

# Analysis

There are **12 marks** available for this section.

## Introduction

In this section you must define **precisely** what your proposed system will do. To do this you will need to:

- Identify the client
- Describe the current system (if any)
- Define the problems with the current system
- Outline (multiple) possible solutions to solve these problems
- Describe possible problems with implementing each possible solution
- Identify what the system must do
- Realistically appraise possible solutions
- Select and justify a solution

The following sections will help you to produce a framework for your Analysis.

## Identify the client

Here you should describe who the system is for, what they do and what kind of organisation they work for. You should also explain why they have an interest in developing this system. In addition, make sure:

- that the role of the client is clear
- that you link the role of the client to the system that is going to be developed

## Describe the current system and define the problems.

In order to describe the current system effectively you will need to know more about it. You will do this by interviewing the client. During the interview you will also discover the problems with the current system.

You will need to at least one and possibly several meetings with your client in order to establish the detailed requirements for the project.

The key to a successful interview is preparation.

Before you meet your client you should prepare a list of questions you want answered so that you are focused on getting the information required during your meeting. Never meet the client for a vague 'chat' as you will come away with more questions than answers!

In addition, give your client a copy of the questions ahead of time so that they can prepare effectively as well - this will make your meeting much more productive.

#### Interview checklist

This checklist of points may help you develop an appropriate set of questions for the interview:

Here is a checklist of points you may want to cover in an Interview:

- Objectives: what is the proposed system to do?
- What are the problems with the current way of doing things?
- What data or information is recorded in the current system?
- How much data is recorded at present?
- What data or information is to be recorded in the proposed system? How much data will the proposed system record?
- How frequently will the data need to be updated?
- Will new records need to be added or old ones deleted? How often?
- Will the changes come in batches or in ones and twos?
- How important is the data or information that is recorded?
- What processes or functions are performed by the current system?
- What processes or functions are to be performed by the new system?
- When should they be done and where?
- What special algorithms do these processes use?
- Which processes should be executed manually?
- What are the inputs to the current system?
- What inputs are required for the proposed system?
- What are the outputs from the current system?
- What outputs will be required from the proposed system?

- Are hard copies required?
- How often will outputs be required?
- What computing resources does the client possess?
- Is the client prepared to purchase hardware/software resources?
- Is security an issues?
- Should there be restricted access to particular areas?
- How are exceptions and errors handled in the current system?
- What errors and exceptions should be reported in the proposed system?
- How should they be reported?
- Are there any constraints on hardware, software, data, methods of working, cost, time, etc.?
- Does the user have a particular solution in mind?

The purpose of the Analysis interview is not design.

During your initial interview with the client focus on discovering what the system is expected to do, **not** how it is supposed to do it. Leave discussions about the design and user interface until a later date - once you have a clear idea of the different inputs, processes and outputs of the system.

## Summarise

From the client interview you will be able to produce a summary which describes the current system and how it works, as well as defining and explaining the problems that the current system has. Keep in mind that it should be possible to **visualise exactly what the current system does** from your description - be as detailed as possible.

In relation to the problems you should clearly list and explain each problem. Make sure that you explain *why* each problem is an issue. Using bullet points to structure this section is helpful.

You should always include your original interview notes as an appendix to your document and look to get copies of any documentation from the client that they use with the current system: order forms, invoices, letters etc. as these will be useful later on during the design stage.

## Investigation

This section should focus on describing both the **current** and **proposed** systems in detail. You should have separate sub-sections for each system.

## Data sources and destinations

This **table** should provide an overview of where the data for the system is coming from (source) and where it ends up (destination). You need to make sure that all data items are identified individually e.g. **first name**, **last name** etc. rather than just **name**.

Think carefully about both the sources and destinations - does the user pass data to the client directly or is there another destination e.g. a form.

The table should be structured in the following way:

| Source | Data                        | Example Data | Destination  |
|--------|-----------------------------|--------------|--------------|
| User   | Amended Patient Information | First Name   | Receptionist |

## Algorithms

This section can be quite challenging as it is difficult to identify algorithms, especially in the current system. Keep in mind that all systems have algorithms, even if they are **manual systems**.

When you have identified some algorithms for your system you should attempt to express them in **pseudo-code** and provide any other annotation that you think is appropriate.

## Data flow diagrams

A data flow diagram (**DFD**) is a good way of summarising the sources and destinations of data and the processing that takes place. These diagrams show how data is **transformed** into information and what data needs to be stored.

A **DFD** will always have sources and destinations and generally the data will be **processed** at some stage in the system. Be sure to use the correct symbols (see **page 236** A2 Bond) and to label all of the arrows appropriately. If required split the diagram up into sub-diagrams so

that it is easier to understand the system.

## Forms

In this section you should include any example **forms, invoices** etc. that you have been provided with to help with your understanding of the system.

You should annotate these forms to explain why you are including them. Remember that they might help inform the design of your user interface later in the project.

## Data Dictionary

A data dictionary is a table that stores meta-data. It stores the names of data items (fields or variables), data types, length, validation criteria and other characteristics such as usage, physical representation, ownership, authorisation and security. It will also show which programs or modules read or write the data.

Below is an example of a data dictionary:

| <b>Name</b>    | <b>Data Type</b> | <b>Length</b> | <b>Validation</b>    | <b>Example Data</b> | <b>Comment</b>  |
|----------------|------------------|---------------|----------------------|---------------------|---|
| SubjectRefCode | String           | 5 chars       | 5 digits, must exist | 54821               | Unique to each subject                                  |
| CentreNo       | Integer          | 4 bytes       | 10000 to 80000       | 12345               | Unique to each centre                                   |
| CandidateNo    | String           | 4 chars       | 4 digits             | 2345                | Unique to each candidate within a centre                |
| CandidateName  | String           | 25 chars      |                      | BLOGGs, Joe Fred    | Full name, Surname in capitals, other names after comma |

## Volumetrics

Volumetrics is an assessment of the volume of data that a system will be required to process and store. For instance: The system should be able to store 200 member records initially and cater for expansion to 1000 members.

## Objectives

Here you should produce bullet point lists which cover the objectives of the project.

- **General objectives** e.g. Clear and easy to use menu structure
- **Specific objectives** e.g. The user should be able to edit customer details
- **Core objectives** i.e. Those specific objectives that the system must do
- **Other objectives** i.e. Those specific objectives that it would be desirable for the system to do if there is enough time available.

## Entity Relationship Diagrams

In this section you must produce diagrams which show the relationships between the data stores (entities) in the proposed system.

## Entity Descriptions

For each of the entities in your ER diagrams you should provide an appropriate entity description showing all **attributes** and **keys**.

## Object Analysis

You must produce an object analysis, which consists of three stages:

- Determine the objects in the problem domain
- Determine the relationships between the objects
- Determine the attributes and the behaviours of each object

## Determine the objects

Think about the the problem and identify possible objects. Examples of objects include: Book, Person, Order, Car.

## Relationship Diagrams

Once you have the objects you must determine the relationships between these objects. Refer to **pages 241-242** of A2 Bond for further details on this.

## Class Definitions

Finally you should define each class - what attributes does a particular class have? what methods will it require? Refer to **pages 242** of A2 Bond for an example.

## Other abstractions (Graphs)

Depending on your problem you may find it helpful to use other graphical representations to help aid understanding of the problem. This is difficult to discuss in general terms but if you think something could be better described in a graphical manner, discuss this with your teacher.

## Constraints

In any project there are various limitations that must be described:

- **Hardware** - what kind of hardware is available? How does this impact the project? What recommendations would you make if additional hardware is required?
- **Software** - state whether the client has stated any preferences or has it been left to you to choose
- **Time** - has the client given you a deadline or is it the project deadline set by your teacher?
- **Knowledge** - how does the user's knowledge of IT constrain the project (if any)?
- **Access rights** - who will be allowed to use the various parts of the system?

## Alternatives and solutions

Look at the possible methods of solutions such as writing a program, customising a database or creating a spreadsheet or web application. List the advantages and disadvantages of each potential solution considering details such as user interface, data storage and manipulation. This should lead to a **justification of your chosen solution**.

## Mark Scheme

| 0-3 marks  | 4-6 marks  | 7-9 marks                                   | 10-12 marks   |
|--|--|---|---|
| Some evidence of an investigation, but somewhat lacking in structure | Evidence of a structured investigation, but with gaps that hinder full understanding | Evidence of a well structured investigation | Evidence of an extensive, well-structured investigation |



| <b>0-3 marks</b>                                      | <b>4-6 marks</b>  | <b>7-9 marks</b>   | <b>10-12 marks</b>  |
|---|---|--|---|
| Some attempt at documenting system requirements       | The set of system requirements falls short of being comprehensive by omission or lack of depth so that only one third to one half of the comprehensive set of system requirements are covered | The set of system requirements falls just short of being comprehensive because of some omissions or some lack of depth | A comprehensive set of system requirements, fully meeting a real end user's needs |
| Some attempt at understanding a real end user's needs | Real end user's needs addressed, but with some omissions so that only one third to one half are considered  | The majority, but not all of a real end user's needs are addressed   | High level of understanding of a real end user's needs                            |
| Some attempt at documenting SMART objectives          | The set of SMART objectives falls short of being comprehensive by omission or lack of depth, so that only one third to one half of the comprehensive set are covered                          | The set of SMART objectives falls just short of being comprehensive because of some omissions or some lack of depth    | A comprehensive set of SMART objectives to the necessary depth                    |