



*Distributed Systems and Middleware Technologies*



RepeTutor

Marta Sanguinetti

Alessio Vivani

## Table of contents

---

<b>1 – Introduction .....</b>	<b>3</b>
<b>2 – Actors and requirements .....</b>	<b>4</b>
Main actors .....	4
Functional requirements.....	4
Non-functional requirements .....	4
<b>3 – Architecture .....</b>	<b>4</b>
Application server .....	5
Erlang server .....	6
MySQL database .....	6
<b>4 – Communication and synchronization .....</b>	<b>7</b>
<b>5 – User interface.....</b>	<b>8</b>

## 1 – Introduction

RepeTutor is an application where students can find tutors who can help them in different subjects. At the same time, it provides tutors with a platform where they can be reached.

RepeTutor allows people to look for tutors filtering them by subject and location.

Students can login and browse tutors, and once they have found someone they want to have lesson with, they can send a message to the selected tutor to find an agreement on topics, date, time and location of the lesson. When an agreement has been found, the tutor can add it to the calendar so that it is visible to both tutor and student and can be used as a reminder.

Chat can be further used to exchange messages about homework, following lessons, etc.

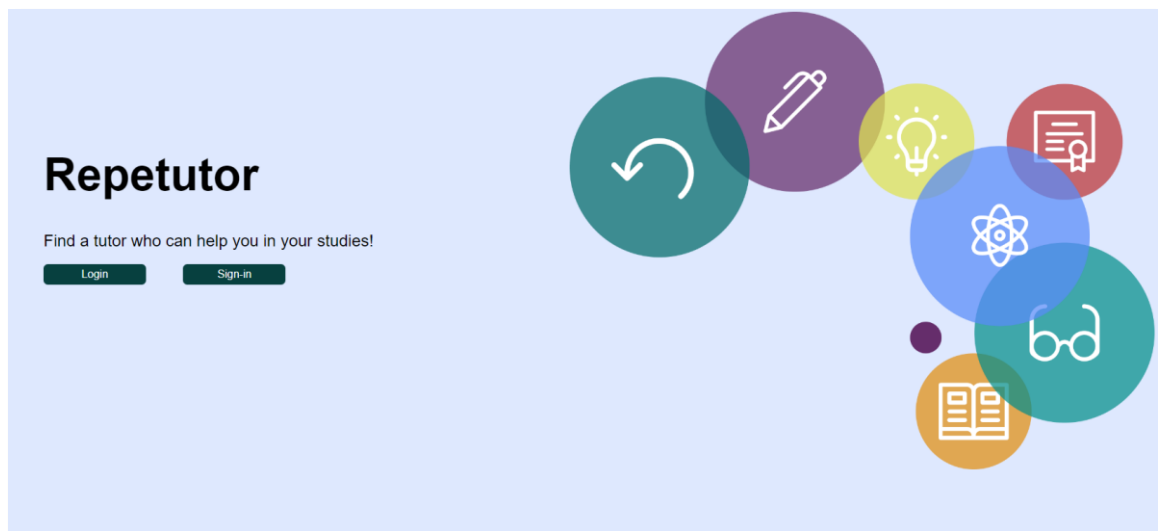


Figure 1: RepeTutor homepage

## 2 – Actors and requirements

---

### Main actors

Main actors of the application are:

- Unregistered User: a user that still hasn't got a profile on the application.
- Registered User: it can either be a student or a tutor and has a profile on the application.
- Admin: the admin of the application.

### Functional requirements

**Unregistered users** can:

- Sign-in as a tutor or as a student;

**Registered users** can:

- Login as a tutor or as a student;

In particular:

A **Tutor** can:

- Edit his/her profile;
- Add/remove a lesson from calendar;
- View the list of past and future lessons;
- Chat with a student;

A **Student** can:

- Browse the list of tutors, filtering them by subject and/or by city;
- Chat with a tutor;
- View the list of past and future lessons;
- Add a comment to a tutor he/she had lesson with.

### Non-functional requirements

The main functional requirements of our web-application are:

- Responsiveness of the application, to guarantee the service with low latency;
- No messages should be lost, neither the ones that are sent to offline users;
- Reachability of the service from any web browser;
- Simplicity of the interface so that the application is easy to use;

## 3 – Architecture

---

RepeTutor application is organized as follows:

- Application server, that has been built with Spring Boot;
- Erlang server, that handles the chat service and communicates with a Mnesia database;
- MySQL database;

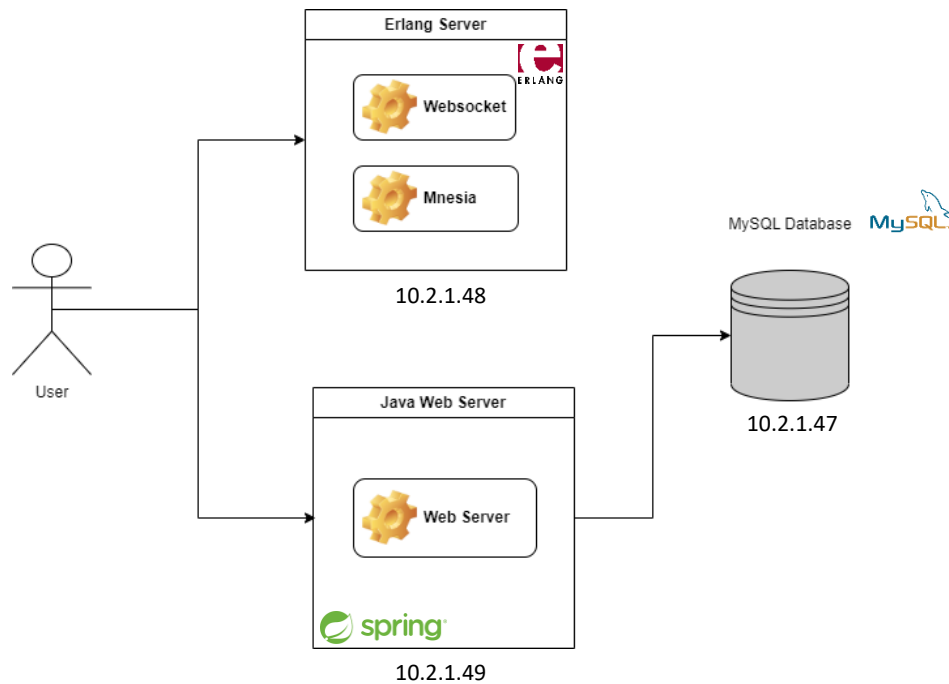


Figure 2: application architecture

Each of them has been deployed on a different remote machine.

### Application server

Application server handles the main functionalities of the application. It is based on Spring Boot, and it is organized as follows:

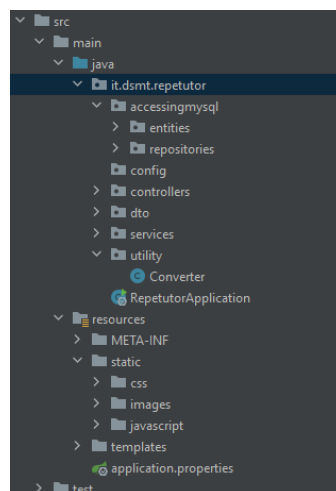


Figure 3: project organization

It follows a **Controller-Service-Repository** pattern, which provides a separation of concerns: the Controller layer is solely responsible for exposing the functionality so that it can be consumed by external entities (including, perhaps, a UI component). The Repository layer is responsible for storing and retrieving some set of data. The Service layer is where all the business logic goes. If the business logic requires fetching/saving data, it wires in a Repository. If someone wants to access this business logic, they go through a Controller to get there.

Moreover, we handled the communication with MySQL database making use of @Entity annotation. JPA **Entities** represent tables stored in the database, and every instance of an entity represents a row in the table.

In the end, we translated entities into **DTOs**, just to keep a logical separation between the two classes since they are used in different ways. Entities are used in the communication with the database, while DTOs are mainly used by controllers.

### Erlang server

The Erlang server handles the whole chatting service. Whenever a generic user goes into the "chat" section it starts communicating with the remote Erlang server.

The communication is handled via cowboy and JavaScript's WebSockets. Once a user opens that web page, he gets the list of users he had a chat with and can then start sending and receiving messages in real-time.

Every time a message is sent it's also stored inside the Mnesia database, which also contains a mapping that associates a user id to an erlang pid. This mapping gets added when a user connects to a websocket and gets removed on exit. This allows to directly send messages to online users, simply by sending such messages to the erlang process whose pid is mapped with that user.

### MySQL database

MySQL database keeps the information related to the users and their lessons.

We have these tables:

Table	Columns
<b>Lesson</b>	<u>Id</u> , Date, Duration, Start, Student, Subject, Tutor
<b>Comment</b>	<u>Id</u> , Comment, Evaluation, Id_tutor, Student_name, Subject
<b>Tutor</b>	<u>Id</u> , Currency, Description, Email, Location, Name, Photo, Price, Psw_hash, Salt, Subjects, Surname, Username
<b>Student</b>	<u>Id</u> , Email, Name, Photo, Psw_hash, Salt, Surname, Username

## 4 – Communication and synchronization

---

By default, Spring Boot controllers are singleton, but this isn't an issue since we exploited HTTP session, whenever we needed to store information about the user and its activity.

In order to not have concurrency problems, it is important that Controllers and in general Singleton beans are stateless since eventual class parameters would be shared between all the threads.

Moreover, in Spring every request is executed in a separate thread. For this reason, Spring can handle correctly concurrent requests and there are no concurrency problems. Indeed, it's possible for Spring to use the same Controller instance in multiple threads, firstly because for each thread, Java creates a private stack memory. The stack memory is responsible for storing the states of the local variables used inside methods during thread execution. This way, Java makes sure that threads executing in parallel do not overwrite each other's variables. Secondly, because controllers set no restrictions or locks at the heap level, the program counter of each thread can point to the same reference of the bean instance in the heap memory. Therefore, both threads can execute the same method simultaneously.

For what concerns synchronization problems, we must highlight that a student could see outdated information about tutors in case he browses the list of tutors and meanwhile a tutor modifies his/her profile or another student adds a comment. We decided to accept this behavior since not seeing updated information doesn't causes any big problem for the student. In the worst case, he will see a slightly different rating or the old version of tutor's profile. Moreover, the student will directly chat with the tutor before booking a lesson so any misunderstanding could be solved that way.

In the end, Erlang server didn't cause any concurrency problems since it is able to handle several threads of execution at the same time. A WebSocket connection is opened for each client who connects, and a new process is spawned. As a final consideration, in case we needed to reduce the load over a single erlang server, we could create a cluster of erlang nodes and balance the load between them.

## 5 – User interface

### Login

### Login

Username:

Password:

Role:

### Sign in

### Sign-in

Sign-in as:

Username:

Password:

Name:

Surname:

Email:

### Sign-in

Sign-in as:

Username:

Password:

Name:

Surname:

Email:

Subjects (separated by comma):

Price per hour:

Currency:

Location:

Description:

0055


### Search tutor


[Search a Tutor](#) [Schedule of lessons](#) [Modify profile](#) [Chat](#) [Logout](#)

### Search your tutor

City:  Subject:

**Valentina Neri** ★ 4.5/5 (4)


 Greco, Italiano, Latino, Storia


 Roma

Sono una docente di 37 anni. Forte di anni di esperienza certificata, svolta anche in collaborazione con un'associazione socio-culturale e un istituto a Roma, insegno con passione e professionalità a studenti di diverso ordine e grado.

Price: 16 €/hour

**Giulia Verdi** ★ 4.0/5 (1)


 Geografia, Inglese, Matematica, Algebra


 Milano

Sono un'interprete e traduttrice di inglese dal 2004. Le mie lezioni sono molto pratiche volte a superare la barriera del soggetto nello speaking. Si ascoltano interviste, si guardano parti di film e si parla.

Price: 20 €/hour

**Eugenio L.**

 Informatica, Elettronica, Programmazione, Matematica

 Milano

Ciao, mi chiamo Eugenio e con due lauree in Ingegneria Informatica ed un dottorato a tema Intelligenza Artificiale appena intrapreso presso il Politecnico di Milano.


Price: 15 €/hour





## Tutor's profile

[Search a Tutor](#) [Schedule of lessons](#) [Modify profile](#) [Chat](#) [Logout](#)

### Valentina Neri

 4.5/5 (4)

 Greco, Italiano, Latino, Storia

 Roma






Sono una docente di 37 anni. Forte di anni di esperienza certificata, svolta anche in collaborazione con un'associazione socio-culturale e un istituto a Roma, insegno con passione e professionalità a studenti di diverso ordine e grado.


Price: 16 €/hour

[Chat](#)

### Comments





Marta




 Italiano

Ottima insegnante. Paziente e coinvolgente. Molto preparata e disponibile

Caterina



 Greco


Mi sono trovata molto bene


## Schedule of lessons


[Search a Tutor](#) [Schedule of lessons](#) [Modify profile](#) [Chat](#) [Logout](#)


### Future lessons

Valentina Neri

 When: 2023-06-10 at 14:00


 Duration: 1


 Subject: Greco


 Email address: valentina@gmail.com


### Past lessons

Valentina Neri

 When: 2023-04-19 at 14:00

 Duration: 2

 Subject: Italiano

 Email address: valentina@gmail.com

[Add comment](#)

## Modify profile

[Search a Tutor](#) [Schedule of lessons](#) [Modify profile](#) [Chat](#) [Logout](#)

### Modify profile

Username:

New password:

Name:

Surname:

Email:

[Modify](#)

# Chat

[Search a Tutor](#) [Schedule of lessons](#) [Modify profile](#) [Chat](#) [Logout](#)

Valentina Neri

Chiara P.

Valentina Neri

Ma avrei bisogno questa estate in vista degli esami di settembre

15/05/2023 10:22

Allora potremmo fare una prima lezione per conoscerci il 10/06/2023 alle 14.00 su teams

15/05/2023 10:23

Certo, va benissimo

Perfetto, aggiungo la lezione al calendario

