FIT2107-S2-2020 Assignment 1

Manual Blackbox Testing & Documentation

This assignment will be completed in pairs

Marks:	15% of your final marks.
Due Date:	Week 5, 4/9/2020, 11:55 pm.
Submission:	(i) The assignment submission must be made through Moodle for this assignment **by the due date**. (ii) The submission of this assignment must be in the form of a single PDF file (or a zip file in case you have more than one files). No other file types will be accepted. (iii) The submission should be named using the authcate id of both members, e.g., **nnnn0001-xxxx0001-assignment1.pdf or nnnn0001-xxxx0001-assignment1.zip**. (iv) Only one member in the pair needs to submit it.
Lateness:	Late penalty of 10% per day after the due date, including the weekends.
Cover Sheet:	A completed individual assignment covered sheet is required with the submission, as per the faculty policy.

1. Goal

This assignment will give you an opportunity to develop a test case from a requirement document, what a professional bug report should be, as well as analyzing bugs before reporting them by applying the Black Box testing techniques we have learnt in this unit, on a software.

2. Artefacts

For this assignment, we require all students to download the ***MyCalendar*** application **(.jar file**) provided on the Moodle assessment page under the Assignment 1 heading. As it is a jar file, it can be run from any operating system using the following command line annotation. ***java - jar MyCalendar.jar.*** We require you all to have **Java 8** installed on your machine as the application is not tested with other versions of Java and may not be compatible with other versions of Java – in this case, the teaching staff may not be able

to help you with running the application. The requirement (specification) document for *MyCalendar* is also provided under the Assignment 1 heading on Moodle.

3. Deliverables

In pairs, you must prepare a *test suite* that contains a set of *test cases* for MyCalendar.

Part 1 Test suites using black box testing techniques (7+4=11 Marks)

Test Suite (7 Marks)

The exact format that you write your test suite is up to you; However, it may include at least following information:

1. A 3 - 10 words description making it clear what your test is about.
2. A tester's name and information (Who conducted the test? etc).
3. Number of test cases and their details in a test suite.
4. Inputs provided
5. Output produced
6. Expected output
7. Screenshots (optional but preferable).
8. Rationale (See the Rationale section for more details).

Using the information provided and extracted in the requirement document, devise test cases using black box testing techniques.

Rationale: (4 Marks)

Your test rationale should clearly explain the strategies used to select your test cases and show in detail how the strategies were used to derive each test case. **For instance, if you use the category-partition method, you should show your categories and partitions, explain how you got them, and show how you turned these into test cases.**

For high marks in this section, your test rationale should clearly show how you used systematic techniques, as presented in this subject to identify and select test cases. A test rationale that says, *"we got tests from using the program"* is unlikely to receive satisfactory marks.

Part 2 Traceability Matrix (2 marks)

Use the information in requirements and test cases to draw a traceability matrix. You can use any format you like. But we expect to at least have a requirement number and the

test cases used to verify the requirements. Perhaps, the recommended way is to extract the requirements into numbers from the specification document and check if the test cases are written for that requirements.

Part 3 Bug Reports (2 Marks)

Write an initial bug report for at least **two** major issues you have identified in your system testing (Hint: if you do not identify at least one issue with the system, you haven't done sufficient testing - do some more poking around informally if your tests do not turn something up). Write issue reports (bug reports) according to the following guidelines:

1. A bug reporter's name and information.
2. A 3 - 10 words description making it clear what the bug is about.
3. Steps to reproduce.
4. Screenshots (optional but preferable)
5. Which program or module has an issue?
6. Any other information which you think is necessary to mention, and
7. Justification (Such as why you think it is a bug etc.)

Part 4 Work Breakdown Agreement (Hurdle)

You must submit the **work breakdown agreement** (hurdle) showing each person's time spent on the project, and the contributions as well as showing what test cases are devised and written by the student. Adjustments will be applied to marks if the WBA or any other information indicates significant discrepancies in contributions.

Groups/Pairs

The assignment will be done in pairs (or, if there are uneven numbers and there is no alternative, groups of three). It should be with people in the tutorial you regularly attend and enrolled in. The teaching staff will create groups and it will be finalised not later than week 3 and announced.

Working as a group

You are expected to work as a group on this assignment and contribute very close to an equal amount of work (unless the special consideration policy outlined below applies). In most cases, if this is followed, students will receive equal marks on those aspects to their teammates.

Marking criteria

- Clarity of testing strategies as explained in test rationale

- Appropriateness of testing strategies (NB: appropriateness is not the same as elaborateness).
- Clarity of test suite documentation
- Consistency of test suite with strategies
- Reasonableness of test suite
- Completeness and quality of test reporting
- Completeness and quality of bug reports
- Traceability matrix matches the requirements and the tests.
- Completion of WBA (hurdle)

Please note that "high number of faults found" is NOT a criterion for marking. Finding more than two faults/bugs will not give you additional marks. While you should be aiming to detect deviations from the spec, whether you trigger the specific bug or bugs I have seeded often comes down to luck.

Special consideration

If a student faces exceptional circumstances (serious illness or injury, family emergency etc.) that prevent them completing the assignment, they may apply for special consideration according to the policy and procedure on the Faculty website:

https://www.monash.edu/exams/changes/special-consideration