# Programming Assignment: Building an Implementation of Dropbox

**Afshan Mehmood**

**March 14, 2025**

## 1 Assignment Information

| | |
|---|---|
| Course: | BSCH |
| Stage / Year: | 4 |
| Module: | Cloud Services & Platforms |
| Semester: | 2 |
| Assignment: | 2 of 2 |
| Date of Issue: | 14-3-2025 |
| Assignment Deadline: | 23-4-2025 |
| 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 building a simplified replica of the Dropbox cloud service using Google App Engine. You will be required to have storage available for users where they can create arbitrary directory structures and can upload files and download files from the service. There should also be some access to some basic file sharing between accounts. It is recommended that the structure of the assignment is closely followed as you will need to get the directory structure working first before you will be able to upload and download files after that.

The score for your assignment will depend on:

- How fully featured, complete, and robust your code is. Along with how well your UI is thought out (80%)
- How well documented your code is (20%)

**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. Copying and pasting code into a PDF does not count as documentation.

There are also a few penalties you should be aware of

- Code without a .git directory will not be corrected.
- Remote repositories are not permitted. You are not permitted to use Github, Gitlab, BitBucket or any other remote repository. Your code will not be corrected if you use one of these.
- Code without documentation will not be corrected.
- A git repository with less than 7 commits will be deducted 5%
- 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.
- The standard late penalties will also apply.

**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 a lot of 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 10% for each one up to a maximum of 30%.

## 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 implementations 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 up to 5 years after you graduate you can be stripped of your degree.
- Logging into someone else's Moodle account, downloading their assignment and uploading it to your own Moodle account.

  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, Git- lab, BitBucket etc. is prohibited.
- If you need to find information online only query about very specific problems you have and don't look for a full assignment or how to.
- 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 then your only permitted source to reference is StackOverflow.
- Please note that AI tools such as ChatGPT will count as plagiarism. Their 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. Application that has a working login/logout service exactly the same as the examples. Any other login system will result in assignment failure. NOTE you must have a

firebase-login.js script setup in the same way as the examples. The same applies for local-constants.py
2. Create collections for a User, Directory, and a File using appropriate data types. When a user logs in for the first time their User document should be created along with a default root directory (path of /)
3. Add the ability for a user to create a directory
4. Add the ability for a user to delete a directory

- **Group 2 tasks (40%)**

  5. Add the ability to change into a directory
  6. Add the ability to go up a directory with the special entry (path of ../)
  7. If a user is in their root directory don't display the special entry (path of ../)
  8. Allow the user to upload a file to the current directory and store it in the cloud storage bucket. If the file already exists ask if they want to overwrite it

- **Group 3 tasks (60%)**

  9. Allow the user to delete a file from the current directory
  10. Allow the user to download a file from the current directory to the local machine
  11. Prevent the deletion of a directory that has files and/or sub directories remaining in it
  12. Add in the ability to detect duplicate files in the current directory (use hashing for this). Highlight the files that match

- Group 4 tasks (80%)

  13. Add in the ability to detect duplicate files in a user's entire dropbox. Display the files and paths that match
  14. Add the ability to share files read only between multiple user accounts
  15. UI design: Well thought out UI design that is intuitive and easy to use.

- Major bugs (presence of one or more of these will be a 10% reduction in marks)
  – It is possible to have two directories of the same name in the same location
  – Deleting a directory removes the wrong directory
  – A file is overwritten without confirming this with the user first.
  – Deleting a file removes the wrong file

## 6 Documentation Brackets (20%)

NOTE: Documentation should be around 1,500 words in length total (20%):
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 as code you will be penalised for this.