ASSIGNMENT 2 FOR FIT3037 Semester 2, 2015

1. General Guidelines

The assignment component of this subject is worth 45% of the total overall assessment. This is the second assignment.

The content of all assignment submissions must be essentially your own work. Please read the "Plagiarism" section from the Unit Information Guide for FIT3037.

2. ASSIGNMENT 2

Due date: 15 October, 2015

Marks: 20%

You have been retained as a Software Engineering consultant to develop a formal specification in Z for the following proposed system.

Ashmon Aero Club Recording System

The Ashmon Aero Club operates from a local airport and operates a number of light aircraft for flying by its members. The club wishes to computerise its aircraft flying records so as to increase efficiency and more effectively service the members. Members who wish to hire an aircraft must be members of the club and also hold an appropriate licence and be endorsed for that type of aircraft and the type of flying intended (eg night flying, aerobatics etc).

However to simplify the system for this assignment we will be content to include the following in our proposed system:

Members details

For each member of the club the system will maintain the members number, type of membership, family name, first name and address. The member number will be simply a sequential numbering of the members in order of joining the club. The club has two categories of membership, flying and social. Obviously social members cannot hire or fly the aircraft.

For the flying members of the club the system will keep a record of the pilots licence category which we will simplify to either of solo pilot or student pilot.

Only solo pilots can hire the clubs aircraft, students must have an accompanying solo pilot on all training flights.

The system must keep a link for each pilot to the flights currently or previously undertaken.

The committee of the club are a small number of members elected to represent the members and run the club. The club should keep a record of those members on the committee.

Aircraft

The club owns a number of fixed wing light aircraft each of which is available for hire by the members. Each aircraft is identified by its registration number (VHxxx, where xxx is a three letter alphanumeric code specific to the aircraft). In addition we need to store whether the aircraft is single or dual engined, the number of passengers it can hold and whether it is currently available for hire. The plane may be unavailable for hire for two reasons, the plane is already on a flight or it is being serviced. For each aircraft we will also maintain a link to each flight it has been used for and the current total number of flying hours.

Flights

The system must maintain a sequential record of the flights undertaken by the clubs aircraft and members. For each flight we will store the aircraft registration, the pilots membership number, the number of passengers and when the flight is completed the number of flying hours involved. To simplify the system we will not allow booking of the aircraft, all hiring must be done at the airport just before the proposed departure time for the flight. We will store the date of the start of each flight, we will ignore the time for this assignment.

Operations

Although you need to include all the data specified above in the state space, you do not need to provide all the functions possibly needed by the system, just those required to carry out these specified operations:

- an operation called Init which will initialise the system to having no records of members, aircraft or flights. This operation is obviously extremely dangerous to the system integrity and should not be allowed to be redone once any records have been stored in the system.
- an operation Add_Member that puts a new members details into the system

- an operation New_Flight that checks that the pilot is licenced, the aircraft is currently available and enters the number of passengers for the flight. The user should enter both the pilots membership number and family name as a check that the correct number is entered, only if both correspond should the flight be allowed.
- an operation Flight_Completed which enters the flying hours for a successfully completed flight and does any associated steps needed. The flight in question should be determined from the plane registration number and the date of the flight as entered by the user. You can assume that there is only one uncompleted flight per aircraft for any date. This operation is carried out immediately on the completion of a flight so that if another flight for the same aircraft takes place later on the same date only the later one will be uncompleted.
- an operation Flying_Hours that calculates and outputs the total number of flying hours for a particular pilot as specified by the user. The user should specify the first and family names to identify the pilot.
- an operation Currently_Flying which outputs the aircraft registration and the associated pilot membership number and family name of all flights currently in the air.
- an operation Committee_Flying_on_Date which accepts a date from the user and outputs for all flights undertaken by a committee member on that date the following: the aircraft registration, the associated pilot membership number, first name and family name

You should provide robust versions of each operation that are capable of handling any possible error conditions. For example, if the pilot is not correctly licenced to undertake a flight an appropriate error message must be given.

Keep your solution as simple as possible, extra complexity will not get extra marks. In particular I do not want a solution that attempts to use higher level tuples to store any data. Each piece of information must be stored using simple sets, relations, functions, injections or sequences.

You may assume a type of DATE exists with an associated function GET_DATE() which returns the current date. You can use GET_DATE() within your schemas in the normal way ie this_date = GET_DATE().

3. Marking Criteria

The submissions will be considered for presentation, conciseness and correctness (both logically and notationally). Versions of the operations that are developed using the Z Schema Calculus will be more highly considered than monolithic versions that account for all conditions within a single schema. You should also add a narrative to explain any complicated schemas or logic that you have used.

You are required to produce an abstract formal specification of the system. Note this means that you do not continue on and produce a Formal Design Using Z. In other words do not do the refinement steps detailed in the lecture notes entitled Formal Design Using Z.

Make sure your submission is your own and different to all other students as copying or collusion will not be tolerated.

4. Reading Materials

Weeks 4 and 5 study materials including lecture notes and Section 4 of study Guide 3

Up to Section 3 of "an introduction to Z and formal specifications" by J.M. Spivey (http://people.csail.mit.edu/dnj/teaching/6898/papers/spivey-intro-to-z.pdf) or up to Section 1.3 of Chapter 1 from spivey.oriel.ox.ac.uk/mike/zrm/zrm.pdf

Sections 21.5, 21.6 and 21.7 from Pressman 7th Ed. or Section 28.6 - Formal Methods Concepts and Appendix 3 - Formal Methods (Applying Mathematical Notation for Formal Specification) from Pressman 8th Ed.

Solutions for Problem 2 of Week 6 tutorial problems

Introduction to Z Notation - http://www.youtube.com/watch?v=qfEe9luJmVE

Z Examples - (1) Document Control System (http://staff.washington.edu/jon/z/dcs.html) and (2) Text Processing (http://staff.washington.edu/jon/z/text.html)

5. General Comments

The submission must be presented in a professional, clear and concise manner. If you need further system information please use your initiative and make reasonable and logical assumptions. Questions of a general nature (for example to clarify some part of the assignment requirements) can also be sent to the discussion forums, note these should not in anyway give solutions or parts thereof. Similarly you are encouraged to ask questions about the Z specification language, it is not simple and no students will have encountered it before.

It is anticipated that this assignment will take in the order of 10-15 hours to complete once you have a basic understanding of formal specification concepts and techniques. You have only about 4 weeks to complete this assignment so I strongly suggest that you start early. It is not something you can understand and complete in just the last few days.

Please look into example solutions provided for Problem 2 of Week 6 tutorial problems, Document Control System (http://staff.washington.edu/jon/z/dcs.html) and Text Processing (http://staff.washington.edu/jon/z/text.html). If after some effort on your part you are having difficulty understanding these sample solutions please ask for clarification of particular points via the discussion forum.

6. Submission Requirements

The assignment must be submitted electronically through the Moodle assignment system in Microsoft Word document format, or rtf format, or as a pdf document. If you are unable to provide one of these formats please contact me by email prior to submission to ensure that I will be able to handle the alternative format.

To alleviate any problems with fonts and symbols for the Z specification all students **must** use the Zed truetype font that is available here. Note this zip file has both truetype and Adobe Type Manager files for both Windows and Macintosh machines. Please make sure you use the truetype font. The archive contains a Readme.txt file that explains how to install the font.

Please ensure that you put your name and student ID on a title page of the submitted document.