An introduction

This is a discussion and interpretation of the specification, which should identify any issues that require clarification. This is where interpretations of the specification should be explained.

Use-Case Model

This is a formal statement of user requirements with Use-Case views, survey descriptions, simple interaction, statechart, class diagrams and CRC cards. Remember the diagrams should be in outline form at this stage.

Analysis Model

This is a technical specification. The class descriptions should be detailed with attributes and methods and relationships identified to define the software architecture. Detail should be added to the interaction and statechart diagrams to show events actions and qualifiers. Non-functional requirements should be identified. The analysis model should identify service and analysis packages as and if appropriate.

Design Model

This should include detailed design classes in which all methods and attributes, and their visibility is identified. Arguments and return data types should be identified and defined. Key implementation mechanisms should be defined. Interaction diagrams and statecharts should also be detailed. The mechanisms for the implementation of non-functional requirements should be defined and designed.

You are not required to provide Implementation and Test Models.

It is important that designs are explained but please be succinct. Also take care to consult the study guide to understand how the design process should be developed at each stage. The easiest way to fail this element is by not addressing one or more steps in the design process.

If the marking scheme is expanded then the following would be an appropriate replacement for the existing design element:

|  |  |
| --- | --- |
| **Introduction**  Interpretation of specification. Identification of appropriate elaboration. | / 5 |
| **The Use-Case Model**  Correct use of notation. Complete set of diagrams. Appropriateness of use-case structure, survey descriptions, CRC cards, interaction diagram(s), class diagram, class identification and statechart diagrams. Remember these are outline diagrams. | /10 |
| **Analysis Model**  Appropriateness of elaboration. Identification of methods and attributes. Definition of class architecture. Identification of non-functional requirements. Consideration and appropriateness of package diagrams. | /10 |
| **Design Model**  Appropriateness of elaboration. The visibility of attributes and methods. The identification of arguments, return data types, class creation and destruction, process forks and joins as appropriate. Definition non-functional requirement implementation. | /5 |