

Booked Library System

Assignment Writeup Guidelines

Software Engineering I



Feb 2022

1 Introduction

Software engineering is concerned with numerous activities that stretch far beyond the programming and debugging of code. It is a collective term that encompasses a range of processes designed to provide context and validation to development, and to better prepare for and sustain the production of software. In fact, programming ability may be far down the list of requisite skills when looking to recruit software engineers. A software engineer is an expert in identifying the need for software, designing appropriate solutions, and documenting design efforts.

1.1 Expectation of Written Representation at an MSc Level

It goes without saying that academic writing uses formal vocabulary as compared to the vocabulary used in day-day communication. Your aim should be to make your text as clear as possible – to present your ideas clearly and concisely and to avoid ambiguity or redundancy. The tone of academic writing is formal with clear focus on the issue or topic rather than your opinion. Academic writing is defined by conventions rather than specific rules. Here is a link that would help you build your written skills. <https://www.bath.ac.uk/professional-services/academic-skills/>.

As future software engineering leaders, we aspire our students to work to a high professional standard and use relevant formal vocabulary to engage in a critical/analytical discourse as part of their report specification. It also goes without saying that **grammar and punctuation** need to be impeccable and that you need to be clear on the accuracy of the required vocabulary for this module. If in doubt please refer to the Sommerville textbook or ask the module leader.

We wish to bring out the best in our students and we highly value originality and individuality in their written work. We therefore, provide you with all the tools that you will need to model your answers but we do not provide model answers to students to template off. We also do not provide model answers on the online forum in response to questions you may ask of us.

1.2 Submission Report

This section details the report deliverable written to high professional standard. At least that is what aspire from our students. Your report must be a maximum of **15 pages**, including diagrams and references. An allowance of +10% is made for this page limit. The marker will not read beyond this. Your report should be written using font size 12 (or other standard size).

Please include appropriate section headings and page numbers in the footer. The core course material is sufficient to pass this assignment well. Reading extra material can be helpful to inspire and refine your solutions. Write from what you know using formal vocabulary and formal definitions, and then move on if needed and if time allows. You are not required to

read beyond the core material, but any additional sources of information should be correctly referenced. Your report should include the sections below. The first page needs to be a cover page with you the assignment title **SE1 Assignment Report** and must not include your name anywhere and the first page is not counted as part of the page count of your written report.

Observing Conventions. Please make sure that you use the correct question numbering shown in the brief and **NOT introduce your own question numbering**.

The assignment deadline can be found on Engage. You must submit a PDF file to the Engage assignment page by the submission deadline shown above. Submissions received after this deadline will be capped at 40% if received within 5 working days. Any submissions received after 5 working days will be marked at 0%. If you have a valid reason for an extension, you must submit an extension request through your Director of Studies – unit leaders cannot grant extensions.

You should leave yourself time to download your file from Engage and check that you have attached the correct file, with the content that you want to be marked.

You are responsible for checking that you are submitting the correct material to the correct assignment.

2 Assignment

You are welcome to ask about a particular process and we will provide a client response where appropriate. However, you do not need to re-post, if you see that someone asks about a process that you have already modelled and made assumptions for. Do feel free to have fun with this assignment and be creative. We will be looking at your requirements, and the accompanying justifications. This is an introductory unit where we are interested in your decisions as a future software engineering leader and understanding of the material.

Posting Online in Forums. We encourage you to post all your questions on the forum but we need to remind you that if responding to you means that we are answering a question which is a potential solution; we may refrain a response and our decision stands. Ultimately, we want you to become confident software engineering leaders who are confident with using their tools and engage in a good academic discourse.

Weekly Class Discussions. Each week, we will be posting online in the weekly class discussions, with hints and examples of how to address the ungraded and ungraded questions so that you can benefit from the additional support and you can benefit from learning and applying yourself to the assignment.

Asking Assignment Questions via Email. In the interest of academic transparency and fairness, we ask all students to post assignment questions online. This is so that every-

one can benefit from the response. We ask that all students please observe and respect this convention.

Scenario: Mirroring the Real World. In the interest of realism, we have worked to provide you a brief that reflects some of the challenges that you could come across in the real world. In the real world you have to work to ascertain the requirements from people, written briefs, from interviews and your role will be ascertain the accuracy and correctness of what is presented to you.

This is where requirements refinement comes into play. The main thing with requirements refinement is working through the requirement until the final requirement is written such that you can translate it directly into system feature - so that will be your check, that a given requirement is complete and final. You can expect some examples from us as part of the online class discussion posts.

2.1 Question 1 - Requirements, 40%

Scope of the brief. It is important to establish the scope of the booked scenario, so you can make some reasonable assumptions about the booked scenario.

The **proposed requirements** placed in the booked library brief **have several issues**. There are **deliberate omissions** in the scenario so that you can recognise what the issues may be. You also have the room to be creative in your approach. You could come up with anything else that you think would be a good and justifiable addition to the system requirements for book loans.

In a real world software engineering project, there would be some interaction with a client, who may want more or less involvement in the process of the system. In this assignment, the role of the client is played by members of the teaching team, who have prepared the brief provided and are now passing decisions to you.

- **Q1a. Identify five issues with the proposed functional/non-functional requirements in the Booking brief.** We want you to reflect and identify what these issues might be and write about them; provide justifications for each point. These issues need to be specific and not based on your opinion. Once you have identified an issue, you will need to explain using reasoning why it is an issue - i.e. what is wrong with it. [10%]

[**Hint:** After reading the requirements, **reflect** and think if the requirement has any omissions, inaccuracies, is it valid?, is it realistic? - can it be implemented into a system feature right from the description of it itself?, does it aim to fulfill what the booked brief is asking of it. **Does it answer the question of what the system is meant to do and NOT how the feature is to be developed and coded (this is not part of requirement analysis).** These are some hints and this list is not exhaustive, so think carefully and strategically. What we are interested in is your ability to recognise an issue and make a reasoned judgement for it, not simply state that there is an issue

because in your opinion there is an issue.

- **Q1b.** A list of **pertinent system and user requirements** for Functional Requirement 4 (book loans) and be clear about separating them into **user requirements** and **system requirements**. The following components are expected as part of your answer. [15%].
 - Traceability Matrix (User and System)
 - Conflict Checks - Identified and Resolved.
 - Pre-Post Conditions Identified
 - Dependencies Identified
 - Constraints on the Requirements Identified
- During the weekly class discussions on the Online Forums, we shall show you how to present system and user requirements and give you ample examples .
- Do not include all system/user requirements, just new ones that you have created.

Q1c. In your requirements, you need to demonstrate that you have provided consistent requirements that show a clear and complete distinction between **what the system shall accomplish (requirement)** and **how a feature is implemented(design)**. You must also show a clear distinction between the **system requirements and user requirements**. Design requirements are outside the scope of this brief and therefore you are not expected to write about these.

Measurability, can be shown with the above five and below.

Validity. The requirements reflect the current needs of the users, bearing in mind that these may have changed over time and through the refinement itself. The final proposed requirements need to be at the highest level of refinement.

Consistency. There are no contradictory descriptions or constraints in the document. For example, all sub-requirements of system requirements are also system requirements and the same be the case of user-requirements.

Completeness. The system requirements include all functions and constraints, are accurate and without omissions. A traceability matrix with conflict checks, identified and resolved, dependencies, pre-conditions and post-condition checks identified is a good way to measure if a given requirement is complete or not.

Realism. Your system requirements are realistic given technology, schedule, and budget (not relevant in the case of booked brief). The central question you need to ask is if I take this system requirement could it be translated into a system feature in software using a suitable programming language?

Verifiability. Tests can be written to demonstrate the system meets each requirement to reduce potential for disputes and hence you need to demonstrate that each system requirements is a testable system feature.

You will be creating the details of these requirements yourself, so the book loaning and any other associated processes are up to you. You are not expected to modify the existing set of requirements, but can indicate if you think your requirements will affect existing ones. This will also be a good way to check if your new requirements have a dependency check, pre-conditions, post-conditions, conflicts and inconsistencies between the new requirements and the ones you have identified.

Measurability can be demonstrated by showing that your requirements have achieved all the of the above as well as meeting the requirements validation criteria set out on the course page. You do need to keep in mind that we do not expect you to demonstrate every single requirement to be measured but you can make a judgement on choosing at least 4 requirements where you have shown that you have measured the requirements. [15%].

2.2 Question 2 - Architectural Design, 30%

This section must include

- **Q2a. Diagrams.** An architectural design for the following 3 architectures using suitable notation: Model View Controller, Client-Server Architecture, Layered Architecture. These architectures diagrams need to reflect how you would design them for booked brief. So, the more detailed and specific you can be, the better it is for you.
- **Q2b. Justifications.** A written justification which includes a comparison between above 3 architectures. It is important that you understand how to write justifications; simply describing your viewpoint is not enough.
- **Q2c. Decision.** In this section, you need to state your decision for the choice of architecture that you propose. Make a case for using critical analytical arguments to propose a particular system architecture for booked brief in your conclusion.
- You can compare the architectures on the following criteria: **Availability Reliability, Testability, Scalability, Security, Agility, Fault Tolerance, Elasticity, Recoverability, Performance, Deployment, Learnability.** These are the 12 main characteristics and I would not expect you to go beyond these but should you have a valid point of comparison then please state what the criteria is for comparison.

Your work should be a critique rather than simply doing a descriptive writing of your points and you have been shown in the study skills post online and what is the difference between descriptive writing and critical writing. I have explained the criteria here below.

- **Availability.** How long the system will need to be available (if 24/7 , steps need to be in place to allow the system to be up and running quickly in case of any

failure). So, which of the 3 architectures offer the best availability for the booked brief?

- **Recoverability.** This is dealing with the business continuity requirements (e.g in the case of a disaster, how quickly is the system required to be online again? This will affect the backup strategy and the requirements for duplicated hardware. So which architecture styles will support faster recoverability when it comes to the booked brief?
- **Elasticity.** It is the ability of the system architecture to handle bursts of requests/traffic. This means can a particular system architecture cope with an exponential increase in the number of requests, so which of the 3 architectures are suitably elastic for the booked brief?
- **Agility.** Agility is the ability for the system architecture to embrace and respond to change in software processes due to significant changes in the business environment or perhaps even new business demands. So which of the 3 architectures are the most agile when it comes to booked scenario?
- **Learnability.** Learnability is how easy it is for the users to learn to use the software and another expansion of the definition is how the developed software can integrate itself into the wider software systems architecture, so another definition is for the software to learn about its environment in order to become self configuring or self-optimizing - this is particular the case when new modules are integrated with existing software modules. So which architecture makes it easier for new modules to be easily integrated?
- **Reliability.** A Software architecture's reliability is the probability of failure-free operation of a computer program for a specified period in a specified environment. Reliability is a customer-oriented view of software quality. A software architecture's reliability is measured in the operational environment with excellent accuracy. So which of the three architectures would be the most reliable in the case of booked library scenario?
- **Testability.** Software testability is the degree to which a software system or a unit under test supports its own testing. To predict and improve software testability, a large number of techniques and metrics have been proposed by both practitioners and researchers in the last several decades. Which of the 3 architectures support testing with ease and how would tests run on one architecture differ from those on another?
- **Security.** Software security is an idea implemented to protect software against malicious attack and other hacker risks so that the software continues to function correctly under such potential risks. Security is necessary to provide integrity, authentication and availability. Which of the 3 architectures are the most secure in the case of the booked scenario?
- **Deployment.** Software deployment refers to the process of running an application on a server or device. Software deployment refers to the process of making the application work on a target device, whether it be a test server, production

environment or a user's computer or mobile device. How do these 3 architectures affect software deployment when it comes to booked scenario?

- **Fault Tolerance.** Software architecture's fault tolerance is the ability of the architecture to not be affected by the fault, and what it takes for the architecture system to recover from a fault that is happening or has already happened in either the software or hardware or in some component of the system architecture in which the software is running in order to provide service in accordance with the specification. So which of these architectures will be highly fault tolerant when it comes to the booked scenario?
- **Scalability.** Architecture scalability is the flexibility and ability of the architecture to grow or shrink to meet changing demands on a business. Architecture scalability is critical to support growth, but also to pivot during times of uncertainty and scale back operations as needed. So, you need to think which architecture styles support better scalability for the booked scenario?
- **Performance.** Performance is an indicator of how well a software systems architecture and its component meet its requirements for timeliness. Timeliness is measured in terms of response time or throughput. The response time is the time required to respond to a request. It may be the time required for a single transaction, or the end-to-end time for a user task. For example, we may require that an online system provide a result within one-half second after the user presses the "enter" key. For embedded systems, it is the time required to respond to events, or the number of events processed in a time interval. The throughput of a system is the number of requests that can be processed in some specified time interval. So, how would the 3 mentioned architectures fare when it comes to the booked scenario?

You may want to refer to the architectural patterns lesson for this. You also need to read the study skills required for the SE1 unit on the online forum post so that you can get an understanding of the expectation of written representation at an MSc Level.

Your justifications should include any **4 out of 12** of the above criteria for comparisons against alternatives. Please be aware that opinionated justifications lacking above mentioned standards for comparison make it harder for us to give you credit for your answers.

Out-of-scope. Hybrid models are out of scope of our syllabus, so please do not discuss these.

Additional Reference Text. The book Software Engineering by Google https://www.amazon.co.uk/Software-Engineering-Google-Lessons-Programming/dp/B08VKJXVHK/ref=sr_1_1?keywords=software+engineering+at+google&qid=1646230724&srefix=software+en%2Caps%2C64&sr=8-1 is a good resource if you want to refer for some of your questions but we do not expect you to use that as your base textbook.

2.3 Questions 3 - System Models, 20%

This section should include:

- **Q3a. Diagrams.** Draw three system models applied to the booked scenario. These are: **Class Diagrams, Sequence Diagrams and Business Process Diagrams**. The system models should use an accurate and relevant notation. It will not be possible to model the entire system in a 15 page document, so be selective and model a few key functionalities. The system models should include suitable and accurate notations for your choice of modelling, use of suitable abstractions and connection to requirements.
- **Q3b. Justification.** The justifications of your system models must include a balanced critique of the aforementioned system models.
- **Q3c. Decision.** In this section, you need to state your decision for the choice of system models that you propose. Make a case for using critical analytical arguments to propose a particular system model for booked brief in your conclusion.

2.4 Presentation Report, 10%

The overall quality of your presentation will be graded for professional presentation, coherence, grammar, punctuation and the quality of your written report.