Assignment 02: Building a Room Scheduler

Barry Denby

February 6, 2024

1 Assignment Information

Course:	MSCC/MSCBD
Stage / Year:	1
Module:	Cloud Platforms & Applications
Semester:	1
Assignment:	2 of 3
Date of Issue:	2024-02-12
Assignment Deadline:	2024-04-14 @ 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 an application that will schedule rooms for use. Your application will support a multitude of rooms and users. As there is potential for overlap in room bookings you will be expected to do validation here to prevent double bookings or rooms. It is expected that your application will be able to do the following things:

- Add and delete rooms. You should not be able to add the same room twice nor should you be able to delete a room that currently has bookings.
- The ability to add and delete bookings for a room. A booking can only be made if it doesn't clash or overlap with other booking for that room.
- It should be possible to view all the bookings for a given day on a room. They should be listed in order of time. When a room is viewed there should be a facility to delete that booking, but only by the user who made the booking.

NOTE that rooms are visible and shared with all users of the system.

The score for your assignment will depend on:

1. How fully features, complete, and robust your code is. Along with how well your UI is thought out (80%)
2. 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 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 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 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 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. 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.
  2. Create documents for a Room, Day, and a Booking. The relationship between these are: A room can have zero or more days, a day can have zero or more bookings. Use an appropriate mechanism to link these documents together. NOTE you are not permitted to use a composite index for these.
  3. Add in a form that permits the user to add a new room.
  4. Add in a display to show the list of rooms that are available to book

- Group 2 tasks (40%)

  5. Add in a form on a seperate page that allows a user to book a room for a given day and time.
  6. Add in a form on the main page that will show all bookings the current user has made on all rooms
  7. Add in a form on the main page that will show all bookings the current user has made on a specific room
  8. When the list of bookings is shown for a user there should be a delete button beside each booking

- Group 3 tasks (60%)

  9. When the delete button is clicked it should delete that booking
  10. When the list of bookings is shown for a user there should be an edit button beside each booking
  11. Editing should be done on a seperate page and the form should be prepopulated with the current data about the booking
  12. Add the ability to delete a room. A room should only be deleted by the user who created it and there should be no bookings on the room.

- Group 4 tasks (80%)

  13. Add in the ability to filter by a single day. It should only show the room bookings for all rooms that fall in that day
  14. When a room is clicked it should show all the bookings for that room
  15. UI design: well thought out UI that is easy and intuitive to use.

- Major bugs (presence of one or more of these will be a 10% reduction in mark)

  - A room with the same name can be added
  - The wrong room booking is deleted
  - The wrong room booking is edited
  - It is possible to cause a clash when a booking is created or edited.

6 Documentation Brackets (20%)

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

1. (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 code you will be penalised for this.