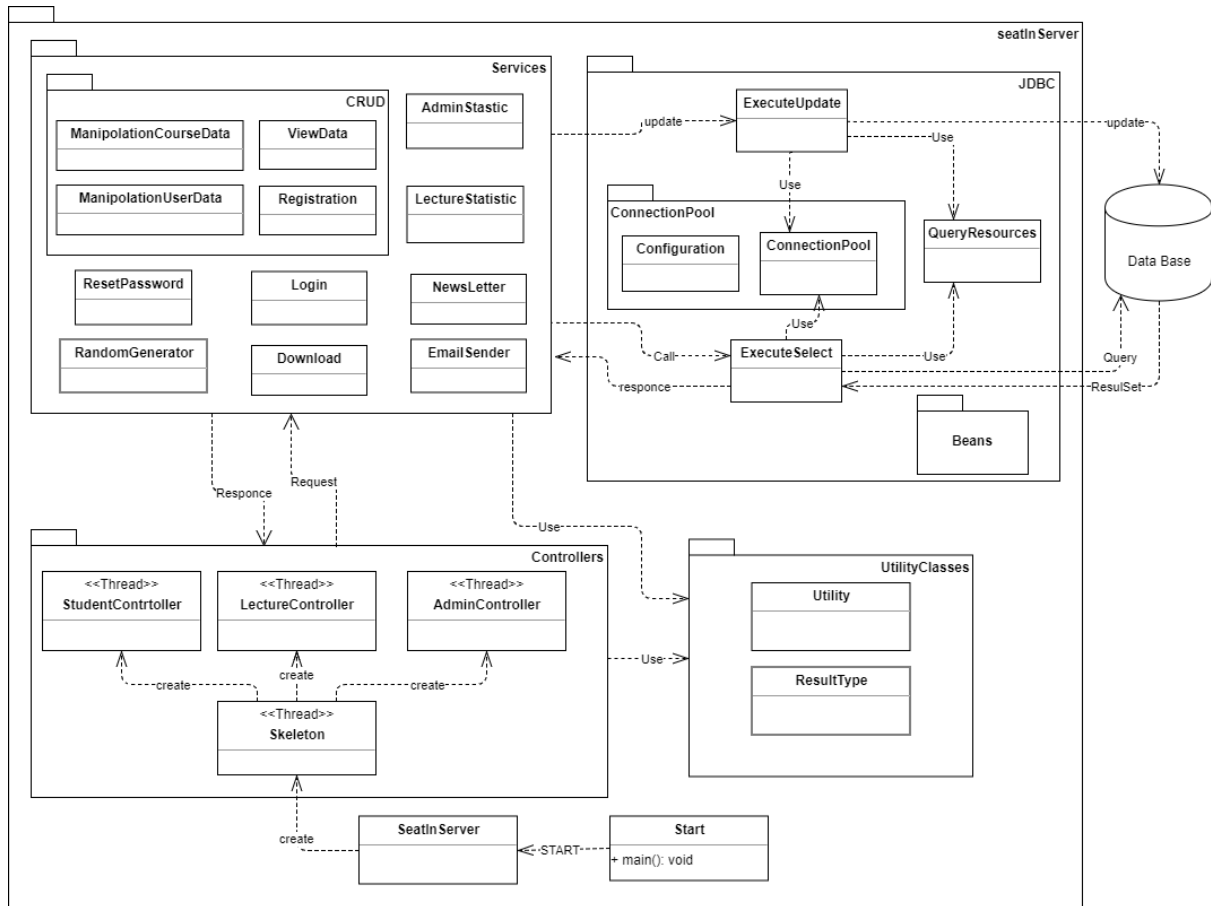


Fig. 2 - Package diagram

<https://drive.google.com/file/d/1dmUFrM8OsiNYRMaL-iuop2NwIY7RTvD/view?usp=sharing>

5.3 seatInServer Class diagram

Nel diagramma in Fig 3. si vuole evidenziare nel modo più dettagliato possibile, la relazione tra le classi.

Nel seguente diagramma viene messa in evidenza la connessione di un qualsiasi utente. Si denota che il server crea il thread dello skeleton, che gestisce i servizi basilari quali login, reset password e attivazione account, che l'utente può usare non essendo nella sessione. Se il login va a buon fine, l'utente riesce quindi ad effettuare l'accesso e, a seconda dei dati inseriti, gli vengono garantiti da un controller i permessi appropriati.

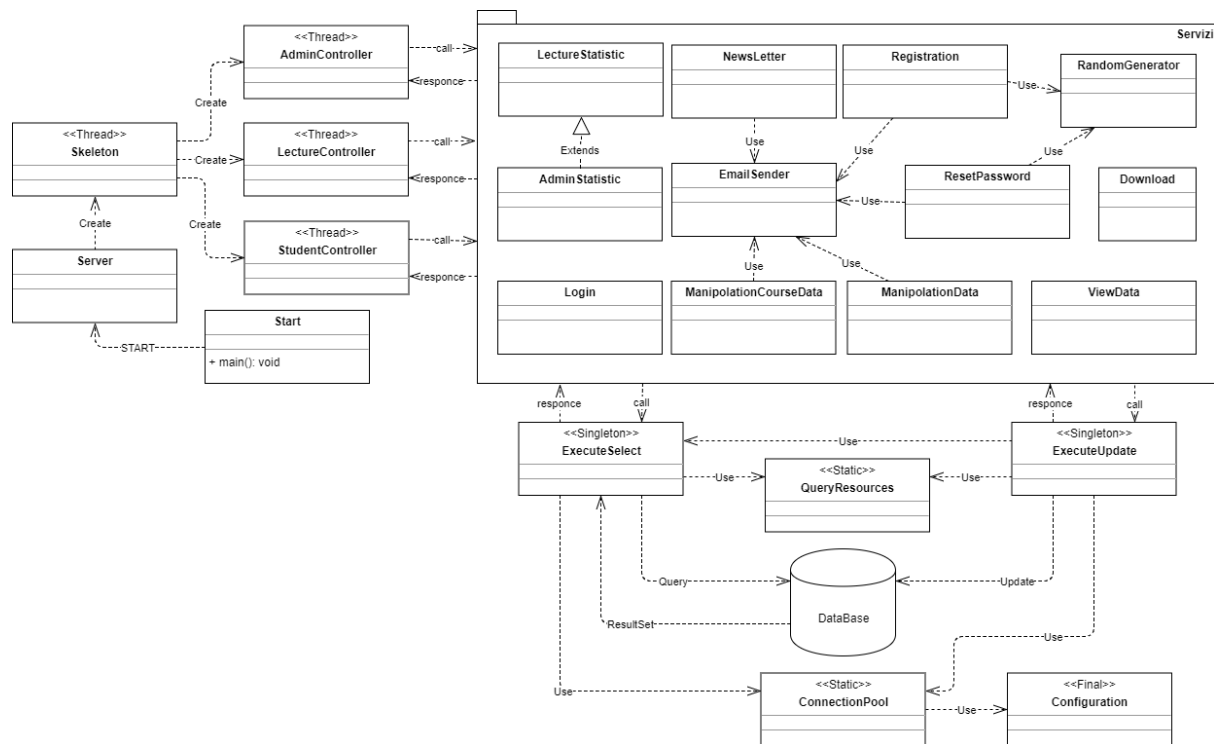


Fig. 3 - Class diagram

<https://drive.google.com/file/d/1G7XMe63IA6L8xsCigp35uGRXUozFkifr/view?usp=sharing>

5.4 seatInAdmin Class diagram

Nel diagramma in Fig. 4 si vuole evidenziare la struttura del client Admin.

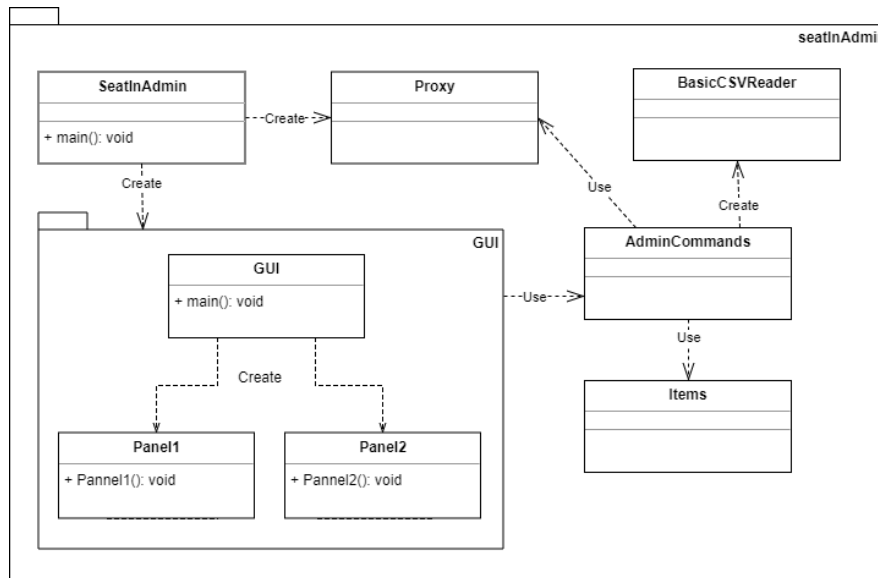


Fig. 4 - Class diagram

<https://drive.google.com/file/d/1UQWGRT02UtML9-IgL2WlszHbYas29JeO/view?usp=sharing>

5.5 seatInUser Class diagram

Nel diagramma in Fig. 5 si vuole evidenziare la struttura del client User, che può essere di tipo studente o il professore.

L'utente di tipo professore è stato considerato come il caso particolare dello studente in quanto utilizza gli stessi servizi, ma in più deve anche gestire i corsi, come la modifica e la pubblicazione dei contenuti ecc.

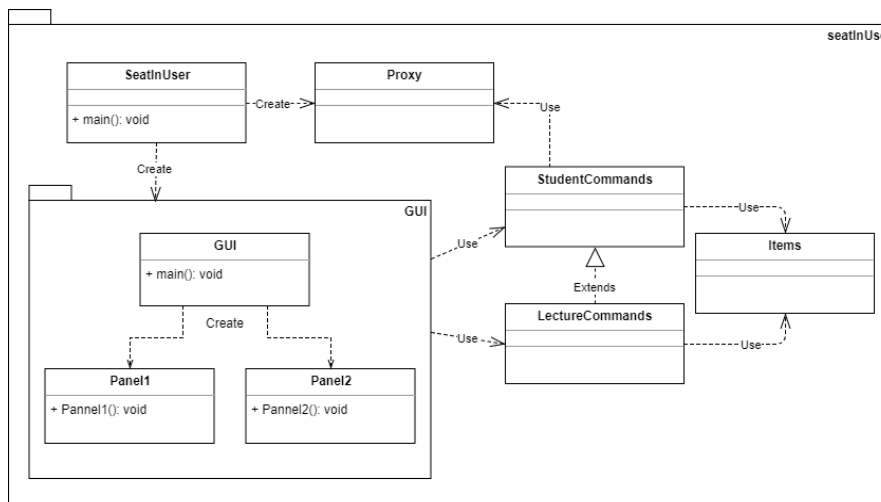


Fig. 5 - Class diagram

<https://drive.google.com/file/d/1qilkbkRBswqS50U5nUsiF4AKcBhbyQph/view?usp=sharing>

5.6 seatInBeans Class diagram

Nel diagramma in Fig. 6 è stato utilizzato il nome Beans in quanto è un nome generale per indicare degli oggetti virtuali che corrispondono ad entità reali, che è esattamente ciò che denota il seguente diagramma.

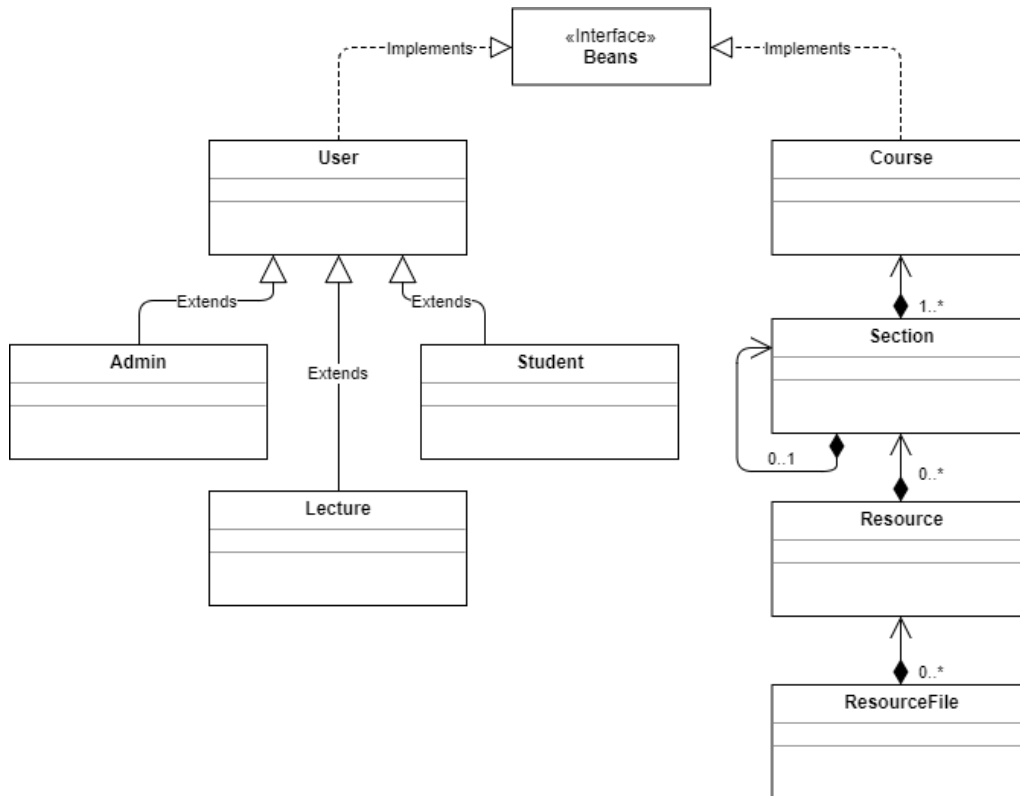


Fig. 6 - Class diagram

<https://drive.google.com/file/d/1akZgaZjBlbTJ84OHqPXXPBFhuxwxrg1V/view?usp=sharing>

6. Organizzazione

Il progetto è stato svolto da un team distribuito in varie località geografiche beneficiando di strumenti orientati al lavoro da remoto e condiviso. I diagrammi sono stati disegnati usando un sito web chiamato draw.io dove i partecipanti potevano vedere, commentare e proporre i cambiamenti ai vari schemi durante le fasi di progettazione.

Questa relazione è stata creata e condivisa usando [Google Doc](https://docs.google.com) un programma gratuito e basato su web di elaborazione e condivisione di testi.

Per collaborare più efficacemente e riuscire a sincronizzare più facilmente le attività del gruppo è stato utilizzato un software chiamato Slack (<https://laboratorio-b.slack.com/>), che ha permesso al gruppo di rimanere in contatto in qualsiasi momento.

L'utilizzo di BitBucket (<https://bitbucket.org/laboratoriob/seatin/src/master/>), una piattaforma sociale per lo sviluppo collaborativo di progetti software, è stata determinante per l'interazione e l'integrazione del codice.

6.1 Divisione del lavoro

Il progetto è stato diviso in tre scaglioni: database, server e client.

Di questi tre gruppi principali ognuno si è occupato di una componente, cercando di rendersi utile nel caso in cui qualcuno avesse bisogno.

Ognuno si è occupato di un lavoro preciso:

Farina Paolo: Si è occupato della realizzazione dell'interfaccia grafica per l'user e ha collaborato nella fase di analisi e progettazione del server.

Strumenti utilizzati:

- Ambiente di sviluppo: Eclipse Neon 3/Oxygen

Muraro Davide: Si è occupato della progettazione ed implementazione del database con query del server e della relativa relazione. Ha collaborato in fase di analisi del server.

Strumenti utilizzati:

- Ambiente di sviluppo: Docker e Kubernetes
- Schemi: draw.io
- Relazione: Google.doc

Roshka Anatoliy: Si è occupato della progettazione e della codifica del server.

Per la codifica server ha utilizzato:

- Sistema di login dedicata al monitoraggio: log4j;
- Per unit testing: framework junit;
- Gestione connessioni e chiamate dei clienti: pattern controller;
- Gestione per la connessione al database: connection pool.

Strumenti utilizzati:

- Schemi: draw.io;
- Mantenimento del codice aggiornato: bitbucket.

Stroppolo Chiara: Si è occupata della realizzazione dell'interfaccia grafica lato client e della stesura della relazione del progetto. Ha collaborato nella fase di analisi e progettazione del server.

Strumenti utilizzati:

- Ambiente di sviluppo: Eclipse Oxygen
- Relazione: Google.doc

7. Informazioni sull'installazione

Il sistema è stato progettato e testato su sistema operativo Windows 10, con PostgreSQL versione 9.6 e versione Java 1.8 (nominata Java 8).

L'applicazione è stata sviluppata utilizzando il linguaggio Java ed è stata implementata con l'ausilio di Eclipse come IDE. In particolare è stato utilizzato Eclipse Oxygen.

7.1 Prerequisiti

- PostgreSQL DBMS installato
- JDK versione 8+
- Maven installato, con PATH impostato correttamente
- Cartella contenente tutti i file necessari all'installazione di seatIn, ottenuta estraendo apposito file zip fornito.

7.2 Installazione database

Attualmente l'installazione del database può avvenire in due modalità:

1. Attraverso un client ad esempio PGAdmin
2. Usando i comandi psql dalla console interattiva

Inoltre sono presenti due files nel pacchetto di installazione: db_creation.sql, contenente la struttura del database, e db_test_data.sql, contenente dati di test (opzionali).

7.2.1 Installazione usando il client PGAdmin

Per questo punto fare riferimento alla guida on-line di [PGAdmin](#).

Le varie fasi sono identificate dai seguenti passaggi:

- Creare un utente e chiamarlo *seatindbuser* con password *seatin.1718*
- Creare il database chiamandolo *seatindb* e assegnare come owner l'utente appena creato
- Eseguire il codice SQL di creazione presente nel file db_creation.sql disponibile al seguente [indirizzo](#).
- Se si vuole installare dei dati di prova eseguire il codice SQL presente nel file db_test_data.sql disponibile al seguente [indirizzo](#).

7.2.2 Installazione usando la console interattiva

- Accedere alla console interattiva

Da linux

```
su - postgres
```

```
$ psql
```

Da Windows

```
c:\path\to\psql.exe -U postgres
```

Una volta collegati alla console eseguire i seguenti comandi:

```
CREATE USER seatindbuser WITH password 'seatin.1718';  
CREATE DATABASE seatindb WITH OWNER seatindbuser;  
GRANT ALL PRIVILEGES ON DATABASE seatindb TO seatindbuser;
```

Per creare la struttura del database eseguire il comando per collegarsi al database appena creato:

```
\c seatindb
```

Caricare il file SQL:

```
-c db_creation.sql
```

Per installare dei dati di test eseguire sempre da console il comando:

```
-c db_test_data.sql
```

Specificare in entrambi il percorso assoluto dei files.

7.3 Installazione applicativo

Per l'installazione dell'applicativo è necessario l'utilizzo del sistema di gestione progetti software chiamato Maven.

Maven è un progetto open source, sviluppato dalla fondazione no-profit Apache Software Foundation, che permette di organizzare in modo molto efficiente un progetto Java, ma può anche essere usato in progetti scritti in C#, Ruby, Scala e altri linguaggi.

È possibile scaricare gratuitamente il progetto dal sito ufficiale; la cui versione attualmente disponibile è la 3.6.0.

Per installare è sufficiente scompattare lo zip, ad esempio nella cartella C:Maven, e aggiungere la variabile di ambiente MAVEN_HOME (MAVEN_HOME=C:Maven).

La guida di installazione è presente sul sito di [Maven](#).

Una volta che il path è correttamente impostato si può procedere nel seguente modo:

- Scompattare il file del progetto con estensione .zip in una cartella a propria scelta;
- Aprire la finestra del terminale ed accedere alla cartella appena creata dove sia presente il file pom.xml, ovvero il Project Object Model;
- Dall'interno della suddetta cartella lanciare i seguenti comandi:
 - mvn clean
 - mvn package
 - mvn javadoc:javadoc
- Al termine dell'esecuzione di questi comandi sono stati generati i file eseguibili di tipi jar relativi a ogni modulo seatIn: seatInServer, seatInUser e seatInAdmin, con la documentazione java relativa ad ogni modulo;
- Entrare nella cartella "seatInServer/target";
- Lanciare il comando: java -jar seatInServer-1.0.jar

In alternativa, l'installazione può essere eseguita tramite ambiente di sviluppo chiamato "Eclipse".

In questo caso i passi da seguire sono i seguenti:

- Da eclipse, importare come “existing Maven Project” la cartella seatIn
- Dopo l’import, effettuare un click destro sul modulo maven nominato seatIn e selezionare Run as->maven clean
- Successivamente, terminata la clean, effettuare nuovamente un click destro sul modulo maven nominato seatIn e selezionare Run as->maven build...
- Nella finestra aperta, nel campo goals digitare i seguenti comandi, separati da uno spazio, e premere Run:
 - – package
 - – javadoc:javadoc

7.4 Avvio applicazione

Per avviare il server (modulo seatInServer):

- Da terminale entrare nella cartella seatIn/seatInServer/target
- Avviare il file jar tramite il comando: `java -jar seatInServer-1.0.jar`

Per avviare il client (modulo seatInUser):

- Da terminale entrare nella cartella seatIn/seatInUser/target
- Avviare il file jar tramite il comando: `java -jar seatInUser-1.0.jar`

Per avviare il client (modulo seatInAdmin):

- Da terminale entrare nella cartella seatIn/seatInAdmin/target
- Avviare il file jar tramite il comando: `java -jar seatInAdmin-1.0.jar`

7.5 Documentazione Java

La documentazione java relativa a ciascun modulo, una volta generata, si trova nella cartella seatIn/”nomeModulo”/target/site/apidocs.

Per visualizzare la documentazione, aprire il file index.html