

PluMA 2.0 Administrator Interface

Students: Ihosvani Noa Zambrano, Emmanuel Egbukuyomi, Brian Gonzalez, Lewis Mesa, Isaac Scott, Javier Fernandez, Joshua Alexander, Leonardo Rodriguez, Max Karey

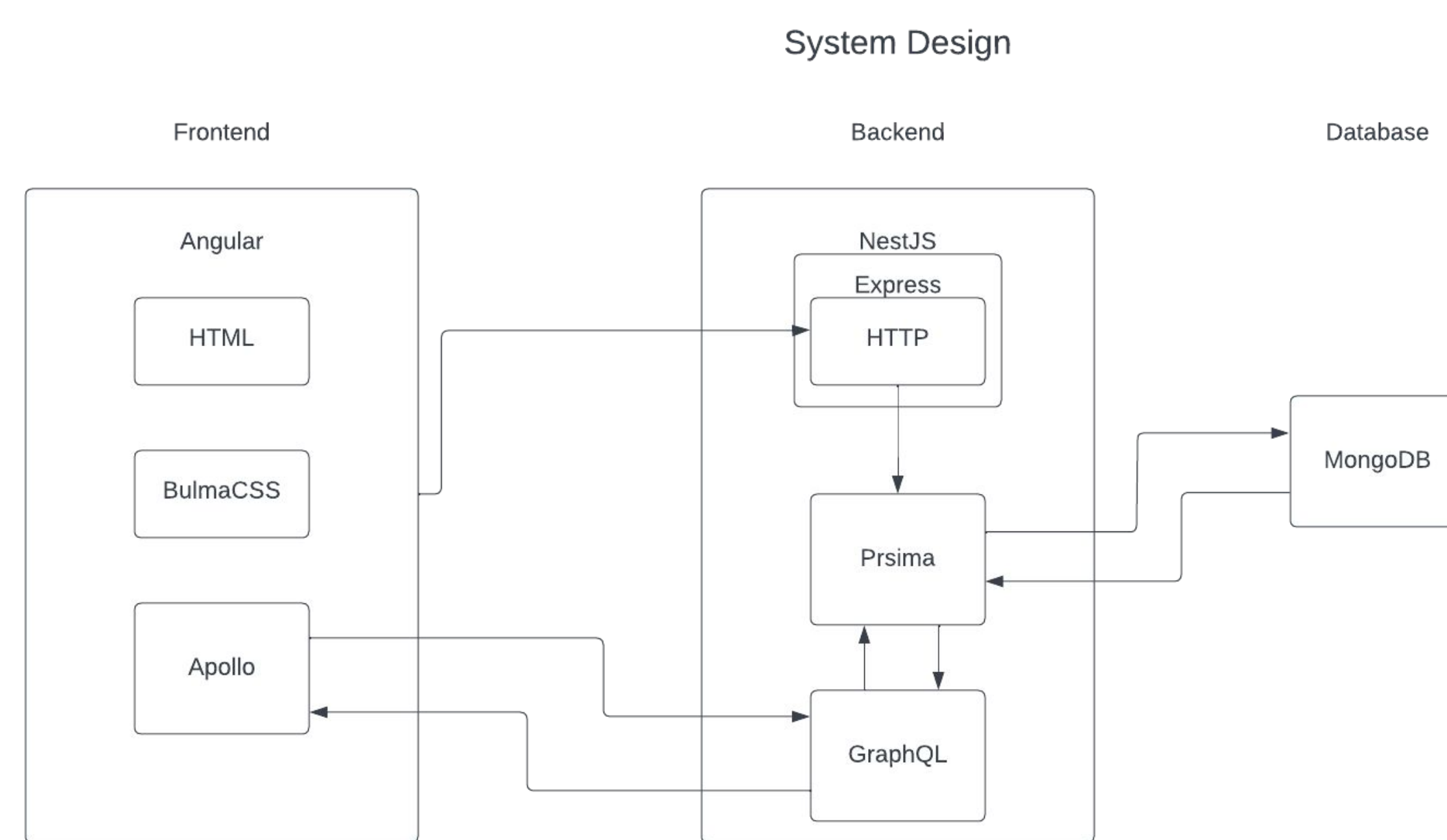
Mentor: Joseph Quinn, SkillCourt

Instructor: Dr. Masoud Sadjadi, Florida International University

Problem

PluMA or Plugin-Based Microbiome Analysis, is a webpage to facilitate the constructions of pipelines which can be used to inspect Microbiomes and allow for re-usability of other programs already created by other developers.

System Design



Individual Contribution

I worked on the creation and testing of several mutations to prisma graphql functions related to administrator and user management. This involved setting up the function properly, properly integrating and importing other PluMA components, and setting up throw cases to handle potential errors. Overall, while the actual functionality itself was rather simple, the actual creation and implementation of some features was still challenging, as PluMA involved so many more components than I was used to. As such, working on PluMA was a good learning experience for working and dealing with big projects.

Register

Email

email@fiu.edu

Email Confirmation

email@fiu.edu

Password

....

Length of password must be at least 8 characters.
Password must include at least one of each:

- lowercase
- uppercase
- number
- symbol

Password Confirmation

....

Display Name

Name

Institution

FIU

Website

website.com

Submit

Current System

The PluMA system runs in Apollo Angular utilizing GraphQL and Prisma to access the mongoDB database; The code style structure is defined by the eslint visual studio code extension. Prisma is utilized to define database schemas and is responsible for communicating with MongoDB which is necessary for the GraphQL API. The rest API is dedicated to authentication of users through first Express taking an HTTP request, then NestJS maintaining the file structure by separating the controller from the service.

Screenshots

ACTIONS	Name	Description	GitHub	Language	Category		
Add Plugin	Admin Plugin	Admin Plugin	Admin Plugin	Java	Admin Category	UPDATE	DELETE
Add Category	Admin plugin 4	Admin plugin 4	Admin plugin 4	Java	Admin Category	UPDATE	DELETE
	Admin plugin 5	Admin plugin 5	Admin plugin 5	Java	Admin Category	UPDATE	DELETE

CATEGORIES FILTER

All

Admin Category

Admin Category 2

Admin Category 3

forJava

forJavaScript

forPython

forTypeScript

Username	Email	Website	Institution	Role	ROLES	DELETE	DEACTIVATE	RESET PASSWORD
Ihosvani	ihosvani@fiu.edu	fiu.edu	fiu	Admin	ROLES	DELETE	DEACTIVATE	RESET PASSWORD
ad	ad1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ae	ae1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
af	af1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ah	ah1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ai	ai1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
aj	aj1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ak	ak1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
al	al1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
am	am1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
an	an1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ao	ao1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ap	ap1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
aq	aq1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
ar	ar1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
as	as1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
at	at1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
au	au1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
av	av1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD
aw	aw1@fiu.edu	fiu.edu	FIU	User	ROLES	DELETE	ACTIVATE	RESET PASSWORD

NEXT 20

Requirements

Throughout the sprints, we worked on the registration component, the admin portal, the capability for a user to import/create a plugin, and authorization handled by role. We have introduced and added additional functionality connecting administrators and the database through both creating new features in the frontend for clients as well as implementing new features to communicate with the database and the backend.

Implementation



Summary

We were successful in completing the base code for all required PBI's. While we were working towards adding in and implementing our new user stories, we found it necessary to adjust and troubleshoot significant portions of the back-end. The Jest Library was implemented during this project cycle, however, unit testing still needs to be completed for most of the stories. Thank you for providing us with the opportunity to work on a real-world type problem and solution with an incredibly knowledgeable mentor.

Acknowledgement

The material presented in this poster is based upon the work supported by Florida International University Knight Foundation School of Computing and Information Sciences We thank Joseph Quinn for assistance, cooperation and mentorship that I/we received throughout this process