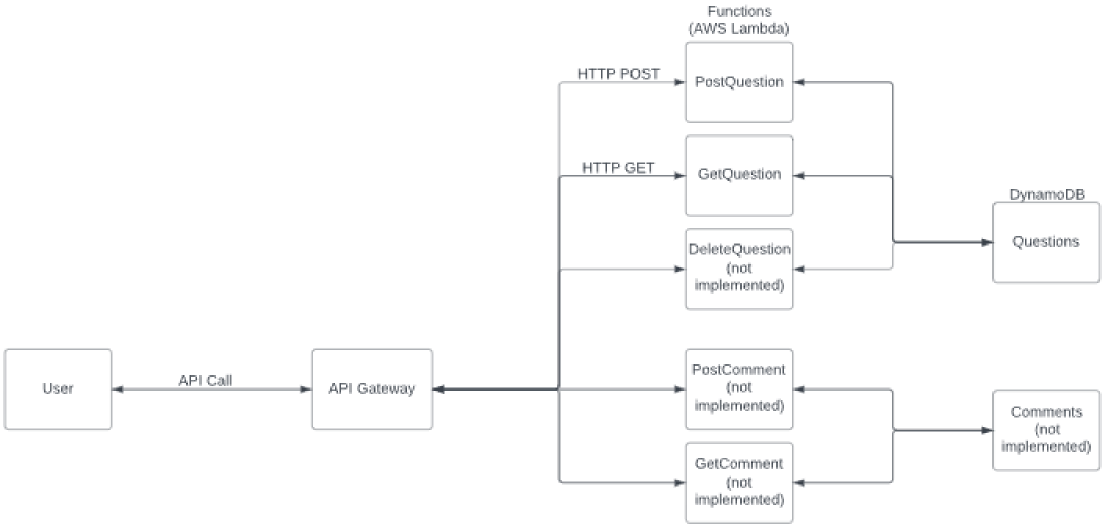
Student Bubble

System Analysis:

Component	Technology	Reason
Backend	lambda	Serverless function provided by AWS to execute code.
FrontEnd	AWS Amplify	Complete serverless solution to host our front end website and can make updates easily
Database	dynamoDB	Serverless NoSQL database that is simple to use for small size operation
API Gateway	REST API using AWS API Gateway	Connects front-end website with backend data stores via REST API's
Policies	IAM	Easy way to change permissions for AWS lambda functions

System Design:



Test Procedures:

Lambda Functions : We used the AWS Lambda functions console to create, test and deploy our GetQuestion and PostQuestion. In order to test our functions for correctness we utilised the inbuilt private test events.

Get and Post API: We used API Gateway within AWS to create the REST API to implement our GET and POST functions. We then linked our lambda functions to the API and deployed it. We then used the inbuilt test function to test the API. We provided the appropriate input and tested our methods.

We tested the overall connection of our back end and the front end by using our individual deployment URL’s and seeing if we obtained our desired outcome.

Goal measures & deployment considerations (Max 350 words):

For this iteration we focused on setting up a web system that can accept user’s questions, and display a list of all questions posted. We believe that this would be the main use and purpose of our website. We used services provided by AWS for our deployment.

The user connects to our front end component hosted on AWS Amplify. The user can input a question title and question description into the forms. When the user presses the ‘Submit’ button, it will trigger a POST API call to the API Gateway then the PostQuestion Lambda function. The Lambda function will communicate with the Questions DynamoDB database to input the data. When the user presses the ‘All Questions’ button, it will trigger a GET API call to the API Gateway then the GetQuestion Lambda function. The function will read data from the Questions DynamoDB, then return to the front end with question information.

Security and Exposure points:

Security structures were the main area we sacrificed this iteration. Our project has no security outside of what is inherently provided as part of a cloud computing system. In future iterations, we plan to add a user authentication system that will allow for the implementation of additional security structures.