

Specs Document - Cloud Team

B Sai Subrahmanyam

Team

- Polisetty Vamsi - Team Member
- Yagnesh Katragadda - Team Member

Overview

The cloud module provides a one-stop storage location for the lab related data. It provides the client(student) a way to submit/upload his/her answer document while it allows the server(lecturer) to view all the submitted answer scripts for the session he/she conducted. It also ensures that the data that the students upload is secure and can be accessed at a later point of time both from the instructor end and the student end.

Required Features

1. To be able to accept answer pdf file's from each of the attendees to the meeting and store them in the cloud.
2. To be able to show all the submission files for all the sessions conducted by a host.
3. To be able to show the attendants all the files he/she submitted in the previous sessions.
4. A UX to help the users to perform these operations.

Introduction

Azure, is a cloud computing service operated by Microsoft for application management via Microsoft-managed data centers. It provides a range of cloud services, including compute, analytics, storage and networking. These applications orchestrate the configuration and operation of virtualized hardware and software on those servers. The orchestration of these servers is what makes Azure so powerful. Azure offers boot-disk-only and full machine VMs where the cloud is boot-disk-only. Both have autoscaling included.

We will be using Azure services by Microsoft to fulfill the requirements of uploading and downloading files as submissions for each session. We will also be storing some metadata related to each session in the cloud for retrieval purposes.

Azure Blob Storage

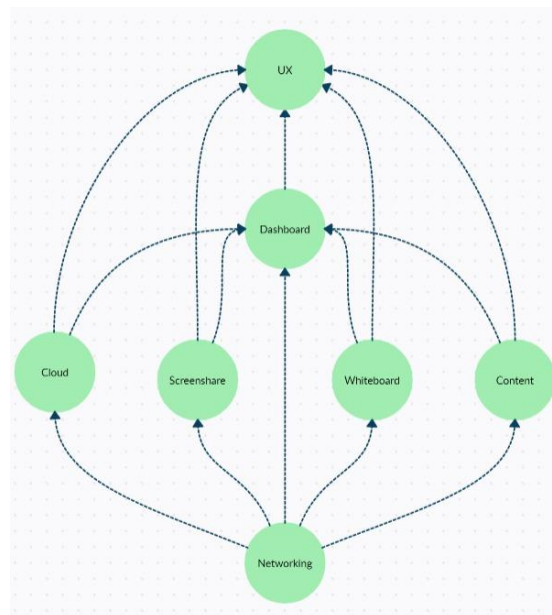
Azure blob storage is an optimized cloud storage service provided by Microsoft especially for the unstructured data. It supports the storage of a wide range of files ranging from images, audio, video, pdfs etc... The access to the Blob storage objects can be done via HTTP/HTTPS requests like Azure Rest APIs etc....

Azure Postgresql Database

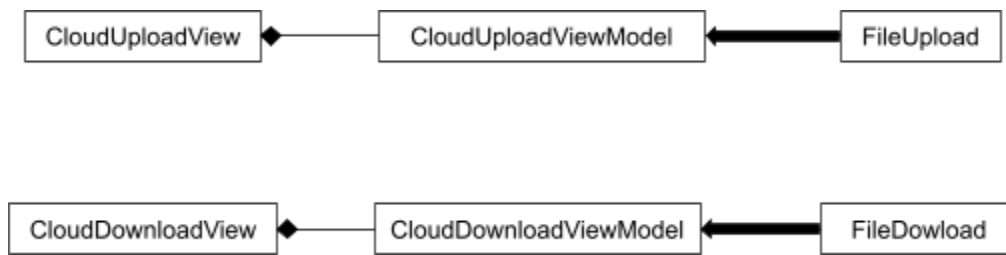
Azure Database for PostgreSQL is a relational database service based on the open-source Postgres database engine. It's a fully managed database-as-a-service that can handle mission-critical workloads with predictable performance, security, high availability, and dynamic scalability. It enables us to use database management functions such as upgrading, patching, backups, and monitoring without user involvement.

UML

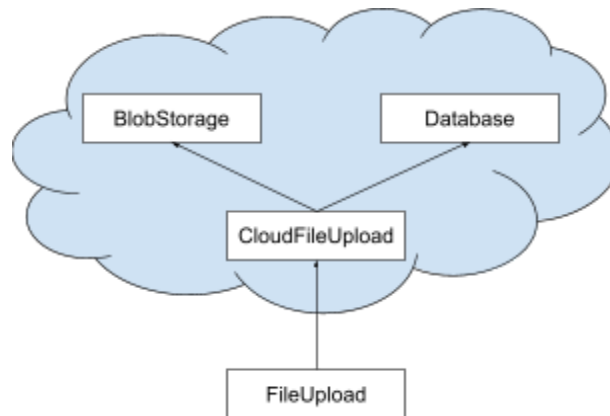
Module Diagram



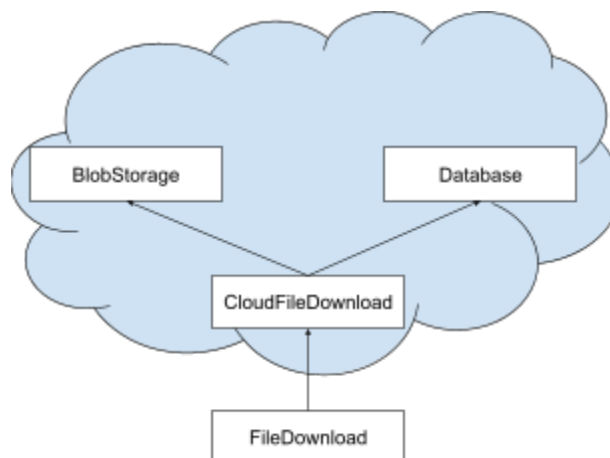
Class Diagram - UX Module



Class Diagram - File Upload Module



Class Diagram - File Download Module



Design

Blob Storage

Azure Blob storage provides us a 3-level hierarchy for storage, i.e., **Account->Container->Blobs**.

1. A storage **Account** provides a unique namespace in Azure for your data. We can access the whole storage account using the URI, <http://plexshare.blob.core.windows.net>
2. A **Container** organizes a set of blobs, similar to a directory in a file system. A storage account can include an unlimited number of containers. We will be having a new container for each new session. We can access the container using the URI, https://plexshare.blob.core.windows.net/<session_id>, where the <session_id> will be replaced with the session id we want to access.
3. A **Blob** is the last level in the hierarchy of the storage, which are the files that we stored in the cloud. A container can store an unlimited number of blobs. We can access each blob/file in the cloud by using the URI, https://plexshare.blob.core.windows.net/<session_id>/<file_name>, where <file_name> is the name of the file we want and the <session_id> is the id of the session to which the file belongs to.

Database

The database is used to store the metadata and respective blob urls, So that we can retrieve required data for the specific whenever they access it. And also to provide consistency and security is to provide the data only which is relevant to the session id provided by the client / user. To reduce the redundancy we use two tables to store all details of the file submissions, session, host and user details.

Session Table

Column name	Id	Session Id	Host Username	Start TimeStamp	End Timestamp
DataType	Integer	UUID	Varchar	Time	Time

Session Table is a database relational schema that maintains the basic details of the session. The session ID is the unique identifier for each session and the respective time periods are stored in the start and the end time stamps. Host Username corresponds to the person who acts as the server for the session. For each new session, a new entry gets added to the session table.

Submission Table

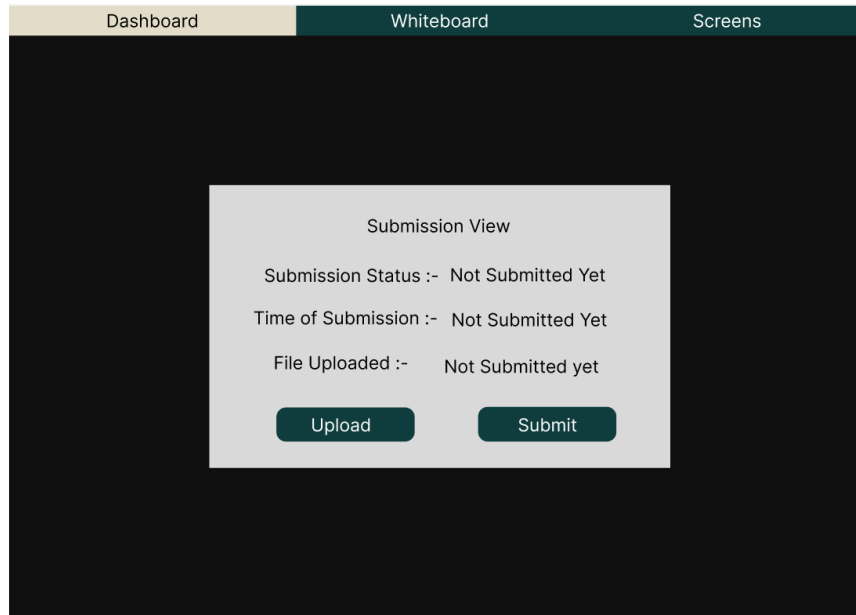
Column name	Id	Session Id	Username	Submission TimeStamp	Blob link
DataType	Integer	UUID	Varchar	Time	Text

Session Table is another relational table that is implemented in the database which maintains the metadata of all the submissions that have been made by the clients(students). The data that is stored in this case are the user who is submitting the data, the session in which the student has made the corresponding submission, Blob link providing the link to the submission file, time at which the user has made the submission and the unique ID to distinguish the submissions made by the users in a particular session.

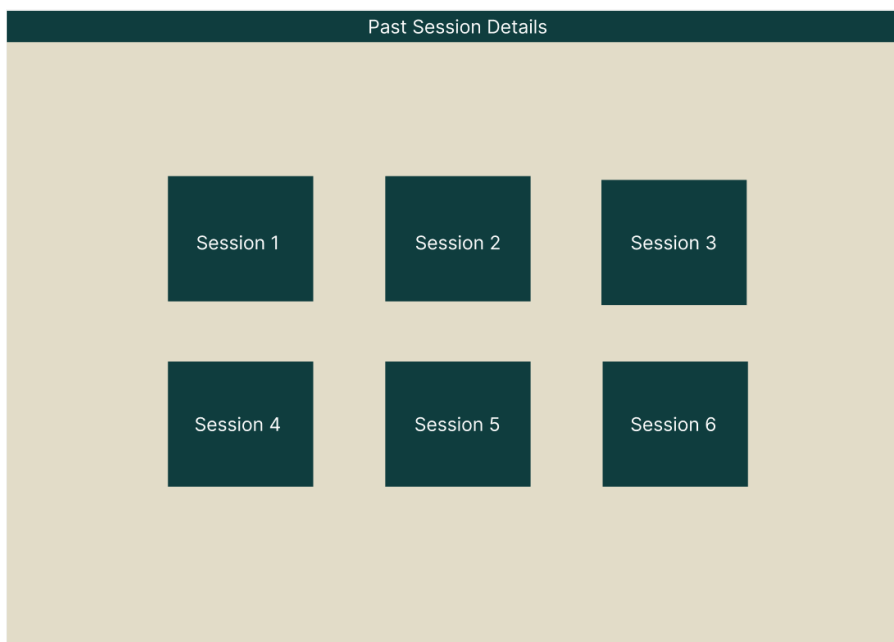
UX

The whole cloud module operations are available for use to users via this UX module. We will be having 2 UX modules each one for File uploading and File downloading.

- The File Uploading module will be available for the clients(students) who are attending a session. There will be an attachment button in the Dashboard UX which on clicking will popup the upload UX. This popup will provide the user with the necessary buttons for uploading and selecting the file to upload. This view also shows the timestamp at which the last submission is made. For current implementation we are considering only one file submission per user.



- The File Download and Viewing module will be available to all the users. There will be a button on the home screen of the application that will redirect to the past sessions page. This page view is different for students and hosts. The host will be able to see all the sessions he/she conducted and on selecting one of the sessions the all the files submitted in that session will be visible along with metadata like the student who submitted. The users who attended sessions will be able to view all his submissions tagged with the session along with the timestamp of when he submitted.



Design Pattern

- Model-View-ViewModel (MVVM) - Using this design pattern to implement Cloud UX. Model has the business logic, ViewModel is where the binding occurs between the view and the model, it captures the events and performs the logic for it and View is the interface that is visible to the user.