A02: Building a room scheduler

April 5, 2018

1 Assignment Information

Course:	MSCC/MSCBD
Stage / Year:	1
Module:	CC
Semester:	2
Assignment:	Assignment 2
Date of Issue:	2018-04-02
Assignment Deadline:	2018-04-18 @ 23:55
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 must be capable of the following things:

- The ability to 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 bookings 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

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
- The standard late penalties will also apply

Very Important: Take note of the milestones listed below. These are meant to be completed in order. If you skip a milestone or trigger one of the failing conditions the following milestones will not be considered for marking. You should be well capable of producing strong and generally robust software by now. For example if there are six milestones and you fail the third one, then the fourth, fifth, and sixth milestones will not be marked. Documentation milestones will be treated separately from Coding milestones.

You should also be aware that marks will be removed 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. We want robust bug free code that also validates all user input to make sure it is sensible in nature.

Also note that the percentage listed after the milestone is the maximum mark you can obtain if you complete that many milestones without error.

4 Coding Milestones (70%)

1. Write the shell of an application that has a working login/logout service (5%)

2. Generate a model consisting of at least two objects: one to represent the information of a room and another to represent the information of a booking (10%)

3. Build a UI that permits the user to add in a new room. This UI should also permit the user to query all of the rooms for their list of bookings. All rooms should be listed and visible to the user. (20%). The failing conditions for this milestone are:

   - Duplicate rooms can be added
   - Not all rooms are shown

4. Add in facilities for adding a booking to a room. This should be done on a seperate page/url. You are expected to validate a room booking before it is accepted (e.g. is in the future, does not clash etc) (35%). The failing conditions for this milestone are:

   - A booking clash or overlap is permitted
   - bookings can be made in the past

5. When the list of bookings is shown for each room there should be a delete button beside each booking. When this is clicked that room booking should be removed (40%). The failing conditions for this milestone are:

   - The booking is not deleted
   - The wrong booking is deleted

6. Modify the room listing facility to include a filter by date. This should only show the bookings for that particular day for all rooms. (50%)

7. Add in a feature of your own choosing. The complexity and quality of implementation will be considered as part of grading. (70%)

5 Documentation Milestones (30%)

8. Document why you designed the UI the way you did. This should detail your choices in widget layout and position and how they make user interaction easier. You are also expected to document your choice of mathematics for how you rendered the custom view. examples of what I'm looking for would be the following

   - Labels and entry boxes have a small horizontal distance to indicate they belong together.
   - The colour scheme was chosen to avoid the main form of colour blindness and produce high contrast for the visually impaired.

Please don't include instructions on how to operate the App or the UI. (15%)

9. Give a high level description of all methods and models in your Python code, the data-structures used and why. (30%)