

Problem Background

- Currently the UPRM's system for enrollment is very tedious and has many flaws. It would be more productive to design a more user-friendly web application and backend that could handle all of the users that are currently in the old system.
- If we were to solve this problem, we would see an increase in productivity with the UPRM's management and students. It would lower the time of enrollment and the amount of requests for assistance to the administration which would allow for them to use that extra free time in managing their assigned tasks. This would also encourage more students to participate in the enrollment process since it would be easier to use.

Countermeasures

- Create a single page web application using React, Typescript, Sass, and GraphQL to access and manage the enrollment process for students and the administration.
- Create a Backend service using Node.js, GraphQL, and Postgres database to administrate everything regarding the enrollment of students and all similar information.
- Make this Backend scalable for future expansion.
- Add payment functionalities using PayPal and Stripe to handle the transactions for enrollment.
- Use GraphQL subscriptions to make real time changes visible to the end user so that they can better assess what is going on in the system and what they are allowed to do.

Target

- Create an easier experience and more robust system to manage enrollments.
- Add more functionality to allow students to take better decisions in their enrollment process.
- Allow students to visualize their current enrollment and what steps they should be taking.
- Add more features that would let students pay for and even keep track of their courses.

Causes

- Current system for managing student enrollment is not scalable.
- Access by students is very tedious and requires extra work to get set up.
- The current system is not user-friendly.
- The system fails after a few thousand users are connected.
- In some cases when students want to go back to the previous page, the system closes and students must log in again.

Check/Evaluate

The countermeasure worked since it allowed for a more user friendly experience and the servers did not go down when being used.

Act/Standardize

Before:

- Students had to wait for the system to react to changes.

Now:

- Students can now use the system without it crashing and have an easier time with it.