README – UCBox

Priyanka Prusty
SID :107167659

August 1, 2019

Contents

1 Introduction

This is document contains the instructions to use the application. This document also include proposed future work.

2 App installation & Usage

This section contains the details about the application usage and testing the app.

1. App could be installed in an Android Phone and used as it is. The details to use the app as it is given in the Section 2.1.

2. As of now user need not create any account with Azure to use this application. The proposed billing and account management system is explained in the future work presented in Section 3.

3. For demo and prototype purposes the data is stored in developer's Azure Storage account.

4. To view the files uploaded in the cloud, click on the "Cloud List" button on the homepage of the application. Because it is based out of unmanaged cloud system, Azure portal would display the blobs uploaded by user and not the actual files. To maintain the uniqueness of the blob, `md5sum` of the file path is appended in the blob name.

5. To test this app using your storage account, you need to make some modifications to the source code and put your own connection string. The steps to create a storage account is given in the Section 4. After you have obtained the connection string replace it in the file `com.pprusty.ucbox.AzureConstants`.

2.1 Use the app as it is

1. Drag and drop a file (preferably an image file) to the emulator (file will be automatically saved in download folder of emulator).
2. Install the App (UCBox).
3. Make sure the device is connected to internet.
4. Go to Settings → Apps & Notifications → UCBox → Permissions → Storage (toggle button on).
5. Now open the App(UCBox) → Click on `Browse` button → It will show the files present in device (check if the download folder is open, if not then click on three bars on the left top corner and select downloads).
6. Select on the file → option to upload or delete.
7. Click on `Upload` button → File uploaded.
8. Then click on `Delete` button → file deleted from emulator.
9. Click on the `Back` button → go back to the home screen.
10. Click on `Browse` → file not present in downloads.
11. Click on `Cloud List` button → show the file in cloud as an item.
12. Click on the item → option to download → `Ok` → file downloaded.
13. Click on `Back` button → go back to the home screen.
14. Click on `Browse` button → file present in downloads.
15. Click on `Back` button → go back to the home screen.
16. Click on `Account Info` button → show the number of uploaded files, number of downloaded files, storage used and approximate cost in $.

3 Future Work

This section contains the proposed Future work, which has been thought of during the length of the course, but could not be achieved in the stipulated timeframe and technical challenges.

3.1 Proposed billing & Account management system

Proposal 1: As per the current implementation, every user data is store in one container of the developer storage account. I would implement an account management system and host it in a web server. The user need to create an account in that webserver with credit card details. Once the credit card is verified The user then need to login using the same credential in the app. The app would verify user credentials and would create a unique container in the developers storage account. Every file upload would per user would be done in the respective container. The monthly billing for a user will be based out of the size of the container.

Proposal 2: The Azure account creation and the storage account creation would be embedded in the app. The user would create the account in the app directly and also the storage account before using the app. However, in the limited time, I did not find any existing Azure api which would create an Azure Account along with choice of Subscription, creating a Resource group and finally creating a Storage Account.

3.2 Parallel Upload of Blobs

To leverage huge upload bandwidth provided by Azure, I propose to split larger files into chunk and upload them to Azure in parallel. Similarly, upon download request by user, these chunks would be downloaded from Azure in parallel and merged before saving.

3.3 Metadata Backup

The user should not lose any data if he/she has to format the device or move to another device. This could be achieved by saving the metadata sqlite db in Cloud and updating it at regular check points. If the user moves to another devices or even format existing device, the app could restore the metadata there by prevent loss of data.

4 Using Azure Storage

This section includes the steps to create a Azure account, then a storage account and finally, get the connection string to use in the project.

1. Create an account in `https://login.microsoftonline.com` .
2. Create an account in `azure.microsoft.com`. Here you might need to give the payment details and choose a subscription. With this account, a user can use lot of Microsoft Azure services starting from Virtual Machine to SQL database. This project only requires Blob Storage.
3. After creation of account in azure.microsoft.com, navigate to `https://portal.azure.com/#home`. It is one-stop portal for all azure services.

4. Create a storage account:

   (a) Click on the storage account on the left menu bar and select Add new storage account.

   (b) It might ask for creating a new resource group if not created first.

   (c) Give a name to the Storage account in the name field.

   (d) It would auto-select the "Location" field.

   (e) Select "Standard" performance.

   (f) Select "RA-GRS" replication policy.

   (g) Click on "Review+Create".

   (h) It will take some time for azure to create the Storage account.

   (i) Once the page says " Your deployment is complete", click on "Storage Accounts" on the left pane and you would see the storage account you have just created.

5. Getting the Connection String

   (a) Navigate to `https://portal.azure.com/#home` on your browser and click on the Storage account on the left pane.

   (b) Select the storage account name you have just created.

   (c) Go to settings of the storage account and click on the "Access keys".

   (d) This would open up a new pane on the right and there will two keys.You could choose any one of them.

   (e) Select the key you want to use and copy the connection string that you have got for the chosen key.