Analysis (10 marks)

**What your report should contain:**

[Description of the problem 2](#_Toc484978860)

[Problem Recognition 2](#_Toc484978861)

[Problem decomposition 2](#_Toc484978862)

[Computational methods 2](#_Toc484978863)

[Stakeholders 3](#_Toc484978864)

[Further investigation 4](#_Toc484978865)

[Investigation into the current system to find more information about the problem 4](#_Toc484978866)

[Diagrammatic representation 5](#_Toc484978867)

[Existing solutions 5](#_Toc484978868)

[Possible approaches to solving the problem 6](#_Toc484978869)

[Proposed solution 7](#_Toc484978870)

[Scope of solution 7](#_Toc484978871)

[Features of solution 7](#_Toc484978872)

[Limitations of solution 7](#_Toc484978873)

[Requirements specification 8](#_Toc484978874)

[Success criteria 8](#_Toc484978875)

[Developer’s hardware/software spec 8](#_Toc484978876)

[User’s hardware/software spec 8](#_Toc484978877)

# Description of the problem

This could be 2-3 sides of A4 (minimum about which the mark scheme says:

*3.1.1 (a) Describe and justify the features that make the problem solvable by computational methods.*

*(b) Explain why the problem is amenable to a computational approach.*

Your project is of limited scope, it will not cover all areas of an organisation - the scope would be too vast. Narrow down the scope of your project to a specific area. Describe the particular area you will be working in, but giving more specific details.

## Problem Recognition

What is the problem? Describe what the scenario and what the user wants to do. What will your new system do instead?

Problem decomposition

Use top down design (Stepwise refinement or other problem decomposition technique) to break down the problem and use divide and conquer to identify manageable chunks. A diagrammatic representation here is good. You could use a modular overview, UML class diagrams etc.

## Computational methods

What other computational methods could be used to solve parts of the problem and why? Possible techniques that you may consider include:

* Abstraction
* Backtracking
* Data mining
* Heuristics
* Performance modelling
* Pipelining
* Visualisation

# Stakeholders

This short description (½ - 1 side of A4) about which the mark scheme says:

*3.1.2 (a) Identify and describe those who will have an interest in the solution explaining how the solution is appropriate to their needs (this may be named individuals, groups or persona that describes the target end user).*

Sample questions that you could use to complete this section include:

* What is the name of the client/organisation?
* What does it do?
* Where is it located?
* How many people does it employ?
* How many shops/head offices are there?
* What are the names and positions held of people who are going to be involved with the project?
* Who are the intended users?
* How will they use the proposed solution?
* How is it appropriate to their needs?

# Further investigation

Your problem definition outlined the existing system. The Investigation develops this but is about defining your client’s requirements – finding out what they want the new system to do and if alternatives exist. There is obviously no point in creating a new system to fulfil a function carried out by one that is already available. This process of investigation is systems analysis.

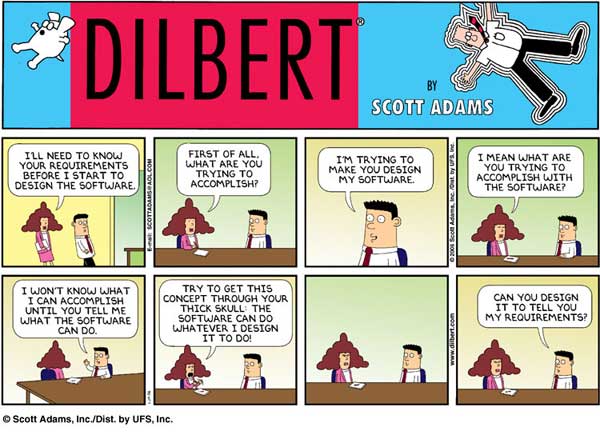
The mark scheme says:

*3.1.3 (a)* ***Research the problem and solutions to similar problems*** *to identify and justify suitable approaches to a solution.*

## Investigation into the current system to find more information about the problem

Firstly, you need to develop an understanding of what the current system is. You need to do some research and find out some detailed information about the system as it currently stands, and the problems with it.

You have to do this by investigating the current system which is in use at the moment by finding out the facts and then recording them for analysis purposes. Remember your intention is to end up with detailed information on the current system and how the problem fits into the current system.



**Fact Finding**

Fact finding is about looking for answers for questions. The questions you should be trying to answer are based around the four areas below:

1. Input
2. Processing
3. Output
4. Storage (file structures)

There are different methods of fact-finding. They include:

* Questionnaires
* Interviews and user meetings/groups
* Observations of the current system in use
* Document analysis

**A sample questionnaire:**

What data is collected for processing?

How does this data originate?

How is it collected and recorded?

How often is it collected?

How is it processed?

Who processes it?

How often is it processed?

What volumes of data are processed?

Is the data processed in batch mode or on demand?

What information is produced?

Who receives the information?

How is it transmitted/presented?

How often is the information provided?

What is the information used for?

How often is the information used?

How reliable and accurate is the information?

Is the information presented clearly, so that it can be easily understood?

What master files/reference files are kept, and how often are they updated?

Do users have on-line enquiry facilities to access these files?

What does the system cost to operate?

What benefits does the system provide?

When you plan your questionnaire, you must think about the possible responses that you may get from the interviewee. This means you will be able to plan follow-on questions, thus gathering more in-depth information.

Do not be afraid in the interview to sketch up a quick diagrams representing how you perceive the system to work. Doing it while you have the user there and the information is fresh in your mind is more beneficial than chasing them up later.

Make use of as many closed and multiple choice questions as possible to narrow down the responses that you get.

## Diagrammatic representation

Once you have found the information, it must be properly recorded so that it can be analysed. You may often use these records to communicate your understanding with the user to make sure that your mental model matches theirs. Methods of fact recording include:

* Flowcharts
* Textual descriptions
* Graphs and charts
* Relevant system diagrams:
* Entity-Relationship diagrams
* Use-Cases
* State Transition diagrams
* Data Flow diagrams

## Existing solutions

Look at existing solutions (provide screenshots of their interfaces) discussing what purpose they serve, how they work, any good points about that solution (to use as takeaways) and any limitations (that you can address in your *better*  software).

# Possible approaches to solving the problem

The mark scheme says:

*3.1.3 (a) Research the problem and solutions to similar problems to* ***identify and justify suitable approaches to a solution.***

What possible methods could you use to solve the problem?

This relates to:

* The technology: What language? What OS? Will your solution be web or desktop based? Why?
* The computational techniques that could be used. There may be several ways that the problem could be approached, what are they, how could they be used?

# Proposed solution

The mark scheme says:

*3.1.3 (b) Describe the essential features of a computational solution explaining these choices.*

*(c) Explain the limitations of the proposed solution.*

## Scope of solution

Provide a brief overview of the boundaries of your solution, i.e. what you will and will not be implementing. The scope of a complete solution may be beyond that of an A Level project. The following two section should provide more detail.

## Features of solution

Based on the research that you have completed, explain what the features of the solution are. Each feature must be related to your research, no feature must be included without having a specific reason for inclusion. The discussion of the features for inclusion must also refer to the different computational techniques outlined above, explaining how/why the final technique chosen is appropriate.

## Limitations of solution

Explain what your solution is not intended do. Why have you decided to not implement particular features?

# Requirements specification

The mark scheme says:

*3.1.4 (a) Specify and justify the solution requirements including hardware and software configuration (if appropriate).*

*(b) Identify and justify measurable success criteria for the proposed solution.*

You should now know what the problem is and it is now time to begin to develop the solution. There are two parts to the solution:

1. The requirements of the solution
2. The approach you are going to use to develop the solution – what software and hardware

## Success criteria

The mark scheme states that the requirements must be justified. This means that it must be possible to backtrack from each of the requirements listed, to the conclusions you have drawn from the investigation, back to the data/problem itself. If you cannot justify a requirement with evidence, you either remove the requirement, or document the evidence for that requirement.

You must produce a list of requirements which are agreed with, and signed-off, by the client. These must be **specific**, **measurable** targets. You cannot have criteria along the lines of “it must be easy to use”; you must be able to demonstrate that you have met the criteria so wording such as “The user is able to output letters to all students who have failed to meet their deadline”.

It may be necessary to redo this section several times, going back your user each time to get more evidence or clarify the requirements. This is normal in the real world so do not fall to the temptation of making it seem as though you got it right the first time, just make sure you document your refinements. The best thing to do is to include the updated requirements spec in between the relevant prototypes, signed off by your client.

## Developer’s hardware/software spec

Think about the software that you will need to develop and test the solution. Based on this, provide a minimum hardware and software specification that will be needed in order to create the solution.

## User’s hardware/software spec

How will your solution be deployed? Based on this, provide a minimum hardware and software specification that will be needed in order to run the solution.