

COSC2276/2277 Assignment 1 Specifications

Deadline: 16th January 2019, Wednesday (11:59 pm)

Compulsory Face2Face demo: 17th / 18th January 2019

Marks allocated to this assignment: 30 (worth 15% of the total marks)

To be attempted individually OR in a group of 2

To be submitted via Canvas

1.1 Introduction:

A school is in need for an **A**ppointment **S**cheduling and **R**eservation (ASR) system. They use a manual system currently, where the reservations are made by an admin and maintained on papers. This system is difficult to use and has various flaws. The school has set aside a budget in the New Year for this software. However, they are cautious as they have never dealt with a software firm before and are afraid that the software may not meet their needs.

After much deliberation, the school committee has approached your firm. After a series of meetings, the firm has been asked to submit a working prototype of ASR.

The Project Manager briefs you with a set of requirements and makes it very clear that the ASR prototype is to be a menu-based console application using **.NET Core 2.1/2.2** written in **C#** with a **Microsoft SQL Server** backend.

If the school committee is pleased with what they see, they will offer your firm a chance to create its web equivalent.

1.2 Scope:

ASR console-based prototype performs the following basic functions-

- a) No login-logout required.
- b) Allows both the Staff and the Students to use the system.
- c) Staff can check the room availability, create & delete booking slots and list their booking schedule.
- d) A student can view the available slots (for each of the staff members), book a slot and cancel the booking.

No other (extra) features are required at this stage; these might be added at a later stage.

1.3 Description:

Here is the detailed description of the system-

The system pre-loads the data (staff and the student details, etc...) from the backend database. **You will be provided SQL scripts.** Please make sure you execute these scripts in your database.

Data Validation Rules-

- a) The Staff ID always starts with a letter 'e' followed by 5 numbers.
- b) The Student ID always starts with a letter 's' followed by 7 numbers.
- c) Email for a staff always ends with rmit.edu.au
- d) Email for a student always ends with student.rmit.edu.au
- e) Each ID (Staff and Student) is unique.

The staff can book any of the 4 available venues for a 1-hour student consultation session. The venues are- Room A, B, C and D.

A staff member can check for the availability of a room.

A staff member can create a slot (which will be made available to the student), they can also delete the slot.

A staff member can also produce a list of their booked slots.

Similarly, a student can view a staff's free (available) slots, make a booking or cancel it.

Business Rules-

- a) Each slot must be of 1-hour duration.
- b) A staff member can book a maximum of 4 slots per day.
- c) The slots must be booked between the school working hours of 9am to 2pm and will always be booked at the start of the hour, e.g., 10:00am, 1:00pm, etc...
- d) Each room can be booked for a maximum of 2 slots per day.
- e) A staff member cannot delete a slot once it has been booked by a student.
- f) A student can only make 1 booking per day.
- g) A slot can have a maximum of 1 student booked into it.

Sample menu and output:

Please open the provided **ASR-output.txt** file. It contains the output of the sample solution. Most of the scenarios have been covered (but not all), feel free to use this format or change it. You are under no obligation to stick to this sample.

Tasks and Marks allocation:

Part A

1.1 (2 marks) Implementation of main menu features

1.2 (4 marks) Implementation of Staff features

Part B

1.3 (4 marks) Implementation of Student features

Part C

1.4 (2 marks) Robust data validation* and exception handling**

1.5 (3 marks) Code quality- comments, consistent indentation, code elegance, code repetition & Object-orientation- encapsulation, coupling and cohesion.

Part D

1.6 (10 marks) Implementation of two coding patterns in the code and an analytical justification in README.txt file (why did you use these patterns? what advantages do they offer? any other important point(s))

1.7 (5 marks) Effective use of a code repository (GitHub/BitBucket etc)

* Validation: Check user input including proper format and type. Ensure that business rules are not violated.

** Exception handling: Your program should not crash under normal usage. For example, entering the input “abc” when a number is expected should not crash the program.

Coding Standards:

- Read the C# coding standard from the following website:
<https://docs.microsoft.com/en-us/dotnet/csharp/programming-guide/inside-a-program/coding-conventions>
- Remember that there are too many to be followed, even if you can stick to 6-10 of the standards, it will be a job well done.
- Do not force yourself to implement every OO feature that you have learnt, use the features wisely and if needed.

Restrictions:

No 3rd party plug-ins may be used in this assignment, without prior approval of the head tutor. Only the use of the `System.Data.SqlClient` package is permitted. You are required to write all your own classes following the principles of object-oriented design.

Submission Procedure:

Each submission must include a README.txt file containing your full name, your Student ID, and any other relevant information (if you are working in a group, then please mention the details of your partner).

Zip the entire solution project folder (with README.txt file included) and submit it via Canvas as a single zipped archive.

Late submissions and Extension-related information:

A penalty of 10% per day of the total marks for each assignment will apply for each day a submission is late, including both weekdays and the weekend. After 5 days, you will receive zero marks for that assignment. Email your lecturer (shekhar.kalra@rmit.edu.au) for extension related queries.

Plagiarism:

All assignments will be checked with plagiarism-detection software; any student found to have plagiarised would be subject to disciplinary action. Plagiarism includes

- submitting work that is not your own or submitting text that is not your own
- allowing others to copy your work via email, printouts, social media etc.
- posting assignment questions (in full or partial) on external technical forums
- copying work from/of previous/current semester students
- sending or passing your work to your friends
- posting assignment questions on technical forums to get them solved

A disciplinary action can lead to

- a meeting with the disciplinary committee
- a score of zero for the assignment
- a permanent record of copying in your personal university records and/or
- expulsion from the university, in some severe cases

All plagiarism will be penalised. There are no exceptions and no excuses. You have been warned.

Assignment-related doubts and consultation, workshops:

Please post your queries in the Discussion Board. Talk to your tutors during tutorial/lab session or during HEAD TUTOR's consultation session- there are two consultation sessions available every week.

We are going to run Assignment 1 assistance workshops; a schedule will be posted on the Canvas. Stay tuned!