



# Allegato Tecnico

Gruppo SWEight - Progetto Colletta

SWEightGroup@gmail.com

## Informazioni sul documento

<b>Versione</b>	1.0.0
<b>Approvatore</b>	Damien Ciagola
<b>Redattori</b>	Enrico Muraro Damien Ciagola Francesco Magarotto
<b>Verificatori</b>	Sebastiano Caccarto Alberto Bacco
<b>Uso</b>	Esterno
<b>Distribuzione</b>	Prof. Vardanega Tullio Prof. Cardin Riccardo Gruppo SWEight

## Descrizione

Documento che ha lo scopo di illustrare la Product Baseline, ponendo attenzione alle scelte architettureali e alla copertura di use case e requisiti funzionali

---

## Registro delle modifiche

Versione	Data	Descrizione	Autore	Ruolo
0.1.0	2019-03-31	Stesura sezione §1 e §2	Damien Ciagola	<i>Redattore</i>
0.0.1	2019-03-30	Creazione scheletro del documento	Damien Ciagola	<i>Redattore</i>

## Indice

<b>1</b>	<b>Introduzione</b>	<b>5</b>
1.1	Scopo del documento	5
1.2	Scopo del prodotto	5
1.3	Glossario	5
<b>2</b>	<b>Installazione ed esecuzione</b>	<b>6</b>
2.1	Maven project	6
2.2	Node.js	6
<b>3</b>	<b>Architettura del prodotto</b>	<b>7</b>
3.1	Design pattern utilizzati	7
3.2	MongoDB Database	7
<b>4</b>	<b>Diagrammi dei package</b>	<b>8</b>
4.1	Model	8
4.2	Controller e service	8
4.3	Service e repository	9
<b>5</b>	<b>Diagrammi delle classi e di sequenza</b>	<b>11</b>
5.1	Inserimento di un utente	11
5.2	Login	13

## Elenco delle figure

1	Model . . . . .	8
2	Controller e Service . . . . .	9
3	Service e Repository . . . . .	10
4	Exercise insert . . . . .	11
5	UML - FrontEnd . . . . .	12
6	Authorization . . . . .	13

## Elenco delle tabelle

## 1 Introduzione

### 1.1 Scopo del documento

Il presente documento ha lo scopo di fornire agli sviluppatori uno specchietto informativo sul design strutturale e logico della piattaforma Colletta. Il documento sarà inoltre corredato da diagrammi UML 2.X delle principali scelte prese dal gruppo SWEight e descriverà le tecnologie utilizzate nella realizzazione dell'applicazione.

### 1.2 Scopo del prodotto

Il prodotto da realizzare consta in un'applicazione web che fornisca uno strumento per creare e svolgere esercizi di analisi grammaticale, e al contempo né raccolga i risultati. I dati raccolti verranno impiegati dagli sviluppatori dell'azienda proponente come strumento per il miglioramento di algoritmi di apprendimento automatico<sub>G</sub>. Nello specifico il prodotto verrà utilizzato da tre tipologie di utenti: le/gli insegnanti che si occuperanno della creazione degli esercizi, gli allievi che potranno svolgere gli esercizi e ottenere delle valutazioni e gli sviluppatori che filtreranno i dati secondo alcuni criteri, e infine li scaricheranno.

Il prodotto si interfacerà con un'applicazione di PoS-tagging<sub>G</sub>, come FreeLing<sub>G</sub>, a cui verrà delegata l'esecuzione dell'analisi grammaticale delle frasi.

### 1.3 Glossario

Al fine di rendere il documento il più comprensibile possibile e permetterne una rapida fruizione, viene allegato il *Glossario\_v3.0.0* in cui sono presenti i termini contraddistinti dal pedice G. Tali termini includono abbreviazioni, acronimi, termini di natura tecnica, oppure sono fonte di ambiguità e pertanto necessitano di una definizione che renda il loro significato inequivocabile. Ogni termine, solo alla prima occorrenza per documento, verrà contrassegnato con la dicitura sopra indicata e rimanderà alla medesima definizione nel *Glossario\_v3.0.0*.

## 2 Installazione ed esecuzione

Il codice relativo alla Product Baseline lo si può trovare al seguente link:

[linkAllaRepo](#).

### 2.1 Maven project

Una volta fatto il clone della repository o dopo aver scaricato lo zip, posizionarsi nella directory "Mockup-V2/Backend" della repo e utilizzare i seguenti comandi:

```
mvn clean install  
java -jar target/colletta-2019-0.0.1-SNAPSHOT.jar
```

### 2.2 Node.js

Per lo sviluppo lato server in linguaggio Javascript ci si avvale dell'ultima versione Long Term Support (LTS) di Node.js, che, al momento della stesura di questo documento, è la 10.15.1 LTS. Node.js è reperibile al seguente link:

[Node.js](#).

Il file package.json contiene tutte le configurazioni e dipendenze del progetto. Per installare tutti i moduli necessari è necessario eseguire il seguente comando nella cartella contenente il file package.json:

```
npm install
```

Per eseguire il progetto è necessario usare il comando:

```
npm start
```

## 3 Architettura del prodotto

### 3.1 Design pattern utilizzati

### 3.2 MongoDB Database



## 4 Diagrammi dei package

### 4.1 Model

Viene di seguito riportato il diagramma delle classi del package model.

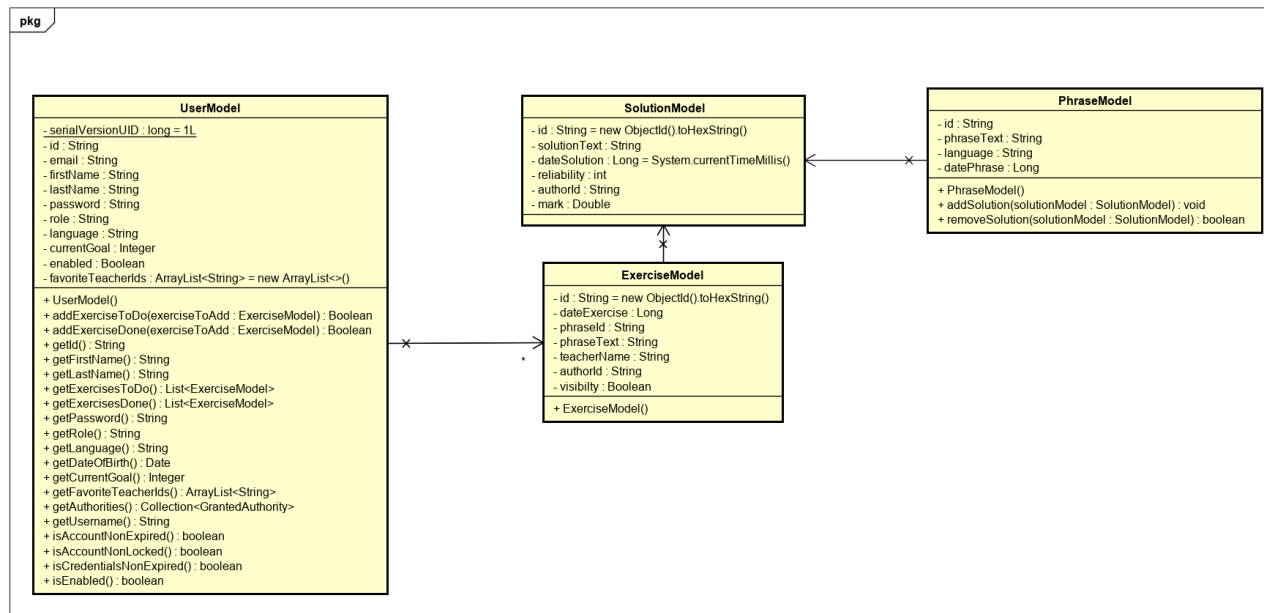


Figura 1: Model

## 4.2 Controller e service

Viene di seguito riportato il diagramma delle classi di package controller e service.

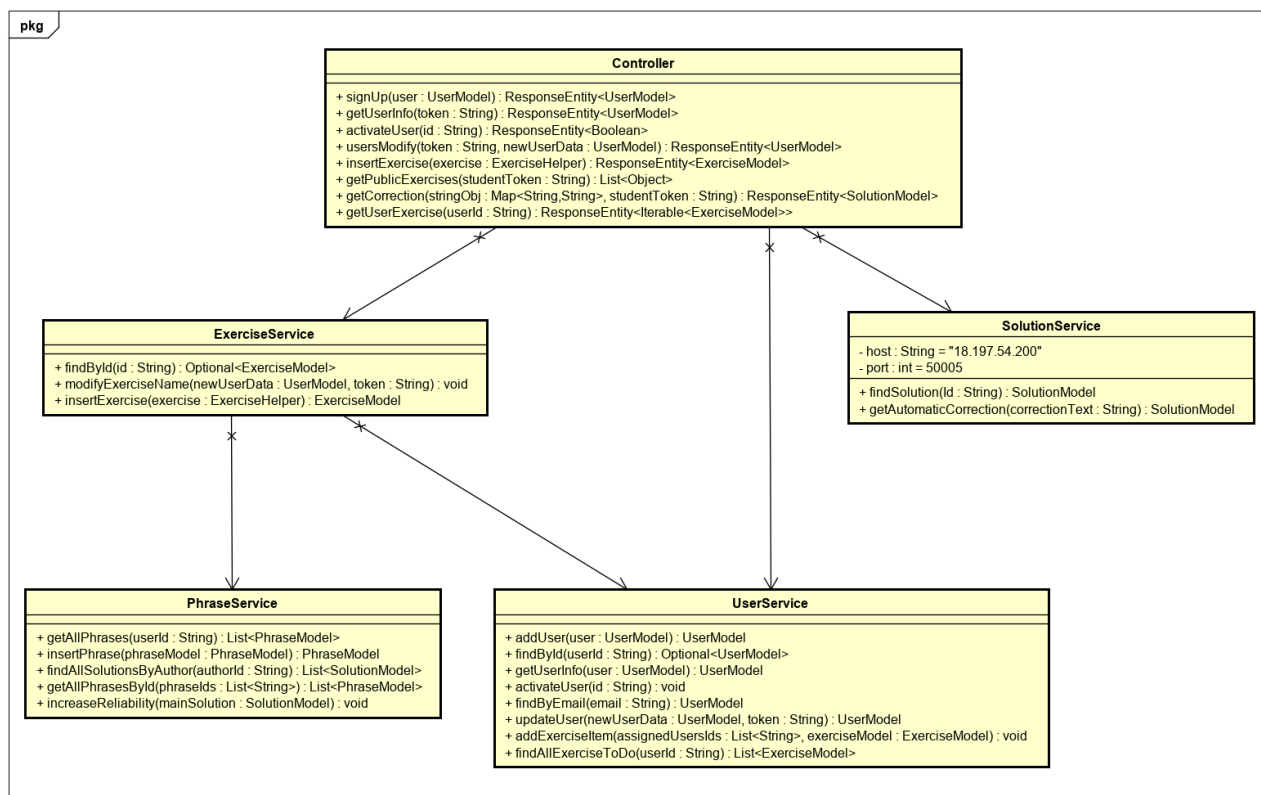


Figura 2: Controller e Service

Viene di seguito riportato il diagramma delle classi di package service e repository.

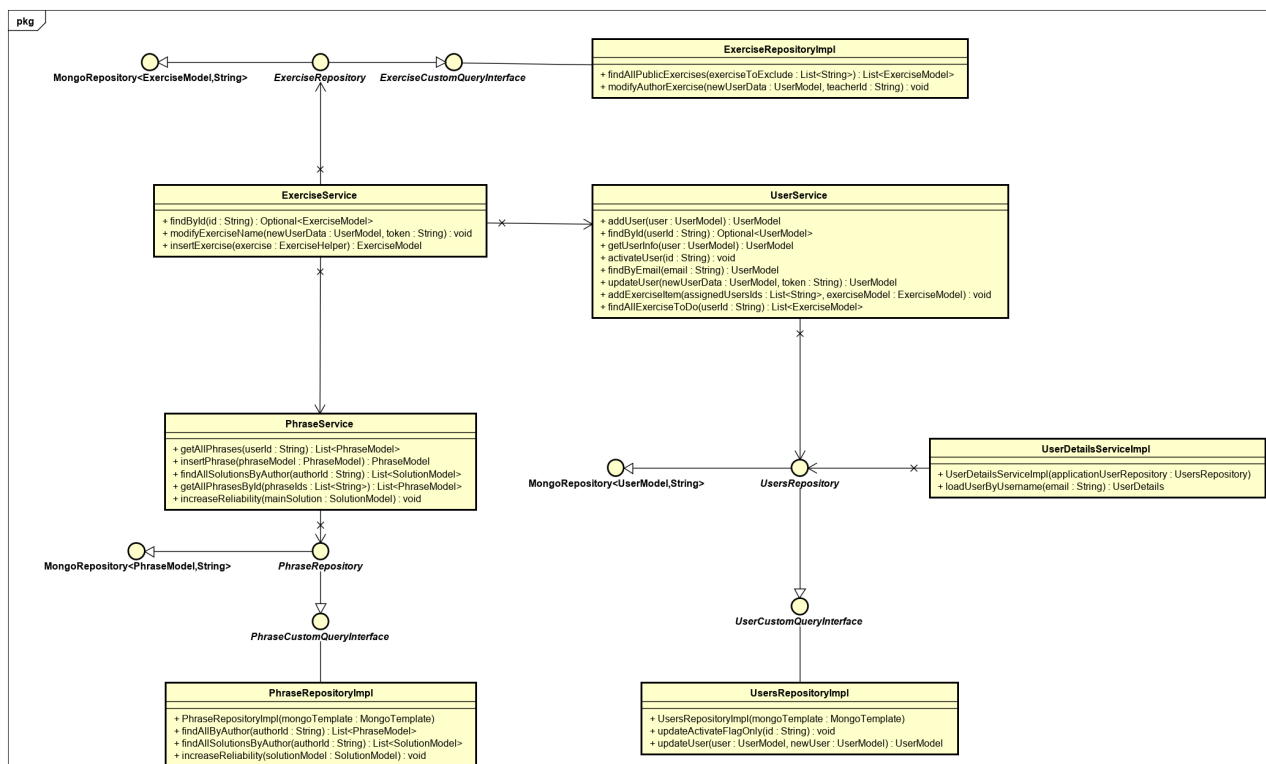


Figura 3: Service e Repository

## 5 Diagrammi delle classi e di sequenza

### 5.1 Inserimento di un utente

Il diagramma di sequenza rappresenta l'azione di inserimento di un esercizio nel sistema

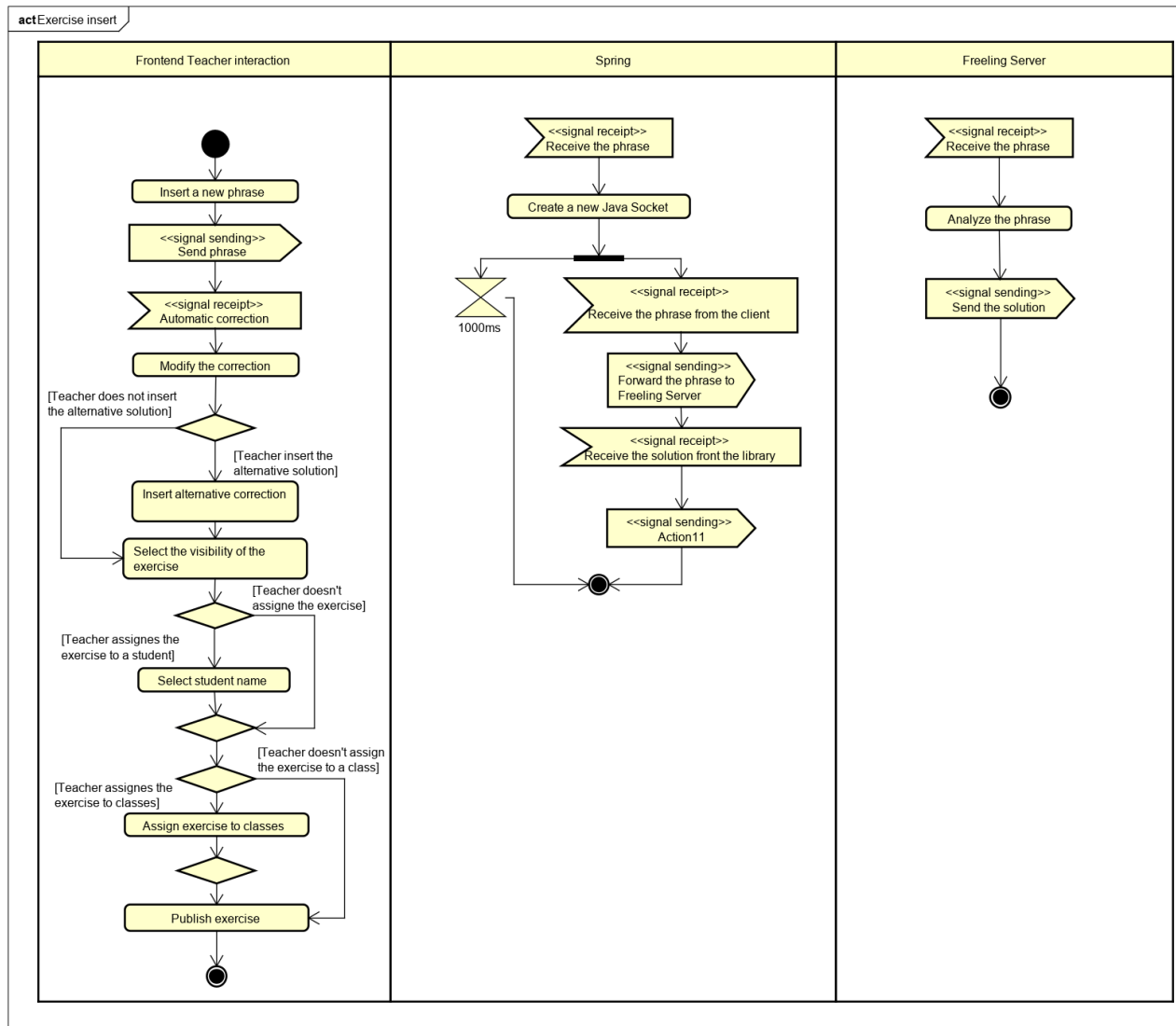


Figura 4: Exercise insert

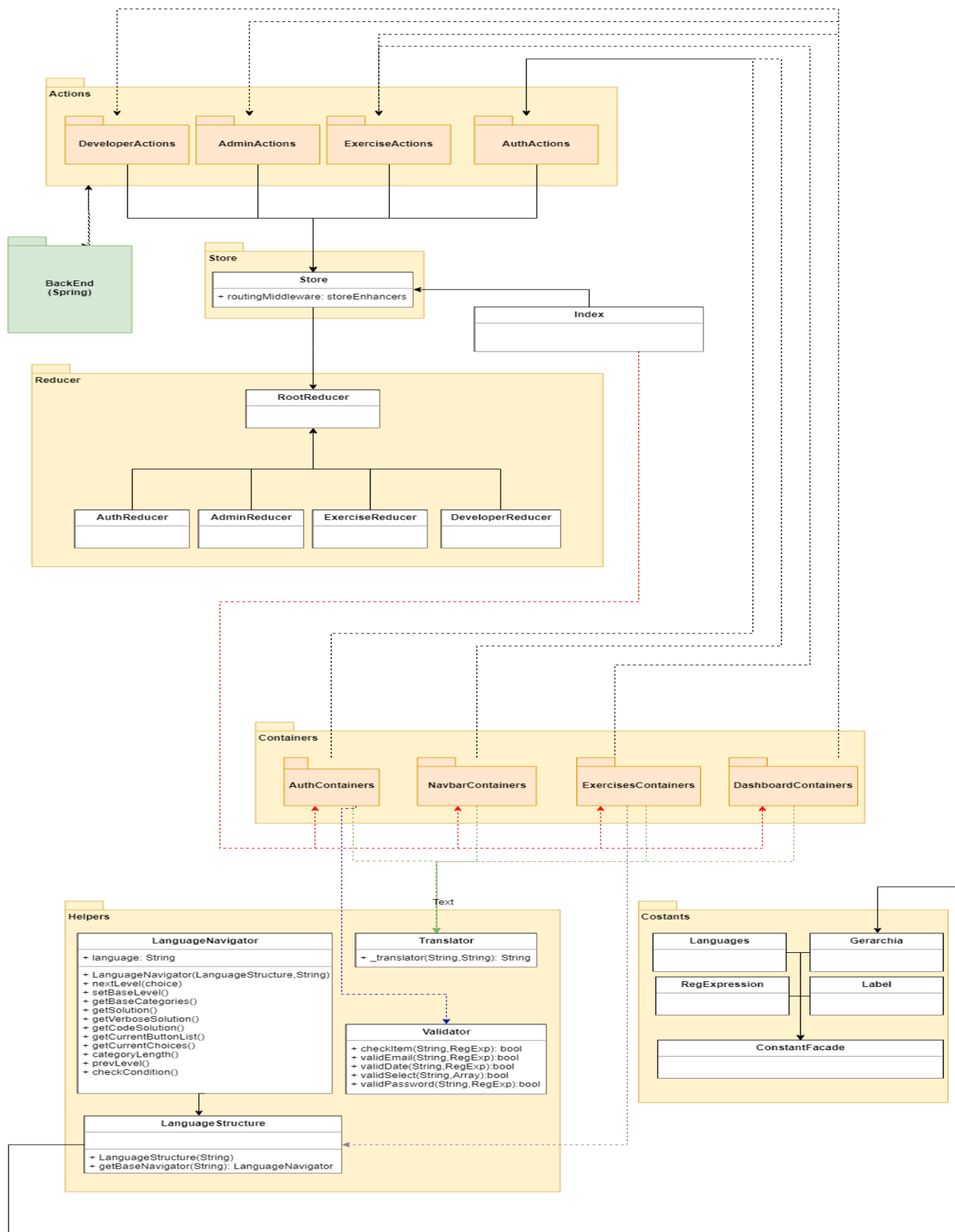


Figura 5: UML - FrontEnd

## 5.2 Login

Il diagramma di sequenza riportato qui di seguito raffigura il processo di login, durante il quale l'utente che vuole accedere può essere autenticato dal sistema.

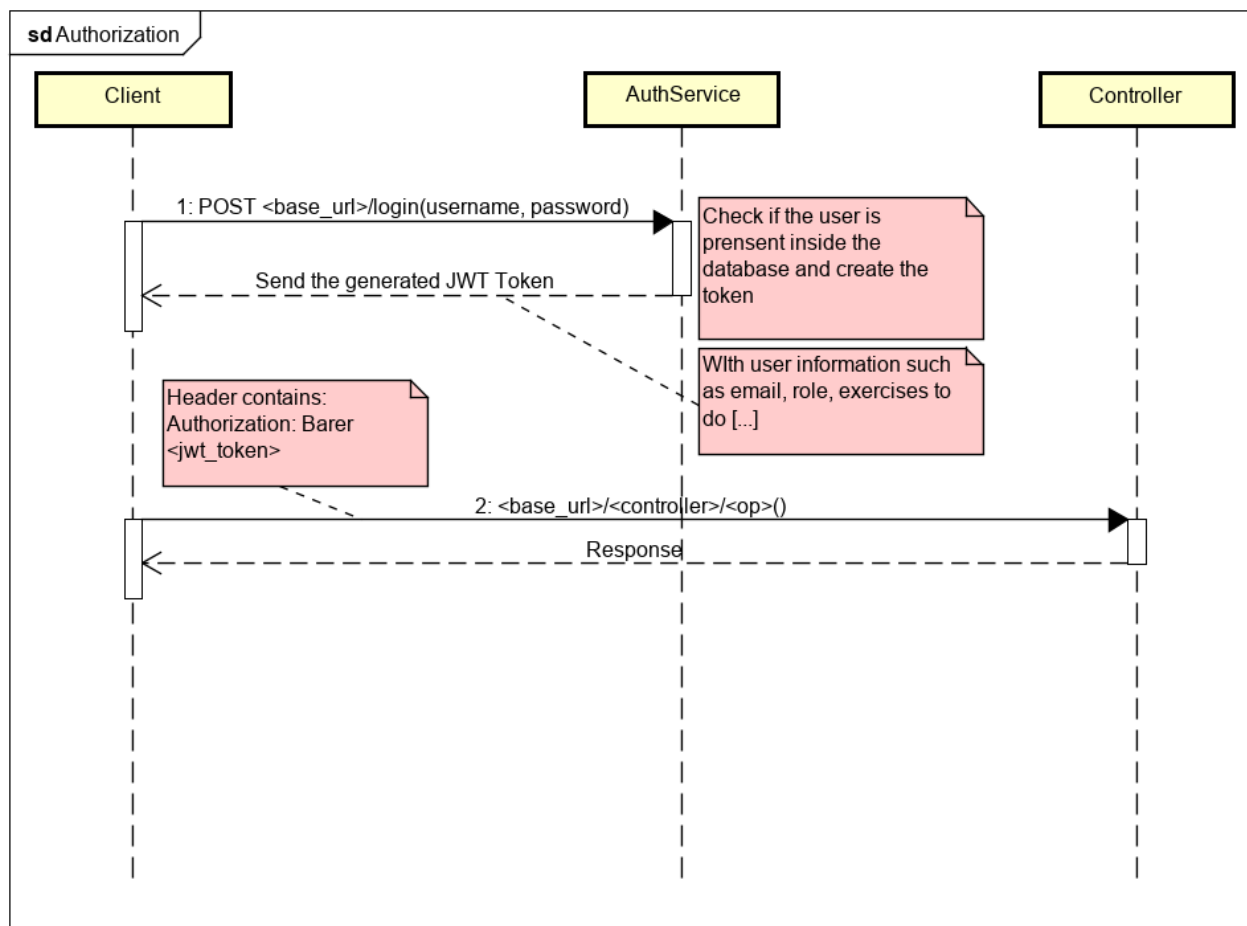


Figura 6: Authorization