FileOcean - CLOUD AGNOSTIC SOLUTION

TEAM: shivahazra_74ea

MEMBERS:
Hrishikesh Baidya - hrishikeshbaidya7@gmail.com
Amit Hazra - shivahazra@gmail.com

THEME: **Cloud-agnostic solution to store and retrieve files**

IDEA/SOLUTION:
We have come up with an api service which can be used as a tool to use storage solutions for multiple cloud providers at a single place without getting complicated to connect each. Using this service to upload and download files to any cloud provider is easy and seamless, and can also generate guest access links for a certain amount of time by the owner of the file.

WHY OUR SOLUTION:
1. We used a Node based framework to build this solution and later it can be broken into a microservice based system to manage different processes efficiently.
2. We focused on requirements for the theme like upload/download files by authorised users, generation of guest access url
3. We have done validation before each request accessing database
4. Used env based system to secure private keys
5. Used file storage as temporary file store before uploading to cloud
6. Used compression to make sure files take less space in the cloud and less cost.
7. We are storing users' cloud credential keys in a database with an encrypted pattern, and no one can get access to it other than the user himself.

API DOC LINK:
https://documenter.getpostman.com/view/3872427/UzBmLRzx

DEPLOYED API SERVER:

https://file-ocean.herokuapp.com/

DATA MODEL DIAGRAM:

Data Model Diagram(Cloud Agnostic Solution)

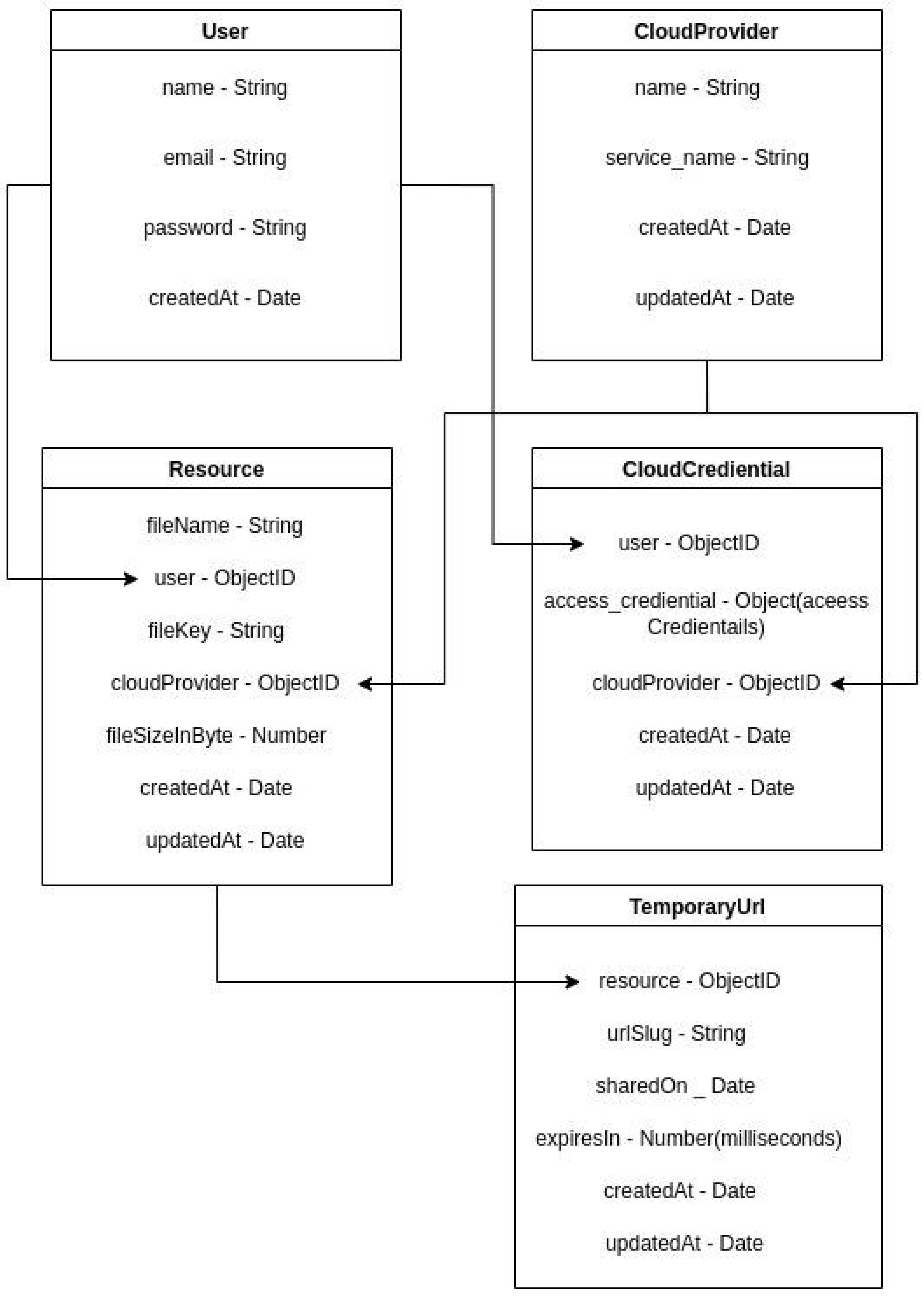

Fig. Data Model Diagram

FLOW DIAGRAM

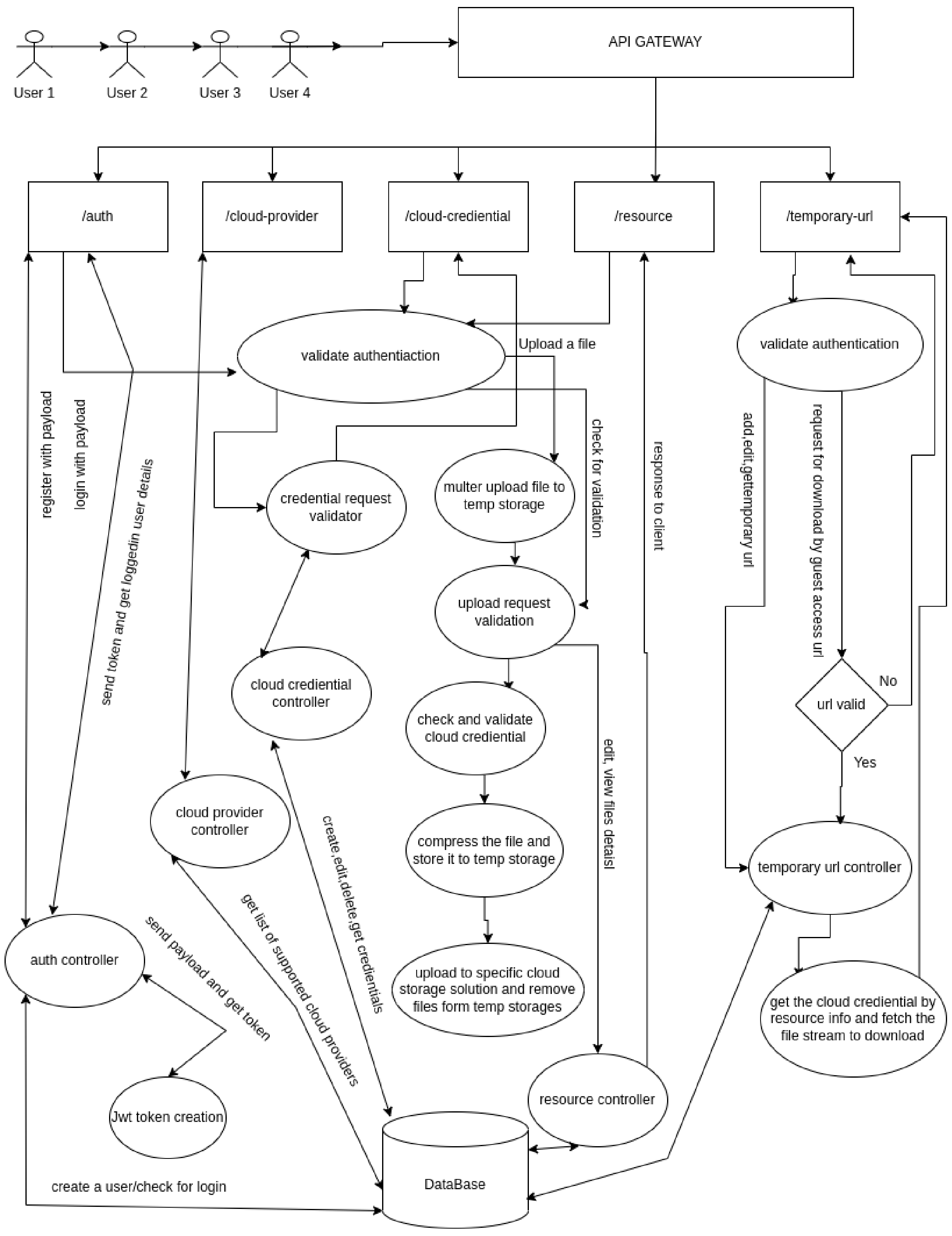
User 1
User 2
User 3
User 4
API GATEWAY
/auth
/cloud-provider
/cloud-credential
/resource
/temporary-url
validate authentiaction
Upload a file
validate authentication
register with payload
login with payload
send token and get loggedin user details
credential request validator
multer upload file to temp storage
check for validation
response to client
add,edit,gettemporary url
request for download by guest access url
upload request validation
url valid
No
Yes
cloud crediential controller
check and validate cloud crediential
edit, view files detaisl
cloud provider controller
compress the file and store it to temp storage
create,edit,delete,get credientials
temporary url controller
get list of supported cloud providers
send payload and get token
auth controller
upload to specific cloud storage solution and remove files form temp storages
get the cloud crediential by resource info and fetch the file stream to download
resource controller
Jwt token creation
create a user/check for login
DataBase

Fig. Rest APi Flow Diagram

BENEFITS:

1. Users can generate credentials for google cloud storage, aws s3, azure blob storage by following the guide document.
2. Users need to save their credentials once and they can seamlessly upload/download to different cloud services without getting into complicated setup.
3. As we are compressing the file before uploading it to the cloud it will reduce storage use.
4. Users can generate a guest access link of some files for a certain amount of time(in minutes).
5. Users can upload multiple files at once.
6. Used Encryption to store user sensitive cloud credential keys.

FEASIBILITY:

1. Most of the required features is completed and working like
   1. Authorised access to files.
   2. Upload File
   3. Download file
   4. Guest User access file by temporary Link
3. Using Node.js + Express.js + Mongodb for developing the API Service.

FUTURE SCOPE:

1. Upload More than 1GB files.
2. Add Files with nested Folder structure.
3. Upload Files with resumable upload strategy.