3.4 A2 Unit F454: *Computing Project*

In this unit, candidates develop their knowledge and understanding of computer systems and the skills studied in AS F451: *Computer fundamentals* and AS F452: *Programming techniques and logical methods* and use the high-level programming techniques studied in AS F452. This project is a substantial piece of work, requiring analysis and design over an extended period of time, which is organised, evaluated and presented in a report.

Candidates choose, in conjunction with their teacher, a well-defined user-driven problem of an appropriate size which enables them to demonstrate their skills in Analysis, Design, Software Development, Testing, Implementation, Documentation and Evaluation, and their interrelation; and to give a completed overall system that solves the problem.

3.4.1 Definition, investigation and analysis

Explanation of the problem to be solved, the user's requirements and how they were obtained.

Evidence of the development of a requirement specification and a clear statement of requirements.

- Define a task
- Research the task and needs of the end-user(s)
- Record findings
- Analyse findings
- Project plan, including specification of the requirements:
  – user
  – hardware
  – software

Candidates should be able to:

a. define the nature of the task to be carried out;

b. identify methods by which to investigate the problem, including questionnaires, observation and structured interviews;

c. record information/data and gather sample documents currently used;

d. identify the current processes and current data;

e. analyse the data and processes: candidates will be expected to use appropriate techniques such as structure diagrams/data flow diagrams/system flowcharts to illustrate their analysis;

f. specify any perceived problems and inefficiencies apparent from discussions with the user and the analysis work carried out;

g. derive the user and information requirements of a system;

h. specify and justify the required hardware;

i. specify and justify the required software.

3.4.2 Design

Detailed system design including:

- data structures
- input-output format
- processes involved.

A clear test strategy including test data needs to be developed to ensure the system meets the design objectives.

A clear design specification:

- Specify the objectives relating them to the requirements specification
- Input design
- Output design
- Design and test plan, and test data
- Data structures/variables
- Algorithms
- Test algorithms

Candidates should be able to:

a. specify the objectives of the proposed system and relate them to the requirements specification;

b. design and document data capture forms and/or screen layouts, drawing up detailed mock-ups of the proposed interface;

c. design and document report layouts, screen displays and/or other forms of output (for example, audio output), drawing up detailed mock-ups of the proposed interface;

d. identify, develop and document a test strategy for the design;

e. select suitable test data for the design;

f. design and document, using appropriate techniques (for example, data flow diagrams), the data structures necessary to solve the inefficiencies/problems indicated in the requirements specification;

g. design and document an algorithm/pseudo-code/top-down diagram or other form of process model;

h. using appropriate techniques, test that the algorithms meet the design objectives.

3.4.3 Software development and testing

Develop a software solution and, using the test plan developed in 3.4.2: *Design*, show that the system works with valid, invalid and borderline data (or, if it does not, under which circumstances it fails).

Test plan clearly cross-referenced to evidence that the system has been tested during development and implementation.

Evidence of user testing.

- Software development
- Alpha testing
- Response to the results of testing
- Beta testing
- Response to the results of beta testing
- Modularisation of code
- Code documentation
- Use of modules, data structures and objects
- In-code documentation
- Code structure
- Applying the test plan and data

Candidates should be able to:

a. develop the rules/methods/algorithms of a design using a programming language;

b. develop the data structures of the design using the appropriate features of a programming language;

c. develop inputs/outputs using the features of a programming language;

d. test and refine the software solution during development, illustrating how the software solution evolves;

e. test the software solution with the user, providing documented evidence that the solution works;

f. produce suitable modular code with full annotation and a description of how the modules combine to create the solution;

g. produce detailed output from the testing, cross-referencing as appropriate with the test plan.

3.4.4 Documentation

The software solution should be self-documenting, with suitable on-screen help and support for the end user(s).

All necessary supporting documentation required to enable the end user to make effective use of the solution.

- User documentation

Candidates should be able to:

a. develop detailed and appropriate user documentation.

3.4.5 Evaluation

Discussion of the degree of success in meeting the original objectives as specified in the requirements specification, including an evaluation of the project management – to include an evaluation of the ease of the use of the package, acceptability to the users, the choice of task, the project plan and desirable extensions/modifications.

- Evaluate results against the requirement specification showing that the objectives in 3.4.2: *Design* have been satisfied
- Evaluate the project management
- Evaluate user responses
- Identify the good and bad points of the final system, including any limitations and necessary extensions or modifications to the system

Candidates should be able to:

a. evaluate the final system against the criteria described in the requirements specification;

b. evaluate the users' responses to testing the system;

c. critically evaluate the project management;

d. identify the good and bad points of the final system highlighting any limitations and necessary extensions or modifications to the system, indicating how these could be carried out.

The written report

See Appendix B for the assessment criteria for this report.

Appendix B: Guidance on Setting and Marking
A2 Unit F454: *Computing Project*

Guidance on Setting Computing Projects

A project should:

- allow candidates to demonstrate their knowledge and understanding of computer systems and the skills in the assessment objectives;

- encourage the sensible use of computers to produce a system, using an appropriate programming language, which is non-trivial, has a substantial coded element and will solve a given problem sensibly within the constraints of the resources available to the candidate.

Note: It may be difficult to quantify the scale of a problem fully. Indeed, before the analysis and design stages have been completed it is not possible to know the depth of the solution. Candidates should note that the emphasis should be on the choice of a real-life problem. Centres are reminded that if they are in any doubt about the suitability of a problem on the grounds of degree of rigour, type of problem or any other criteria, they should contact the board at the earliest opportunity for advice.

- show the successful completion of a whole task from its definition, involving a third party user, to its acceptance and evaluation by that user. Projects that involve much repetitive design, analysis or especially implementation, leading to unwieldy reports, are to be discouraged;

- involve all elements of the skills of definition, analysis, design, development, testing, implementation, documentation and evaluation. The project must provide sufficient opportunity for the candidate to demonstrate the programming skills developed as part of the AS syllabus and in the A2 F451. Projects need not be 'stand alone': the enhancement or modification of an existing system are acceptable, provided that all these elements are covered; this may in fact lead to work that is more likely to reflect a real-world situation;

- involve a third party user, who will provide information for the analysis, use the implemented solution and contribute towards its evaluation. The third party is likely to be a user (or potential user) of a computer system for a designated purpose. Whilst a teacher could act as the third party user, this arrangement is far from ideal. Candidates should be encouraged to look beyond school life into either the businesses and companies in the community of the surrounding area or to focus groups. The emphasis is on analysing an existing system or area for development, and producing a computer-based solution to fit the needs of the user.

Candidates should make the final choice of context for the project, although the supervisor should give guidance about project suitability. This should include guidance on the appropriateness of implementing a stand-alone or networked small computer system or other available facility. In a well-organised project, the candidate focuses on the production of an overall system analysis and design. The solution implementation must include the use of a high-level programming language. The additional use of pre-written modules and toolkits, applications software and programmable packages may be appropriate. Brief descriptions of the programming languages and any additional software packages used, together with reasons for their selection, should be included in the report.

For the coded element of the solution, the candidate should:

- annotate listings;
- explain each selection of the program with appropriate algorithm descriptions, which should be language independent;
- define variables by name, type and function where appropriate;
- define clearly, and identify the purpose of, functions subroutines and procedures.

Where the solution has used additional software packages not involving programming, candidates should:

- explain each section of the solution with appropriate algorithm descriptions;
- define the purpose and interrelationship of modules within the system;
- clearly annotate the results produced.

Test data should be devised and used systematically to test the package thoroughly. The choice of test data used, and the reason for choice, should be included. A description of the methods of testing should also be included, together with evidence of testing.

The projects should contain the title, a contents list, a description and justification of investigation, analysis, design and methods used, an evaluation and bibliography. Pages should be clearly numbered. Appropriate evidence of development, testing and implementation must support the report, for example screen dumps or photographs of screen layouts and printouts, paper-based user documentation and suitable evidence from the third party user to show that the system has been developed satisfactorily. Any evidence submitted to demonstrate the development of the solution must be able to be assessed without the use of any specific hardware or software.

Candidates should choose a well-defined user-driven problem of an appropriate size, which enables them to demonstrate their skills in Analysis, Design, Development, Testing, Implementation, Documentation and Evaluation. The project should involve the skills attained by studying the other modules of this specification, specifically the programming skills studied in the other units.

The computing projects must involve programming, and in some cases may also involve the selection and installation of hardware.

Guidance on Marking Computing Projects

Computing projects are assessed as follows:

(a)	Definition, Investigation and Analysis	**[14 marks]**
(b)	Design	**[16 marks]**
(c)	Software Development and Testing	**[30 marks]**
(d)	Documentation	**[10 marks]**
(e)	Evaluation	**[10 marks]**

(a) Definition, Investigation and Analysis [14 marks]

(i) Definition – nature of the problem to be investigated [3 marks]

A candidate should not expect the examiner to be familiar with the theory and practice in the area of the chosen system. There should be a brief description of the end user (for example, firm or business) involved; and the current methods used or details of the area for development that may form the basis of the project. A clear statement of the origins and form of any relevant data should be given. At this stage, the exact scope of the project may not be known and it may lead to the arrangement of an interview with the user.

3 marks	Excellent description with all elements present.
2 marks	Some description of both the stages of study and end user involved.
1 mark	Vague description of the end user or area for development.

(ii) Investigation and Analysis [11 marks]

This section is the 'systems analysis'. The question is not how a system performs detailed tasks, but rather how the project progresses from the original data to the results. The candidate should describe how the user requirements were ascertained (including detailed planning of the investigation). The results of the investigation should be recorded accurately and analysed carefully to show how the candidate has arrived at the requirements specification. The specification must be detailed and should include the user, hardware and software requirements of the proposed solution.

9–11 marks	Excellent user involvement with detailed recording of the user's requirements. All other items must be present, showing a thorough analysis of the system to be computerised. A detailed requirements specification, including full justification for the approach and hardware and software requirements, has been produced.
6–8 marks	Good user involvement and recording of the data collection methods. Most of the necessary items have been covered. However, one or two items have been omitted. A requirements specification is present with some attempt to justify the approach based on the results of the investigations but with some omissions, eg hardware and software requirements.
3–5marks	Some evidence that an attempt has been made to identify the end-user requirements and some recording of it has been made. Attempts at some of the other items have been made. An attempt has been made to develop a requirement specification but with little attempt to justify this based on the results of the investigation.
1–2 marks	Some elements have been discussed but with little or no user involvement.

(b) Design [16 marks]

(i) Nature of the solution [6 marks]

A detailed systems design (including appropriate diagrams) should be produced and agreed with the users. Proposed record, file and data structures should be described and design limitations should be included. Design of data capture forms, input formats (with examples of screen layouts) and output formats should be included. A detailed summary of the aims and objectives should also be included. These items are the design specifications, which should be agreed with the user.

5–6 marks	A clear set of objectives with a detailed and complete design specification, which is logically correct. There is evidence to show that the end user has seen and agreed these designs. There are also detailed written descriptions of any processes/modules and a clear, complete definition of any data structures. The specification is sufficient for someone to pick up and develop an end result using the software and hardware specified in the requirements specification.
3–4 marks	The major objectives of the new system have been adequately summarised, *but omissions have been made*. There is a brief outline of a design specification, including mock-ups of inputs and outputs, and the process model has been described (including a diagram: structure diagram, data flow diagram or system flowchart). There is some evidence that the end user has seen these designs. However, there is a lack of completeness with omissions from the process model, inputs and outputs. Data structures have been identified but there may be inadequate detail.
1–2 marks	Some vague discussion of what the system will do, with a brief diagrammatic representation of the new system.

(ii) Algorithms [5 marks]

Detailed language-independent algorithms should be developed together with evidence that the algorithms have been tested to ensure they meet the design objectives.

5 marks	A complete set of algorithms with evidence to show that they have been assessed by the candidate to show that they will meet the design specification. (Evidence should show how these algorithms form a complete solution and that they have been tested for functionality using appropriate techniques.)
3–4 marks	A complete set of detailed algorithms covering the system as specified.
1–2 marks	Some vague algorithms detailing how the system will be developed.

(iii) Test strategy [5 marks]

A detailed test strategy and plan, together with appropriate test data, should be developed and documented. It is vital to produce test cases and to show that they work. To do this, it is necessary not only to have test data, but to know what the expected results are with that data.

5 marks	A detailed test strategy and plan covering all aspects of the system with data to test under normal, extreme and abnormal circumstances.
3–4 marks	A detailed test strategy and a plan covering several aspects of the system but with inadequate data to effectively test the system, eg data covers only normal circumstances or covers only a limited part of the design specification.
1–2 marks	A vague discussion of how the system might be tested.

(c) Software Development and Testing [30 marks]

(i) Software Development [16 marks]

A technical description of how the solution relates to the design specification produced and agreed with the user should be included. It is the responsibility of the candidate to produce evidence of their development work. This section must show how the candidate tested each section during development and the responses to this alpha testing. The code must be documented adequately to explain its function and there must be clear evidence of how modular code has been used to develop the final solution.

13–16 marks	There is complete evidence showing how the solution was developed using suitable alpha testing at each stage to inform the process. The modular code is fully annotated indicating clearly the purpose of each section and the interrelationship between the sections. The developed solution fulfils all of the design specification.
9–12 marks	Program listings are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is sufficient annotation evident to illustrate how the solution was developed for a particular purpose and indicate the purpose of sections of code. The code will be modular and there will be good evidence to show how testing was used during the development process to inform each stage. The developed solution fulfils the design specification but there are some minor flaws in the solution.
5–8 marks	Program listings are provided in the form of printouts. Data structures are illustrated as part of the listings where appropriate, detailing their purpose. There is some annotation evident to illustrate how the solution was developed and some limited evidence that some testing took place during development. The developed solution has significant flaws and only partially fulfils the design specification. The code may be linear but with some annotation indicating how the code relates to the problem and some limited evidence of alpha testing.
1–4 marks	Program listings are provided in the form of printouts but with no annotation or evidence of alpha testing. The developed solution does not fulfil the design specification. There is some evidence of system development.

(ii) Testing [14 marks]

An attempt should be made to show that all parts of the system have been tested, including those sections dealing with unexpected or invalid data as well as extreme cases. Showing that many other cases of test data are likely to work – by including the outputs that they produce – is another important feature. Evidence of testing is essential. The beta testing should cover all aspects of the test plan produced in the design section, which should cover all aspects of the design specification. The examiner must be left in no doubt that the system actually works in the target environment. This evidence may be in the form of hardcopy output (possibly including screen dumps), photographs or any format that does not require access to any specific hardware or software. The end user(s) must be involved in this process and evidence of end-user testing is required.

11–14 marks	The testing covers as many different paths through the system as is feasible, including valid, invalid and extreme cases. The testing covers all aspects of the design specification and the test plan from the design section. There is clear evidence of end-user testing.
8–10 marks	There is evidence of testing covering most aspects of the design specification but with omissions, eg test data does not include erroneous data for all tests or there is limited evidence of end-user testing.

5–7 marks There is limited evidence of testing based on a badly developed test plan with clear omissions. There is no description of the relationship between the test plan and the testing in evidence.

1–4 marks A collection of hardcopy test run outputs with no clear link to the test plan and covering few aspects of the system. No evidence of end-user testing.

(d) Documentation [10 marks]

Quality of Written Communication is assessed in this documentation. Much of the technical documentation will have been produced as a by-product of design and development work and also as part of writing up the report to date. The software solution should also include sufficient on-screen help to enable the end user to make use of the system. Some additional supporting documents will be necessary including initial set-up, getting started and troubleshooting guides, to ensure the end user can implement the solution.

8–10 marks Candidates will provide detailed and accurate documentation. The documentation will be well presented, in a structured and coherent format. The documentation will cover all aspects of the system, with no omissions, including installation, typical use, troubleshooting, and backup. The on-screen help and supplementary documentation makes a complete guide to the solution and is well presented and easy to follow. Subject-specific terminology will be used accurately and appropriately. There will be few, if any, errors of spelling, grammar and punctuation.

4–7 marks Candidates will provide clear documentation. The documentation will be well presented. There is clear on-screen support to enable the end user to use the system. The supporting documentation and on-screen help is well presented and covers most aspects of the system operation with only one or two omissions, eg troubleshooting or backup. Some subject-specific terminology will be used. There may be occasional errors of spelling, grammar and punctuation.

1–3 marks Candidates will provide superficial documentation, with weak supplementary user documentation covering few aspects of the system. The information will be poorly expressed and limited technical terms will be used. Errors of grammar, punctuation and spelling may be intrusive.

(e) Evaluation [10 marks]

(i) Discussion of the degree of success in meeting the original objectives [4 marks]

This discussion should demonstrate the candidate's ability to evaluate the effectiveness of the completed system. The original objectives stated in the requirements specification should be matched to the achievements, taking into account the limitations. User evaluation is also essential and should arise from direct user evaluation.

3–4 marks A full discussion, taking each objective mentioned in **(b) (i)** and explaining the degree of success in meeting them (indicating where in the project evidence can be found to support this), or reasons why they were not met.

1–2 mark Some discussion about a number of objectives, but some omissions or inadequate explanation of success or failure.

0 marks No discussion present.

(ii) Evaluate the user's response to the system [3 marks]

It is important that the user is not assumed to be an expert in computer jargon, so some effort must be made to ensure that the system is user-friendly. It will be assumed that the user will have considerable knowledge of the underlying theory of the business or area being computerised. Clarity of menus, clear on-screen help and easy methods of inputting data are all examples of how the system can be made user-friendly. Here marks are awarded for the degree of satisfaction that the user indicates in the acceptance procedure. Could the system or its results be used? Was the system specification achieved? Do any system faults still exist? The candidate should evaluate the user's response to the final version of the system.

3 marks	The user indicates that the system could be used but there are some faults which need to be rectified. The candidate provides a detailed discussion of how these inadequacies may be dealt with. OR A fully functional user-friendly system has been produced. The user indicates that the system fully meets the specification given in section **(a)**, and there are no known faults in the system.
2 marks	The system is, in the main, user-friendly, but there is room for improvement (eg no on-screen help has been provided). The user indicates that the system could be used but there are some faults which need to be rectified. The candidate has made some limited attempt to discuss how these inadequacies may be dealt with.
1 mark	The system does not meet the design specification and the end user is not able to make use of the system. The candidate briefly discusses these issues in terms of their project management.

(iii) Desirable extensions [3 marks]

As a result of completing the system, the candidate should identify the good and bad points of the final system, highlighting any limitations and necessary extensions to the system, and indicating how the extensions could be carried out.

3 marks	The candidate clearly portrays the good and bad points of the system indicating the limitations, possible extensions and how to carry out the extensions.
2 marks	The candidate clearly identifies good and bad points and any limitations.
1 mark	The candidate identifies the obvious good points of the system and possibly some bad points or limitations.