



School of Science and Technology

COURSEWORK ASSESSMENT

MODULE CODE	:	SOFT30121
MODULE TITLE	:	Advanced Analysis and Design
MODULE LEADER	:	Dr Caroline Langensiepen
TUTOR(S)	:	Dr Caroline Langensiepen Dr Tom Hughes-Roberts
COMPONENT	:	1 of 1
TITLE	:	System analysis, design and implementation
LEARNING OUTCOMES ASSESSED	:	1 (part), 2 (part), 3, 4 (part), 5-7
WEIGHTING	:	70% of the overall module mark
DISTRIBUTION DATE	:	Teaching Week 1
SUBMISSION DATE	:	First submission – Thursday 23 rd November 2017 Main report, code and exe – Friday 16 th February 2018 Demonstration of working code/tests – TBA within 2 weeks of submission
(CLASS TEST DATE	:	In week starting 5 th March 2018)
SUBMISSION METHOD	:	Electronic, via Dropbox on NOW by end of day (ie midnight)
NOTE	:	The usual University penalties apply for late submission and plagiarism. Please consult your student handbook for further details.

I. Assessment Requirements

- The coursework is to be performed in groups. However all contributors in a group will be assessed individually in order to evaluate their contribution.
- Groups must be indicated to module leader via email with subject line "AAD Group" **by Wednesday 11th October 2017, 5pm**. Anyone not in a group by that point will be put into a group. Groups should be of 4 or 5 people.
- Any student submitting any work for this module is deemed to be asserting it as their own, except where explicitly attributed. The deliverable must include a statement asserting this – the format of the statement is specified in the document "Disclaimer". Hand-in will be via the NOW dropbox. End of day means any submission on/before that date will be accepted.
- The first submission will be by end of day on **23rd November 2017**.
- The report must comprise a software requirements specification where the individual requirements conform to the needs of being unambiguous, testable, clear etc. as will be covered in taught module. The document must also contain detailed justifications against the brief for the requirements, with an initial scoping and suggested architecture. The justification can include evidence from interviews, existing products etc.. Though the requirements specification will be written as a common submission from the team, the rest of the report (justification, scoping, evidence etc.) must be **individual** submissions. In other words, the team must work together to define their requirements, but they must work as individuals to clearly explain the reasons for the requirements, the scope of the system, and their suggestion for an architecture. Note that because the system requirements have already been provided, the software requirements should be at a level that they could be provided to developers so that they would know what they were going to build.
- This report will be a maximum of 15 pages (font no smaller than 9 point) and worth **25%** of the overall assignment mark (the assignment forms 70% of the module mark). End of day means any submission on/before that date will be accepted.
- Each student must submit their report in 2 parts – the requirements specification worked out by the group in one file, and their justification, scoping, suggested architecture in another file. This will avoid problems with Turnitin giving false plagiarism scores due to the group work.
- The remainder of the assignment will be handed in by end of day **16th February 2018**. All document limits assume font no smaller than 9. All group documents must be handed in via only one team member's Dropbox.
 1. Documentation of the analysis and design. No more than 20 A4 pages **including** diagrams.
 2. A working implementation of the design - all source and executable files to be supplied electronically (including IDE project files) in order to test for functionality and/or plagiarism automatically. The system must be easily runnable for testing.
 3. A user help document. It must be not more than 3 pages of A4.
 4. An acceptance test plan. This should be no more than 8 pages. We will expect that any of the tests in the test plan may be run successfully on the supplied code. If this is not the case, you must provide a list of the tests that would/would not be passed.
 5. An **INDIVIDUALLY WRITTEN** critique of the product or process (no more than 4 pages). You may choose to perform a detailed assessment against a single criterion (eg usability) or a more general assessment against a basket of criteria. (these should be submitted independently via individuals' Dropboxes).
 6. A contribution form from each of the team members (these should be submitted independently via individuals' Dropboxes).

The degree of effort and thought will be taken into account when allocating marks.

- The mark for the second part will be mainly derived from a demonstration of the assignment products and an interview about their design and structure. The documentation will be used to help

Advanced Analysis & Design

- in this process, and the individual critique used to help distinguish between individuals.
- Implementation may use existing low level classes from wherever the students choose to find them, provided they are also open source and covered by the same GPL as the product to which they will be applied. However such reuse must be fully attributed, and the design documentation should show good reason why the classes were appropriate for this problem. A lack of attribution will be considered as plagiarism and usual action will be taken.
 - Unnecessary complexity for its own sake in an implementation will lose marks. However the design should take account of system issues such as error handling and performance.
 - Documentation for the test plan and test procedures should be as concise as possible, while still being as prescriptive as necessary for a tester. Innovative structure and presentation that assist in this will be credited, so no format is provided here.
 - The following brief summary (next page) is indicative of the expected standards. Actual pieces of work may have a spread of achievement. More detail is provided in the marking grid on NOW. Please note that different sections in the marking grid do not have equal weighting.
 - The Phase Test will occur in the week of **5th March 2018**. It will be 2 hours, and will include a compulsory question related to the assignment in order to cross check that plagiarism has not occurred and to provide more individual assessment related to the coursework. The test is worth the remaining 30% of the module mark.

II. Assessment Scenario/Problem

People who have suffered a stroke are told that they should not drive for a certain period, and are then sometimes assessed for their readiness to drive by a clinical expert. This assessment (which is of cognitive readiness, not their physical limitations) can be very expensive to administer – particularly if it uses the live driving experience as at the Derby assessment centre. A cognitive test was developed at the University of Nottingham that has been shown to be an accurate indicator of fitness to drive (Nouri FM, Lincoln NB. Predicting driving performance after stroke. *BMJ: British Medical Journal*. 1993;307(6902):482-483). The test is in the form of a physical set of board games, and require a tester to record the time taken, add up the passes/fails and then perform a complicated calculation to predict fitness to drive.

The aim of this assignment is to construct a tablet based version of the tests, including the additional path-forming test used by other research groups, which would then generate the results automatically. The information should be logged in a secure database for access by clinicians, since they may have to report the bad news to people considered unfit to drive. In addition, to help clinicians conduct research on the way people handle the tests, it should provide the option to log all mouse movements and clicks while doing the tests. As for the algorithm used to calculate the results, it has been shown that a different calculation can be used with the same tests on people with mild cognitive impairment (dementia) (Lincoln, Nadina B., et al. "The assessment of fitness to drive in people with dementia." *International journal of geriatric psychiatry* 21.11 (2006): 1044-1051). It would therefore be desirable that the algorithm could be switched for different patients relatively simply.

In order that this software can be provided relatively cheaply for research, it should use NO RESOURCES that are not royalty free (preferably conforming to the GPL). In addition, since trials of the existing tests have occurred all over the world, change of language and available road signs (for some tests) should be easy to carry out.

A copy of the paper based tests will be provided for examination. In addition, a draft version of two of the games on a tablet will be provided, along with feedback from clinical users. Please note that it is possible that one or more of the final products may be used in future research. If so, the authors of the tablet application will be acknowledged, but

Advanced Analysis & Design

SOFT30121

4

they will not be able to hold copyright in the application, because it needs to be based on the paper-based games which are copyright of University of Nottingham.

III. Assessment Criteria

Part 1 - Requirements

Level	Expectation
1 st (Excellent) / Exceptional 1 st *	Comprehensive suite of requirements covering all areas and stakeholders. Requirements well stated and show high quality throughout eg they are clear, unambiguous, achievable, testable. Clear understanding of functional and non-functional requirements, and needs of different stakeholders. Thoughtfully and cogently justified, as evidenced by reference to literature, other products, initial needs and other well-founded arguments*. Scope well defined. Proposed architecture clearly appropriate to problem. Proposed solution as evidenced by the requirements list shows innovation and novelty in both its documentation and content, and would leave reader in no doubt as to its capabilities.*
2i (Very Good)	Full set of requirements covering the main stakeholders and their needs. Requirements clearly stated and generally of a high standard, though there may be a few omissions, implying some lesser use cases not considered, or not fully addressed. Some consideration of functional vs non-functional requirements. Justifications well considered, though may not have a comprehensive basis in evidence. Scope mostly clearly defined. Proposed architecture sound. Proposed solution as evidenced by requirements list appears to satisfy needs capably.
2ii (Good)	Set of requirements concentrate on main stakeholder and may only partially address the needs of other stakeholders, or may miss some key requirements needed to achieve the goals of the system. Expression of requirements is mostly clear and there may be some quality issues with some statements. Justifications mostly plausible but not entirely adequately based. Architecture outlined reasonably. Proposed solution as evidenced by requirements list appears to satisfy some of users' needs, while leaving some question over others.
3 (Sufficient)	List of requirements has some substantial holes in terms of coverage of usage and stakeholders. Requirements statements rushed or somewhat incoherent, or stated at too general a level to convey to the reader what it proposes. Justifications resorting to generalised statements rather than evidence based. Limited idea of architecture. Proposed solution as evidenced by requirements list leaves reader in some doubt as to whether it can wholly satisfy users' needs
Fail (Insufficient)	Limited set of requirements statements, leaving large areas uncovered. Some statements may leave reader in doubt as to what was meant due to lack of clarity or coherence. Proposed solution as evidenced by requirements list leaves reader with little confidence that they can address the users' problems/needs.

Part 2 – System

Level	Expectation
Exceptional 1st	Work close to publishable standard. Exceptional understanding shown, with superb attention to needs of the users, producing a product that is excellently designed and built, with all its behaviour and interfaces targeted at the users. Design and implementation coherent, ingenious and of exemplary quality in terms of modifiability, maintenance etc.
1 st (Excellent)	Fully developed (but not verbose) documentation, with excellent use of diagrams. Decisions rationally taken and explained. Extensive and excellent use cases chosen, with pertinent secondary use cases. Design shows excellent choice of abstractions and operations with real innovation and inventiveness. Implementation wholly successful, includes strategic decisions regarding

Advanced Analysis & Design

	error handling etc. Reuse fully justified. Design and implementation show 'elegant simplicity'. Thoughtful and helpful user documentation
2i (Very Good)	Documentation largely complete. Decisions reasonably explained. More limited use cases covered, but well used, showing ability to relate concepts together appropriately. Design reasonable, but may miss some aspects of classes. Implementation reasonably complete though may miss some cases. User documentation reasonably clear.
2ii (Good)	Documentation reasonable but lacking completeness in some aspects. Use cases may not cover appropriate functionality, not pertinent. Implementation partially covers function points, but lacks coherence, and may not have a clear or matching design. User documentation lacking some elements/clarity.
3 (Sufficient)	Documentation attempted but does not fully describe the product. Choice of use cases is fair. Implementation successfully achieves a little, does not match design as documented or design is inadequately described. User documentation basic, tending to descriptive rather than analytical
Fail (Insufficient)	Documentation perfunctory - product description attempted, but process unjustified. Poor choice of use cases. Implementation may not work at all, or at best achieves 1 or 2 minor functions. User documentation perfunctory.

IV. Feedback Opportunities

Formative (Whilst you're working on the coursework)

Meetings will be held where you will present your progress. The tutor will give informal verbal feedback. All must attend at least one meeting where they will present their work.

Summative (After you've submitted the coursework)

You will receive specific feedback regarding your coursework submission together with your awarded mark when it is returned to you. Please be aware that there is a tradeoff between the level of detail of feedback and the speed of return.