Université Gustave Eiffel

REPORT FOR THE PROJECT OF

Gustave Tutoring Application

BOUHAOUALA Sana
BOUGHZALA Ghailene
BOUSSIF Melek

2023-2024

Table of Contents

Abstract:

The Eiffel Tutoring Solutions project epitomizes a strategic initiative conceived to address the escalating demand for academic support within the esteemed confines of Gustave Eiffel University. This comprehensive report systematically delineates the conceptualization, realization, and outcomes of a distributed Java application, meticulously grounded in the Java RMI middleware. The principal aim is to seamlessly facilitate the nexus between students seeking academic fortification and erudite tutors poised to disseminate their specialized knowledge. Moreover, the project envisions extending its services beyond the precincts of the university through the judicious incorporation of the GustaveTutorService web service. This auxiliary service promises to broaden accessibility, welcoming students from disparate institutions. The report articulates the intricacies of the architectural framework, interfaces, interactive dynamics, encountered challenges, and innovative solutions that have manifested throughout the iterative phases of development.

1. Introduction:

1.1 Project Overview:

The Eiffel Tutoring Solutions project started because we saw a big need for a helpful tutoring system at Gustave Eiffel University. Imagine it like a digital place where students and teachers can connect easily to share knowledge. We built this using a special kind of computer language called Java, and it works like a smooth dance between different parts of the system.

Our main goal was to create a friendly space where tutors (the teachers) can easily sign up and share what subjects they're good at. On the other side, students can easily search for these tutors, see when they're available, and ask for help. We wanted to make it all work smoothly like a well-organized library of knowledge right at your fingertips.

The whole idea is to make learning and teaching feel like a friendly chat, like you would have with a friend. This is our way of making sure everyone at Gustave Eiffel University can learn and help each other out in a really cool and easy way.

1.2 Project Goals:

Our main goals for the Eiffel Tutoring project are like weaving a really nice blanket that combines teaching help and clever technology:

- Helping Students and Tutors Connect Easily: Imagine a website that is like a friendly meeting place for students and tutors. We want it to be easy for them to find each other and work together.
- Making a Special Kind of Computer System: We're using a computer language called Java to build our website. It's like building a smart and fast system that can quickly respond to what students and tutors need.
- Letting Tutors Control Their Info: Tutors, who are the teachers, can easily sign up and share what subjects they're good at. They can also decide when to teach and how much they want to be paid. It's like giving them the keys to control their own part of the website.
- Helping Students Find and Talk to Tutors: It's like having a super simple map for students. They can easily search for tutors, see when they're available, and ask for help. It's like ensuring students can quickly find the right person to guide them.
- Making it Work for Everyone, Everywhere: We don't want to limit this cool learning space to just one group. We're dreaming big! We want students from different schools or areas to join in. It's like opening the doors to a bigger community of learners. Adding Something Extra Cool - Real-Time Money Exchange: We're adding a special feature! When students need to pay for tutoring, we've made it easy. They can pay in their own money, and we'll magically change it to the money the tutor wants. It's like having a really handy money translator right there on the website.
-

So, our project is like making a comfortable, easy-to-use place where students and tutors meet, with a sprinkle of smart technology to make it even better!

2. System Design:

2.1 System Architecture:

In creating our tutoring platform, we designed a strong and reliable structure using something called Java RMI. Think of it like a sturdy framework that supports and connects all the different parts. This framework helps our system run smoothly, just like a well-organized orchestra.

We have two main parts in our system – one for tutors (the teachers) and one for students. It's like having two separate rooms, each with its own computer space, called Java Virtual Machines (JVMs). These rooms keep things tidy and organized.

Then, there's this special service called GustaveTutorService. It's like the center of everything. This service helps bring in people from outside the university too, making it a big community. It works like a special web service, connecting our inside world with the broader outside world.

2. System Design:

2.2 Elements and Interfaces:

- Tutors: Our tutors are like the heart of our system. They are knowledgeable people who want to share what they know. To join, they go through a simple sign-up process where they share what subjects they can teach and when they're available. It's like setting up their own special space for teaching. The space they use, which we call the tutor interface, is where they can talk about what they're good at, when they're free, and how much they want to get paid.

- Students: Students are the ones eager to learn. We made a special area for them to find tutors and learn more easily. It's like creating a simple map for them to explore and find the right tutor. They can see when tutors are available and easily make appointments or payments. We want their experience to be like navigating a friendly and helpful space.

- GustaveTutorService: This special service is like the bridge between our university space and the larger world. It helps students from different places join in. It also does something cool – ensuring that when students pay, tutors translate the money correctly. It's like having a helpful translator for money matters.

2. System Design:

2.3 Interactions:

- Tutor Registration: When a tutor wants to join, they undergo a simple sign-up process. It's like filling out a form online. The information they share is sent securely using special Java RMI rules.

- Student-Tutor Interaction: Students and tutors can easily talk to each other through our system. It's like having a smooth conversation where information flows seamlessly. The Java RMI system helps make sure everything runs in harmony.

- GustaveTutorService Interaction: External users (people from outside the university) can easily join our platform through GustaveTutorService. It's like welcoming new friends. The service also does something really helpful – it connects with an external Web Bank service to handle money matters, making it easy for everyone, no matter where they are.

3. Implementation

3.1 Applications and Web Services:

- Tutor Application:
- In our Java RMI setup, we created two essential applications – one called "client" and the other called "server." The server application is like the main hub where we keep all the information about tutors and their appointments. It's like the brain of our system, handling all the important data.

- Student Application: On the other side, the client application is where students hang out. It's designed to be user-friendly and has all the tools students need to find tutors and manage appointments. It's like a cozy corner for students to explore and connect with their tutors.

- Banking Task Management Application: To handle the financial side of things, we incorporated an application that manages banking tasks. It's like having a separate wizard that handles all the money-related activities. We made it even more efficient by using an API called FXTOP, a special tool helping us manage currency exchange smoothly.

3. Implementation

3.1 Applications and Web Services:

- GustaveTutorService: We introduced a web service called GustaveTutorService. This is like the welcoming door for people from outside the university to join our tutoring platform. It's not just a door; it's also a smart translator. When students pay for tutoring, this service connects with an external Web Bank service to handle money matters. It ensures that payments are smooth and can be done in different currencies. It's like a helpful guide for everyone to connect and share knowledge.

3. Implementation

3.2 Graphical Interfaces:

- **Tutor Interface:** Within the server application, we've organized the tutor's interface to be easy to navigate. We used JavaFX to make sure it's user-friendly. Tutors can easily manage their appointments and other important details. It's like giving tutors their control panel where they can effortlessly handle their tutoring responsibilities.

- **Student Interface:** For the client application, we also used JavaFX to create an interface that's simple and intuitive. Students can easily explore tutor profiles, book appointments, and make payments hassle-free. It's like having a friendly map for students to navigate their learning journey.

3. Implementation

3.3 Scenario:

Now, let's dive deeper into the scenario of a student using our platform:
Imagine Sana, a student eager to get some help with her physics homework. She opens the client application, where she can easily search for physics tutors. After browsing tutor profiles, she finds Ghailene, a physics whiz with great reviews. Sana decides to book an appointment with him for the next day.

Behind the scenes, the "Server" application manages this appointment seamlessly. It updated Ghailene's schedule and notified him of the new appointment. Now, here's where the banking task application comes in. When it's time for Sana to pay for the tutoring session, the application, with the help of the FXTOP API, ensures a smooth transaction, converting currencies effortlessly if needed.

Throughout this scenario, our distributed system works harmoniously, making Sana's journey to find and connect with a tutor not just educational but also enjoyable.

4. Challenges and Solutions:

Challenges

- Integration with JavaFX: Integrating our applications with JavaFX posed a significant challenge. It's akin to assembling a complex puzzle, and a particularly tricky part was the rebind process. This step, crucial for connecting different elements of our system, required meticulous attention.

- Web Service API Integration: Another challenge we encountered was integrating our platform with the FXTOP API for currency exchange. Ensuring seamless communication between our applications and the external API required careful coordination.

- Choice of Architecture: Selecting the right architecture for our distributed system was a critical decision. It's like choosing the blueprint for a house. We needed an architecture that could handle the complexity of tutoring interactions and financial transactions.

4. Challenges and Solutions:

Solutions

- **Overcoming JavaFX Integration Challenges:** Facing the intricacies of JavaFX, especially the rebinding process, demanded dedicated effort. We invested time and resources to understand JavaFX thoroughly, ensuring a smooth integration. It's like patiently solving a puzzle to create a beautiful picture.

- **API Web Service Integration:** Integrating with the FXTOP API required careful planning and execution. We established clear communication channels between our banking task application and the API, ensuring a reliable and secure exchange of currency information. It's like setting up a secure bridge between different parts of our system.

- **Architectural Decision-making:** Choosing the right architecture was akin to crafting the blueprint for our platform. We conducted thorough evaluations to select an architecture that could handle the complexities of tutoring services and financial transactions. It's like ensuring our house is beautiful and structurally sound.

5. Conclusion

In concluding this comprehensive report on the GustaveEiffel Tutoring project, we find ourselves at the intersection of innovation and education. Our journey began with a vision to create a platform that transcends traditional boundaries, connecting students and tutors in a seamless knowledge exchange.

Gustave Eiffel Tutoring has evolved into more than a solution; it embodies symbiotic academic growth. Tutors find a platform to share their expertise, students discover avenues for academic support, and the community thrives on the dynamic interplay between these two pillars of education.

The architectural choices, leveraging Java RMI for distributed functionality and JavaFX for user interfaces, form the backbone of a robust and user-friendly system. The implementation details, from tutor and student applications to the integration of real-time currency exchange, showcase the technical prowess harnessed in building this platform.

5. Conclusion

The user manual, crafted precisely, serves as a beacon, guiding users through the platform's intricacies. From installation to troubleshooting, it is a testament to our commitment to a user-centric approach. Clear language, visual aids, and a logical structure ensure that users can unlock the full potential of Gustave Eiffel Tutoring with ease.

As we conclude this report, we'd like to let it serve as a stepping stone to future enrichments. The world of technology and education is ever-evolving, and Gustave Eiffel Tutoring stands ready to adapt and grow. Future enhancements include AI-driven recommendations, enhanced interactivity, or even partnerships with educational institutions worldwide.

Thank you for being part of this academic revolution.