

# PROJECT PROPOSAL

## CSCI 5409: CLOUD COMPUTING Summer 2022

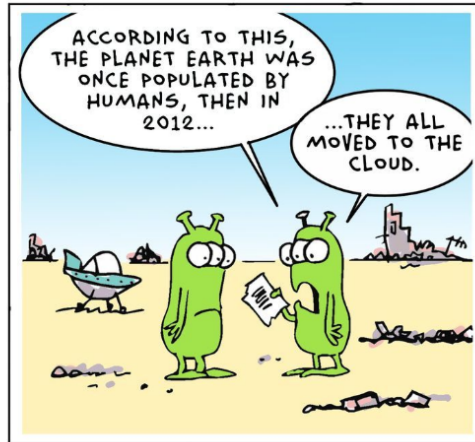


Image source: [AT&T cybersecurity blogs](#)

**Mr. Robert Hawkey,**  
Instructor & Faculty at Computer Science

**Assigned TA:**  
Kamran Awaisi

### **GROUP 16: CLOUDVENGERS**

#### **Members:**

**Ruhi Rajnish Tyagi**  
B00872269, [rh679297@dal.ca](mailto:rh679297@dal.ca)

**Adil Dinmahamad Otha**  
B00900955, [ad842343@dal.ca](mailto:ad842343@dal.ca)

**Saurabh Jayeshbhai Das**  
B00911733, [sr850847@dal.ca](mailto:sr850847@dal.ca)

# INDEX

<b>1. Overview.....</b>	<b>3</b>
<b>2. Feature Description.....</b>	<b>4-8</b>
2.1 Teacher Login and Signup.....	4
2.2 Preparing the quiz.....	5
2.3 Making quiz live.....	6
2.4 Student Login and Signup.....	7
2.5 Student screen interaction.....	7
2.6 Send email to parents/friends.....	8
2.7 Progress reports to teacher.....	8
<b>3. Core Cloud Components Used.....</b>	<b>9-10</b>
3.1 Cloud Formation.....	9
3.2 AWS ECR.....	9
3.3 AWS Lambda.....	9
3.4 AWS S3.....	9
3.5 ECS.....	9
3.6 Amazon Cognito.....	9
3.7 Amazon SNS.....	9
3.8 Amazon API Gateway.....	9
3.9 Amazon Lex.....	10
3.10 Amazon DynamoDB.....	10
3.11 Amazon Secrets Manager.....	10
<b>4. Services Used.....</b>	<b>11</b>
<b>5. Project Timeline.....</b>	<b>12</b>
<b>6. Analysis Of Deployment and Delivery Model.....</b>	<b>13-15</b>

6.1 Describe the deployment model for the cloud-based software system you intend to build. Why did your group pick this deployment model?.....	13
6.2 What are the benefits of your deployment model?.....	13
6.3 What are the drawbacks or risks of your deployment model?.....	14
6.4 Describe the delivery model for the cloud-based software system you intend to build. Explain the trade-offs of this model in comparison to other delivery models (IaaS, FaaS, PaaS, SaaS). For example, how much control do you have over your cloud provided infrastructure?.....	14
<b>7. References.....</b>	<b>16-17</b>

## **1. OVERVIEW:**

Since the world moved to online mode after pandemic, schools and universities also figured ways to arrange online classes, quizzes, and exams. Our app is based on something similar. It a web-based application where teachers can customize the quiz and make it available for students to attempt. Teachers and students both need to sign up to create profiles and login to create quiz and attempt quiz respectively. It is a similar implementation to Brightspace quiz practices.

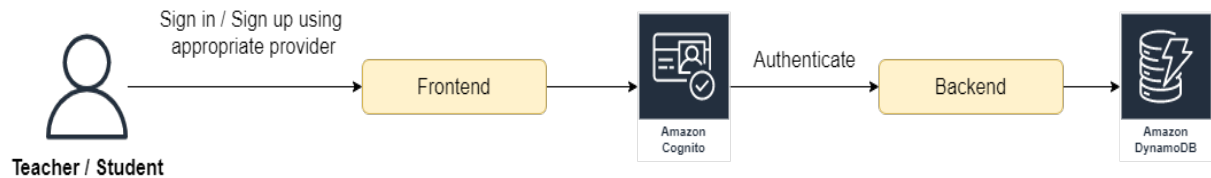
When teachers are new to the application, a chatbot supported by Amazon Lex service can help with tutorials on how to create a quiz, how to make it live and how to look at progress reports of students after quiz is done i.e., how to operate the application. Teachers can see marks of the students after the quiz is attempted by t. them. Also, the quiz data (when teacher makes quiz) will be kept persistent in Amazon DynamoDB. When the quiz ends, by support of Amazon SNS, the system will send email notifications to their parents/guardians to highlight their performance on a particular test.

A NodeJS based Express app is being chosen as the backend service to deliver the above functionality since it works well in an asynchronous way, thus single-threaded architecture works well. The frontend will be served via an Angular app that will provide a user-friendly UI.

## 2. FEATURE DESCRIPTION:

### 2.1 Teacher / Student Login and Signup:

Teachers and students can register into the system if they are first time users and then can sign in later. Upon successful login they can access the functionalities offered to them. This role-based user login will decide what functionalities are offered to the user by the system

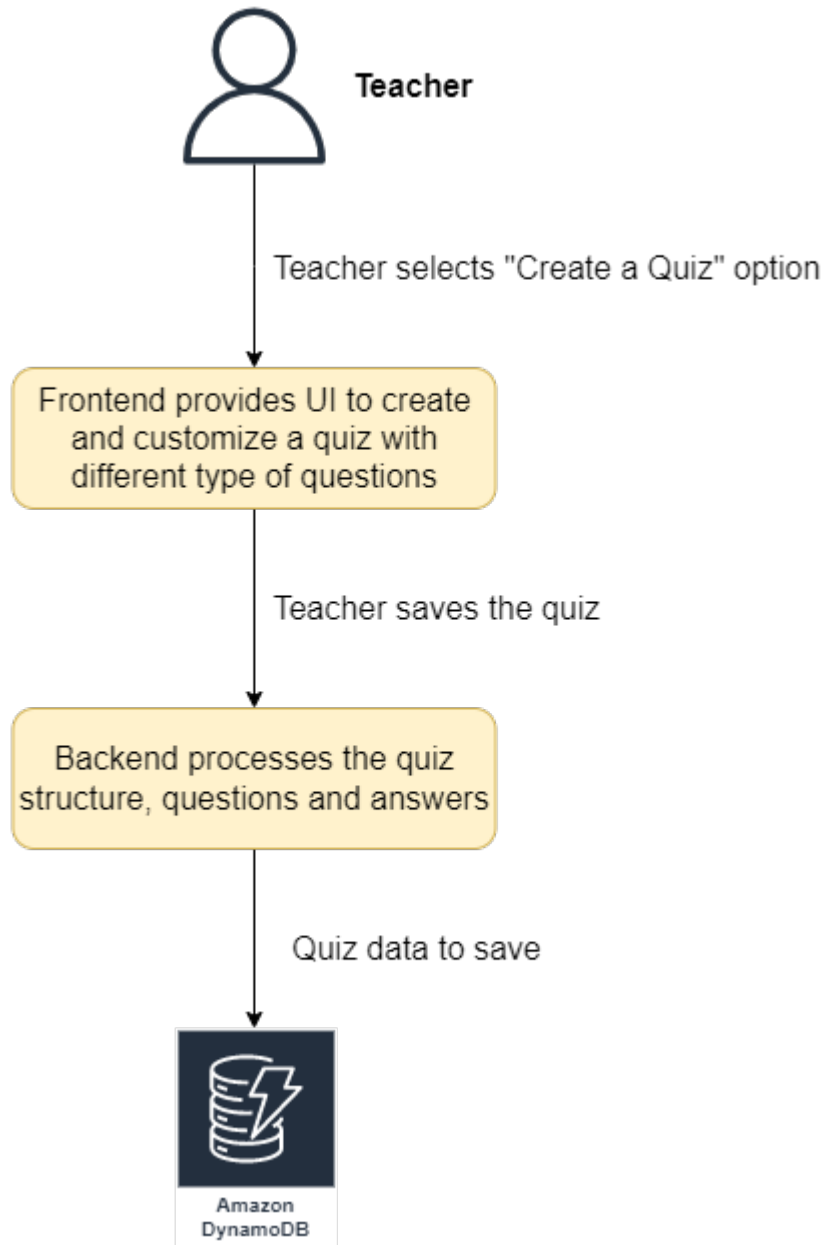


*Figure 1. Login/Registration flow*

AWS Cognito will be used to have users' data in our system.

## 2.2 Preparing the quiz:

Each teacher can make a quiz for the students of their class. For this, they will be presented with a form wherein they can create different types of questions (single correct, multiple correct, short answer). This quiz can be made active as soon as all the questions and their respective answers have been made available.

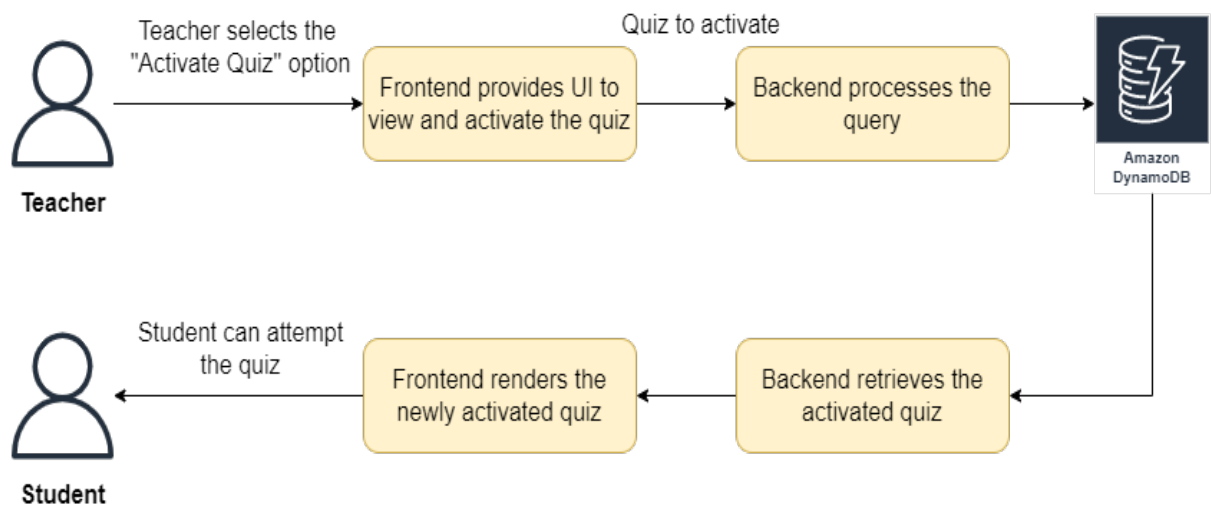


*Figure 2. Flow for creating a quiz*

### 2.3 Making quiz live:

In case the teacher decides to just prepare the questions and answers now in the system as a draft quiz and wants to make the quiz active later then this can be done via this feature. The drafted quiz will show up only to the teacher and not the students of the class. The teacher can then make the quiz active for the students, only then it will be available to make attempts. Only registered students for that class will be able to attempt that quiz.

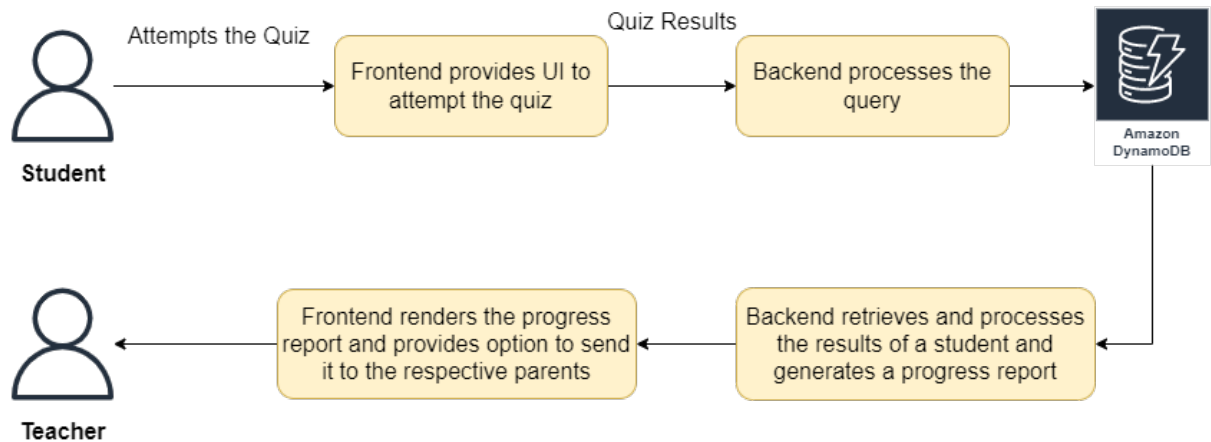
Each quiz generated by the teacher will be in the DynamoDB.



*Figure 3. Flow for activating the quiz*

## 2.4 Progress reports to teacher:

Students attempting the quiz will be evaluated. The results of the quiz will be stored in the system and visible to the teacher. The teacher can see all the quiz results for a particular student of their class using the system.



*Figure 4. Flow for generating progress report*

## 2.5 Student attempting the quiz:

All the quizzes that the teacher has made available for students to attend will show up in the student's dashboard of quizzes. From here the students can attempt the quiz. Upon attempt, the student will be prompted with the quiz reports and this will also send the results to the DB. The teacher will be able to see the marks of each student for their class.



## 2.6 Send progress report as email to respective parents:

Teachers have the option of sending the overall progress reports of all the quizzes attempted so far for an individual student to their parents / guardian. They can also view this overall progress reports themselves.

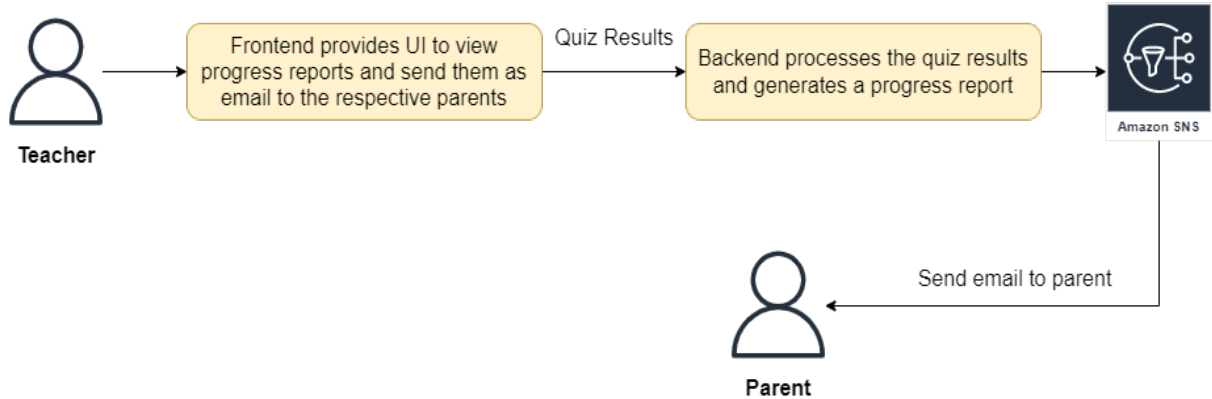


Figure 5. Flow for sending progress report to the parents

## 2.7 Chatbot for guiding the first-time users

The site will also have a chatbot that will guide the users according to their roles about the actions they can perform on the site. When teachers are new to the platform, it may be difficult to navigate the dashboard and view their quiz records, Lex will answer these queries of teachers to help them get familiar with the app.

## 2.8 Notifying when there are new quizzes posted

Students will be able to see the quizzes that they got to attempt but to make the use-case of the app more intuitive, whenever a new quiz is posted by the teachers, all the students would be notified via an e-mail that the quiz is posted and is available to attempt.

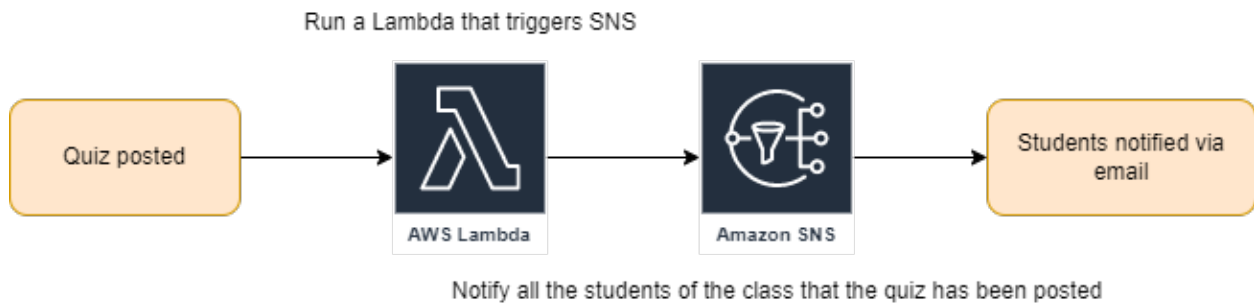


Figure 6. Flow for sending emails to students after quiz is posted

### **3. CORE CLOUD COMPONENTS USED:**

- 3.1 AWS CloudFormation:** AWS CloudFormation is a service to model the AWS infrastructure using code. Using this, we can code the required services, resources, and their dependencies so that we can launch them together using just a single code [1].
- 3.2 AWS Elastic Container Registry:** ECR is the container repository for Docker images that we want to push over to AWS. Images pushed in here can be pulled by ECS to run them as containers and serve the app [2].
- 3.3 AWS Lambda:** AWS Lambda is a serverless, event-driven compute service that allows to run code for almost any form of application or backend service without having to provision or manage servers [3].
- 3.4 AWS S3:** Amazon S3 is an object storage service with industry-leading scalability, data availability, security, and performance. You can use Amazon S3's administration tools to optimize, organize, and configure data access to meet your specific business, organizational, and compliance needs [4].
- 3.5 AWS Elastic Container Service (ECS):** Amazon ECS is a fully managed container orchestration service that makes deploying, managing, and scaling containerized applications simple [5].
- 3.6 Amazon Cognito:** Amazon Cognito allows you to add user sign-up quickly and easily, sign-in, and access management to your online and mobile apps [6].
- 3.7 Amazon SNS:** The Amazon Simple Notification Service (Amazon SNS) is a fully managed messaging service for both A2A and A2P communication. Your publisher systems can fanout messages to a wide number of subscriber systems for simultaneous processing using Amazon SNS topics, such as Amazon SQS queues, AWS Lambda functions, HTTPS endpoints, and Amazon Kinesis Data Firehose. You can send messages to people at scale using the A2P functionality, which includes SMS, mobile push, and email [7].
- 3.8 AWS API Gateway:** API Gateway handles traffic management, CORS compliance, authorization and access control, throttling, monitoring, and

API version management, as well as other activities involved in accepting and processing hundreds of thousands of concurrent API calls [8].

- 3.9 Amazon Lex:** Amazon Lex is a fully managed artificial intelligence (AI) service that lets you design, build, test, and deploy conversational interfaces in apps using advanced natural language models [9].
- 3.10 Amazon DynamoDB:** Web apps generate huge amount of unstructured data. This is because of the increased interaction of user with the system and thus the data is so variable that storing it using NoSQL DBs makes more rational than SQL DBs. DynamoDB is a key-value and document type database that will be used to store all the app data for the project [10].
- 3.11 AWS Secrets Manager:** All database credentials and API keys can be securely saved using this service [11].

#### 4. SERVICES USED:

CATEGORY	SERVICE	APPLICATION FEATURE
Compute	<ul style="list-style-type: none"><li>• AWS Elastic Container Service</li><li>• AWS Lambda</li></ul>	<ul style="list-style-type: none"><li>• AWS ECS will host web application and support microservices.</li><li>• In our scenario, the students can use email notifications provisioned by Lambda to share their test results.</li></ul>
General	<ul style="list-style-type: none"><li>• AWS SNS</li><li>• Amazon Lex</li><li>• Amazon Cognito</li></ul>	<ul style="list-style-type: none"><li>• For students to send emails to their parents/guardians, we need an email service which is provided by AWS SNS.</li><li>• AWS Lex will support the teachers in making familiar to the application use.</li><li>• Cognito will allow signup and login for teachers as well as students.</li></ul>
Network	<ul style="list-style-type: none"><li>• AWS API Gateway</li></ul>	<ul style="list-style-type: none"><li>• AWS API Gateway enables access to AWS services and data stored in the AWS Cloud.</li></ul>
Storage	<ul style="list-style-type: none"><li>• AWS S3</li><li>• AWS DynamoDB</li></ul>	<ul style="list-style-type: none"><li>• AWS S3 will store static files such as images.</li><li>• It will act as a persistent database like storing application related data like user details, marks on quiz and so on.</li></ul>

## 5. PROJECT TIMELINE:

SPRINT	DATES	TASKS	PROJECT COMPLETION
<b>Sprint 1</b>	30 <sup>th</sup> May – 10 <sup>th</sup> June	<ul style="list-style-type: none"><li>• Repository setup for frontend and backend.</li><li>• Spinning up services and connecting them together.</li><li>• Making the frontend for both user roles.</li><li>• CloudFormation.</li></ul>	25%
<b>Sprint 2</b>	13 <sup>th</sup> June – 24 <sup>th</sup> June	<ul style="list-style-type: none"><li>• User registration and login.</li><li>• Quiz drafting and making it live.</li><li>• Sending notifications to students.</li><li>• Students attempting the quiz and seeing their result.</li></ul>	35%
<b>Sprint 3</b>	27 <sup>th</sup> June- 8 <sup>th</sup> July	<ul style="list-style-type: none"><li>• Report generation on teachers' portal.</li><li>• Sending reports to the parents/guardians.</li><li>• Lex chatbot for navigation.</li></ul>	30%
<b>Sprint 4</b>	11 <sup>th</sup> July – 15 <sup>th</sup> July	<ul style="list-style-type: none"><li>• UI enhancements.</li><li>• Final documentation.</li></ul>	10%

## **6. ANALYSIS OF DEPLOYMENT AND DELIVERY MODEL:**

### **6.1 Describe the deployment model for the cloud-based software system you intend to build. Why did your group pick this deployment model?**

Our system uses the “Public Cloud” deployment model which means that it is a publicly accessible cloud environment owned by a third-party cloud provider. In our case, the cloud provider is AWS.

The reason behind the selection of this model is that the cloud provider will be responsible for the management of the physical resources. Also, AWS provides the ability to scale the physical resources as per our needs. Besides this, the pay-as-you-go feature is quite cost effective and flexible compared to buying our own physical IT resources.

Also practically, to deploy a private cloud infrastructure, we would need physical IT resources that are up and running 24x7 which is impossible unless we spend huge amount of money on setting that up. This is also the very reason why can't think of a hybrid cloud (private + public) deployment model even if we wish to do so because that would mean the inclusion of a private cloud. Many enterprises spin up hybrid cloud deployment model using a cloud-bursting strategy to have normal loads on the private cloud and peak loads to the public. But this isn't possible in an academic setting.

### **6.2 What are the benefits of your deployment model?**

Some of the benefits of using the “Public Cloud” deployment model are [12]:

- *Cost Effectiveness:* We only pay for the services we use based on the amount of time, requests made and, memory used.
- *Maintenance:* The cloud provider is entirely responsible for the maintenance of the physical IT resources.
- *Scalability:* Using public cloud allows us to select from a plethora of services and the ability to scale them as per our needs. For example, scaling a server horizontally or vertically to accommodate a large user-base.

- *Reliability and Resiliency:* The services provided by the cloud providers such as AWS are highly reliable and fault tolerant. For example, even if some server nodes go down in one region, the requests will be redirected to the nodes in another region.

### 6.3 What are the drawbacks or risks of your deployment model?

Some of the drawbacks of using the “Public Cloud” model are [13][14]:

- *Data security and Privacy:* Due to multitenancy in public cloud, the services are shared by a multitude of businesses and organizations. This is a cause for concern in terms of data security and privacy. When data access is simple, public clouds don't let users see where their data is stored or who has access to it.
- *Lack of Options:* Even though cloud providers provide us with a lot of options while selecting a service, some companies might have unique needs that may be difficult to configure in that service.
- *Loss of Control:* It's practically out of reach when you outsource to the public cloud. Any IT management configuration and other components are delegated to a group that is not directly involved in day-to-day operations.

### 6.4 Describe the delivery model for the cloud-based software system you intend to build. Explain the trade-offs of this model in comparison to other delivery models (IaaS, FaaS, PaaS, SaaS). For example, how much control do you have over your cloud provided infrastructure?

Our system uses the Software-as-a-Service deployment model. It is also referred to as "on-demand software." Users often access it using a thin client and a web browser. Vendors can manage everything in SaaS, including apps, runtime, data, middleware, OSes, virtualization, servers, storage, and networking, and end users have to utilise it.

Our system provides a service that can be used by schools to provide an online quiz portal that will be managed by the teachers and used by the students. This makes it a reusable service that many schools can use for a subscription fee in the future.

The benefit of using this model, from the end-users' perspective, is that they will get a ready-made product to use without the need to configure or install it.

Trade-offs[15]:

1. Level of control:

- SaaS: consumption-related configuration
- IaaS: full administration
- FaaS: limited administration (create, update, execute)
- PaaS: limited administration

2. Level of functionality:

- SaaS: Access of front-end UI
- IaaS: Full access to IT resources linked to virtualized infrastructure.
- FaaS: Some providers provide you control over the function code, and others give you control over memory restrictions.
- PaaS: Administrative control over IT resources is moderate.

3. Level of maintenance required:

- SaaS: Using and configuring the cloud service
- IaaS: Setting up and configuring virtual infrastructure, managing and monitoring any needed software
- FaaS: Developing, testing, deploying, connecting functions to events/triggers
- PaaS: Developing, testing, deploying, managing cloud services and cloud-based solutions



## 7. REFERENCES:

- [1] "Provision Infrastructure As Code – AWS CloudFormation – Amazon Web Services," *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/cloudformation/>. [Accessed: May 28, 2022]
- [2] "Fully Managed Container Registry – Amazon Elastic Container Registry – Amazon Web Services," *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/ecr/>. [Accessed: May 28, 2022]
- [3] "Serverless Computing - AWS Lambda - Amazon Web Services", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/lambda/>. [Accessed: 27- May- 2022]
- [4] "AWS S3", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html>. [Accessed: 27- May- 2022]
- [5] "Fully Managed Container Solution – Amazon Elastic Container Service (Amazon ECS) - Amazon Web Services", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/ecs/>. [Accessed: 27- May- 2022]
- [6] "Amazon Cognito - Simple and Secure User Sign Up & Sign In | Amazon Web Services (AWS)", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/cognito/>. [Accessed: 27- May- 2022]
- [7] "Amazon Simple Notification Service (SNS) | Messaging Service | AWS", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/sns/>. [Accessed: 27- May- 2022]
- [8] "Amazon API Gateway | API Management | Amazon Web Services", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/api-gateway/>. [Accessed: 27- May- 2022]
- [9] "Conversational AI and Chatbots - Amazon Lex - Amazon Web Services", *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/lex/>. [Accessed: 27- May- 2022]

- [10] “Fast NoSQL Key-Value Database – Amazon DynamoDB – Amazon Web Services,” *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/dynamodb/>. [Accessed: May 28, 2022]
- [11] “AWS Secrets Manager | Rotate, Manage, Retrieve Secrets | Amazon Web Services (AWS),” *Amazon Web Services, Inc.*, 2022. [Online]. Available: <https://aws.amazon.com/secrets-manager/>. [Accessed: May 28, 2022]
- [12] “Public cloud vs private cloud vs hybrid cloud: Microsoft azure,” Public Cloud vs Private Cloud vs Hybrid Cloud | Microsoft Azure. [Online]. Available: <https://azure.microsoft.com/en-us/overview/what-are-private-public-hybrid-clouds/>. [Accessed: 28-May-2022].
- [13] “Advantages and disadvantages of the public cloud,” 165 Halsey Street, 22-Jan-2021. [Online]. Available: <https://www.165halsey.com/advantages-and-disadvantages-of-the-public-cloud/>. [Accessed: 28-May-2022].
- [14] M. Roomi, “5 advantages and disadvantages of public cloud: Limitations & benefits of public cloud,” HitechWhizz, 30-May-2020. [Online]. Available: <https://www.hitechwhizz.com/2020/05/5-advantages-and-disadvantages-risks-benefits-of-public-cloud.html>. [Accessed: 28-May-2022].
- [15] R. Hawkey. (2022). Deployment & Delivery Models [PowerPoint Slides]. Available: <https://dal.brightspace.com/d2l/le/content/222417/viewContent/3008532/View>
- [16] “Diagrams.net - free flowchart maker and diagrams online,” Flowchart Maker & Online Diagram Software. [Online]. Available: <https://www.draw.io/index.html>. [Accessed: 28-May-2022].