Assignment 3: Building a replica of Instagram

Barry Denby

January 27, 2023

1 Assignment Information

Course:	MSCC/MSCBD
Stage / Year:	1
Module:	Cloud Platforms & Applications
Semester:	2
Assignment:	3 of 3
Date of Issue:	2023-02-13
Assignment Deadline:	2023-05-07 @ 23:55 (End of week 12)
Assignment Submission:	Upload to Moodle
Assignment Weighting:	24% of Module

2 Introduction

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

In this assignment you will be building a simplified replica of Instagram. You will be required to build a system where users can create posts. All posts must have an image attached to it (PNG or JPG only) and some text associated with it. There will be a system here where users can follow each other to see their posts. Evetually this will lead to a timeline where the posts of a user and the posts of users they follow will be merged together. Users will also be permitted to comment on other posts.

Here the main task you will add in is that you will have to deal with and manage multiple data storage mechanisms. The Cloud Storage Bucket will solely be used for managing your images. whereas the Datastore will be used to manage all other information.

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.

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 10% 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 12% for each one up to a maximum of 36%

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. **NOTE: if you are caught participating in either side of such a transaction upto 5 years after you graduate you can be stripped of your degree.**
- 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 ten 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.
- If you need to refer to anything online your only permitted source to reference is StackOverflow.
- Please note that AI tools such as OpenAI or ChatGPT will count as plagiarism. There use is strictly prohibited

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.

Also be aware that if you are caught for plagiarism you will not get another opportunity or a second chance to resubmit the assignment.

If you see the words **"pending review"** in your assignment feedback it is 99% likely that you will be called to a plagiarism meeting.

5 Coding Tasks (80%)

- Group 1 tasks (20%)

  1. Write the shell of an application that has a working login/logout service.
  2. Create models of users and posts using appropriate datatypes.
  3. Add facilities to your models to map who a user is following and who is following this user.
  4. When a user logs in for the first time all of their models should be initialised.

- Group 2 tasks (40%)

  5. Add in a profile page for a user that shows a reverse chronological sequence of their posts.
  6. Add the ability for a user to create a post. The post must take an image upload (PNG or JPG formats only) and a caption.
  7. Add in clickable numbers for the number of followers and following these should redirect to a seperate page when clicked.
  8. The seperate page should either show the list of followers or list of following in reverse chronological order.

- Group 3 tasks (60%)

  9. On the profile of another user there should be a follow/unfollow button that will either follow or unfollow that user when clicked.
  10. Add in the ability to search by profile name. The search should match the beginning of the profile name not the username.
  11. The search should return a list of usernames that should be clickable where it will bring you to that user's profile page.
  12. Create the main timeline page that contains the last 50 posts in reverse chronological order from the current user's and following user's posts. Posts should display in a vertical timeline.

- Group 4 tasks (80%)

  13. Add the ability for user's to add comments on any post but limit this to 200 characters maximum.
  14. Comments should be shown in reverse chronological order. each comment should show the username that wrote it.
  15. Only 5 comments should be displayed unless a user clicks an expand button to show the rest of the comments.
  16. UI design: UI that is easy and intuitive to use.

- Major bugs (presence of any one of these will be a 12% reduction in mark up to a maximum of 36%)

  - None

6 Documentation Brackets (20%)

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

1. (0 to 15%): 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. (15 to 20%): Document every datastructure and model you have used in your code and why you chose them.