Assignment Guidance and Front Sheet

This front sheet for assignments is designed to contain the brief, the submission instructions, and the actual student submission for any WMG assignment. As a result the sheet is completed by several people over time, and is therefore split up into sections explaining who completes what information and when. Yellow highlighted text indicates examples or further explanation of what is requested, and the highlight and instructions should be removed as you populate ‘your’ section.

This sheet is only to be used for components of assessment worth more than 3 CATS (e.g. for a 15 credit module, weighted more than 20%; or for a 10 credit module, weighted more than 30%).

**To be completed by the student(s) prior to final submission:**

Your actual submission should be written at the end of this cover sheet file, or attached with the cover sheet at the front if drafted in a separate file, program or application.

|  |  |
| --- | --- |
| **Student ID** |  |

**To be completed (highlighted parts only) by the programme administration after approval and prior to issuing of the assessment; to be consulted by the student(s) so that you know how and when to submit:**

|  |  |
| --- | --- |
| **Date set** | 22rd November 2021 |
| **Submission date (excluding extensions)** | 07th February 2022  12 noon (mid-day) |
| **Submission guidance** | **Submission requirements**   * You must submit a **project report** **indicating the Student ID number** in the title of the submission, i.e., ***0000001\_Report.pdf***. * The report must be in **PDF** format. * The report must be submitted via Tabula and must not be Zipped. * You must zip all the codes used for your implementation README file, and submit the zipped file **indicating the Student ID number** in the title of the submission, i.e., ***0000001\_Code.zip***. * You **must** check if the report and the zipped file have been uploaded successfully. * You must include the assessment front sheet in your report.   **Report Requirements**   * The report should include a title page, table of contents in the report. * The report should include your student ID. * There is no page limit as long as it fits the total number of words for the report. * The report should follow a logical and well-defined structure with headings and subheadings. * Diagrams in the report should be clearly labelled and well-presented. * Appendices will not normally be marked but they must not include material essential to the argument developed in the main body of the work. |
| **Marks return date (excluding extensions)** | Within 20 working days after the submission deadline. |
| **Late submission policy** | If work is submitted late, penalties will be applied at the rate of **5 marks per University working day** after the due date, up to a **maximum of 10 working days** late. After this period the mark for the work will be reduced to 0 (which is the maximum penalty). “Late” means **after the submission deadline time as well as the date** – work submitted after the given time even on the same day is counted as 1 day late. |
| **Resubmission policy** | If you fail this assignment or module, please be aware that the University allows students to remedy such failure (within certain limits). Decisions to authorise such resubmissions are made by Exam Boards. Normally these will be issued at specific times of the year, depending on your programme of study. More information can be found from your programme office if you are concerned. |

**To be completed by the module owner/tutor prior to approval and issuing of the assessment; to be consulted by the student(s) so that you understand the assignment brief, its context within the module, and any specific criteria and advice from the tutor:**

|  |  |
| --- | --- |
| **Module title & code** | WM393 Software Development Life Cycle |
| **Module owner** | Dr. Jianhua Yang |
| **Module tutor** | Dr. Jianhua Yang & Dr. Young Saeng Park |
| **Assessment type** | Written report & Implementation |
| **Weighting of mark** | 60% of the total module mark |

|  |
| --- |
| **Assessment brief** |
| **Scenario**  Refer to the scenario in the assignment 1.  **Task**  Based on the function you selected in the assignment 1, the assignment 2 extends on the assignment 1 and additionally performs the following two tasks. First, you have to produce an individual report for the function (board) that you selected in the assignment 1. You should include the contents from your assignment 1 in the individual report but needs to improve the contents for further requirements and implementation. The detail instructions for the individual report are described below. Second, you have to implement partial or close to a workable version of software based on the requirements and architecture design. Remember that it is not only the design of user interface, but you should implement some required features for the function you selected. The language decision is up to you, but the implementation should be based on the requirements defined in your individual report you have created. The detail instructions for the implementation are given below.  **[ Individual Report ]**  Here are the instructions of how to produce your individual report:   * You should include the contents for your function defined in the assignment 1 such as the functional requirements, non-functional requirement, user interface, etc. * If necessary, you are free to improve the contents for your function previously defined in the assignment 1. * If necessary, you are free to include some other contents related to your function from the assignment 1 such as the management of functions control, the management of users, non-functional requirements, etc. * To improve the structural view in the architecture design, you must include at least one class diagram for your function. * To improve the behavioural view in the architecture design, you must include at least one sequence diagram for your function. * To include implementation technical review, you must include some important parts of your code and technical description with some screen captures. Remember that the code is not included in the total word counts but the description is included. So, put the code inside a table with a caption in order to make it an easy to recognise. * To verify your implementation code, you must include at least one test case for some functions with testing code and description. Remember that the testing code is not included in the total word counts but the description is included. So, put the testing code inside a table with a caption in order to make it an easy to recognise. * You must include the progression of your work. It is about how effectively you evolve your work considering agility, milestones, etc. * You must include your current status and future work. The current status and future work are about what you have completed based on the functional, non-functional and user interface requirements, and what you need to improve in future. * You must include a short conclusion. * You can add more appendixes if necessary. Remember it is not included in the total word counts. * This is only a proposed report format, but you can change the format if necessary.   + Cover page   + Table of Contents   + Introduction   + Overall description   + Requirements   + Design   + Technical Detail   + Test Design   + Progression of Work   + Current Status and Future Work   + Conclusion   + Appendixes   **[ Implementation ]**  Here are the instructions of how you implement your software and how you submit your code:   * The implementation must be based on the requirements defined in your report. * You should include comments in your code to provide better understanding and analysis of your code for readers. * You should include a README file which includes how to setup and how to execute your software and additional information required for your software. There is no format restriction in the README file such as text only, markdown, json, xml, etc. As long as it is understandable by readers, it will be acceptable. * However, if any problems do occur due to not providing the detail and clear explanations of setup and execution of your software in a README file, it is your responsibility for the problems. * In a case that your software is available online, the README file should include how to access your software. * You should provide a zip file containing all source code but should not include dependencies. Instead, you should provide the instructions in the README file explaining how to install them if you need dependencies for your software. * In a case that it is difficult to execute or access your software, you should include as many screen-captured images as possible in your report or provide the images separately with your code in a zip file. * You must include some test codes in the zip file and also include some description regarding these tests in your report. * It is important that your code has to be well-arranged. If it is necessary, please consider refactoring your code before submitting it. |

|  |  |
| --- | --- |
| **Word count** | 2500 words ± 10%. Word count is defined as the number of words contained within the main body of the text which include titles, headings, abstracts, summaries, in-text citations, quotations, and footnotes.  Items excluded from the word count are acknowledgements, tables of contents, a list of acronyms, meeting notes, a glossary, a list of tables, or figures.  Exceeding the work count:  For more than 10% up to and including 20% a deduction of 10 percentage points will be applied. For more than 20% up to and including 30% a deduction of 15 percentage points will be applied. More than 30%, The work will be assigned a grade of 0. |
| **Module learning outcomes (numbered)** | 1. Demonstrate a sound understanding of a range of software process models and symbolic representations. 2. Discriminate scenarios where different design patterns and software testing strategies can be applied, and critically evaluate these patterns and testing solutions. 3. Apply the range of software tools used tools for configuration management, version control and software build. 4. Discriminate the key concepts and techniques used in the Agile Manifesto and Scrum, carefully design and critically evaluate project plans using these techniques. 5. Distinguish current and emerging XP, Lean, and Kanban values, principles, and practices, use these techniques to analyse and optimize process workflow. |
| **Learning outcomes assessed in this assessment (numbered)** | LO3, LO4, LO5 |
| **Marking guidelines** | First class report is expected to be exceptional works by demonstrating excellent analysis of the project scenario and task, excellent specification of functional and non-functional specification, excellent design of system architecture, and excellent partial or close to a workable version of software.  Second class report is expected to be works by demonstrating good analysis of the project scenario and task, good specification of functional and non-functional specification and excellent design of system architecture, and good partial or close to a workable version of software.  Report that presents a limited quality work by demonstrating some relevant knowledge with partially evident critical thinking will be deemed as third class.  Work that is below the standard required for the appropriate stage of an Honours degree will be deemed as fail.  \*\* Detailed marking rubrics can be found in the assignment brief. |
| **Academic guidance resources** | **How to seek further help**  Students are strongly advised to ask tutors via Moodle forum  <https://warwick.ac.uk/services/library/students/your-library-online/>  Numerous online courses provided by the University library to help in academic referencing, writing, avoiding plagiarism and a number of other useful resources.  **Referencing**  Follow the University of Warwick referencing guidelines, found via the links:  <https://warwick.ac.uk/services/library/students/referencing/referencing-styles>  <https://warwick.ac.uk/fac/soc/al-archive/leap/writing/referencing/intext/>  Should you experience difficulties likely to seriously impact your ability to complete any module work, please see the website section for Mitigating Circumstances and Reasonable Adjustments at:  <https://warwick.ac.uk/services/aro/dar/quality/categories/examinations/policies/u_mitigatingcircumstances/> |