

# Web Services Project

*Completed by :*

**Heni WALHA  
Anis BOUHAMED  
Maissa BOUZID**

---

---

*Subject :*

**\* Design and Implementation of Eiffel Tutoring Solutions \***

---

---

*Academic Year :*

**2023 - 2024**



---

## ACKNOWLEDGMENTS :

We express our heartfelt gratitude to Dr. Zargayouna Mahdi, whose guidance and expertise have been instrumental in shaping the vision and execution of Eiffel Tutoring Solutions. Dr. Mahdi's unwavering support and insightful feedback have been a beacon throughout this short journey.

We extend our sincere appreciation to the entire team involved in the development of this project. Each team member has contributed their unique skills and dedication, creating a collaborative environment that fostered innovation and excellence. This project stands as a testament to the collective effort and commitment of the team. Thank you all for your invaluable contributions and unwavering support.



---

# TABLE OF CONTENTS

<b>LIST OF FIGURES</b>	<b>iv</b>
<b>LIST OF ACRONYMS</b>	<b>1</b>
<b>1 INTRODUCTION</b>	<b>2</b>
<b>2 SYSTEM DESIGN</b>	<b>4</b>
2.1 UML class diagram : . . . . .	4
<b>3 USED TECHNOLOGIES</b>	<b>5</b>
3.1 Remote Method Invocation (RMI) : . . . . .	5
3.2 Web Services (Tomcat and Axis) : . . . . .	5
3.3 JavaFX : . . . . .	5
3.4 Deep Learning Model : . . . . .	6
<b>4 IMPLEMENTATION</b>	<b>7</b>
4.1 Application Server : . . . . .	7
4.1.1 Server Components : . . . . .	7
4.1.2 Workflow : . . . . .	7
4.2 Java Application for Students : . . . . .	8
4.2.1 Components : . . . . .	8
4.2.2 Intelligent System : . . . . .	8
4.2.3 Workflow : . . . . .	8
4.3 External Student (Dynamic Web Project) : . . . . .	9
4.3.1 Components : . . . . .	9
4.3.2 Workflow : . . . . .	9
4.4 Gustave Tutor Service (Dynamic Web Project) : . . . . .	9
4.4.1 Workflow : . . . . .	9
4.5 Bank Server (Dynamic Web Project) : . . . . .	9
4.5.1 Components : . . . . .	9

## TABLE OF CONTENTS

---

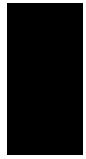
4.5.2	Workflow : . . . . .	10
4.6	BankService (Dynamic Web Project) : . . . . .	10
4.6.1	Workflow : . . . . .	10
4.7	Conclusion . . . . .	10
<b>5</b>	<b>CHALLENGES FACED</b>	<b>11</b>
5.1	Whole Project Architecture : . . . . .	11
5.1.1	Description : . . . . .	11
5.1.2	Solution : . . . . .	11
5.2	Converting Student Feedback to Ratings . . . . .	11
5.2.1	Description : . . . . .	11
5.2.2	Solution : . . . . .	12
5.3	Notifying a Student on the Waiting List : . . . . .	12
5.3.1	Description : . . . . .	12
5.3.2	Solution : . . . . .	12
<b>6</b>	<b>ADDITIONAL FUNCTIONALITIES :</b>	<b>13</b>
6.1	JavaFX Interfaces for Enhanced User Interaction : . . . . .	13
6.1.1	Description : . . . . .	13
6.1.2	Benefits : . . . . .	13
6.2	Mailing Service for Waiting List Notifications . . . . .	19
6.2.1	Description : . . . . .	19
6.2.2	Benefits : . . . . .	19
6.3	Fine-Tuned BERT Language Model . . . . .	19
6.3.1	Description : . . . . .	19
6.3.2	Benefits : . . . . .	20
6.4	Conclusion . . . . .	20
	<b>CONCLUSION</b>	<b>21</b>



---

# LIST OF FIGURES

2.1	class diagram . . . . .	4
6.1	Student sign up . . . . .	14
6.2	Student sign in . . . . .	14
6.3	Tutor sign up . . . . .	15
6.4	Tutor sign in . . . . .	15
6.5	Book appointment . . . . .	16
6.6	Tutor details . . . . .	16
6.7	Student feedback . . . . .	17
6.8	Tutor interface . . . . .	17
6.9	Student interface . . . . .	18
6.10	send mail . . . . .	19



---

# LIST OF ACRONYMS :

- RMI : Remote Method Invocation
- JVM : Java Virtual Machine
- UI : User Interface
- NLP : Natural Language Processing
- SOAP : Simple Object Access Protocol
- BERT : Bidirectional Encoder Representations from Transformers
- API : Application Programming Interface
- REST : Representational State Transfer
- IMDb : Internet Movie Database

---

# INTRODUCTION

Eiffel Tutoring Solutions emerges as a response to the escalating need for a comprehensive tutoring platform within the academic realm of Gustave Eiffel University. At its core, the objective is to seamlessly connect students in search of academic support with proficient tutors, facilitated through a distributed Java application leveraging Java RMI middleware. Crucially, the application operates distinct Java Virtual Machines (JVMs) for both tutors and students, ensuring an optimized and efficient user experience.

Within this dynamic ecosystem, tutors are empowered to register on the platform, showcase their areas of expertise, and establish their availability and rates. Simultaneously, students gain the ability to explore tutor profiles, conduct targeted searches, and seamlessly schedule appointments based on their academic needs. A unique feature of the platform is the implementation of a waiting list, which intelligently notifies students when high-demand tutors become available, enhancing accessibility.

Taking a bold stride beyond the confines of Gustave Eiffel University, the platform extends its reach through GustaveTutorService, offering its invaluable services to students from diverse institutions. The evolution doesn't stop there; a sophisticated payment system is integrated, complete with currency exchange rates. Collaborating seamlessly with Web Bank, the platform ensures thorough verification of fund availability and facilitates secure payment transactions, adding a layer of financial integrity to the tutoring experience.

This comprehensive report delves into the intricacies of the design and implementation of the Java application, meticulously examining elements, interfaces, and interactions that shape the user experience. Furthermore, the document embarks on an exploration of potential enhancements, considering the integration of graphical interfaces for both tutors and students. The ultimate aim is to elevate user experience and usability, making educationally enriching interactions seamless and engaging.

In essence, this report serves as a guiding narrative, offering valuable insights into the multifaceted functionalities of the tutoring platform. It not only delineates the strides taken to bring this educational solution to fruition but also underscores the platform's commitment to providing a holistic and user-centric approach to academic support.



# SYSTEM DESIGN

## 2.1 UML class diagram :

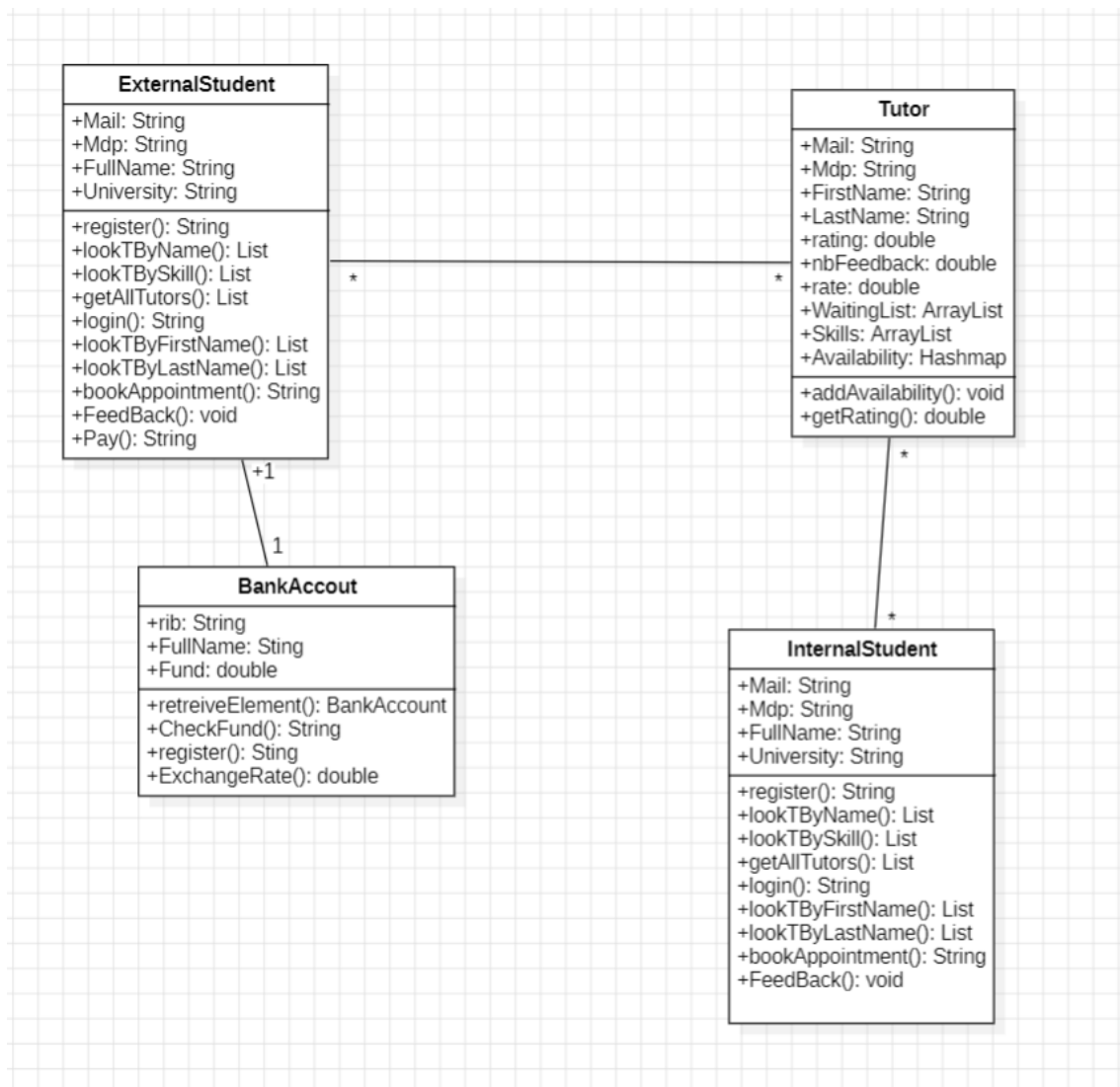


FIGURE 2.1 – class diagram

---

## **USED TECHNOLOGIES**

### **3.1 Remote Method Invocation (RMI) :**

Eiffel Tutoring Solutions relies on Java RMI middleware, a powerful mechanism for remote object communication. This allows the platform to seamlessly connect tutors and students across distributed Java Virtual Machines (JVMs), fostering real-time and efficient interactions. Key RMI classes, including 'Remote', 'RemoteException', and 'Registry', are harnessed to facilitate this distributed communication.

### **3.2 Web Services (Tomcat and Axis) :**

Tomcat, acting as the web server, and Axis, serving as the web service framework, form a robust combination. Tomcat hosts the platform's web services, while Axis streamlines the creation and deployment of SOAP-based services. This integration empowers Eiffel Tutoring Solutions with the capability to manage tutor-student interactions and facilitate seamless data exchange between various components.

### **3.3 JavaFX :**

JavaFX, the platform for building rich and interactive graphical user interfaces (GUIs), plays a pivotal role in enhancing the user experience for tutors and students. Leveraging JavaFX components such as 'Scene', 'Stage', and UI controls, the platform ensures an intuitive and

visually appealing interface. This integration extends beyond mere functionality, focusing on creating an engaging and user-friendly environment.

### **3.4 Deep Learning Model :**

The integration of a deep learning model introduces a layer of intelligent decision-making to the tutoring platform.

This thoughtful selection and integration of technologies define the robust foundation of Eiffel Tutoring Solutions. Each technology contributes distinct capabilities, collectively ensuring a seamless, intelligent, and user-centric educational solution.

---

# IMPLEMENTATION

## 4.1 Application Server :

### 4.1.1 Server Components :

- **Tutor Class** : contains Tutor information
- **Service Class** :
  - Hosts various methods for students to search for tutors by name or skill.
  - Manages tutor registration, login, availability listing, and feedback retrieval.
  - Notifies students on waiting lists via email when a session is booked.

### 4.1.2 Workflow :

1. Tutors register and log in, utilizing the Tutor Class functionalities.
2. Tutors list their expertise and set availability using the Service Class methods.
3. Students search for tutors using specified criteria (name, skill) via the Service Class.
4. Students book appointments, and those on waiting lists are notified of available sessions.
5. Tutors check feedback from their students to gauge their performance.

## **4.2 Java Application for Students :**

### **4.2.1 Components :**

- **Student Class :**

- Manages student registration and login.
- Allows students to provide feedback in natural language.

### **4.2.2 Intelligent System :**

- **BERT Language Model :**

- Evaluates feedback and assigns ratings using natural language processing.
- Calculates overall ratings for teacher sessions.

### **4.2.3 Workflow :**

1. Students register and log in, utilizing the Student Class functionalities.
2. Students provide feedback in natural language.
3. BERT Language Model processes feedback, assigns ratings, and calculates overall session ratings.
4. Students can retrieve the rating of a specific teacher.

## **4.3 External Student (Dynamic Web Project) :**

### **4.3.1 Components :**

- **External Student Class :**

- Shares functionalities with the internal student class.
- Allows external students to register, login, and book sessions.

### **4.3.2 Workflow :**

1. External students register and log in, leveraging the shared Student Class functionalities.
2. External students book appointments and interact with tutors using the same methods as internal students.

## **4.4 Gustave Tutor Service (Dynamic Web Project) :**

### **4.4.1 Workflow :**

1. Students use this service to pay tutoring fees in their preferred currency.
2. Gustave Tutor Service contacts the Web Bank service to verify fund availability.
3. It contacts another web service to get the exchange rate and convert currencies for payment.

## **4.5 Bank Server (Dynamic Web Project) :**

### **4.5.1 Components :**

- **BankAccount Class :**

- Manages bank account information.
- Facilitates bank account registration.

### **4.5.2 Workflow :**

1. Users register bank accounts through the BankAccount Class.
2. The Bank Server deducts the payment amount after successful transactions.

## **4.6 BankService (Dynamic Web Project) :**

### **4.6.1 Workflow :**

1. Gustave Tutor Service consumes this web service to process payments.
2. BankService verifies fund availability and facilitates payment transactions.

## **4.7 Conclusion**

This detailed breakdown highlights the intricate functionalities and interactions within Eiffel Tutoring Solutions, demonstrating how each component contributes to the overall success of the platform.

---

# CHALLENGES FACED

## 5.1 Whole Project Architecture :

### 5.1.1 Description :

- The overarching challenge was designing a coherent and scalable architecture for Eiffel Tutoring Solutions that accommodates various components like RMI, web services, JavaFX, and deep learning integration.

### 5.1.2 Solution :

- Regular feedback loops and revisions were conducted to ensure a robust and adaptable architecture.

## 5.2 Converting Student Feedback to Ratings

### 5.2.1 Description :

- Converting qualitative student feedback into a numerical rating posed a significant challenge. Designing a system that accurately interprets the sentiment of natural language feedback and assigns a meaningful rating was intricate.



### **5.2.2 Solution :**

- Leveraging advanced natural language processing (NLP) techniques, the project adopted the BERT (Bidirectional Encoder Representations from Transformers) language model. BERT's ability to capture contextual nuances in language facilitated the conversion of textual feedback into a quantifiable rating, providing more informed insights for prospective students.

## **5.3 Notifying a Student on the Waiting List :**

### **5.3.1 Description :**

- The problem is how to notify a student on the waiting list when his booked teacher is available.

### **5.3.2 Solution :**

- Implementing an email notification system to inform students on the waiting list about the availability of a high-demand tutor by using a web service request.

---

# **ADDITIONAL FUNCTIONALITIES :**

## **6.1 JavaFX Interfaces for Enhanced User Interaction :**

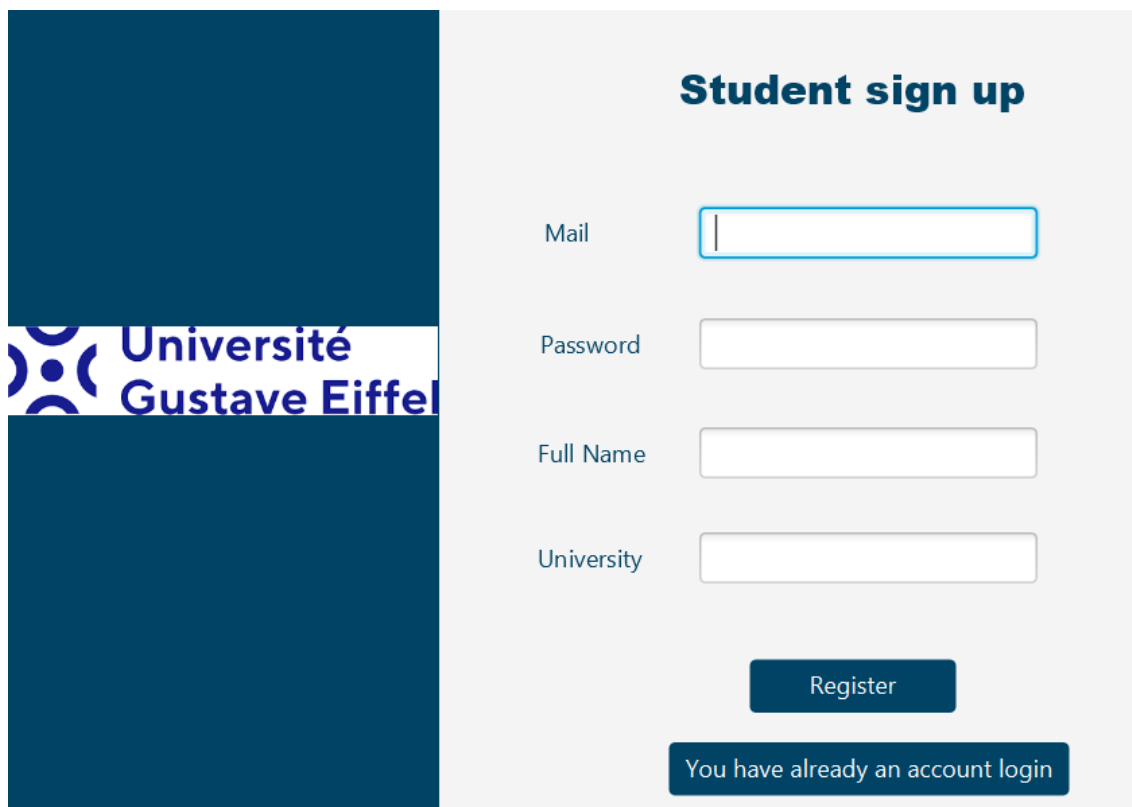
### **6.1.1 Description :**

- To elevate user experience, JavaFX interfaces were implemented, providing an intuitive and visually appealing environment for both tutors and students.

### **6.1.2 Benefits :**

- JavaFX components such as 'Scene' and UI controls enhanced the platform's usability.

- Intuitive interfaces allowed users to seamlessly navigate through functionalities like tutor registration, availability listing, and appointment booking.



The image shows a 'Student sign up' form. On the left is a dark blue vertical banner with the 'Université Gustave Eiffel' logo and name. The form itself is on a light gray background. It has a title 'Student sign up' in bold dark blue. Below the title are four input fields: 'Mail' (with a blue border), 'Password', 'Full Name', and 'University'. At the bottom of the form are two buttons: a dark blue 'Register' button and a dark blue button with white text that says 'You have already an account login'.

**Student sign up**

Mail

Password

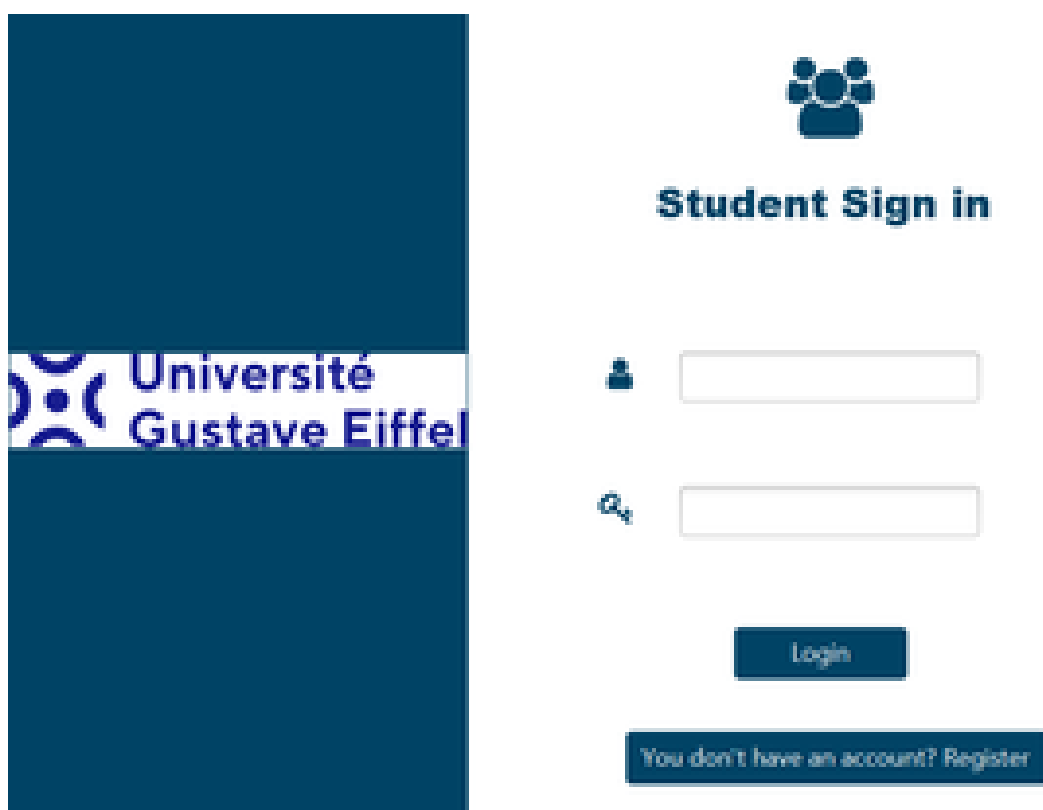
Full Name

University


Register

You have already an account login


**FIGURE 6.1 – Student sign up**




The image shows a 'Student Sign in' form. On the left is a dark blue vertical banner with the 'Université Gustave Eiffel' logo and name. The form is on a light gray background. It has a group of three people icon above the title 'Student Sign in' in bold dark blue. Below the title are two input fields: the first has a person icon and the second has a magnifying glass icon. At the bottom are two buttons: a dark blue 'Login' button and a dark blue button with white text that says 'You don't have an account? Register'.



**Student Sign in**

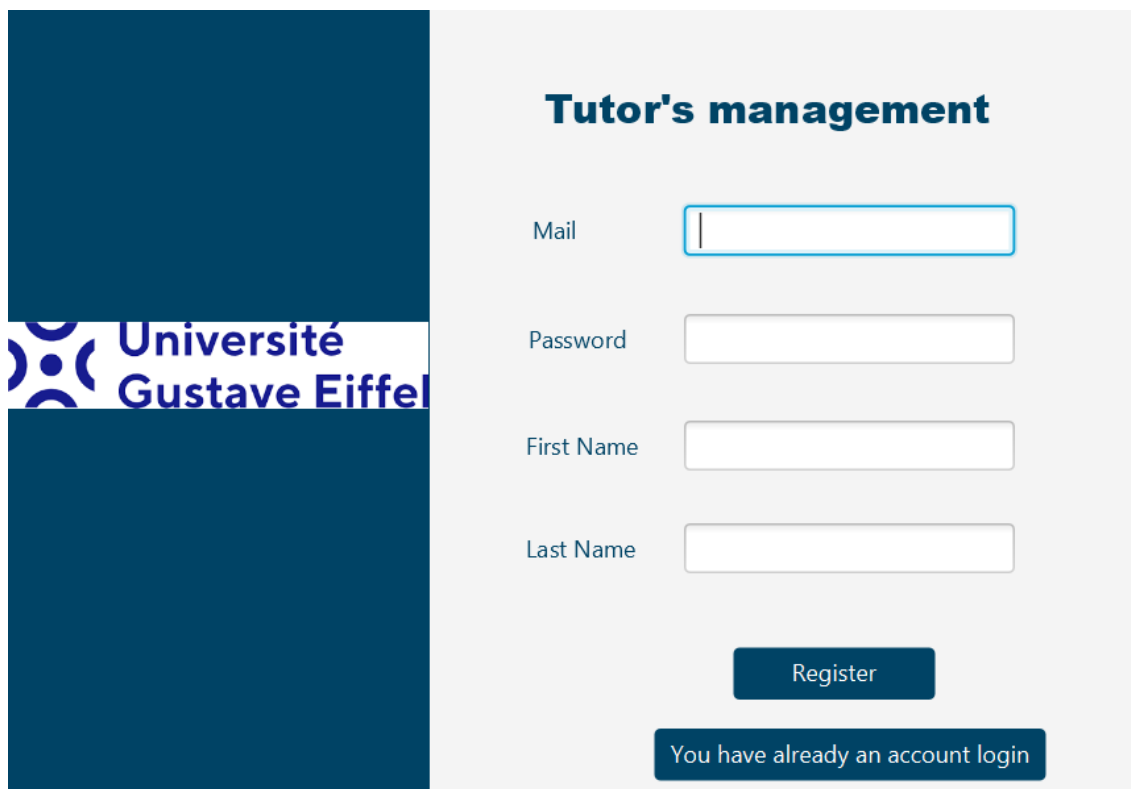




Login

You don't have an account? Register

**FIGURE 6.2 – Student sign in**



The screenshot shows a web interface for 'Tutor's management' at Université Gustave Eiffel. On the left is a dark blue sidebar with the university's logo and name. The main content area has a light gray background. At the top, the title 'Tutor's management' is in bold dark blue. Below it are four input fields: 'Mail' (with a blue border), 'Password', 'First Name', and 'Last Name'. A dark blue 'Register' button is positioned below the fields. At the bottom, a dark blue button contains the text 'You have already an account login'.

**Tutor's management**

Mail

Password

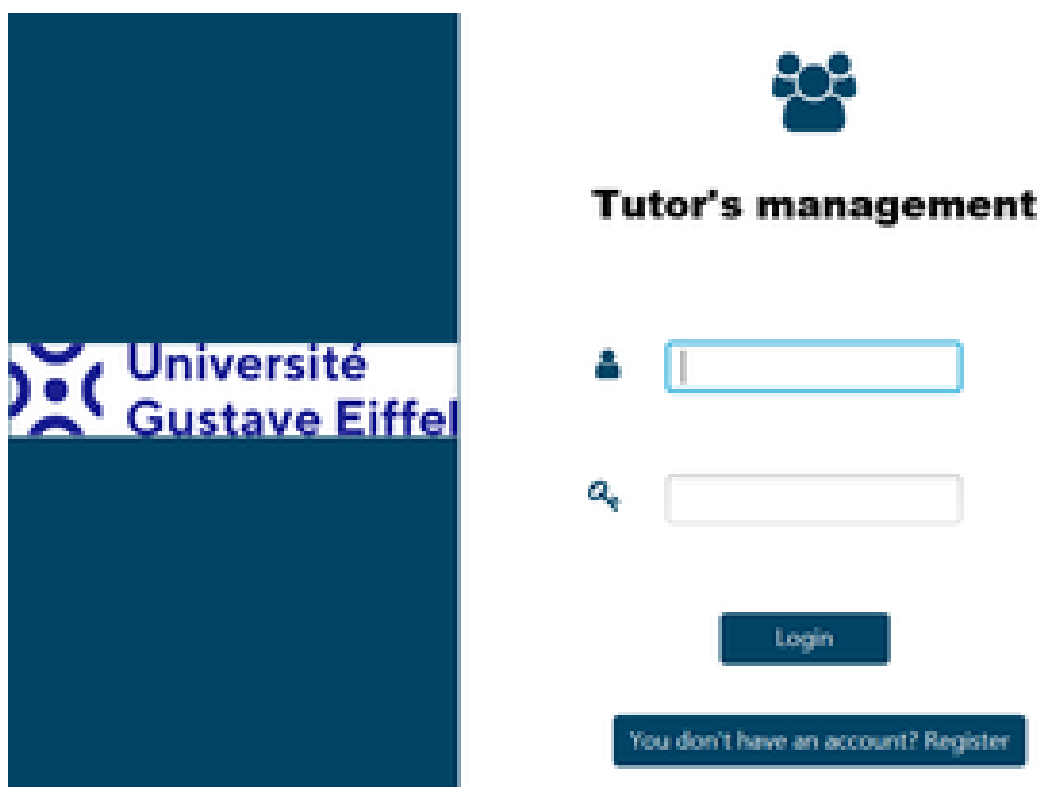
First Name

Last Name


Register

You have already an account login


FIGURE 6.3 – Tutor sign up




The screenshot shows a web interface for 'Tutor's management' at Université Gustave Eiffel. On the left is a dark blue sidebar with the university's logo and name. The main content area has a light gray background. At the top, there is a group of three people icon, followed by the title 'Tutor's management'. Below the title are two input fields: the first is preceded by a person icon, and the second by a magnifying glass icon. A dark blue 'Login' button is positioned below the fields. At the bottom, a dark blue button contains the text 'You don't have an account? Register'.



**Tutor's management**





Login

You don't have an account? Register

FIGURE 6.4 – Tutor sign in


## *Book Appointment*

Choose Availability :

Date  From  To

Availabilities

**FIGURE 6.5 – Book appointment**



Mail :

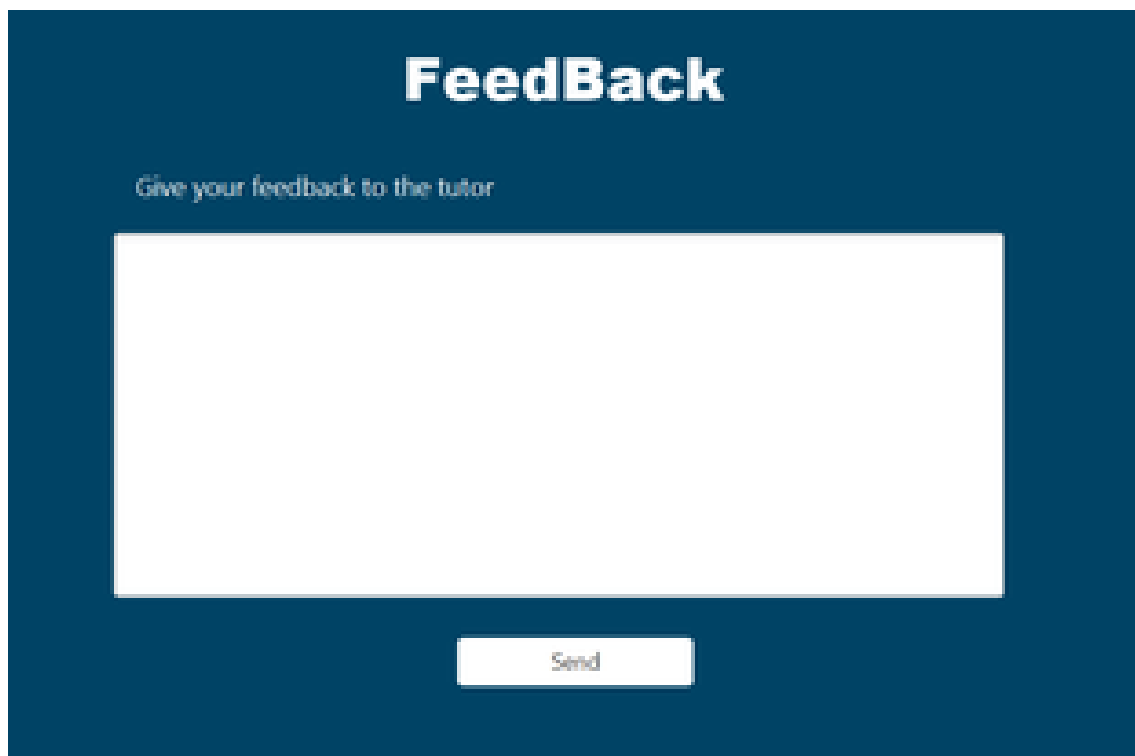
Full Name : Mathieu Hugo

Skills :

Availabilities :

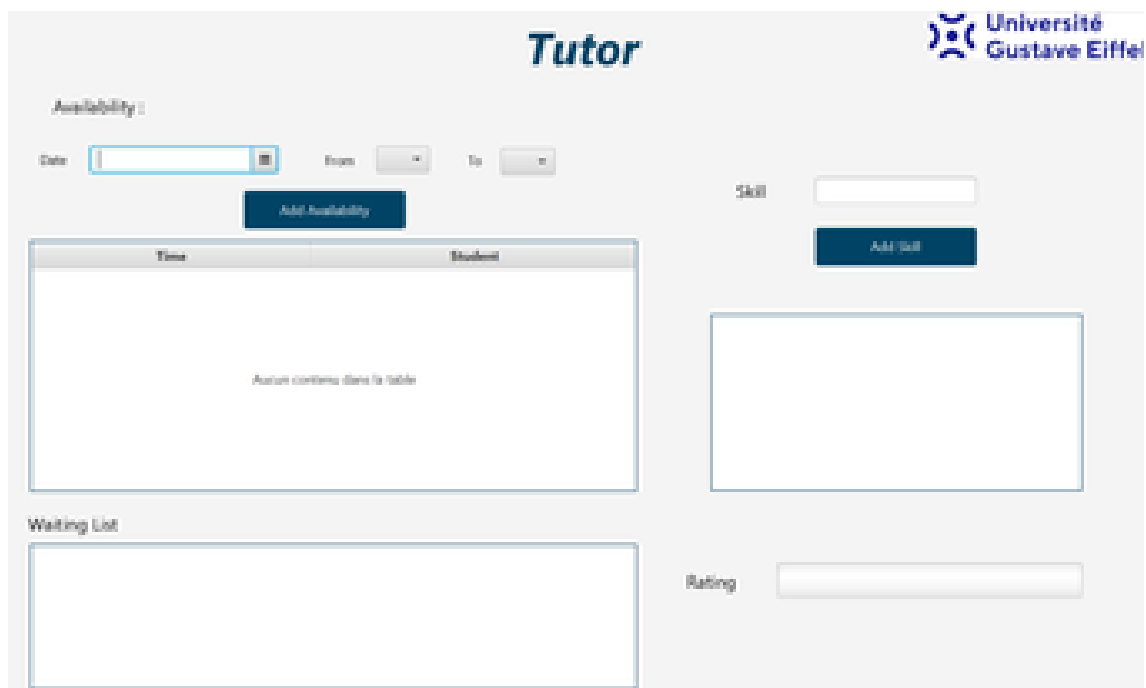
Rating:

**FIGURE 6.6 – Tutor details**



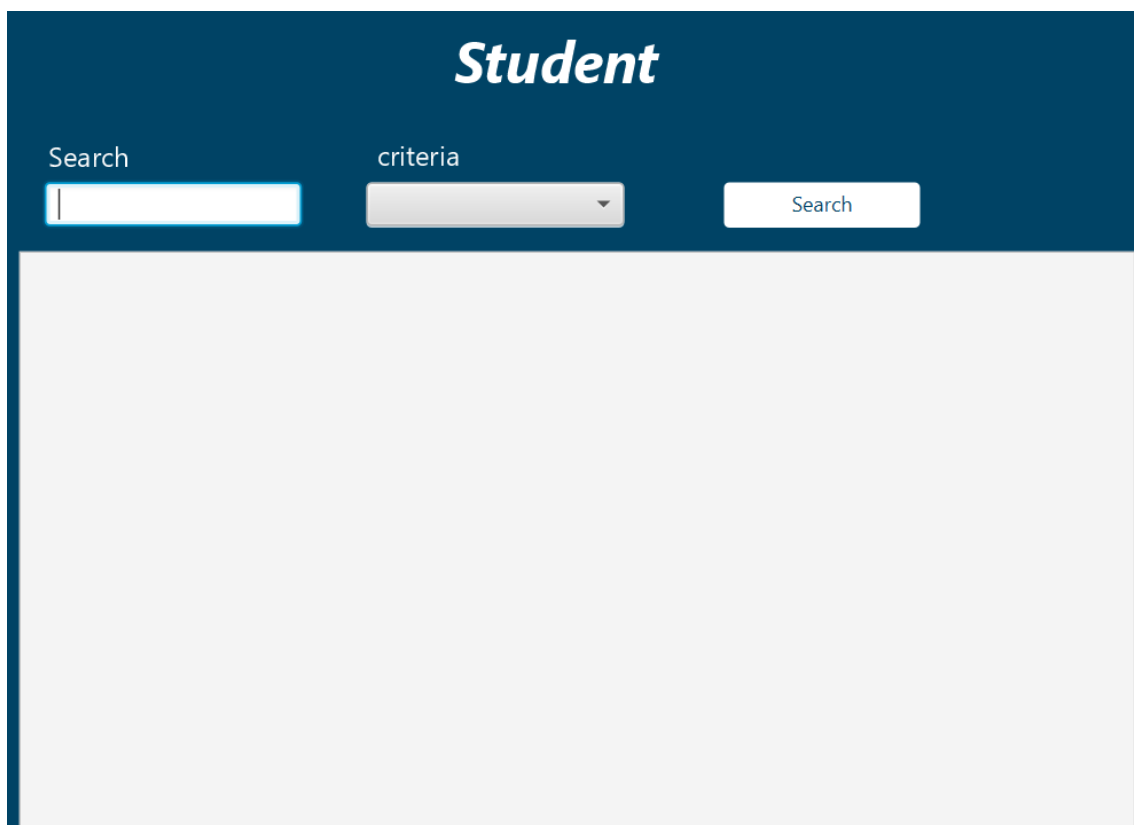
The image shows a 'FeedBack' form with a dark blue background. At the top, the word 'FeedBack' is written in large white letters. Below it, the text 'Give your feedback to the tutor' is displayed in a smaller white font. A large white rectangular box is provided for the student to enter their feedback. At the bottom center of the form, there is a white button with the text 'Send' in dark blue.

**FIGURE 6.7 – Student feedback**



The image shows the 'Tutor' interface with a light gray background. At the top center, the word 'Tutor' is written in a large, bold, dark blue font. In the top right corner, there is a logo for 'Université Gustave Eiffel' consisting of a stylized 'U' and 'E' icon followed by the text 'Université Gustave Eiffel'. Below the title, there is a section for 'Availability :'. It includes a 'Date' field with a calendar icon, 'From' and 'To' time selection buttons, and an 'Add Availability' button. To the right of this section, there is a 'Skill' field and an 'Add Skill' button. Below the 'Availability' section, there is a table with two columns: 'Time' and 'Student'. The table is currently empty, and a message 'Aucun contenu dans la table' (No content in the table) is displayed in the center. Below the table, there is a 'Waiting List' section with a large empty rectangular box. To the right of the 'Waiting List', there is a 'Rating' field with a horizontal slider bar.

**FIGURE 6.8 – Tutor interface**



**FIGURE 6.9 – Student interface**

## 6.2 Mailing Service for Waiting List Notifications

### 6.2.1 Description :

- A mailing service was integrated to notify students on the waiting list when a tutor became available, ensuring timely communication.

### 6.2.2 Benefits :

- Real-time email notifications provided waiting students with immediate updates on session availability.

- The mailing service enhanced user engagement by keeping students informed about relevant opportunities.

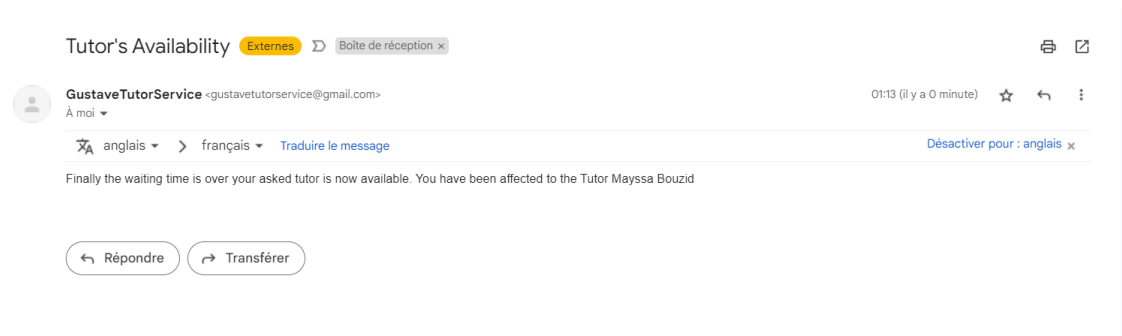


FIGURE 6.10 – send mail

## 6.3 Fine-Tuned BERT Language Model

### 6.3.1 Description :

- The project adopted a fine-tuned version of the pre-trained BERT large language model, specifically tailored for sentiment analysis using the IMDb dataset.



### **6.3.2 Benefits :**

- The fine-tuned BERT model processed student feedback, providing nuanced sentiment analysis for accurate session ratings.
- Leveraging the Hugging Face web service endpoint streamlined the integration, ensuring efficient and reliable sentiment analysis.

## **6.4 Conclusion**

These additional functionalities not only contributed to a more user-friendly and engaging platform but also enhanced the intelligence and responsiveness of Eiffel Tutoring Solutions, creating a well-rounded and sophisticated educational environment.



---

## CONCLUSION

In summary, Eiffel Tutoring Solutions represents more than a technological initiative, it embodies the spirit of collaboration and a shared commitment to enriching the educational landscape. This project faced multifaceted challenges, each serving as a springboard for innovation and growth. The creation of a robust architecture harmonizing RMI, web services, JavaFX, and advanced NLP was a significant achievement, forged through iterative refinement and resilient teamwork.

Our introduction of user-centric functionalities, like intuitive JavaFX interfaces and a mailing service for waiting list notifications, focused not on extravagance, but on simplicity and practicality to ensure an effortless user experience. Incorporating a fine-tuned BERT language model exemplifies our strategic use of advanced tools for nuanced sentiment analysis, enhancing session ratings.

However, beyond technological milestones, Eiffel Tutoring Solutions embodies humility, a thirst for learning, and collective dedication. It's not solely about lines of code but the countless hours, challenges overcome, and lessons absorbed.

As we conclude this phase, it marks not an end but a beginning. Eiffel Tutoring Solutions may not revolutionize, but it signifies an ongoing commitment to enhancing educational experiences. It symbolizes the belief that technology, thoughtfully applied, fosters supportive and engaging academic environments.

Our gratitude extends to Dr. Zargayouna Mahdi, our team, and all involved. Together, we've sown seeds of innovation and commitment to an educational journey transcending boundaries. We envision a future where technology empowers education, fostering a supportive ecosystem.