Bournemouth University

Faculty of Science and Technology

Department of Computing & Informatics

Unit Title: Software Engineering (COMP5054)		
Assessment Title: Software Engineering Team Project		
Unit Level: 5	**Assessment Number:** 1 **of** 2	
Credit Value of Unit: 20	**Date Issued:** 29/09/2022	
Unit Leader: Suzy Atfield-Cutts	**Submission Due Date:** 16/12/2022	**Time:** 12:30 PM
Other Marker(s): N/A	**Submission Location:** Turnitin (+ large file submission box)	
Quality Assessor (QA): Benjamin Gorman	**Feedback Method:** Brightspace	

This is a group assignment which carries 50% of the final unit mark.

ASSESSMENT TASK

To conduct a Software Engineering (SE) project of choice as part of a team, that follows software engineering principles. Your team will produce a product with supporting documentation (report) in accordance with a software engineering model. Your team is free to choose the topic of your project.

REPORT

The report should be a maximum of 4,000 words long (excludes references, tables and figures), and contains at least:

- The following documentation reflecting the development of the chosen product:
  - Requirements Specification
  - Design
  - Implementation information
  - Test Plan
- Evidence of applying the chosen SE model in the form of references to process and planning documentation, such as (the following are examples only as not everything will apply to all SE models):
  - Prioritisation of requirements
  - Gantt Chart defining phases or sprints, with updated versions
  - Change management process or system used eg VCS
  - Risk assessment
  - Project Management tools used
  - Any other information regarding the management of the project.

PRODUCT

All team members should participate in a discussion to select the product to be developed. When considering the topic and complexity of the product, consider the skills and interests of **ALL** team members.

Suitable types of products include:

- An original web or mobile application.
- A re-engineering of an existing software product.

SCOPE & CRITERIA

As a minimum:

- The scope of the requirements specification should at least include The create, review, update, delete (CRUD) functions of at least two types of records eg products and users.
- On creation of records there must be at least 10 inputs by users across the two types of records. At least four of these should be of specified formats e.g., date, time or email, and of different types i.e.: two date attributes count as one format. This will give you scope to evidence good testing practice in your test plan.
- User interfaces should not be left as default formats and application of guidelines should be documented.

Examples of applications built in recent years by student teams (or part thereof) include:

- Personal library book tracker - online
- Personal trainer/gym booking system – online
- Fitness exercise plan - online
- Meal planner - phone app
- Weather forecast site - online
- Sports fans social media site - phone app
- Sports fans live results - online
- Stock management system - online

The final product should be in working order. Where a product is incomplete the working section should consist of functionality documented as being of highest priority.

To successfully produce a working product (or part thereof), your team will manage the project by selecting and following an SE model. You will be supported and guided through this process during lab sessions.

PROFESSIONAL APPROACH

Each week lectures and labs will include knowledge and exercises to help you develop the skills needed to build the project. You will demonstrate a professional approach to software engineering by systematically completing lab tasks each week, regularly contributing to the development of the project, and by following the team project development plan. With your team, you will attend weekly meetings with staff during your lab sessions. During these meetings team progress will be recorded. Any questions you have may also be raised here.

TEAMS AND DEADLINES

Each team should aim to consist of **FOUR students.**

Teams will be finalised by 9am on 10th of October 2022 by self-enrolment on Brightspace. Students not self-enrolled into a team by the deadline will be allocated to a team. Teams of fewer than 4 students may be allocated additional team members at that point.

Team members are expected to agree on the product they intend to develop and inform staff **by 5pm, 14th October 2022**.

Summary of deadlines	
10th October 2022 - 9am	Self-enrolment in Teams on Brightspace
14th October 2022 - 5pm	Teams agree on the product and inform staff
16th December 2022 -12.30pm	Submission deadline

SUBMISSION FORMAT

The submission consists of three parts: 2 as a team and 1 by individuals.

1. **Team Submissions**
   IMPORTANT: In the spirit of the team project, no one student can be made responsible for the group submission. The responsibility for ensuring the submission is completed lies equally with all team members, as do the consequences of late or non submission.
   One student will submit on behalf of the team to the Brightspace group submission, and will provide an electronic submission of the following two items:
   - **Report** - a maximum of 4000 words excluding references, tables, figures, diagrams and appendices, to be submitted as PDF or MSWord document via TurnitIn on Brightspace.
   - **Product code** - to be submitted as a .zip file **(not .rar or any other compressed format)** via a large file submission box on Brightspace.

2. **Individual Submission**
   Each Individual student will submit a Individual Contribution Form (see below), as a PDF or a Microsoft Word document via Brightspace.

MARKING CRITERIA
The following criteria will be used to assess the assignment:

Criteria		Available Marks (/100%)
Software Engineering project report including the following sections:		
	Requirements Specification • Should contain detail sufficient for the type of project and align with the chosen SE model.	25%
	Design and Implementation • Should contain Implementation details to enable the set up of the product for review by staff. • Should contain sufficient details for the project to be maintainable by other software engineers.	10%
	Testing • Should contain at least a test plan based on requirements and an overview of conducted tests.	20%
	SE Model and Structure • Should reflect a professional approach, structure, style, and application of the chosen software engineering model.	20%
Software Engineering product • Complexity and completeness of the product. • Quality of source code and comments.		25%

A rubric as a detailed guide to quality for this assessment is attached in **Appendix B**.

Please, refer to Bournemouth University's Generic Assessment Criteria available on Brightspace that will guide you in more general terms, regarding the expected quality of your work.

INTENDED LEARNING OUTCOMES (ILOs)
This unit assesses your ability to:

1. Demonstrate an understanding of the nature of software and the software engineering process.
2. Demonstrate critical awareness of the process of developing and enhancing software applications.
3. Demonstrate an appreciation of the role of quality standards in the development of professional software applications.
4. Demonstrate an appreciation of characteristics of software projects.

QUESTIONS ABOUT THE BRIEF
Your questions regarding the brief will be handled during lectures, labs, on Teams and in Brightspace team discussion boards, by the teaching team (Dr Suzy Atfield-Cutts).

Unit Leader Signature Suzy Atfield-Cutts

PENDING EXTERNAL EXAMINER REVIEW
This version of the coursework assessment brief has been approved internally by Bournemouth University. However, **please note** that it is yet to be approved by the unit's external examiner. Therefore, there may be some minor changes made to this assessment brief as a result of external examiner comments.

A **new version** of the brief will be published by the Unit Leader and an announcement will be posted on Brightspace to notify you once the final version has been made available.

Group Assessment - Individual Contribution Form

Department of Computing & Informatics

Unit Name: Software Engineering (COMP5054)
Element: Coursework 01 - Submission

This is an **optional form**, meaning that if your group believes all members have contributed **fully and equally** towards to the group element of the assessment, there is **no requirement to complete this form**. The below form should be completed **individually** by **ALL** members of the group, for the Unit Leader to consider when marking, if there were significant issues with engagement or contribution from one or more group members.

Please Note: In the event where forms are submitted, the following will apply:

- If **only part of the group** submits this form, the Unit Leader reserves the right to adjust any member's marks accordingly, based on the evidence **detailed in the form below**.
- Any group where the forms submitted provide **insufficient information** to support Unit Leader to make a judgement, the Unit Leader reserves the right to award every group member an **equal mark**.
- Any group where the forms submitted provide **clear contradictory information**, the marks awarded will be at the discretion of the Unit Leader.

IMPORTANT: Every member of the group takes **full responsibility** to work collaboratively and professionally as a group. The Unit Leader will provide guidance on the next steps if there is a dispute between group members, Unit Leaders would not normally be expected to intervene to resolve disputes between group members.

Your Name and Student Number	**Contribution** (Please Circle)			**Comments** (Required when less than full contribution)
	None	Partial	Full	

Other Group Member Names and Student Numbers	**Contribution** (Please Circle)			**Comments** (Required when less than full contribution)
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	
	None	Partial	Full	

Your role within the group and actions/tasks which you completed or significantly contributed to:
Any other comments you would like to make on your group, particularly anything related to the group work:

Student Signature: _____ **Date:** ____ / ____ / _____

Help and Support

Undergraduate Coursework Assessments

If a piece of coursework is not submitted by the required deadline, the following will apply:

1. If coursework is submitted within 72 hours after the deadline, the maximum mark that can be awarded is 40%. If the assessment achieves a pass mark and subject to the overall performance of the unit and the student's profile for the level, it will be accepted by the Assessment Board as the reassessment piece. This ruling will apply to written coursework and artefacts only; This ruling will apply to the first attempt only (including any subsequent attempt taken as a first attempt due to exceptional circumstances).
2. If a first attempt coursework is submitted more than 72 hours after the deadline, a mark of zero (0%) will be awarded.
3. Failure to submit/complete any other types of coursework (which includes resubmission coursework without exceptional circumstances) by the required deadline will result in a mark of zero (0%) being awarded.

The Standard Assessment Regulations can be found on **Brightspace or via** https://www1.bournemouth.ac.uk/students/help-advice/important-information (under Assessment).

Exceptional Circumstances

If you have any valid **exceptional circumstances** which mean that you cannot meet an assignment submission deadline and you wish to request an extension, you will need to complete and submit the online Exceptional Circumstances Form together with appropriate supporting evidence (e.g. GP note) normally **before the coursework deadline**. Further details on the procedure and links to the exceptional circumstances forms can be found on **Brightspace or via** https://www1.bournemouth.ac.uk/students/help-advice/looking-support/exceptional-circumstances. Please make sure that you read these documents carefully before submitting anything for consideration. For further guidance on exceptional circumstances please contact your Programme Leader.

Referencing

You must acknowledge your source every time you refer to others' work, using the **BU Harvard Referencing** system (Author Date Method). Failure to do so amounts to plagiarism which is against University regulations. Please refer to https://libguides.bournemouth.ac.uk/bu-referencing-harvard-style for the University's guide to citation in the Harvard style. Also be aware of Self-plagiarism, this primarily occurs when a student submits a piece of work to fulfill the assessment requirement for a particular unit and all or part of the content has been previously submitted by that student for formal assessment on the same/a different unit. Further information on academic offences can be found on **Brightspace** and from https://www1.bournemouth.ac.uk/discover/library/using-library/how-guides/how-avoid-academic-offences

Additional Learning Support

Students with **Additional Learning Needs** may contact the Additional Learning Support Team. Details can be found here: https://www1.bournemouth.ac.uk/als

Primary Research (Undergraduate Levels)

You should not be conducting any primary research (i.e. carrying out an investigation to acquire data first-hand, for example, where it involves approaching participants to ask questions or to participate in surveys, questionnaires, interviews, observations, focus groups, etc.) unless otherwise specified in the brief. However, if there is a genuine requirement to collect primary research data you will require ethical approval before doing so. In the first instance, please discuss with the Unit Leader. The collection of primary data without appropriate ethical approval is a serious breach of Bournemouth University's Research Ethics Code of Practice and will be treated as Research Misconduct.

IT Support

If you have any problems submitting your assessment please contact the IT Service Desk - +44 (0)1202 965515 - immediately and before the deadline.

Disclaimer

The information provided in this assignment brief is correct at time of publication. In the unlikely event that any changes are deemed necessary, they will be communicated clearly via e-mail and Brightspace and a new version of this assignment brief will be circulated.

Software Engineering (L5) Assessment – Coursework – Rubric (Appendix B)

Section	<30	30-40	40-50	50-60	60--70	70-80	80-90	90-100
Report overall	Report missing	Lacks any / has little structure Content is minimal or very vague/ summarised	Muddled structure or lacks headings. Informal style throughout Content rarely relevant and some missing	Structure attempted but mostly informal and not always useful Content mostly relevant to headings	Some sections merged or not named well. Some informality in style Content occasionally not in correct section	Structure is clear and well named. Occasional informality. Content relevant to headings	Structure is clear and well named. Good professional style Content highly relevant to headings	Well considered structure Excellent professional style Content all relevant to headings
SE Model and Structure	Unclear which SE model was followed. Lacks most key documentation.	The model chosen is unclear or applied very loosely.	The model is implied as a mix of various models, or loosely applied. Lacks documentation of significance.	Model is implied and applied with unexplained deviations. Lacks formality in documentation for some elements	Model is implied and applied with some minor deviation. Includes a range of formal documentation	Model is applied well in the main. All relevant documentation included	Model is closely applied. All relevant documentation included and of good quality	The model chosen is clearly applied. Demonstrates excellent Project Management skills through thorough documentation
Requirement Specification	Missing requirements specification.	Few requirements or vague / summarised. Informal style and unusable.	Incomplete. Some requirements are vague or summarised. Informal style. Some parts are useable.	Some errors / ambiguity / missing requirements Informal style but still useful.	Minor errors / ambiguity. Clear and formal structure. Some inconsistencies.	Unambiguous requirements with few errors. Clear, formal structure. Minor inconsistencies.	Unambiguous requirements. Clear, formal structure. Occasional inconsistencies.	Unambiguous and thorough requirements. Clear, formal, consistent structure.
Design and Implementation	Design missing Implementation plan missing	Design does not have enough information to be useful. Implementation lacks information to be useful to other engineers.	Design lacks some information required by a developer. Implementation missing a lot of information but could guide an experienced engineer.	Design covers main requirements. Rarely references requirements. Implementation missing some information but could guide an experienced engineer.	Design covers many requirements and sometimes references requirements. Implementation lacks several details but is useful for other engineers.	Design covers most requirements and often references requirements. Implementation lacks a few details but is useful for other engineers.	Design covers all requirements and references requirements with occasional errors. Implementation is detailed.	Design covers all requirements and references requirements. Implementation is formal and detailed
Testing	Test plan missing.	Very few tests Only for expected use. Incorrectly applied. Test plan missing essential information. Tests not run. No reference to requirements.	Some useful tests. Most for expected use. Reruns of tests with different data recorded as a single test. Missing essential columns. Tests not run. No reference to requirements.	Mostly useful tests. Some for unexpected use. Runs of a test with different data merged as a single test. Missing data /misuse of test plan Most tests not run. Little reference to requirements.	Most requirements tested. Several for unexpected use. Each run of a test with different data is recorded as a seperate test. Occasional misuse of columns. Most tests run and reference requirements.	All requirements tested. Many for unexpected use. Correct recording of tests. Wording could be improved. All tests run once. Some have been rerun. Some to reference requirements.	All tests have been run and accurately recorded. Tests all cross reference to requirements Most failing tests have been fixed and rerun.	Provides evidence that all requirements tested, and failing tests have also been rerun following fixes. Accurately recorded. Tests all cross reference to requirements and reruns of tests refer to original first run.
Product	Product missing	Minimally functional. Detrimental code style. No comments.	Partly functional. Poor code style. No comments in code.	Mostly functional. Eg out of time /technical difficulties. Acceptable code style and documentation (comments).	Mostly functional. Eg out of time /technical difficulties. Acceptable code style. Informal documentation (comments).	Product functional. Good style of code. Some formal code documentation (comments).	Product functional. Good code style. Mostly formal code documentation (comments).	Fully functional in scope. Excellent code style. Formally documented throughout.