Development Assignment: Building a cloud based photosharing service

Barry Denby

February 23, 2022

1 Assignment Information

Course:	BSCH
Stage / Year:	4
Module:	Cloud Services & Platforms
Semester:	2
Assignment:	Programming Assignment
Date of Issue:	2022-03-28
Assignment Deadline:	2022-05-08 @ 23:55 (End of week 12)
Assignment Submission:	Upload to Moodle
Assignment Weighting:	30% of Module

2 Introduction

NOTE: read the whole assignment brief first before implementing it contains very important information

In this assignment you will be tasked with building a cloud based service in which users can upload and organise photos in galleries. They will also be able to share individual images or galleries with other users of the service.

This will require the use of two storage mechanisms:

- the Datastore will be used to store information and metadata about the galleries and images along with information as to who can see what galleries.
- the Cloud Storage Bucket will be used for storing the images themselves

These along with the use of Firebase for authentication is a requirement of the assignment. Usage of other authentication or storage mechanisms will result in an immediate fail.

Users will also be limited in how much they can also store on the system. To keep things manageable and testable we will limit this to 50 MB beyond which a user cannot store anything more unless they remove some images from storage

NOTE: This is an individual assignment. It is not a group assignment. You can discuss ideas/algorithms but you cannot share code/documentation

3 Submission and Penalties

You are required to submit two separate components to the Moodle

- An archive containing your complete Google App Engine Python project. The accepted archive formats are: zip, rar, 7z, tar.gz, tar.bz2, tar.xz. The use of any other archive format will incur a 10% penalty before grading.
- A PDF containing documentation of your code. **If you do not provide documentation your code will not be marked.** Copying and pasting code into a PDF does not count as documentation.

There are also a few penalties you should be aware of

- Code that fails to compile will incur a 30% penalty before grading. At this stage you have zero excuse to produce non compiling code. I should be able to open your project and be able to compile and run without having to fix syntax errors.
- The use of libraries outside the SDK will incur a 20% penalty before grading. You have all you need in the standard SDK. I shouldn't have to figure out how to install and use an external library to get your app to work
- **An omission of a git repository attached to your email address that is registered for GCD will result in your application and documentation not being graded.**
- The standard late penalties will also apply

You are also required to submit as part of your archive a working Git repository.

- When I unpack your archive there should be a .git directory as part of it.
- This should be a fully working **local** git archive. It should not require access to a remote repository
- You are not permitted to upload your work to Github, Gitlab, or any other publicly visible git repository (assignment will be marked as a zero if it is)
- If you need a remote git repository the only permitted one is the college provided Gitlab which can be found at gitlab.griffith.ie
- There must be a minimum of seven commits in the git repository, one per completed bracket.

Very Important: Take note of the groups listed below. These are meant to be completed in order. Groups must be completed in full before the next group will be evaluated. Completed will mean that all tasks in the groups are visible and testable. If a single one is not visible and testable further groups will not be considered. e.g. if there are four tasks in Group 1 and task 3 is skipped or not visible or testable then Groups 2, 3 and 4 will be ignored. Documentation

will be treated separately irrespective of how many Groups you have completed.

You should also be aware that I will remove marks for the presence of bugs anywhere in the code and this will incur a deduction of between 1% and 15% depending on the severity. If you have enough of these bugs it is entirely possible that you may not score very many marks overall. I want robust bug free code that also validates all user input to make sure it is sensible in nature. Please be aware of the major bugs section. If any of these bugs are present in your application you will lose 20% for each one up to a maximum of 60%

4 Plagiarism

Be aware that we take plagiarism very seriously here. Plagiarism is where you take someone else's work and submit it as if it was your own work. There are many different ways plagiarism can happen. I will list a few here (this is not exhaustive):

- Finding something similar online (full implementation or tutorial) that does the same job and submit that.
- Finding something similar online (full implementation or tutorial) and transcribing (i.e. copying it out by hand)
- Working together on an individual assignment and sharing code together such that all implementation look the same.
- Getting a copy of someone else's code and submitting/transcribing that
- Paying someone to do your assignment
- Logging into someone elses Moodle account, downloading their assignment and uploading it to your own Moodle account.

I've had to deal with many cases of plagiarism over the last six years so I can spot it and diagnose it easily, so don't do it. To prevent plagiarism include but not limited to the following:

- Do all your code by yourself
- Don't share your code with anyone, particularly if anyone looks for a copy of your code for reference.
- Don't post your code publicly online. Remember the use of GitHub, Gitlab, BitBucket etc is prohibited.
- If you need to find information online only query about very specific problems you have don't look for a full assignment or howto.
- Change the default password on your Moodle account. The default password can be determined if someone is connected to you through social media or they get one or two details from you.

Be aware that if you submit your assignment you accept that you understand what plagiarism is and that your assignment is not plagiarised in any way.

5 Coding Tasks (80%)

- Group 1 tasks (20%)

  1. Application with a working login and logout service.
  2. Datastore objects that represent a user, gallery and an image.
  3. Ability for a user to add a gallery from the root view.
  4. Ability for a user to delete a gallery from the root view

- Group 2 tasks (40%)

  5. Ability for a user to upload an image to a gallery while in a gallery
     - Images should be stored in a cloud storage bucket
     - This should only accept JPG or PNG images
  6. Ability for a user to delete an image from a gallery while in a gallery
  7. Gallery should show thumbnails of uploaded images arranged in a grid
  8. Image should be loaded and displayed if a user clicks on it

- Group 3 tasks (60%)

  9. While in a gallery a user can request and display the duplicate images in that specific gallery
  10. While in the root view a user can request and display the duplicate images in all their galleries
  11. The total storage size of a user's images is updated whenever an image is added or removed
  12. If a user goes over 50MB of total storage they are prevented from uploading further until they clear their storage

- Group 4 tasks (80%)

  13. Ability for a user to share a single image with another user
  14. Ability for a user to share a gallery with another user
  15. A user should be able to look at images shared with them
  16. A user should be able to look at galleries shared with them

- Major bugs (presence of any one of these will be a 20% reduction in mark up to a maximum of 60%)

  - A user is able to add a gallery with the same name
  - A user can delete a gallery but it deletes the wrong one
  - A user can delete a gallery that is not empty without a dialog first asking the user to confirm that they have images that will be deleted
  - A user deletes an image but the system deletes the wrong one
  - Images or thumbnails do not display
  - A user can delete an image that was shared with them
  - A user can delete a gallery that was shared with them

6 Documentation Brackets (20%)

NOTE: Documentation should be around 1,500 words in length total

1. (0 to 10%): Document every method in your code from a high level perspective. i.e. give an overview of what the method does. Do not copy and paste code you will be penalised for this.
2. (10 to 20%): Document every datastructure and model you have used in your code and why you chose them.