**Draw It or Lose It Memory Management Assignment**

**Memory Management:**

In order to properly manage the allocation of memory for the Draw it or Lose it application, it will be important to address the way the different functions of the application will be organized. In order to do this, we must first look at the general structure of the application and break it down into the different compartments of service that the app will provide. For example, the application can be seen to have different compartments for the game authorizer objects and functionality for actually logging into the game client as well as the REST API’s and controllers for both the client and user side of the application to aid in the connection between the backend, server-based side of the app to what the client may actually be interacting with on the front-end side. Because of the compartmentalization that can be seen in the structure of this application, it would be beneficial to use a microservices style architecture to implement the application. This is because “These services are created to serve only one specific business function…Furthermore, they are completely independent of each other, meaning they can be written in different programming languages and use different databases.” (Arsov, 2017) Though different languages will not be needed in the development of this application, it will be important to have the different microservices involved in this application to be structured in this type of architecture.

**Storage Management:**

Because the Draw It or Lose it application requires the rendering and storage of approximately 1.6 GB of data, 200 pictures all approximately 8 MB in size, proper memory management will be crucial to the application’s proper functionality. In addition to the memory required for the basic rendering of these images, it is also important to take into consideration of the storage space the client’s will need to actually run the Draw it or Lose it application on their device. Because of the amount of storage space that is required for these elements of the application, it is suggested to use a serverless approach to dictate how the application stores information. This will allow any backend operations of the application to be managed by cloud vendors rather than having to mange your own servers. With this amount of information to be stored on servers, allow of headaches can be avoided by relying on cloud services to store the information required by the app.

**Comparison:**

The difference primary difference between memory and storage management systems is that memory management deals with the short term purposes of the application such as connecting the client to the different functionalities of the application during runtime whereas the storage management system is responsible for the way the application is actually stored and where the contents of the application can be accessed during runtime, whether it be a cloud storage service, local disk drive, USB drive, etc. When these two systems work together optimally, they make an application’s use by any given client seamless and creates a smooth working environment for the user.

Arsov, Kristijan (2017). Microservices vs. SOA — Is There Any Difference at All? *https://medium.com/microtica/microservices-vs-soa-is-there-any-difference-at-all-2a1e3b66e1be*