TutorMe.io

Diego Figueroa - [**diego.figueroa3@upr.edu**](mailto:-diego.figueroa3@upr.edu)

Luis Rivera - [**luis.rivera157@upr.edu**](mailto:luis.rivera157@upr.edu)

Andres Hernandez - [**andres.hernandez2@upr.edu**](mailto:andres.hernandez2@upr.edu)

# Introduction

TutorMe.io is an educational web app inspired in the *Uberization* of Tutoring. The common business scheme of tutoring is as goes: student pays for tutoring session(s), student schedules session(s) with tutor, student keeps going until he improves his knowledge. For the tutor, the problem with this scheme is that as the student registry scales, it becomes harder to keep up with every student’s progress. For the student, the problem with this scheme is that it becomes to harder to set meetings during the week due to unexpected disruptions, or schedule conflicts with the tutor so there can be times when the student and the tutor won’t meet.

TutorMe.io looks to solve this problem by offering an open platform for tutors to publicly showcase their services under their own domain, providing crucial features like a Public Course Wall, Message Box to communicate with other students or tutors, Assignments section to post course work and Lessons section to post class videos and other information. For the students’ side, they can see the recorded session whenever they can and as many times they may need to understand the topics. It is also convenient, since you can find a tutor for any of your classes on the same web page and you can pay your tutor a set amount without the hassle of having to pay with cash. The student will also be able to communicate easily with his tutor though the public wall and private messages. The advantage of communicating through the public wall is that you can benefit from other student’s questions, just like in a real classroom.

This app will give the student all the advantages of a real private tutor, without any of the inconveniences.

# Client Side

The client app will be a web page that works well on both computers and mobile devices. When you first go to the website, you will find a brief description of the app, the option to log in, and the option to register as a student or as a tutor.

If you log in with a student account you will find a summary of your courses at the moment, including if there are new lessons or tasks for any one of them. You will also be able to find new courses to enroll in, as well as your personal messages. If you find a course you want to enroll in, you can easily review the course, the cost, talk to the tutor if you want, and eventually pay to enroll. Then if you select a specific course, you get a list of the different lessons available and tasks associated with each one, if any. You can select a lesson to watch a video or read the material that the tutor uploaded. You can make public comments on each individual lesson that your tutor can then answer. You can also send the tutor, or another student a private message.

If you log in with a tutor account you will find a summary of the courses you are teaching at the moment, including if any student left a comment in any lesson. You will also be able to create new courses, as well as your personal messages. If you are going to create a new course, you can easily set up the cost and description. Then, if you select a specific course you are teaching, you can see enrolled students and comments. You can also add a lesson to the course, and make public comments on each individual lesson. You can also send a private message to any student.

To implement the client app, we will be using AngularJS as the front-end framework, along with Twitter Bootstrap.

# Server Side

The server side application should be able to authenticate users using their unique email and password. The server side is inspired in RESTful services implemented using Java Play and must be capable of receiving requests asynchronously and respond with requested information in JSON format, along with a status code that complies with HTTP standards. The server side should be able to store all it’s user information in a PostgreSQL database and keep sessions live. The relation schema is as follows:

* Users - authentication and accountability
* Tutors
* Students
* Courses
* Assignments
* Lessons
* Deadlines - records that describe a time limit for objects like Assignments or Lessons
* Walls
* Posts
* Comments
* Favorites
* Messages

# Division of Labor

* Diego Figueroa
* Development of features related to Users, Tutors, Students and Courses
* Design Landing, Signup/Login, Homepage, Course and Profile pages
* Implement Signup/Login Auth scheme
* Implement Settings module
* Andres Hernandez
* Development of features related to Walls, Posts, Comments, Favorites, and Messages.
* Design the look of Walls, Posts, Comments, and Favorites.
* Implement a messaging system.
* Design the messages page.
* Luis Rivera
* Developments of Assignments, Lessons, and Deadlines.
* Design Assignments and Lessons pages.
* Implement an assignments system.
* Implement a file storage solution to save the files uploaded by users.