Assignment 2: Buliding an application that will replicate Calendar

Barry Denby

January 30, 2023

1 Assignment Information

Course:	MSCC/MSCBD
Stage / Year:	1
Module:	Cloud Platforms & Applications
Semester:	2
Assignment:	2 of 3
Date of Issue:	2023-02-13
Assignment Deadline:	2023-04-16 @ 23:55 (End of week 9)
Assignment Submission:	Upload to Moodle
Assignment Weighting:	16% 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 calendar application that will be capable of storing events for a user on any given day. It will also be possible to share events with other users. It should be possible for a user to schedule events in any given day. If an event is to be shared with multiple users it should be possible to find a common free time on their shared calendars to prevent overlap.

This assignment will require you to use a many to many relationship between calendars and events. A calendar can only belong to one user but can be shared amongst users. an event can belong to one or more calendars. Think very carefully how you will structure your data here as it will directly affect how easy or hard it will be to query this information at a later stage.

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
  1. Application that has a working login/logout service
  2. Datastore models that represent a user, calendar, and an event. Events must have a start time, end time, name, and notes. Calendar must have a name.
  3. When a user logs in by default show the next 7 days of their personal calendar. it should be possible to move forward or back a week in the calendar.
  4. Add in a form/page to add an event to the calendar. it should be able to take in the start time, end time, name, and notes.
- Group 2 tasks
  5. Add the ability to edit or delete an event from the calendar.
  6. Should two or more events overlap either partially or in full the UI should highlight these clashes. How they are highlighted is up to you.
  7. Allow a user to add another calendar to their list of calendars.
  8. There should be the ability to delete or edit a calendar. If a calendar has events on it then confirm before deleting.
- Group 3 tasks
  9. Add the ability to share a calendar with another user. If they accept they should be able to see,that shared calendar.
  10. The owner of a calendar should be able to remove a user from their calendar.
- Major bugs (presence of any one of these will be a 12% reduction in mark up to a maximum of 36%)
  - A user can create two of their own calendars with the same name.
  - A user can add themselves to a calendar.
  - A calendar with events can be deleted without first confirming with the user that they want to delete it.

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.