AC51003

SOFTWARE ENGINEERING

Assignment 1

ANALYSIS MODELS

Header image: https://blog.planview.com/top-6-software-development-methodologies/

Introduction

This document describes your first assignment in AC51003. It is a team-based assignment. This document will detail what you are required to do, the deadline for doing so, the intended learning outcomes of the task, the marking criteria, and other useful information and resources.

Assessment Summary

Please find below a summary of the key information about this assignment, including deadline(s).

Assignment Name:	Analysis Models
Deadline for Submission:	23.59 on Sunday 3rd March 2024
Hand-in Method:	Submit a ZIP file that contains the following: - A Use Case Diagram - Use Case Specifications and Requirements - A (brief) STRIDE Analysis - Candidate Class Analysis - Class Diagram Please submit ONE file per team. Please see Microsoft Word templates on My Dundee available for you to use.
Date That You Will Receive Feedback:	Within the University's 3-week policy which for this assignment is Sunday 24th March 2024
Feedback Type:	Written feedback and a grade
Penalty for Late Submission:	One grade point per day late (meaning if a submission is one day late and marked as a C2 it will receive a C3 grade). A day is defined as each 24-hour period following the submission deadline including weekends and holidays. Assessments submitted more than 5 days after the agreed deadline will receive a zero mark (AB).
Academic Policies:	By submitting this assignment, you are bound by the University of Dundee policies in relation to Academic Standards and Integrity. Please refer to Policy Statements, Regulations and Guidance for more information.
Percentage of Module:	This assignment is worth 30% of your total grade.

Learning Outcomes

This assignment addresses the following learning outcomes of this module:

- To demonstrate understanding of software specification and design principles, practices, and notations.
- To apply methods of software requirements specification; conforming to accepted industry practices.
- To demonstrate awareness of software security issues during software analysis and design.

- To comprehend and selectively utilize visual notations that pertain to the analysis and design of software systems.
- To employ analysis techniques: to formulate and depict structural designs for a software system.
- To demonstrate awareness of and to apply software design principles and software quality characteristics.
- To critically evaluate the quality of a software design against known design principles and quality characteristics
- To develop and apply problem-solving, communication, team-working, reporting, time-management, and independent study skills.

Task Description

In your coursework for AC51003 you will be performing a complete analysis and design for a software system. This assignment focuses on the initial analysis phase where you will identify and specify the requirements for the system and represent your understanding of the problem in the form of a 'domain model', i.e., an analysis class diagram. You should work together in your team to produce the following:

- A Use Case diagram.
- Use Case specifications (e.g., the basic and alternative flows for each of the system Use Cases).
- A STRIDE analysis.
- A supplementary requirement specification, to capture any remaining requirements that have not already been captured in the Use Case specifications above.
- An analysis of the candidate classes for the system, utilizing methods such as noun analysis, analysis class stereotypes, etc.
- A UML class diagram, to convey a structural design for the system and showing the relevant classes and relationships between them. Your design should make considerations for quality too, e.g., principles such as cohesion, the single-responsibility principle.

The system that you are being asked to design is for software that controls a drinks Vending Machine. See the separate document 'Vending Machine System Description' for more details of what this system is, and how it works.

What to submit

Please submit the following:

1. **A Use Case Diagram**

   Identify the actors and the use cases for the system and then use a UML tool such as Visual Paradigm to create a use case diagram which shows the actors, use cases, and relationships between them. You can submit your Use Case diagram as an image file or as an image embedded into a document (Word or PDF). Please also include your Visual Paradigm project file in case that needs to be consulted too. NOTE: you will find Microsoft Word templates that you can use if you wish, to submit these components.

2. A STRIDE Analysis (Threat Modelling Task)

This is intended to be a smaller component of your work and requires you to consider the topic of threat modelling. Use the Microsoft STRIDE categories to identify potential areas of security threat for your application. Try for at least one risk per STRIDE category. Summarize what the risk(s) are and suggest a way(s) to mitigate it. Don't spend too long on this but consider it at least.

3. Use Case Specifications (Basic and Alternative Flows) and Supplementary Requirements

Submit a document which contains the specifications for all your system use cases, conveying their basic and alternative flows. The specifications can be written using a standard word processor. Please see the Microsoft Word templates on My Dundee as a starting point, but you can use other formats if you prefer. The Use Case specifications should cover the majority of your requirements, particularly the functional requirements. However, they may not cover all of them, such as some non-functional requirements. In that case, please include a supplementary specification for these requirements, to ensure that your overall requirements specification is complete! The supplementary requirements should be written in the format stipulated by Standard 830-1998 - IEEE Recommended Practice for Software Requirements Specifications (and which is now superseded by Standard IEEE/ISO/IEC 29148-2011 - ISO/IEC/IEEE International Standard - Systems and software engineering -- Life cycle processes -- Requirements engineering). The supplementary requirements can be added into your Use Case Specification document – see the templates available for this on My Dundee.

4. Candidate Class Analysis

Please submit a document which shows your analysis of, and identification of the Classes for the system design. At the very least, this should include a noun analysis, e.g., a table showing candidate classes that are being accepted or rejected, based upon nouns identified within the system description. For additional depth and scope, you should consider other methods too, e.g., Analysis Class Stereotypes, Larman's Conceptual Class Categories, or others.

5. UML Class Diagram(s)

Having identified the classes for the system design, depict your structural design in the form of a UML class diagram – showing the classes identified and any relationships between them. You can consider additional layers of quality in your design too, such as design principles and UML packages. When you submit your assignment, the class diagrams can be copied and pasted from Visual Paradigm into Word or PDF documents. And/or, you can also export class diagrams as image files from Visual Paradigm. Regardless, please submit your Visual Paradigm project file too so that your class diagram can be consulted directly if required.

6. **A completed Peer Review by each member of the team.**
   Each member of the team must submit their own peer assessment which shows the contributions made by everyone. The online peer review system will be used for this. The results of the peer review *may* be considered when finalizing individual grades for the assignment. For example, those who have contributed less may receive a smaller proportion of the group's overall mark.

How to submit

Assignments should be submitted to My Dundee within the relevant section of the 'Assessment Area'. Please collate all the files for your team's submission into a single ZIP file. Please name this file after your team, e.g., "Team1.zip". One person from your team should submit the file on behalf of the team. If group-based submissions have been enabled on My Dundee, it will be possible to submit the ZIP file as a group (one person can upload, but everyone in the group will see it). If group-based submissions have not been enabled on My Dundee, it will require one person in the group to submit on behalf of everyone else (meaning only that one person can see what was submitted, therefore, make sure everyone else in the group has a copy of what was submitted too).

Component Weightings

The following weightings are applied to the components in this assignment:

Component	Weighting
Use Case Diagram	15%
STRIDE Analysis	5%
Requirements (Use Case Specifications and Supplementary Requirements)	25%
Candidate Class Analysis	20%
Class Diagram	35%

Helpful Resources

You will find answers to common questions for this assignment in an associated document. There is also a document with details of common issues for you to be aware of when completing your work, as well as templates for you to use. Please find below a list of documents associated with this assignment.

Document	Description
Vending Machine System Description.pdf	A description of the system you are designing.
AC51003 - Assignment 1 - COMMON ISSUES.pdf	Some common issues to look out for with this assignment.
AC51003 - Assignment 1 - ANSWERS TO COMMON QUESTIONS.pdf	Answers to common questions about this assignment.
Template for Use Case Specification Document.docx	A template for Use Case specifications.

Document	Description
Template for Supplementary Requirements Specification.docx	A template for a supplementary requirements specification.
Template for Use Case Specification WITH Supplementary Requirements - Variant 1.docx	A template for Use Case specifications which combines the supplementary requirements within it too. The supplementary requirements are placed under each use case they relate to.
Template for Use Case Specification WITH Supplementary Requirements - Variant 2.docx	A template for Use Case specifications which combines the supplementary requirements within it too. The supplementary requirements are placed at the end, after all use cases.

Marking Criteria

The tables below provide detailed marking criteria which are used to assess the different components of your work. An indicative grade range is noted but this is dependent upon the extent to which the criteria are met and criteria will be combined selectively where required, e.g. if you have provided evidence of enhancements that go beyond the core requirements (an indicator of potential excellence) BUT there is very poor coverage of the system requirements (an indicator of a poor outcome) then an outcome will obviously lie somewhere in between these two points in the grade continuum.

Use Case Diagram

Classification	Indicative Grade Range	Criteria – Use Case Diagram
Excellent	A5 to A1	All actors identified correctly covering the expected external entities. All use cases identified correctly and covering the expected functionality at an appropriate level of size / granularity – use cases should be self-contained objectives or sub-objectives for a system which can stand alone in their own right (even if part of a larger process) and deliver something of value to actor. A complete use case diagram which conveys the actors and use cases. Actors connected correctly to their relevant use cases with the primary and secondary actors clearly distinguishable. Actors named appropriately, reflecting the role they undertake in relation to the system. Use cases named appropriately reflecting their main objective / outcome and stated from the perspective of the actor that initiates them. Correct use of UML notation to convey the actors and use cases. Correct and justifiable use of <<include>> and <<extend>> relationships where applicable (noting that these relationships may not always be required). The use case diagram corresponds correctly with the use case specifications.
Very good	B3 to B1	Very good identification of use cases and actors but some non-core elements may be missing. The use cases are an appropriate level of size / granularity. Naming of actors and use cases are very good but some are not stated from the perspective of the actors or reflecting the roles of the actors sufficiently.

Classification	Indicative Grade Range	Criteria – Use Case Diagram
		Modelling of the use cases is very good but some aspects of notation may be missing or incorrect, including distinguishing primary and secondary actors. There is very good use of <<include>> and/or <<extend>> relationships (where applicable) but there are some questionable decisions in terms of justifying the use of these relationships. There is very good correlation between the use case diagram and use case specifications.
Good	C3 to C1	There is good identification of use cases and actors, but some elements may be missing, including what could be 'core' elements. The use cases are at a good level of size / granularity but there is some fragmentation in the model. Naming of actors and use cases are good but some are not stated from the perspective of the actors or reflecting the roles of the actors sufficiently. Modelling of the use cases is good but some aspects of notation may be missing or incorrect, including distinguishing primary and secondary actors. The use of <<include>> and/or <<extend>> relationships is good (where applicable) but the justification of these may be questionable and/or these are not modelled correctly. There is good correlation between the use case diagram and use case specifications but various aspects which are out of sync.
Poor	D3 to D1	There is poor identification of use cases and actors, various elements are missing or incorrect. Various use cases have a poor level of size / granularity which has resulted in a fragmented model. Naming of actors and use cases is poor. Modelling of the use cases is okay but the diagram is difficult to understand or does not convey the system functionality correctly. Use of <<include>> and <<extend>> (where applicable) is okay but mostly unjustified and misunderstood. The correlation between the use case diagram and use case specifications is okay but various aspects are out of sync.
Very Poor	BF to MF	There is very poor identification of use cases and actors, various elements are missing or incorrect. Many use cases have a poor level of size / granularity which has resulted in a fragmented model. Naming of actors and use cases is poor. Modelling of the use cases is poor or missing. There is little or no correlation between the use case diagram and use case specifications.

STRIDE Analysis

Criteria – STRIDE Analysis
Ensure that you identify at least ONE risk for each of the STRIDE categories with a suggestion for how to mitigate it. For extra marks, consider more!

Use Case Specifications

Classification	Indicative Grade Range	Criteria – Use Case Specifications
Excellent	A5 to A1	Complete specifications for all required use cases and which correlates with the use case diagram. A complete, valid basic flow of events for each use case which presents the main success scenario in an expected format, and which clearly indicates how each use case begins and ends and which clearly shows the relevant system actors involved. A complete set of alternative flows identified for each use case covering all relevant error scenarios, variations in options, choices, decisions, and additional behaviors. The alternative flows documented in an expected format and including cross-references from where they occur within the basic flow of events or within other alternative flows. A correct level of detail in the use cases - as a form of requirements specification, Use Cases must provide sufficient detail of what the system is doing, albeit in a more informal way than a traditional requirements specification. The use cases shouldn't omit details which are important to understanding the true extent of the system's functionality. A correct communication style in the use cases - Use cases should communicate the intended behavior of the system in a way that is accessible to both technical and non-technical stakeholders on a project and should avoid the use of technical jargon and should avoid making commitments to user interface choices unless already stipulated by the client. For extra marks, you can consider 'enhancements' which go beyond the basic requirements of the system, e.g., you can consider additional use cases or features that you think would make the system more robust, usable or more appealing.
Very good	B3 to B1	A very good set of specifications but some areas of system functionality are missing or incomplete. The basic flows of events are very good but there are some documentation issues, e.g., missing details of how the use case begins or ends or structural issues. The alternative flows are very good, but some are missing and/or have not been documented appropriately. The level of detail in the use cases is very good but there are some aspects of functionality that are missing or need to be expanded. The communication style in the use cases is very good, clear, and accessible. There is very good correlation between the use case specifications and use case diagram. There is very good correspondence between the content of the use case specifications and the requirements specification for the system.
Good	C3 to C1	A good set of specifications but there are various areas of system functionality missing or incomplete and which may include 'core' areas. The basic flows of events are good but there are some documentation issues, e.g., missing details of how the use case begins or ends or structural issues. The alternative flows are good but there are a number of these missing and/or have not been documented appropriately.

Classification	Indicative Grade Range	Criteria – Use Case Specifications
		The level of detail in the use cases is good but is missing core aspects of functionality and/or some of the incidental requirements of the system. The communication style in the use cases is good. There is good correlation between the use case specifications and use case diagram. There is good correspondence between the content of the use case specifications and the requirements specification for the system but there are various areas of difference, e.g., alternative flows identified in use cases are not mentioned in the requirements specification.
Poor	D3 to D1	A poor set of specifications with various core areas of system functionality missing or incomplete. The basic flows of events are okay but are missing details or are poorly structured. The alternative flows are okay, but a large number are missing or poorly documented. The level of detail in the use cases is poor and is missing core aspects of functionality. The correspondence with the requirements specification is poor.
Very Poor	BF to MF	The specifications are missing or very poor. The documentation has been attempted but is poorly detailed and/or poorly structured. Alternative flows are missing or very poor. There is very poor correspondence with the requirements specification and the use case diagram.

Supplementary / Traditional Requirements

Classification	Indicative Grade Range	Criteria – Supplementary Requirements
Excellent	A5 to A1	A complete set of supplementary requirements to address those not covered within the Use Case specifications, both functional and non-functional. Each requirement is clear, concise, verifiable, and unambiguous. Each requirement is written using correct formal language and conforms to the format expected of IEEE Std 830-1998. The requirements are presented in clear, well-organized sections avoiding conflicts and redundancies, with cross-referencing where required. For extra marks, consideration may be given to 'enhancements' which go beyond the core requirements, e.g., ideas of your own.
Very good	B3 to B1	A very good set of requirements but some aspects are missing. Most requirements are clear, concise, verifiable, and unambiguous. Most requirements conform to IEEE Std 830-1998. The presentation of the requirements is very good but is lacking some structure and organization.
Good	C3 to C1	Coverage of the requirements is good but there are various aspects missing or not fully elaborated and which may include core aspects of the system functionality Various requirements lack clarity and/or are not measurable.

Classification	Indicative Grade Range	Criteria – Supplementary Requirements
		There is good but inconsistent compliance with IEEE Std 830-1998. The presentation of the requirements is good but could be improved in several areas.
Poor	D3 to D1	There is poor coverage of the requirements and many core aspects are missing or not fully elaborated. Many requirements are missing the expected language or format. Compliance with IEEE Std 830-1998 is poor. The presentation of the requirements is poor.
Very Poor	BF to MF	There is very poor coverage of the requirements Requirements haven't been written using the expected language and format. Compliance with IEEE Std 830-1998 is very poor or non-existent. The presentation of the requirements is poor. There is poor correlation between the requirements and use cases.

Candidate Class Analysis

Classification	Indicative Grade Range	Criteria – Candidate Class Analysis
Excellent	A5 to A1	Identification of all relevant nouns within the system description which could reveal the required classes in the system design as well as any relevant fields or attributes within those classes too. Presentation of all relevant nouns in the form of a candidate class refinement. Correct selection of all relevant classes (and fields/attributes) based on guidelines given. Reasons provided for keeping any nouns as candidate classes Reasons provided for rejecting any nouns as candidate classes Consideration of any relevant <<boundary>>, <<entity>>, and <<control>> classes. Consideration of conceptual class categories as an approach to candidate class analysis. Correct correlation between classes identified and the class diagram.
Very good	B3 to B1	Very good coverage of the relevant nouns but some important concepts are missing. Very good refinement of candidate classes but some classes have been rejected which could be useful in the system design. A very good audit trail of the decision-making process but some aspects may be missing, e.g., there isn't full reasoning for decisions made. Additional class analysis techniques have been partially considered. The correlation between the class analysis and the class diagram is very good but there are some unexplained differences, e.g., classes appearing on the diagram which weren't in the earlier analysis or classes missing from the diagram that were identified during the analysis.
Good	C3 to C1	Good coverage of the relevant nouns but various concepts are missing and/or the list presented is limited. Good refinement of candidate classes but various classes have been rejected which could be useful in the system design.

Classification	Indicative Grade Range	Criteria – Candidate Class Analysis
		A good audit trail of the decision-making process but only partially documented in terms of reasoning. Additional class analysis techniques haven't been considered. The correlation between the class analysis and the class diagram is good but there are various unexplained differences, e.g., classes appearing on the diagram which weren't in the earlier analysis or classes missing from the diagram that were identified during the analysis.
Poor	D3 to D1	The noun list is limited or incomplete. The refinement of candidate classes is poor – many classes have been rejected which could be useful in the system design. The audit trail of the decision-making process is very limited. Additional class analysis techniques haven't been considered. The correlation between the class analysis and the class diagram is good but there are various unexplained differences, e.g., classes appearing on the diagram which weren't in the earlier analysis or classes missing from the diagram that were identified during the analysis.
Very Poor	BF to MF	The noun list is extremely limited or missing. The refinement of candidate classes is very poor – many classes have been rejected which could be useful in the system design. The audit trail of the decision-making process is very limited in scope or is missing. Additional class analysis techniques haven't been considered. The correlation between the class analysis and the class diagram is poor or non-existent.

Class Diagram

Classification	Indicative Grade Range	Criteria – Class Diagram
Excellent	A5 to A1	A complete set of classes presented for the system design. A complete set of relationships between classes which may include associations, composition, inheritance, dependencies, and unidirectional associations where applicable. Correct use of UML notation to represent the various relationship types. Each relationship has a descriptive name or role names specified and which helps to tell the 'story' of the system. Each relationship has multiplicities specified. The class names are an accurate reflection of the relevant entities in the problem domain and indicative of their purpose in the system. The classes are named according to object-oriented conventions, e.g., singular nouns or noun phrases with the first letters of each word capitalized and multiple words bunched together. Coverage of the system functionality is excellent – it is clear how the entirety of the system functionality will be facilitated in terms of classes presented and the relationships between them. Design quality is excellent – the classes and relationships ensure that the system is maintainable and extensible.

Classification	Indicative Grade Range	Criteria – Class Diagram
		Each class is cohesive (singularly focused) and in line with design principles such as the open-closed principle or similar. Any known extensibility requirements have been considered and are represented in the design. The diagram is readable and well laid out. Additional considerations have been made for bringing higher levels of organization into the design, e.g., use of Packages. Additional enhancements to the design have been considered, e.g., use of UML stereotypes to convey boundary, entity, and control classes; more advanced user interface considerations (e.g., additional boundary classes); consideration of more advanced approaches to design quality, e.g., design patterns.
Very good	B3 to B1	A very good set of classes are presented in the class diagram, but some are missing. A very good set of relationships between classes but some are missing. Very good use of UML notation to represent the various relationship types but some aspects are missing or incorrect. Very good specification of relationships but some are missing descriptive names and/or multiplicities and/or don't convey a natural 'story' for the system. The class names are very good but could be improved in terms of how well they reflect the relevant entities in the problem domain and/or in terms of how well they conform to object-oriented naming conventions. Coverage of the system functionality is very good but there are some areas where it is unclear how the classes and relationships between them will achieve the stated functionality. Design quality is very good but there are some parts of the design where maintainability and/or extensibility are limited or compromised. The cohesion of the classes is very good but there are some where cohesion could be improved. Extensibility requirements have been considered but not fully achieved within the design. The diagram is readable and well laid out.
Good	C3 to C1	The class diagram is good but limited in scope in terms of the classes and relationships presented. Good use of UML notation to represent the design. The specification of the relationships is good, e.g., with descriptive names and multiplicities. The class names are good but could be improved in terms of how well they reflect the relevant entities in the problem domain and/or in terms of how well they conform to object-oriented naming conventions. Coverage of the system functionality is good but there are issues with the cohesion in the system classes, i.e., there should be a larger number of smaller, more focused classes in the design. Design quality is OK but there are various parts of the design where maintainability and/or extensibility are limited or compromised.

Classification	Indicative Grade Range	Criteria – Class Diagram
		Extensibility requirements have been considered but not fully achieved within the design. The diagram is readable and well laid out.
Poor	D3 to D1	The class diagram is very limited in scope in terms of the classes and relationships identified. Use of UML notation is OK but there are various mistakes and aspects missing. The specification of the relationships is OK, but many are missing descriptive names and/or multiplicities. The class names are OK but could be improved in terms of how well they represent the relevant entities in the problem domain and their role in the system design. Coverage of the system functionality is limited and there are issues with the cohesion in the system classes. Design quality is poor, this has not been considered to any great extent. Extensibility requirements have not been considered. The diagram is poorly laid out.
Very Poor	BF to MF	A very poor representation of the system design with many aspects missing or not considered. Use of UML notation is OK but there are various mistakes and aspects missing and/or limited in scope. The specification of the relationships is very poor, most are missing, and many are missing descriptive names and/or multiplicities. The class names are a poor reflection of the relevant concepts in the domain and are not indicative of their purpose in the system design. Coverage of the system functionality is very poor or non-existent. Design quality has not been considered. Extensibility requirements have not been considered. The diagram is poorly laid out.