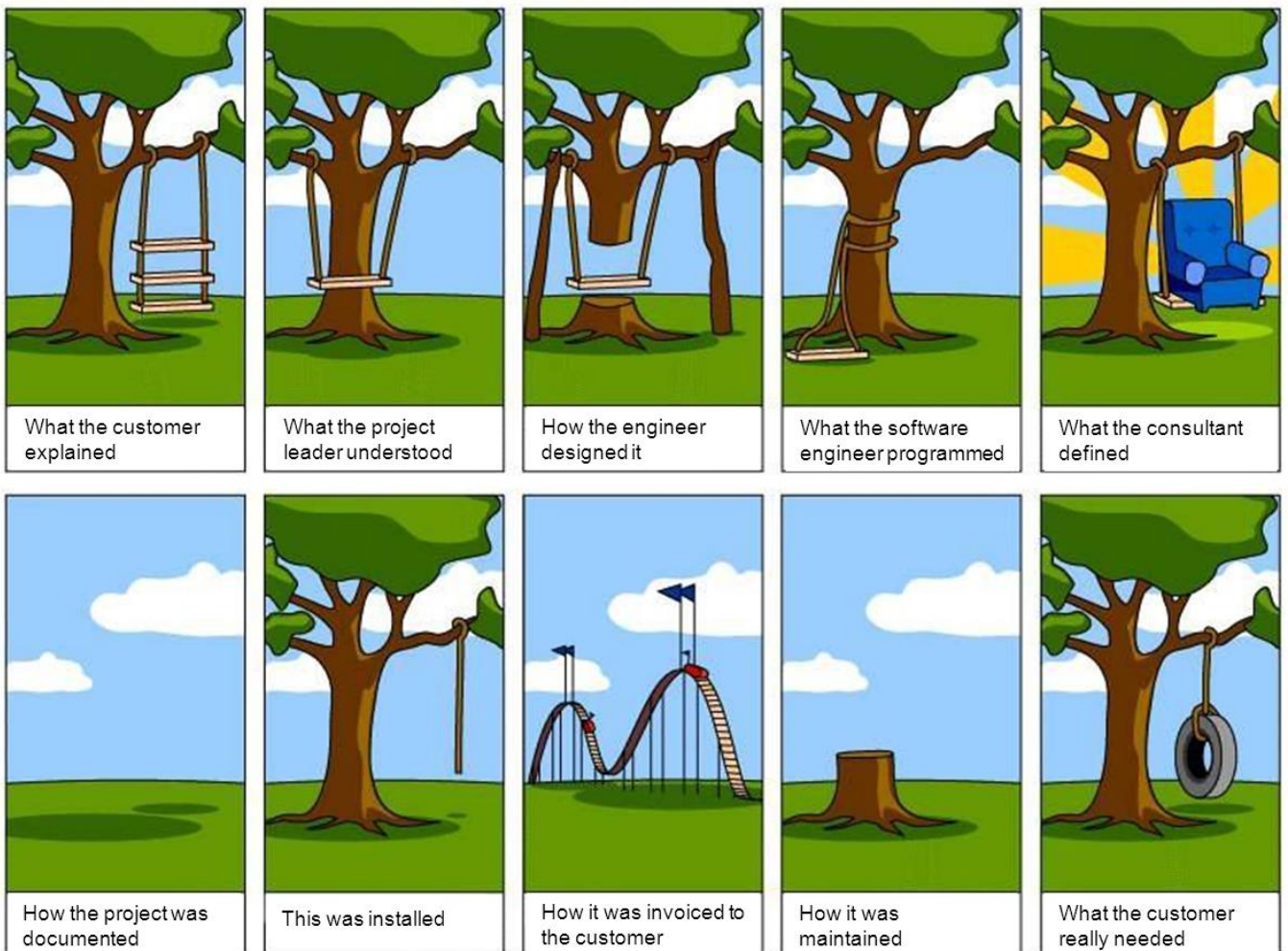


Advanced Higher Computing Science

SDD: Analysis



SDD: Analysis: Contents

SDD: Analysis: Contents	1
Purpose	2
Feasibility Study	3
Economic Feasibility	3
Time Feasibility (sometimes called “schedule feasibility” in older SQA docs)	4
Legal Feasibility	4
Technical Feasibility	5
User surveys	6
Determining End-User Requirements for your AH project	6
Survey Design (not in AH specification)	6
Project Planning	8
Scheduling: Gantt Charts.....	9
Scheduling: Critical Path Analysis (not explicitly in AH but useful).....	11
Resources	14
Requirements Specification	15
Purpose	15
End User Requirements	15
Scope, Boundaries and Constraints (also Assumptions).....	15
Functional Requirements	19
SQA Examples	21
Appendix 1: problem analysis (SDD, DDD and WDD)	21
UML.....	25
What Is UML?.....	25
Why Use UML?.....	25
What Does Modelling Mean? (not explicitly in AH but you need to understand this).....	25
Purpose of Use Case Diagrams	26
How to Draw Use Case Diagrams.....	26
Use Case Questions.....	32
Analysis Questions	37
SQA Questions	38
OLD PRELIM Questions	43
SQA Exemplar Projects and Marker Commentary: Analysis.....	45
Candidate 1 Evidence.....	45
Candidate 1 Commentary	52
Candidate 2 Evidence.....	53
Candidate 2 Commentary	58

Purpose

The first stage of software development is making sure that you have fully understood the problem that you are being asked to solve and what your solution is supposed to achieve.

In identifying the purpose, you are defining the problem and describing what your completed solution will be able to do.

You are clearly describing what the software will be used for.

Do not forget to include a description of:

- what is it to do
- who will be using it
- why they will be using it
- the main input(s) and output(s).

Note that this is a description of *what* your solution will do rather than *how* it will do it.

At this stage, it is NOT a detailed technical description of how your solution will work, rather, think of it more like describing your solution from the point of view of the client/user with a high-level description of the overall goal and objectives.

For your AH project, do not forget to discuss integration. It would also be helpful to describe how your project is meeting the AH project requirements i.e. the specific project criteria that you are meeting.

Feasibility Study

The creation of a new system can take months (or even years) and cost thousands (or millions!) of pounds. The basic purpose of a feasibility study is to ascertain whether such expenditure of time and money is likely to prove worthwhile and whether the objectives of the problem definition can actually be realised – what some companies want might be unrealistic and unachievable.

A feasibility study should be relatively cheap and quick to carry out. It should be complete in itself and the client does not have to commit to any further expenditure or long-term development by commissioning one – the client has no legal or contractual obligation to take developments further.

Another advantage of a feasibility study is that it can present the client with a range of possibilities from which a choice can be made. Often, several different solutions will be considered in a feasibility study before a recommendation is made.

Economic Feasibility

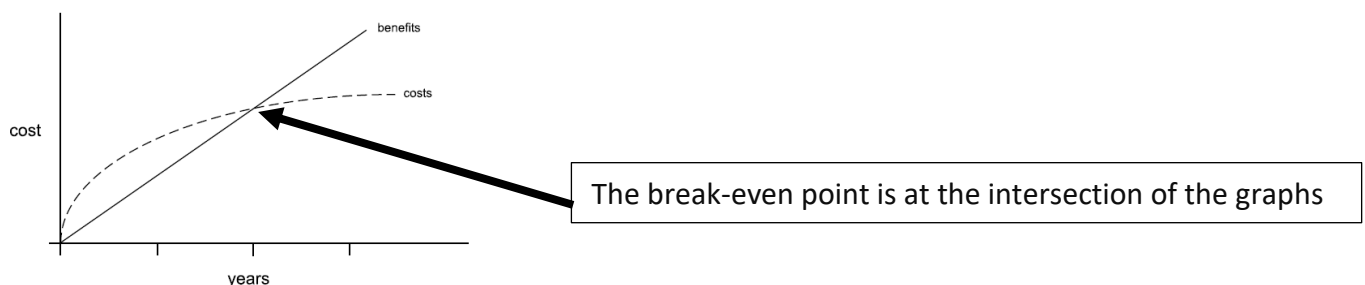
This deals with the **cost implications** involved. Management will want to know how much each option will cost, what is affordable within the company's budget and what they get for their money.

A **cost-benefit-analysis** is part of the budgetary feasibility study. If the project is not cost-effective then there is no point proceeding. Setting up a new computer system is an investment and involves capital outlay.

The costs of a new system include:

- the costs of acquiring it in the first place (consultancy fees, program development including cost of any resources required for development, etc.)
- the costs of installing it (disruption of current operations, cost of new equipment, alteration of workplace, etc.)
- the costs of maintaining it which also includes training.

In the long term, management will also want to know the '**break-even point**' when the new system stops costing money and starts to make money. This is extremely difficult to quantify. However, an accurate estimate of a system's operational life span is a valid option and will rely solely on the knowledge and experience of the systems analyst involved.



Tangible benefits that management would certainly be looking for in the new system would be:

- reduced running costs
- increased operational speed
- increased throughput of work
- better reporting facilities

Note that not all the costs and benefits lend themselves to direct measurement. These are called *intangible benefits* e.g. new systems generally affect the morale of the staff involved, for good or ill.

Time Feasibility (sometimes called “schedule feasibility” in older SQA docs)

Obviously, time is a main factor in the development of a new system.

Questions to be asked at this stage might include:

- How long will the proposed system take to develop?
- Will it be ready within the specified time-frame?
- When is the best time to install?

For example, a project might have to start within six months; assuming it would take three months to purchase and install the required hardware and software and a further six months to train the end users. Such a project is not technically feasible because of shortage of time so it would not go ahead unless some of the time constraints were reviewed and changed.

Legal Feasibility

This has to do with any conflicts that might arise between the proposed system and legal requirements. For example:

- How would the new system affect contracts and liability?
- Are health and safety issues in place?
- Would the system be legal under laws such as GDPR?
- Would the system be legal under other laws local to the UK or wider jurisdictions where your software may be used or used to store data from inhabitants of those countries e.g. EU or USA?
- What are the software licensing implications for the new system?

Software licensing can be quite a thorny problem. Licences can be purchased as: client licence (per seat), server licence, network licence or site licence and the period of operation may be annual or perpetual. Software vendors vary in their licensing regulations so this has to be fully investigated.

Technical Feasibility

The feasibility study must ascertain what technologies are necessary for the proposed system to work as it should. Perhaps suitably advanced technologies do not yet exist. Unless it is the object of the project to design a system to use such advanced technology, this would rule the project out as being a non-starter. It would be a foolish move for a feasibility study to evaluate technologies which are either under development or undergoing testing.

Given that suitable technology does exist, the feasibility study must establish if the organisation already has the necessary resources. If not, the feasibility study must make clear what new resources the organisation would have to acquire. This will also involve determining whether the hardware and software recommended will operate effectively under the proposed workload and in the proposed environmental conditions. Will new staff be required or will existing staff require training?

The development of a new system involves risks of one kind or another. Every understanding that might be reached could carry the risk of some misunderstanding:

- software companies and their clients often have different vocabularies and consequently they appear to be in perfect agreement until the finished product is supplied.
- management may have unrealistic visions of computer systems.

The feasibility study is where idealism meets reality.

Further issues might include the training of personnel to use the new system, consideration of service contracts, warranty conditions and the establishing of help desk facilities for inexperienced users.

User surveys

As part of the analysis process, it is important to find out what your end-user requirements are. User research is important because in order to design a successful solution, you need to know who your users are and what they want to achieve with the potential system.

You need to identify the users' goals, needs and capabilities as well as features that they like/loathe in similar existing systems and new features that they would like to see included in your development.

- The results of your user survey will determine the “End-User Requirements” section of your requirements specification document. This section outlines what the users want to be able to do with the new system.
- You may also encounter ideas and suggestions that you had not thought of and this may provide additional content for the scope and boundaries sections of your requirements specification.
- The end-user group is important because their requirements will help to determine the user-interface design and these requirements will also have an influence on your test plan.

Determining End-User Requirements for your AH project

You could make a paper-based questionnaire or an electronic one using SurveyMonkey (<https://www.surveymonkey.co.uk>).

Examples of user surveys for an AH project here:

<https://www.surveymonkey.co.uk/r/3XL2825>

<https://www.surveymonkey.co.uk/r/8B3XZLL>

Note that they lack sufficient detail for AH level (particularly about the user completing the questionnaire e.g. age range; experience level; gender etc. which may have an influence on their responses) and contain spelling mistakes. The second survey is very brief and vague – I doubt much useful information could be collected from it.

Survey Design (not in AH specification)

Surveys normally consist of two types of questions:

Closed Questions: These questions get the quantitative data from users. Responses to closed questions do not tell you about the context, motivation or reason for the response. These questions are normally responded to via a checkbox, radio button or drop-down menu. The data obtained can be easily visualized with the help of graphical representations.

Open Questions: These questions provide the qualitative data about a user's behaviour, actions etc. It tells us how the user thinks about a problem by providing them with an opportunity to explain their reasoning. These questions required a text box to explain the cause. Qualitative responses tend to take a lot longer to analyse but provide rich details.

HINTS:

- Try to ask neutral questions and avoid leading questions
- Keep the language and context clear and unambiguous
- Avoid grammatical and spelling errors
- Be clear about your reasons for doing the survey and what the data will be used for
- Respect respondents' privacy and anonymity. For your AH project, knowledge of your end-users' ages and genders may be important or it may not. Do not collect unnecessary data.
- If questions are not applicable to some respondents e.g. because of their answer to the previous question, make this clear.
- Use open questions to obtain detailed qualitative responses e.g. if you were asking users if they used a similar product to the one you are going to develop for your AH project and you ask the question "Do you use x?", the answer would be yes or no. Analysis of the results of this question would result in a quantitative number. If you ask "How would you describe your experience of using x?", you will get descriptions explaining features they liked/loathed, reasons why they haven't used it at all, features they would like to see added etc. This will give you excellent qualitative feedback that you can incorporate into your Requirements Specification (e.g. as part of end-user requirements, scope or boundaries)

Project Planning

If the client agrees to the suggested solution in the Feasibility Study and the results of the User Survey suggests that the proposed solution is one that would meet End-User Requirements, the developer will then carry out a more detailed investigation which will result in the Requirements Specification document.

The Requirements Specification document forms the basis of a legally binding contract between the developer and the client. In very simple terms, it will state that the project will be completed according to the agreed specification within a certain time period and for a certain cost. As soon as the contract is agreed and signed by both the client and the developers, the clock starts ticking.

The contract may also include penalty clauses e.g.

- *If the project takes longer to develop than agreed, then there may be a financial penalty that the developer has to pay;*
- *If the client wants changes made to the specification during the development, then the developers will charge the clients more for the extra work involved in making the changes.*

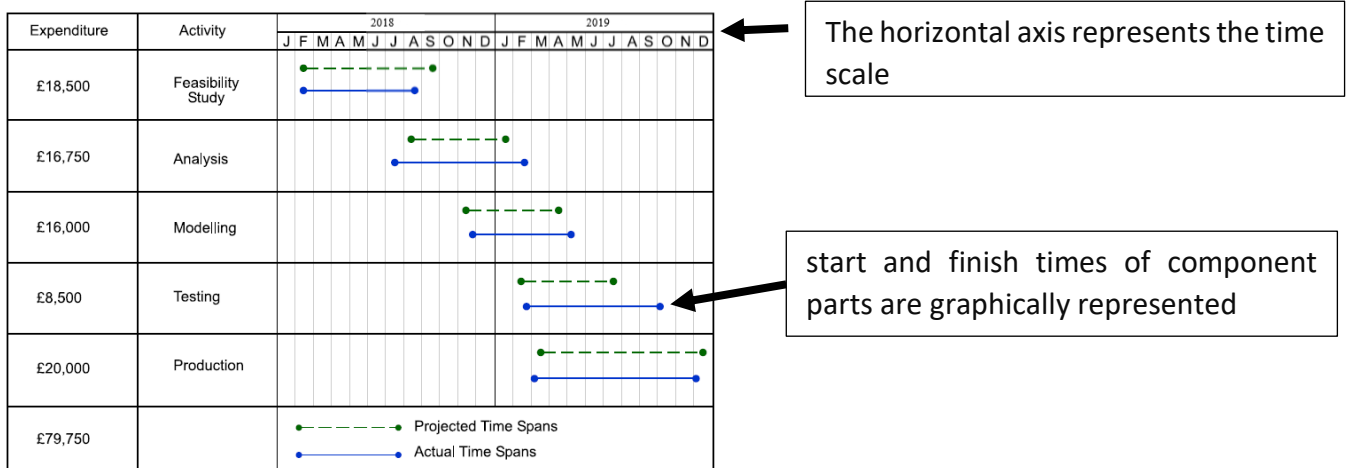
It may also allow for a bonus if the developer completes and delivers the project ahead of time.

It is vital that the project is developed in an orderly manner – this requires careful **planning** to organise the human and physical resources needed to complete the project.

Large project developments will involve the developers being divided into small teams. The project itself will be divided into many small parts, each assigned to a team. **Scheduling** involves making sure that no team is left idle waiting for another team to finish their part.

Scheduling: Gantt Charts

In many cases of project management, the scheduling component can be aided by means of a **Gantt chart**. A Gantt chart is used to plan and schedule projects involving several **concurrent tasks**.



Advantages of a Gantt chart:

- Shows at a glance the progress of a project.
- Helpful for working out the order in which tasks need to be carried out (the **schedule**).
- Helpful for managing the **dependencies** between tasks (which parts depend on earlier parts).
- Helpful for working out which parts can be completed **simultaneously** (the parallel activities) because they have no dependencies at that particular point in the development.
- Allow you to identify the **resources** (see p14) needed to complete the project, along with the times when these resources will be needed
- Help you work out the quickest possible time in which a project can be completed
- Help you identify the "critical path" for a project. This is the sequence of tasks that must be completed on time if you are to complete the project by a particular date.
- When a project is under way, a Gantt Chart will help you to monitor whether the project is on schedule. If it is not, it will help you to pinpoint the remedial action necessary to put it back on schedule.

Using a Gantt chart makes it easy to identify which activities are **sequential** and which are **parallel**. An essential concept behind project planning (and Critical Path Analysis as well) is that some activities are dependent on other activities being completed first e.g. it is not a good idea to start building a bridge before you have designed it! These dependent activities need to be completed in a sequence, with each stage being completed before the next activity can begin. These dependent activities are called *sequential* or linear.

Other activities are not dependent on completion of any other tasks. These may be done at any time before or after a particular stage is reached. These are nondependent or *parallel* tasks.

Rather than hand-drawing a Gantt chart, you can use a Gantt chart tool instead. This is a piece of software that draws the chart for you e.g. GanttProject (free download from <https://www.ganttproject.biz/>) or Microsoft Project (available as part of Office 365).

Watch: Introduction Into GanttProject 2.5 (15:05): <https://youtu.be/5rHCSa5ad34?t=15>

Download: GanttProject for Beginners – GanttProject Support:

<https://help.ganttproject.biz/uploads/default/original/1X/ae5fa52935b40a26960ee3c350dbbb311822b542.pdf>

(also available here: X:\Computing\S6 - AH Computing Science\New AH SDD 2019)

For your project:

1. List all activities in the plan

The first step is to list all of the tasks that need to be completed to deliver the project. For each task, show the earliest start date, estimated length of time it will take (and/or estimated due date), and whether it is parallel or sequential. And if tasks are sequential, show which previous stage or stages they depend on.

Task	possible start	Length	Type	Dependent on...
1. High level analysis	week 1	5 days	sequential	
2. Selection of hardware platform	week 1	1 day	sequential	1
3. Installation and commissioning of hardware	week 3	2 weeks	parallel	2
4. Detailed analysis of core modules	week 1	2 weeks	sequential	1
5. Detailed analysis of supporting utilities	week 1	2 weeks	sequential	4
6. Programming of core modules	week 4	3 weeks	sequential	4
7. Programming of supporting modules	week 4	3 weeks	sequential	5
8. Quality assurance of core modules	week 5	1 week	sequential	6
9. Quality assurance of supporting modules	week 5	1 week	sequential	7
10. Core module training	week 7	1 day	parallel	6
11. Development of accounting reporting	week 6	1 week	parallel	5
12. Development of management reporting	week 6	1 week	parallel	5
13. Development of management analysis	week 6	2 weeks	sequential	5
14. Detailed training	week 7	1 week	sequential	1-13
15. Documentation	week 4	2 weeks	parallel	13

2. List resources that each activity requires e.g. people, equipment, new knowledge etc.

3. Use a Gantt chart tool to create your Gantt chart.

TIP: Be prepared for unforeseen problems with your project – try to build some flexibility into your scheduling. Any delay will have a ‘ripple’ effect potentially causing even more delays further down the time line.

The development company’s managers will monitor progress daily, updating and assigning resources to try to keep the project ‘on time’ and ‘on budget’.

TIP: GanttProject has an option for you to export to CSV so that you can analyse your data in a spreadsheet. You can also generate a pdf report. See p8 of the Intro for Beginners pdf.

Scheduling: Critical Path Analysis (not explicitly in AH but useful)

As with Gantt Charts, Critical Path Analysis (CPA) helps you to plan all tasks that must be completed as part of a project. The critical path is the sequence of linked tasks that has the longest total duration. The critical path has no 'slack' and determines the end date of the project.

Critical Path Analysis is an effective and powerful method of assessing:

- What tasks must be carried out.
- Where parallel activity can be performed.
- The shortest time in which you can complete a project.
- Resources needed to execute a project.
- The sequence of activities, scheduling and timings involved.
- Task priorities.
- The most efficient way of shortening time on urgent projects.

The advantage of using CPA over Gantt Charts:

- CPA formally identifies tasks which must be completed on time for the whole project to be completed on time (these are the tasks on the critical path)
- CPA also identifies tasks which can be delayed for a while, if resources need to be redeployed to catch up elsewhere.

The disadvantage of CPA compared to Gantt charts is:

- the relation of tasks to time is not as immediately obvious as with Gantt Charts. This can make them more difficult to understand.

As with Gantt Charts, the essential concept behind Critical Path Analysis is that you cannot start some activities until others are finished. These activities need to be completed in a sequence, with each stage being more-or-less completed before the next stage can begin. These are *sequential* activities.

Other activities are not dependent on completion of any other tasks. You can do these at any time, before or after a particular stage is reached. These are non-dependent or *parallel* tasks.

For your project (remember that Critical Path Analysis is not a requirement):

1. List all activities in the plan

The first step is to list all of the tasks that need to be completed to deliver the project. For each task, show the earliest start date, estimated length of time it will take (and/or estimated due date), and whether it is parallel or sequential. And if tasks are sequential, show which previous stage or stages they depend on.

Task	possible start	Length	Type	Dependent on...
1. High level analysis	week 1	5 days	sequential	
2. Selection of hardware platform	week 1	1 day	sequential	1
3. Installation and commissioning of hardware	week 3	2 weeks	parallel	2
4. Detailed analysis of core modules	week 1	2 weeks	sequential	1
5. Detailed analysis of supporting utilities	week 1	2 weeks	sequential	4
6. Programming of core modules	week 4	3 weeks	sequential	4
7. Programming of supporting modules	week 4	3 weeks	sequential	5
8. Quality assurance of core modules	week 5	1 week	sequential	6
9. Quality assurance of supporting modules	week 5	1 week	sequential	7
10. Core module training	week 7	1 day	parallel	6
11. Development of accounting reporting	week 6	1 week	parallel	5
12. Development of management reporting	week 6	1 week	parallel	5
13. Development of management analysis	week 6	2 weeks	sequential	5
14. Detailed training	week 7	1 week	sequential	1-13
15. Documentation	week 4	2 weeks	parallel	13

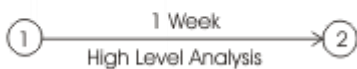
2. Plot the activities as a circle and arrow diagram

Critical Path Analyses are presented using circle and arrow diagrams.

In these, circles show events within the project, such as the start and finish of tasks. Circles are normally numbered to allow you to identify them. An arrow running between two event circles shows the activity needed to complete that task. A description of the task is written underneath the arrow. The length of the task is shown above it. By convention, all arrows run left to right.

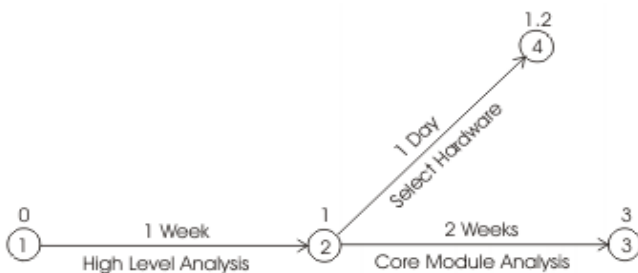
An example of a very simple diagram:

Figure 2: Simple Circle and Arrow Diagram



This shows the start event (circle 1), and the completion of the 'High Level Analysis' task (circle 2). The arrow between them shows the activity of carrying out the High Level Analysis. This activity should take 1 week.

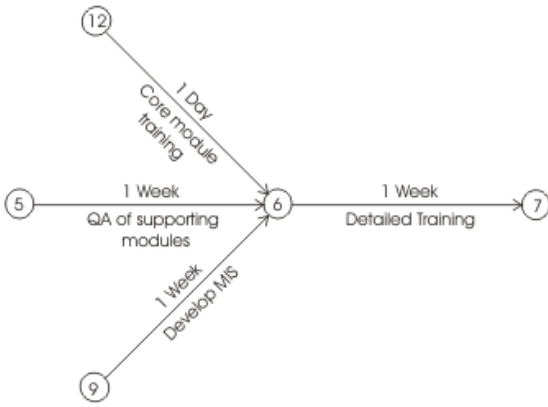
Where one activity cannot start until another has been completed, you start the arrow for the dependent activity at the completion event circle of the previous activity. An example of this:



Here the activities of 'Selecting Hardware' and 'Core Module Analysis' cannot be started until 'High Level Analysis' has been completed.

Circle and Arrow Diagram showing two activities that cannot be started until the first activity has been completed.

Another example:



Here, activity 6 to 7 cannot start until the other three activities (12 to 6, 5 to 6 and 9 to 6) have been completed.

TIP: GanttProject has an option to display the critical path but it does not use standard Critical Path Analysis notation.

Resources

Resources include particular people, equipment, additional information (e.g. do you need to ascertain particular facts? Where from?), additional knowledge (e.g. do you need to learn new programming language syntax? Do you need to purchase a new book to do this?) etc. that you require in order to successfully develop your solution.

Examples (not exhaustive!):

- Access to:
 - Individuals
 - Organisations
 - Web resources
- Hardware
- Software
- Data storage requirements
- Access to Web Services e.g.
 - Web server
 - Database server
 - Mail server
- Backup strategy

Requirements Specification

The requirements specification is how your solution will work rather than just what it will do. It should contain the technical detail lacking in the “purpose” section.

Note: The Requirements Specification document is often the basis of a **legal contract** between the client (customer) and the software company writing the software. It therefore must be **clear, detailed and unambiguous** and **signed by both parties** in order to avoid future litigation.

Purpose

A high-level description of the problem and the goal, objectives and deliverables of the solution. See p2 for more details. Don't forget to include any integration.

End User Requirements

These describe what users expect to be able to do with the system.

This section will be informed by common sense and the results of your User Survey.

Users may want features that you do not have time to develop (see time feasibility) and are outwith the scope of the project. Include these in the boundaries section (see below).

You will already have broadly described your end-users in the purpose section. In the End User Requirements section, you can give more details about your end users, what they expect to be able to do and if different end users require different features, different interfaces, different levels of security etc. These different features, interfaces and security levels should also be stated in the scope section.

Scope, Boundaries and Constraints (also Assumptions)

- Scope clarifies what the project must cover.
- Boundaries clarify what the project will not cover.
- Constraints are restrictions that apply to the development.

Scope

This is a list of the **deliverables** (what the **full development** must contain).

It must be **detailed** and **unambiguous** to avoid future litigation.

Because it is a list of what the development must contain, the scope section goes beyond just describing what the project will accomplish. The scope section should also include *everything* that will be handed over to the client.

For example (in no particular order!):

- What information should be included in the system? (not as detailed as functional requirements section)
- What information is being output by the solution? (not as detailed as functional requirements section)
- What system(s) is it being designed to run on e.g. OS version(s)?
- Are there different features for different users?
- Have you included security? Are there different levels for different users?
- Interface – do different types of user require different interfaces?
- What is being integrated?
- Test plan including test table
- End-user test report (is there a % of acceptable errors?)
- Evaluation report
- Designs (list the designs you will be including e.g. use case diagrams, wireframes, low fidelity prototypes, high quality mock-ups, flowcharts, class diagrams, structure diagrams, pseudocode, entity-relationship diagrams, data dictionaries etc. Remember to briefly explain what the particular design is for e.g. wireframe of customer login screen etc.)
- Structured listing (for your AH project)
- Completed program (not required to be submitted to SQA for AH)
- Do you also have to include a user guide (training manual)?
- Time limit (see Project Planning p8)
- Cost

Boundaries

This sets out in detail the limits of your project. i.e. clarifies explicitly what it will **not** cover.

You (or the results of your user survey) may come up with some ideas that might be reasonable suggestions to include in your project but you do not have time to develop them. You can detail these in the boundaries section as being explicitly excluded.

You should also detail technical boundaries e.g. while the scope should state what OS version(s) you are developing/testing for, boundaries should explicitly state that the system will not run on any other operating system. Also think about file sizes e.g. while

your scope should state the maximum file size your system can handle, the boundaries should explicitly state that files larger than your maximum will not be handled.¹

Scope and Boundaries Brief Example of Things to Think About:

Suppose your project was to develop an expert system giving students guidance on job opportunities which they should consider after graduating from University.

The **scope** of the project would be to create an expert system. Then it would be necessary to describe the range of jobs and degrees that would be included in the system, the level of information that would be output by the system (does it suggest contact addresses as well as simply job types), the types of questions that the user will be asked. Does it cover all degrees, or is it only for students with Computing Science degrees, and so on ...

Sometimes it is also helpful to spell out exactly what will NOT be covered. All these things will define the **boundaries** of the system. So, for example, a clear statement could be made which states that the system will NOT cover advice on jobs for those with medical and veterinary degrees, or jobs overseas.

The scope and boundaries could also refer to technical issues. For example, they might state that the resultant system will run on any computer capable of running any version on Windows after Windows XP, but not on any other operating system.

Why is it important to clarify the exact scope and boundaries?

Real world answer: Proper scope definition is critical to a project's success. It establishes the boundaries of what the project will and will not accomplish. The scope statement eliminates any confusion or ambiguity that might still exist after considering the project's goal, objectives and high-level deliverables statements. Poorly defined scope leads to "scope creep", which means that the project's objectives change as it progresses. These changes inevitably lead to increased work effort, which in turn causes project delays, cost overruns, poor team morale and/or customer dissatisfaction.

AH project answer: Proper scope definition is essential to ensure that you embark on a realistic project. If you don't define the scope and boundaries, you won't know when you have finished implementing the project, you won't be able to evaluate it properly, and finally, you will lose marks!

¹ Please note that when discussing boundaries, I frequently use the phrase "the system *should not*...". "Should" is a modal verb that can be used to imply desirability or probability BUT it can also be used to express necessity or correctness. I am (correctly) using the word "should" to express necessity (<https://www.dictionary.com/browse/should?s=ts>).

Constraints

Constraints are restrictions that apply to the development. These restrict the changes made to design decisions during the development. Time, scope and cost are the main constraints of project management; however, depending on the type of development, other constraints may apply, for example:

Technical constraints

- knowledge and/or availability of development tools and programming language
- the operating system or platforms that will be used to deliver the working solution
- hardware considerations such as capacity
- non-functional requirements such as performance considerations

Business constraints

- schedule and timescales that must be met
- available budget
- composition and makeup of the development team
- software licensing restrictions or requirements

Further constraints

economic considerations

political issues

Assumptions (not explicitly in AH)

Ideally, all ambiguities and resultant assumptions should be clarified with the client prior to starting the Design stage. This avoids problems with the client later on.

Although not part of the AH syllabus, be aware that it can be necessary to clearly outline any assumptions that have been made prior to design and implementation.

For SQA, you should state any assumptions you have made, particularly if you find the question to be a little vague. This helps to explain your reasoning for your answer.

Functional Requirements

This section of the Requirements Specification is where you **specify inputs, processes and outputs**. The functional requirements outline what your project should do (the mandatory features) in more technical detail than the Purpose or Scope sections.

It is still relatively top level, however, with more detail produced during the Design stage e.g. you might want to create a top-level structure diagram to show the expected program structure but detailed pseudocode would not be produced until the Design stage.

For your project, the easiest (!) way of producing a functional specification is to list all the project's activities/possible scenarios. Then clearly list the project's inputs, processes and outputs. You might find this relatively straightforward as long as you remember to include relevant details for all parts i.e. the program, database and/or website (remember to include integration!).

However, you might find it easier to approach this with a separate "sub-section" for each activity/possible scenario. You should then put a small descriptive paragraph explaining what this "sub-section" is for (i.e. explain the activity/possible scenario) and then list its inputs, processes and outputs.

Examples of inputs, processes and outputs:

- Inputs
 - o Each expected input should be listed along with the source of the input (screen, file, etc.).
- Processes
 - o The principal processes should be outlined e.g.
 - Opening connection to SQL database
 - Reading files
 - Forming and executing SQL queries
 - Calculations
 - Sorting
 - Searching
 - Interpreting the results from an SQL query
 - All required validation
 - Writing files
 - Closing connection to SQL database
- Outputs
 - o Each expected output to the user, either as a new or updated screen or webpage, including error messages.

Things to remember/consider (even if just discussed in boundaries):

Security e.g. think about different levels of security (access) for different types of user e.g. customers; site administrators etc. Remember to close database connection.

Integration (have you noticed that I keep mentioning this?)

Interface - Will all users of your solution use the same interface or will customers have a different interface from site administrators.

Data structure requirements.

And finally....

You should also produce a **Use Case diagram** clearly showing the actors, use cases and relationships between them (see p26-31 for notes on how to draw a use case diagram).

SQA Examples

https://www.sqa.org.uk/files_ccc/AHCourseSpecComputingScience.pdf

****Note that the examples below use the heading “Analysis” rather than Purpose****

Be aware that both of these examples are very brief and simplistic. You are expected to provide significantly more detail in your Requirements Specification as well as more accompanying documentation in your project (which must therefore be listed in the scope section)! Read this booklet for more information.

Appendix 1: problem analysis (SDD, DDD and WDD)

Worked example of a requirements specification (SDD) – p54 of SQA Course Spec doc

Analysis

The purpose of a program is to allow the end user to search for an item on an unsorted list of data. If a match is found, the program will display the row of data for the item.

Scope

This development involves creating a modular program. The deliverables include:

- a detailed design of the program structure
- a test plan with a completed test data table
- a working program
- the results of testing
- an evaluation report

Boundaries

- the program will read the data (itemID, price, and number in stock) from a sequential file

In my opinion, the way this is written means it is not a boundary but rather belongs in scope. For a boundary, it should be rewritten:

- the program will only read the data from a sequential file, not a random or binary file
- the program will only read the data itemID, price and number in stock NOT <<other data items that will not be read if there are any>>

- assuming the data is accurate, there is no need to implement input validation

End-user requirements

End users will expect:

- to enter an itemID while the program is running
- the data corresponding to the itemID to be displayed
- a user interface that is clearly labelled and easy to use for all user types

Functional requirements

Functional requirements are defined in terms of the inputs, processes, and outputs listed below. All inputs are imported from a sequential file and all outputs displayed on the screen. The program is activated by double clicking on the file icon and then selecting “Run” from the menu. Each process should be a separate procedure or function that is called from the main program.

Inputs

- itemID
- price
- number in stock

Processes

- read in data from an external file to a 2D array
- sort the data in order of itemID from low to high
- search the 2D array for the required itemID, based on the end-user input

Output

- if a match is found, the data (itemID, price, and number in stock) will correspond to the end-user input
- if no match is found, a suitable message will inform the end user

Constraints

The constraints that apply to this development are:

- Live Code, Python, or Visual Basic must be used to develop the program.
- The working program will run on the Windows operating system.
- The work must be completed within 8 hours.

NOTE THAT THIS IS MISSING A USE CASE DIAGRAM!

Worked example of a requirements specification (DDD) – p56 of SQA Course Spec doc Analysis

GoGoGadgets.com is a company specialising in quirky and unusual gadgets that are available for purchase through its online catalogue.

Before customers can make a purchase, they must first register with the GoGoGadgets website and be allocated a unique customerID.

Customers can browse the product range through an online catalogue. Each item is categorised as one of the following: Toys, Gizmos, Office Distractions, Personal Grooming, and Computer Accessories. All items cost less than £50.

A database is required to store details of customers, items, and orders.

Scope

This development involves creating a relational database. The deliverables include:

- a detailed design of the database structure
- a test plan with a completed test data table
- a working database
- the results of testing
- an evaluation report

Again, these are blurring “scope” and “boundaries”. Scope is what the project will include e.g. “each item will cost £50 or less”. Boundaries are what it will exclude e.g. “no item will cost over £50” (also think about input validation as a functional requirement).

Boundaries

- the database will contain a maximum of 10 000 items
- each item will cost £50 or less
- all items should be categorised as one of the following: Toys, Gizmos, Office Distractions, Personal Grooming, and Computer Accessories
- users must enter a valid e-mail address to register

End-user requirements

End users (customers) will expect queries that enable them to:

- register as a user and store their details in the database
- search for items based on the category of the item
- search for items based on the name of an item
- sort items by price (low to high), price (high to low) or rating

End users (administrators) will expect queries that enable them to:

- edit the price of items
- edit customer contact details
- add and remove details of individual items
- remove details of customers from the database
- view details of all orders placed each month

Functional requirements

Functional requirements are defined in terms of the inputs, processes and outputs listed below.

Inputs (customers)

- register: user e-mail, password, password re-entered, firstName, lastName, address, and postcode:
 - search details: category
 - search details: itemName
- sort details: field (price or rating) and order required (ascending or descending)

Inputs (administrators)

- edit item details: itemID and price
- edit customer details: customerID, address, postcode, and e-mail
- add item details: ID, description, category, and price
- delete item details: itemID
- delete customer details: customerID
- monthly orders: month

Processes

- auto generate customerID whenever a new customer registers
- queries to:
 - insert records into the Customer and Item tables
 - sort item details in order of price and rating
 - delete a specific customer and an item record from the database
 - edit records in the Customer and Item tables
 - search Item table
 - display details of all orders placed in a particular month

Output

- confirmation of successful:
 - insertions
 - deletions
 - edits
- answer tables showing details of:
 - sorted items (sorts)
 - required items (searches)

Constraints

The constraints that apply to this development are:

- The Oracle MySQL server must be used to develop the database.
- The working database will run on the Windows operating system.
- The work must be completed within 15 hours.

NOTE THAT THIS IS MISSING A USE CASE DIAGRAM!

UML

What Is UML?

Unified Modelling Language (**UML**) is a **graphical standardised design language** used to **model a system**. The system may be software or non-software.

UML provides a **clear visual representation** of the system making that system easier to understand and maintain.

Using UML, you can model just about any type of application, running on any type and combination of hardware, operating system, programming language, and network. Built upon fundamental object-oriented concepts including *class* and *operation*, UML is particularly useful in *object-oriented software development* where it is used to create visual models of software systems.

Why Use UML?

Large enterprise applications (i.e. ones that execute core business applications and keep a company going) must be more than just a collection of code modules. They must be structured in a way that enables scalability, security and robust execution under stressful conditions. Their **structure** (their **architecture**) must be defined clearly enough that maintenance programmers can (quickly!) find and fix a bug that shows up long after the original authors have moved on to other projects.

Clearly defined structure is a good way of dealing with complexity.

Another benefit of structure is that it enables **code reuse**. Design time is the easiest time to structure an application as a collection of self-contained modules or components.

What Does Modelling Mean? (not explicitly in AH but you need to understand this)

Modelling is the designing of software applications before coding i.e. the act of visualising, specifying, constructing and documenting the analysis and design of a system.

Modelling is an essential part of large software projects, and helpful to smaller projects as well. A model plays the analogous role in software development that blueprints and other plans (site maps, elevations, physical models) play in the building of a skyscraper.

Using a model, those responsible for a software development project's success can assure themselves that business functionality is complete and correct, end-user needs are met, and program design supports requirements for scalability, robustness, security, extendibility (and other characteristics), **before implementation** in code renders changes difficult and expensive to make.

Surveys show that large software projects have a huge probability of failure - modelling is a way to visualize your design and check it against requirements before your team starts to code.

There are currently thirteen types of UML diagrams which are used to represent **dynamic** and **static** views of a system model.

UML diagrams can be divided into two categories:

- **behaviour** diagrams (**dynamic** view)

The dynamic (behaviour) view represents collaboration among objects and changes to internal states of objects through sequence, activity and state machine diagrams.

Behaviour diagrams represent the **functionality** of the software system and emphasise ***what must happen*** in the system being modelled.

There are seven diagram types representing general types of behaviour.

For AH, you only need to know about **Use Case Diagrams**.

- **structure** diagrams (**static** view)

The static (structure) view includes class diagrams and composite structure diagrams, which emphasize static structure of systems using objects, attributes, operations and relations.

Structure diagrams are used in documenting the **architecture** of software systems and emphasise the ***things that must be present*** in the system being modelled.

There are six diagram types representing structural information.

For AH, you only need to know about **Class Diagrams**. You will learn about class diagrams in the Design stage booklet.

Purpose of Use Case Diagrams

Use Case diagrams are used to show the dynamic behaviour (functionality) of the system (dynamic behaviour means while it is running). It shows the **processes** that must be in the system being developed as well as the **actors** that interact with them.

Remember that the actors are external to the system.

How to Draw Use Case Diagrams

Use Case diagrams describe the functionality (dynamic behaviour) of a system in terms of:

- System boundary
- Actors
- Use cases
- Relationships

For your AH project, you must indicate the relevant actors, use cases and relationships. **You must ensure you are indicating any integration** e.g. with files, database etc.

System Boundary

The system boundary represents the limits of the system being developed: only those actors and processes to be considered are illustrated within the system boundary.

In a UML case diagram, a system boundary is shown as a rectangle. All components of the use case diagram are shown inside the system boundary. Inside the boundary at the top left, remember to write the name of the system.

Actors

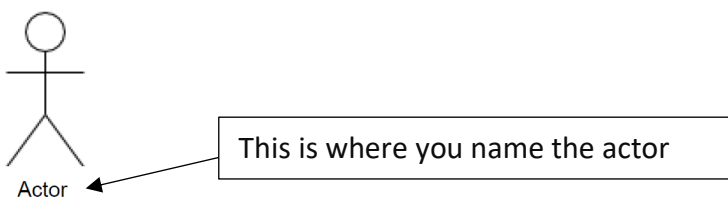
An actor interacts with the system being developed.

The actor may be a **human** or an **entity** that **interacts** with the system, for example another system or server. Actors are **external** to the system being developed.



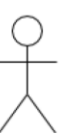
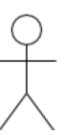




An actor performs a role in a system and may be a primary or secondary actor:

- A primary actor is one that uses the system to achieve a goal, for example a customer buying an item.
- A secondary actor is one that supports the system in delivering the goal, for example a bank used to pay for the item.

A UML case diagram shows an actor by using the symbol:



Some examples of actors (SQA Course Specification p59):

Human	Systems software	Hardware	Timer (clock)
 Customer	 Payroll	 Phone network	 Scheduled backup
 Passenger	 Library	 Server	 Scheduled anti-virus

Note that even the systems software is represented by a little stick person. Don't forget to LABEL your actors!

Don't forget that you need to show integration in your AH project – you will need actors to represent e.g. a database if you have one.

Use Cases

A use case **describes a process** (an action) or a **sequence of processes** (actions) that must be in the system being developed.

A UML case diagram shows a use case using a **named ellipse**. The name describes some observable or useful result to an actor. Examples of naming are Update Subscription, Manage Account, and Place Order.



Use cases help to determine the requirements of the system under consideration, by **describing the functionality** (processes) that the system will provide.

Use case functionality (process) may be initiated by an actor or may be started by the system itself, providing a useful result to an actor.

Relationships

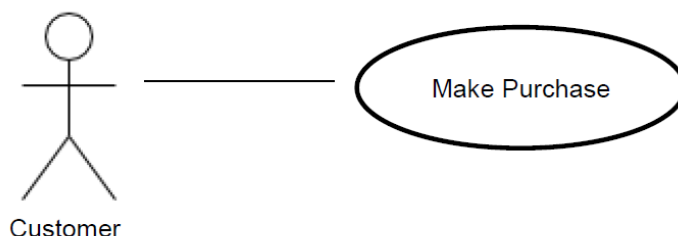
A use case diagram can have five types of relationship:

- association between an actor and a use case
- generalisation of an actor
- extend between two use cases
- include between two use cases
- generalisation of a use case

Association between actor and use case

Each actor **must be associated with at least one use case**, although it can be associated with many use cases.

A solid **line with no arrowheads connects an actor to a use case**.

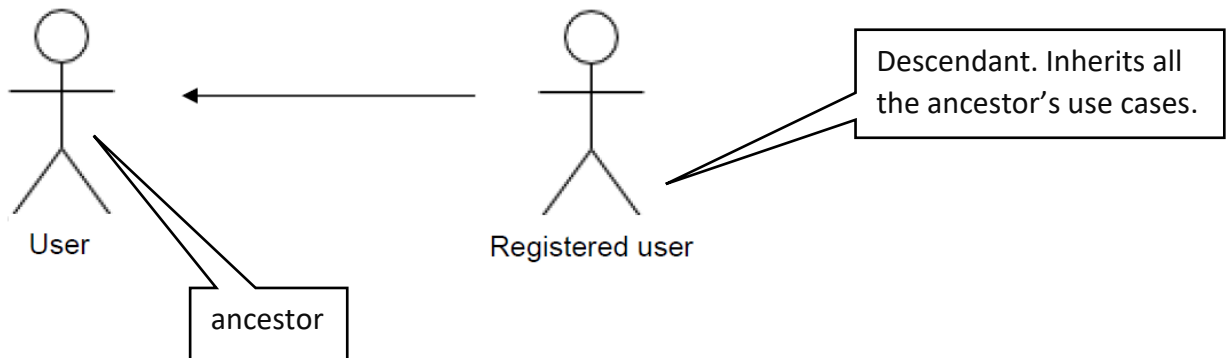


Generalisation of an actor

Generalisation of an actor means one actor can inherit the role of another actor.

The **descendant actor inherits all the use cases of the ancestor.**

A **solid line**, with a **single solid arrowhead pointing at the ancestor actor**, connects a descendant actor to the ancestor actor.



Extend between two use cases

Extending a basic use case **provides additional functionality** to the system.

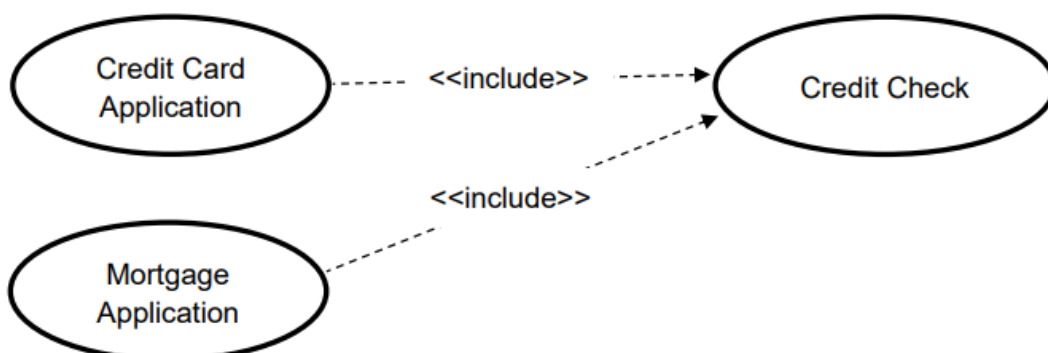
An extended use case is connected to a basic use case using a **dashed line**, with a single solid **arrowhead pointing at the basic use case**. The label **<<extend>>** is placed on the line.



Include between two use cases

An included use case is **part of** the basic use case. It is a **mandatory process**, as the basic use case is incomplete without it.

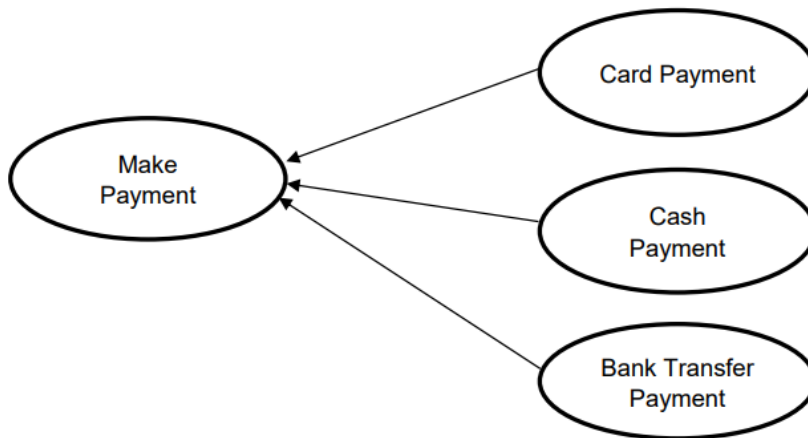
An included use case is connected to the basic use case using a **dashed line** with a single solid **arrowhead, pointing at the mandatory common basic use case**. The label **<<include>>** is placed on the line.



Generalisation of a use case

This is similar to generalisation of an actor.

A **solid line**, with a **single solid arrowhead pointing at the ancestor use case**, connects a descendant use case to the ancestor use case.



The easiest way to think about generalisation of a use case is to identify two or more use cases that have commonalities in behaviour, structure and purpose. When this happens, you can describe the shared parts in a new (often abstract) use case, that is then specialized by descendant (child) use cases.

Example: https://sceweb.uhcl.edu/helm/RationalUnifiedProcess/process/modguide/md_ucgen.htm

Consider the following step-by-step outlines to use cases for a simple phone system:
Place Local Call

- 1. Caller lifts receiver.**
- 2. System presents dial-tone.**
- 3. Caller dials a digit.**
- 4. System turns off dial-tone.**
- 5. Caller enters remainder of number.**
- 6. System analyzes the number.**
7. System finds corresponding party.
8. System connects the parties.
- 9. Parties disconnect.**

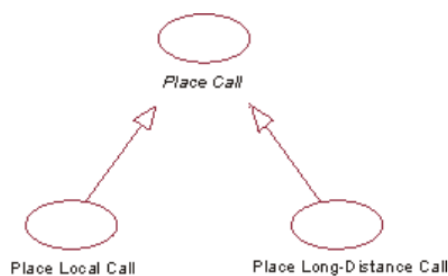
Place Long-Distance Call

- 1. Caller lifts receiver.**
- 2. System presents dial-tone.**
- 3. Caller dials a digit.**
- 4. System turns off dial-tone.**
- 5. Caller enters remainder of number.**
- 6. System analyzes the number.**
7. System sends number to other system.
8. System connect the lines.
- 9. Parties disconnect.**

The text in bold is very similar in the two use cases. If the two use cases are pretty much identical, you should consider merging them into one use case, where alternative sub-flows show the difference between local calls and long-distance calls.

If, however, the difference between them is of some significance or there is a value in clearly showing in the use-case diagram the relationship between local call and long-distance call, we can extract common behaviour into a new, more general use case, called Place Call.

In a use-case diagram, the generalization-relationship created will be illustrated as follows:



The use cases Place Local Call and Place Long-Distance Call are inheriting from the abstract use case Place Call.

Recommended Resource:

<http://www.lucidchart.com>

Watch:

Lucidchart: UML Use Case Diagram Tutorial (13:23): <https://youtu.be/zid-MVo7M-E?t=33>

Derek Banas: UML 2.0 Tutorial (12:46): <https://youtu.be/OkC7HKtiZC0?t=28>

Read:

<https://slideplayer.com/slide/12059295/> (slides 9 – 20)

<https://creately.com/blog/diagrams/use-case-diagram-relationships/>

<https://www.smartdraw.com/use-case-diagram/>

Be aware that the smartdraw illustration of include and extend is not as per SQA instructions.

<https://stackoverflow.com/questions/1696927/whats-is-the-difference-between-include-and-extend-in-use-case-diagram>

TOP TIP:

For your AH project, you must indicate the relevant actors, use cases and relationships. You must ensure you are indicating any integration e.g. with files, database etc.

Use Case Questions²

Q1. Requirements: Develop a swimming portal for a swimming coach.

- The system must allow scheduling for swimming lessons.
- The system must allow swimmers view their competition time and results.
- The system must provide a payment system for termly fees.

Persona

Avril is a swimming coach and has a hectic life, swimming competitions are approaching and as an instructor she wants to schedule extra training hours for 15 kids, this should also fit into her extremely hectic work program.

User Story

Goals

She wants a quick way of booking pupils that require extra lessons.

She wants to be able to review available dates quickly on a portable device. She also wants swimmers to login to the system, make and pay for bookings, edit their details, view their previous competition time.

User Scenario

When Avril receives requests from swimmers for dates they would like extra training, she wishes to use the App to pull up a calendar for the next two months, to add a swimmers name as well as specific time for their training. She wants the app to alert each swimmer immediately with the scheduled information; each swimmer must then accept or reject the time allocated.

Each swimmer should be given generated login information to use on the system. Using this login information, swimmers will be able to login to the system, view their upcoming swim meets, add/change their personal details, to view data kept for competition times in each swimming technique. Abby should also be able to view all information entered by each swimmer.

Swimmers can also use the system to pay for each individual extra training session.

Task

Avril requires a swimming portal app urgently, you must design a use case diagram for the above scenario, your diagram should include at least one example for each of the following structures:

- a) <<include >>
- b) <<extend >>
- c) generalization

² Use Case Diagram questions 1 – 6

Q2. Requirement: Develop a holiday booking App

The system must

- maintain a database of trips available
- Allow users to search for particular locations
- Allow users to view reviews
- Allow users to book a particular trip using various payment methods
- Allow users to change the textual language if they wish to do so.

Persona

Ben is a new dentist in Aberdeen, he is originally from Spain. He is ecstatic about his new job and looking forward to getting to know Scotland.

User Story

Goals

Ben would love to be able to take his family up to the highlands on holiday he wishes to book a hotel and transportation on any digital device.

He would also really like the information translated to Spanish.

User Scenario

Ben will like an app he can use to book the trip for his family. He wishes to be able to search for most popular locations in the Highlands. He also wants to select Spanish for language translation, and the ability to use the app to compare prices and reviews posted by other users. Finally, Ben wants to be able to select the number of travellers, date and use various payment methods (credit card, debit card or wire transfer) details to pay for the trip.

Task

The travel company wants to develop this app so it can be accessible to busy people like Ben. You have been tasked to design a use case diagram for the above scenario. Include examples of each of these constructs in your diagram:

- a) <<include >>
- b) <<extend >>
- c) Use case generalization

HINT: you will find the Spanish language translation tricky as it is unclear whether you have to develop this yourself or interact with another actor e.g. Google Translate.

Q3. Requirement: Develop an additional module for a school management information system (MIS)

The system must:

- Allow the Senior Leadership Team (SLT) to browse for available teachers within periods
- Generate absence reports
- System must receive and/or notify teachers for cover requests
- Allow teachers update result and attendance data for pupils
- Allow SLT to view exam results and attendance data.

Persona

Eileen is a new deputy head teacher; she has just been promoted to the job and is very enthusiastic about taking the role. She is looking forward to getting to know the teachers and pupils in the new school.

User Story

Goals

To assist with the responsibilities of her new role Eileen wishes to have additional functionality added to the school's management information system (MIS) application to organise cover for classes due to teacher absences. She also wants to be able view the number of classes each teacher has covered.

She wants a system in which she can count and view each individual pupil's absences, and termly results and prelim exam mark.

User Scenario

Eileen wants an additional module added to the school's management information system which can be used for generating necessary reports. She wants the ability to be able to receive cover requests from teachers; search for available teachers for cover within that period/ lesson as well as receive future cover requests for up to 2 months. Eileen wants to count the number of cover each available teacher has completed to date so she can select teachers with the lowest numbers.

Eileen wants to receive alerts from teachers requesting absence reports for a pupil. She wants to use the system to search for the pupil and retrieve the absence reports for those specific dates.

Eileen also wishes to use the system to search for pupils' exam and prelim reports. The system should identify any omissions within the results and notify the subject teachers requesting these results. Teachers should also be able to update results in the system.

Task

Design a use case diagram for the schools MIS, Include these constructs in your diagram:

- a) <<include >>
- b) <<extend >>
- c) generalization
- d) association

Q4. Write an effective use case for a system to check and show the winnings for a monthly raffle. The customer enters their ticket number, the system detects that it matches the winning number of the month, registers the user as this month's winner, sends an email to the sales manager, congratulates the customer and gives them instructions on how to collect the prize.

Q5. Review the scenario below and model it with a use case diagram. Use at least one of each of the following constructs:

- a) <<include >>
- b) use case generalization
- c) association

Scope: A FAST CASH cashpoint machine

Context of use: To withdraw cash

Scenario

Jemma, taking her two daughters to day care on the way to work, drives up to the Cashpoint Machine, runs her card across the card reader, enters her PIN code, selects FAST CASH, and enters £35 as the amount. The Cashpoint issues a £20 and three £5 bills, plus a receipt showing her account balance after the £35 is withdrawn. The cashpoint resets its screens after each transaction with FAST CASH, so that Jemma can drive away and not worry that the next driver will have access to her account. Jemma likes FAST CASH because it avoids the many questions that slow down the interaction. She comes to this particular Cashpoint because it issues £5 notes, which she uses to pay the day care, and she doesn't have to get out of her car to use it.

Q6. You have been asked to develop software for an insurance company to satisfy the scenario below, create a use case diagram modelling this software, taking into account the preconditions. Where it would be applicable, use these constructs in your diagram:

- a) <<include >>
- b) <<extend >>

Scope: "System" means A claims-capturing computer system

Context of use: To Capture loss fully

Primary Actor: Administrator

Preconditions: Administrator already logged in.

Scenario

Craig is a new driver who has just passed his driving test, on his way home soon after dropping off a friend, he turned at a junction and took a quick left and unknowingly collided with a van. Following the accident Craig tries to make a claim on his insurance.

Craig sends details of the accident to his insurance company. Upon receiving the details an administrator will first verify Craig's policy details to make sure he is entitled to make this claim. Then the administrator will search to check to see if the accident / claim have already been recorded. If it is a new claim the administrator will enter the details as a new claim and assign it to a claims adjuster. The claims adjuster will then be able to view the claim details and process the claim. To process the claim the claims adjuster will either approve or reject it based on their review of the details.

TOP TIP:

Do not lose your use case answers. You will need them to create class diagrams during the Design booklet.

Analysis Questions

Q1: In the context of systems development, explain what is meant by a feasibility study and who carries it out?

Q2: Identify **four** kinds of feasibility and describe their differences.

Q3: Why is it important that a feasibility study should generate a report?

Q4: Explain what is meant by the term *cost-benefit analysis*.

Q5: What is the purpose of a Gantt chart?

Q6: Explain what is meant by the scope of a software project.

SQA Questions

Specimen (2019)

1. A new website is being developed for owners of electric cars.

Unregistered users of the website will be able to see the location of public charging stations.

Registered users of the website will be able to connect to one of the chargers and pay for the use of the charger via the website. Once the charge has completed, registered users will be able to view their charging history with details of locations, prices and charge level after each charge.

Draw a use-case diagram for this website.

3

2019 (Discontinued)

2019

4. A coach company runs a daily journey from Edinburgh to London. Users of the company website can check seat availability, book a seat and make payments online. Registered users can login to edit or cancel a booking.

(a) With reference to the information above, draw a use case diagram to represent the requirements of the online booking system.

3

2017

1. An annual shot put competition is to take place in October.

The program used each year to record and process competition data is being updated to allow participants to compete in three different categories: Senior, Adult or Junior.

(a) The organisers have reviewed feedback about the existing program and essential changes have been identified. Part of the outline plan for the development work is described below.

The development aims to begin on Monday September 5, 2017, by identifying:

- the user requirements
- the business requirements
- the scope and boundaries of the development

These three tasks can be completed concurrently. Identification of the user requirements and the business requirements will each take 5 days whereas identification of the scope and boundaries will only take two days.

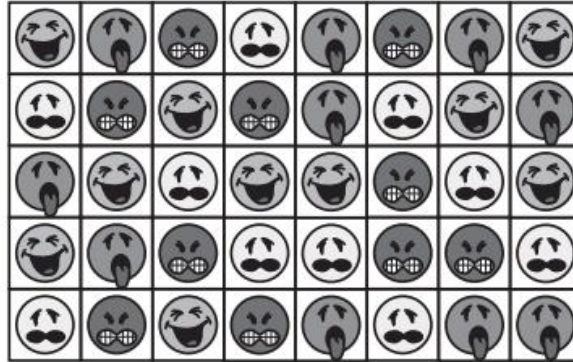
The functional requirements can only be identified once the previous three tasks have been completed. This work will take three days.

Create a Gantt chart to schedule the first four development tasks outlined in the description above.

2

4. Ali has been asked by a local primary school to create a program which allows young children to express their feelings. The program will be based on a pairs matching game.

When the program starts, 40 images representing the four most common feelings will be displayed and the child asked to match any two of the images.



- (a) At the start of the development, Ali needs to research the four most common feelings experienced by young children.

Explain how a user survey could be used for this purpose.

2

- (e) As a self-employed developer, Ali is concerned about his intellectual property rights in relation to the program.

Describe one form of automatic protection Ali receives when creating the program.

2

2016

1. The owners of a monthly magazine decide to update the company website. The current website allows users to access online versions of articles printed in the monthly magazines.

- (a) Requirements for the updated website are listed below.

The updated website will allow **all** users to:

- access a maximum of five free articles every month
- search for articles over 12 months old
- subscribe to the full service using a secure payment system

The updated website will allow **subscribed** users to:

- log-in to gain access to the full service
- access any number of articles
- search for articles without restriction
- renew their subscription at a reduced rate using a secure payment system

Draw a use case diagram to represent these requirements.

3

2. Radio Lowden plays songs from the years 1990 to 1999 inclusive. The songs played by the radio station must have featured in the official UK top 40 singles chart from these years.

(a) Using the above example, explain the terms scope and constraints. 2

(b) The management of Radio Lowden has commissioned a developer to create a new website for the radio station. One of the pages of the new website will give access to playlists from recent radio programmes.

(i) The developer suggests that the layout and interface of the website belonging to a rival radio station could be copied and used by Radio Lowden.

Discuss whether this is acceptable practice. 2

2015

1. A hovercraft company, ScotiaFlow, plans to open routes to transport passengers to islands in the Inner and Outer Hebrides within the next six months. It wants customers to be able to reserve and pay for seats using touch screen kiosks located at the departure terminals, or online using a specially-developed application.

A software development company is approached to create the program.

(b) A *feasibility study* is then undertaken.

(i) As well as reporting on cost and time issues, describe **two** other aspects of this development that may have been investigated. 2

(ii) Describe **two** ways that ScotiaFlow's managers could make use of the results of the feasibility study. 2

(c) ScotiaFlow considers the findings and decides to go ahead with the project. A requirements specification document is produced.

(i) The requirements specification contains the functional requirements and data requirements.

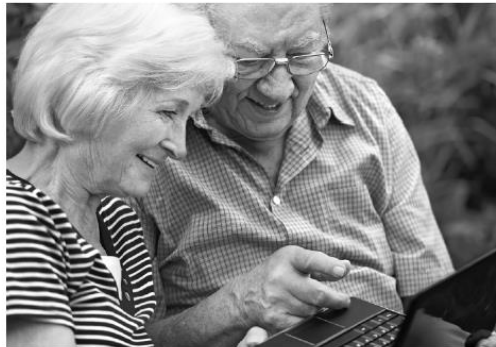
State **one** other item that may be included. 1

(ii) Describe **two** ways in which this document could protect the software development company against additional demands from ScotiaFlow. 2

(d) A project plan, which includes details of project deadlines, is created to manage the development. State **one** other element of this plan and explain its significance in managing progress. 2

2014

1. The Simply Amazing Retired People group has recently trained all its members in the use of the Internet and e-mail. Traditionally the club had a notice board where members could pin up postcard-size advertisements about objects for sale, publicise entertainment events, promote exercise activities for retired people and so on. The group now has an electronic version of this notice board on their website so that members can upload their own notices and include advertising videos for events.



The group employed a software development company to create the electronic noticeboard.

- (a) The *scope and boundaries* for the electronic noticeboard were clearly identified by the development company.

Describe the difference between the scope and the boundaries of a project. 2

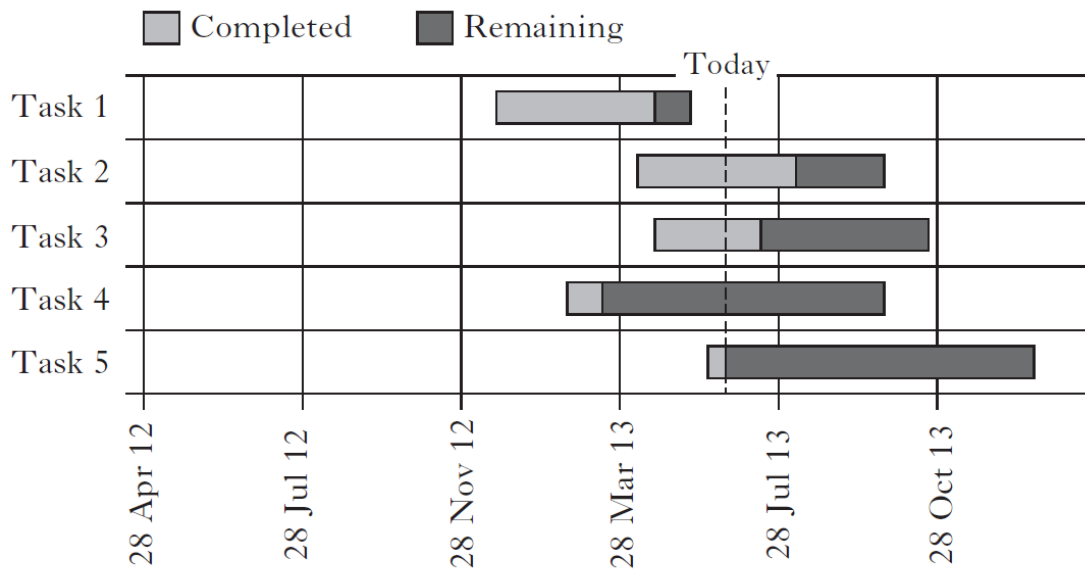
- (b) As part of the development process, the developers identified *sub-tasks* for the project. Explain why it was necessary to do this. 2

- (c) Describe **two** issues related to legal feasibility that the development company needed to consider before proceeding with the development of the electronic noticeboard. 2

2013

1. The Scottish Government is considering the introduction of a new electronic voting system for elections. The government appoints a small team to undertake a *feasibility study*. Some of the concerns include:
 - voters' accessibility to a suitable computer system
 - the possibility of electoral fraud
 - the cost of such a system.

- (b) (i) People will not be charged to vote. Explain why aspects of *economic feasibility* would still need to be considered. **1**
- (ii) Name **one** other type of feasibility and explain why it would be considered for the voting system. **2**
- (c) The government decides to proceed and a full investigation results in a Requirements Specification document.
- (i) The requirements specification contains details of the scope, boundaries and constraints of the voting system. Explain what is meant by scope, boundaries and constraints. **3**
- (ii) State one other item specified in the Requirements Specification. **1**
- (e) The team use a number of *project management techniques*. An example is shown below.



The chart shows how much of a task has been completed. State **two** other project management techniques that are incorporated in this diagram. **2**

2012

1. A mobile phone company will launch its new smartphone in six months and will introduce multiplayer gaming. These multiplayer games will allow users to interact with other smartphone users in the same gaming environment. The smartphone will allow users to:

- connect with other smartphones
- choose their character in the game
- interact and communicate with other players during the game.

The mobile phone company approaches GeekySoft, a games development company, to develop the game.

(a) GeekySoft begins by considering whether the capabilities of the smartphone can deliver realistic game play.

(i) State the **type** of feasibility being considered. 1

(ii) State **one** characteristic of the smartphone that might be considered when investigating this type of feasibility. Explain why it would be considered. 2

OLD PRELIM Questions

Prelim 2014-15

1. TeenTV is a television channel which is launching a new digital service. The digital service will be available on a number of devices such as smart TVs and mobile devices. Users will be able to post comments, upload photos/video and vote on the programmes to shown on the TV channel. The service is provided by an application, developed for each of the platforms, which makes use of the same media, logos and colour schemes to ensure a consistent “look and feel” across all platforms.

A software development company has been commissioned to produce the applications to support the service.

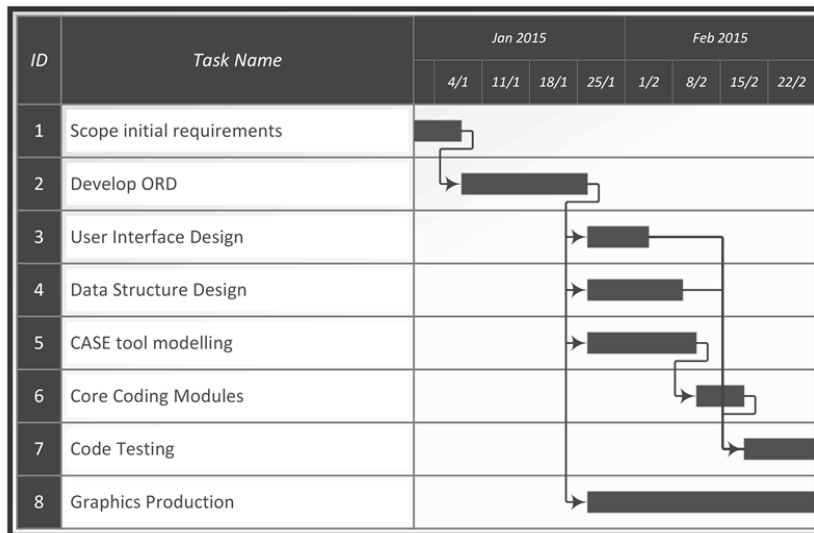
(a) The *scope*, *boundaries* and *constraints* for the applications were clearly identified by the software development company.

Describe the difference between the scope, boundaries and constraints of a project. 2

(b) (i) The service will be free to use. Explain why *economic feasibility* factors would still need to be considered before the development of the applications. 1

(ii) Name **one** other type of feasibility and explain why it would be considered for the applications. 2

(d) *Project management techniques* are used by the project team. An example is shown below.



State **two** project management techniques that are incorporated in this diagram.

2

Prelim 2007-08

MacroSift is a large multinational company which intends to improve the quality, accuracy and communication of information generated and shared by its employees. It proposes using handheld personal organisers and mobile phones to automatically receive and send updated information about employee activities.

When an employee enters a company building the device will automatically login to a wireless network and record the employee's location. Once logged in, the device will download any appointments, emails, notes and tasks for the employee to view. Any information that the employee has entered in the device will be uploaded to the central database to be shared with other employees as required.

(a) The project team will undertake a *feasibility study* which will consider both economic and schedule aspects of feasibility.

(i) Name and describe **two** other aspects of feasibility that may be investigated.

2

(ii) Suggest **two** factors that the project team might investigate when considering economic feasibility.

2

(d) User interface design is an important part of the design phase of the project in addition to producing the design for the detailed logic of the software.

(i) State **three** factors that will influence the project team when designing the user interface for software accessed on handheld personal organisers and mobile phones.

3

SQA Exemplar Projects and Marker Commentary: Analysis

https://www.understandingstandards.org.uk/AdvHigher_images/computing_science/ComputingScienceAdvHProject2019AnalysisOfTheProblemCandidateEvidence.pdf

Candidate 1 Evidence

Description of Problem

I intend to develop a quiz based on well-known historical events. The quiz will have three levels of difficulty so that the user can select a level on either their historical knowledge or how much of a challenge they want to set themselves. I intend to include the following features: login facilities, help screens, each level of difficulty will have a quiz selected at random, each quiz will have multiple choice questions and a leader board that can be sorted.

The end-users of my quiz will specifically be S1-S6 students who study history at school but also anybody who is interested in history and wants to test their knowledge.

My project meets the advanced higher computing requirements because it will have a user interface suitable for S1-S6 students which will be easy to use and will validate all inputs. My project will interface with stored data within a database for the questions (using SQL) and also with a text file to store the login facilities (these will be processed using an array of records). My project will also use a sort algorithm to arrange the details leader board details.

Scope

The scope of my project will include:

1. A completed design with pseudocode, data dictionary, query design and wireframes showing the intended interface for the quiz
2. A working quiz and a database with multiple choice quiz questions
3. Completed test plan with descriptions of a test persona, test cases and expected outputs
4. The results of final testing
5. An evaluation report

Constraints

There are a number of technical, economic and time constraints that will apply to this development.

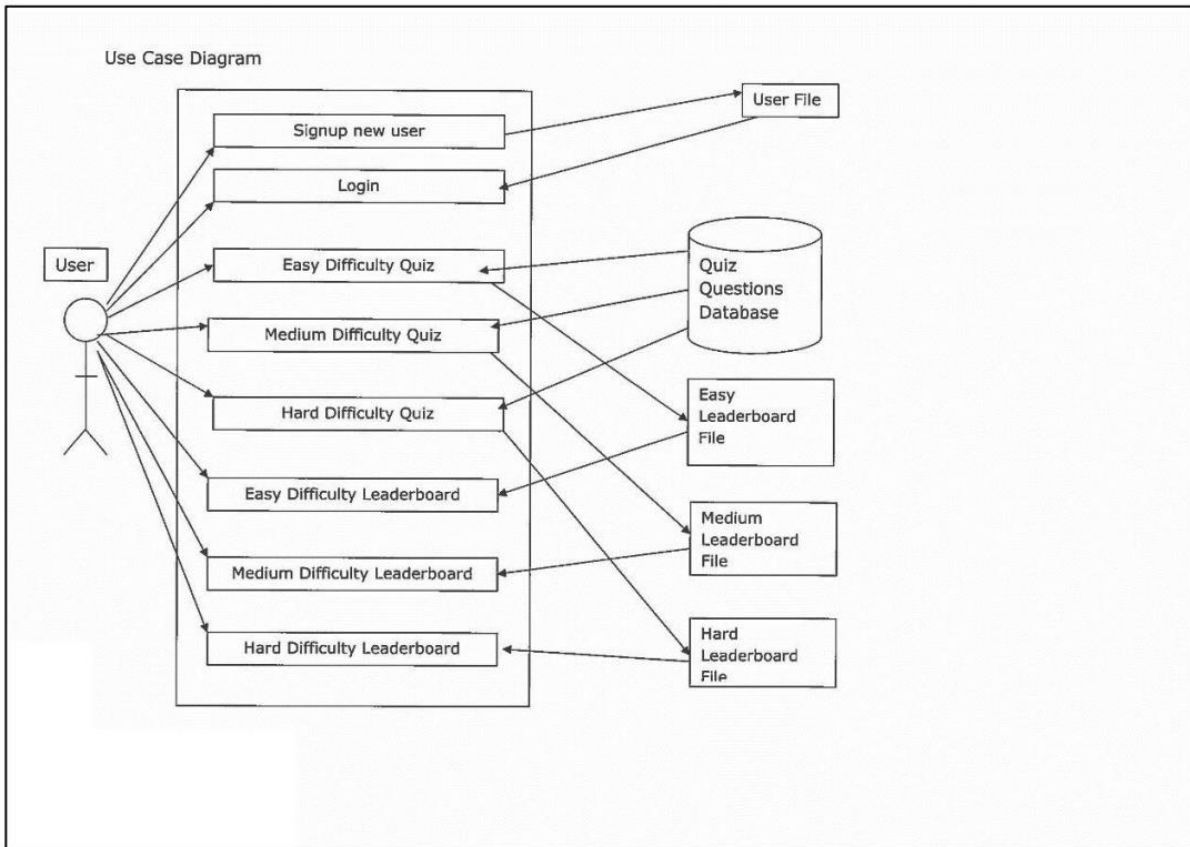
1. I will use LiveCode version 9 to create the quiz because I have several years' experience using this language
2. The final solution will run on the Windows systems with ODBC to access my quiz database.
3. Microsoft Access will be used to store my quiz questions
4. There will be no costs involved in the development of my project because I am using software that is fully licensed and readily available in school
5. I will ensure that my project is completed by the deadline of 29th March as it will need to be delivered to SQA for marking. The submission deadline is 4th April 2019.

Boundaries

My working solution will contain:

1. A system to allow users to sign up with a new username and password combination
 - The user cannot have a password of less than 4 characters
2. A system to allow users to log in with their unique username and password
3. All user input must be validated
4. Three quizzes of different levels of Easy, Medium and Hard will be available
5. Each quiz will contain 10 questions that will be presented to the user in a random order
 - Users will not be able to choose how many questions per run through of the quiz they do
 - Users will only be able to choose their difficulty level, they will not be able to choose which questions they are asked
6. The score achieved by each user will be stored in the leader board file corresponding to the correct difficulty level
7. A leader board that will display the top 10 scores and usernames for each difficulty level
 - The leader board for each level will be sorted in descending order of score

UML Use Case Diagram



Requirements Specification

For my system to be able to fulfil its requirements, it must meet several criteria:

Login System Requirements

Users of my system will be able to create a unique username-password combination to access the quiz. The login functionality will include:

1. The Login page is the first screen that users will see, and will allow the user to enter
 - a) A valid username and
 - b) A valid password
2. The user will then press the "Login" button, which will check that the username and password they have entered is valid by:
 - a) Checking that neither field is blank
 - b) Checking that the password entered contains more than 4 characters
 - c) Importing stored usernames and passwords from the login file
 - d) Checking that the username and password combination exists in the login file
3. If the user passes this validation:
 - a) The username will be stored in a variable to be accessed throughout the program
 - b) The user will be directed to the Home page to select one of three levels of difficulty options for the quiz

Login screen inputs

- o Users will login by entering their:
 - username
 - password
- o Users will be able to add a new user by inputting a:
 - username
 - password
- o User button click to:
 - login to the quiz system
 - sign up as a new user

Logout System Requirements

Users of my system will be able to log out of my system from the Home page. The logout functionality will include:

1. Clearing the variable that stores the username of the currently logged in user
2. The user is brought back to the Login page
3. A message will be displayed telling the user that they have logged out

Home screen inputs

- o User button click to:
 - indicate which of the three levels the user wants to play
 - logout of the quiz system
- o Message displayed when the user logs out

Quiz System Requirements

The users will begin the quiz by choosing a difficulty from the Home page from the choice of “Easy”, “Medium” or “Hard”, which will then take the user to the relevant Quiz page for that level of difficulty. They will then press the “Play” button to start the quiz. The quiz will provide the following functionality:

1. The username of the currently logged in user will be displayed at the top corner of the page
2. When clicked, the “Home” button will allow the user to return to the Home page
3. A “Clear” button will be provided to clear the output box between successive attempts
4. Once the “Play” button is pressed:
 - a) A list of 10 questions will be randomly selected
 - b) Each question will be displayed one at a time in an output field on the Quiz page
 - c) The list of 4 possible answers for each question will be displayed below the question in the same output field
 - d) A pop-up box will be used to allow the user to enter the number of their chosen answer

- e) The number entered will then be compared with the correct answer stored in the database
- f) If the answer is correct, a message stating that they're correct will be displayed and one will be added to their score
- g) If they're incorrect, then the score will remain the same and a message stating that they're incorrect will be displayed
- h) Once the quiz is finished, the score that the user achieved out of 10 will be displayed
- i) The username and score of the user will be added to the leader board file for the chosen level of difficulty

Quiz screen inputs

- o User button click will be used to:
 - start the quiz
 - return to the Home screen
 - clear the leader board display area
- o Users will type their answer to each question into the pop-up box that opens Quiz screen outputs
- o Username of the current user will be displayed
- o Each question will be displayed along with the 4 possible answers
- o Correct or incorrect message will be displayed
- o The user's quiz score will be displayed

Leader Boards Requirements

The users will be able to view a leader board for each difficulty of the quiz "Easy", "Medium" and "Hard". The leader board functionality will include:

1. The username of the currently logged in user will be displayed in the top corner
2. A "Return" button will be provided to return to the Home page
3. A "Clear" button will be provided to clear the leader board display area between displays
4. When the "Display" button is pressed:
 - a) The contents of the file storing the scores for the user's chosen difficulty will be imported into the program and sorted in descending order
 - b) The top 10 scores will be displayed with the username of those who got them

Leader board inputs

- o User button click will be used to:
 - display high scores and matching user names
 - clear leader board display array
 - return to the Home screen

Leader board outputs

- o Username of the current user will be displayed

o Sorted score and matching username on the leader board

Project Plan

Here is my detailed project plan with list of tasks and time allocation.

Task	Sub Task	Duration	Start Date	Target Date
Analysis 6 days	Read requirements for AH project and create idea	1 day	23 rd Oct	23 rd Oct
	Write outline of project	0.5 day	24 th Oct	24 th Oct
	Create scope, boundaries and constraints	0.5 day	24 th Oct	24 th Oct
	Create UML use case diagram	1 day	25 th Oct	25 th Oct
	Create user requirements description	1 day	26 th Oct	26 th Oct
	Create functional requirements	1 day	30 th Oct	30 th Oct
	Project plan	1 day	31 st Oct	31 st Oct
Design 14 days	Interface design - Create wireframes	5 days	1 st Nov	8 th Nov
	Design Structure Diagrams	2 days	9 th Nov	11 th Nov
	Create Pseudocode	5 days	12 th Nov	19 th Nov
	Design SQL	2 days	20 th Nov	21 st Nov
Implementation 30 days	Create User Interface	5 days	28 th Nov	5 th Dec
	Create Program Listing	10 days	6 th Dec	20 st Dec
	Create DB structure	5 days	5 th Jan	12 th Jan
	Create File Structure	5 days	13 th Jan	19 st Jan

	Write description of new skills	0.5 day	19 th Jan	19 th Jan
	Log of ongoing testing	Ongoing through implementation	28 th Nov	20 th Jan
Testing 8 days	Test plan	3 days	21 nd Jan	23 th Jan
	Carry out testing	3 days	26 th Jan	28 th Jan
	Describe testing encountered problems	2 days	1 st Feb	2 nd Feb
Evaluation 2 days	Write evaluation	2 days	8 nd Feb	9 th Feb
Finalise evidence 5 days	Finalise analysis	1 day	15 th Feb	15 th Feb
	Finalise design	1 day	16 th Feb	16 th Feb
	Finalise implementation	1 day	17 th Feb	17 th Feb
	Finalise testing and evaluation	1 day	18 th Feb	18 th Feb
	Submit	1 day	19 th Feb	19 th Feb

Resources Required

The resources I will need at each stage of the development are listed below.

Analysis	<ul style="list-style-type: none"> • Microsoft Office Word 2007 • Google Chrome • Gantt project 2.7
Design	<ul style="list-style-type: none"> • Microsoft Office Word 2007 • Google Chrome
Implementation	<ul style="list-style-type: none"> • Microsoft Office Word 2007 with ODBC • Livecode 9.0.0 • Microsoft Office Access 2007 (connection will be achieved used ODBC access) • Notepad
Final Testing	<ul style="list-style-type: none"> • Microsoft Office Word 2007 • Livecode 9.0.0 • Microsoft Office Access 2007
Evaluation	<ul style="list-style-type: none"> • Microsoft Office Word 2007

Candidate 1 Commentary

The evidence for this candidate has achieved the following marks for each part of the analysis stage.

Description of the problem (2 marks)

The candidate was awarded **2 marks** because the detailed description of the problem meets all of the evidence requirements, including integration. This meets the requirements of an SDD project that integrates with the DB content of the course.

UML use case diagram (2 marks)

The candidate was awarded **2 marks** because the use case diagram indicates relevant actors, use cases and relationships. Integration with the database has been indicated.

Requirements specification (4 marks)

The candidate was awarded **2 marks** because the requirements specification is missing some details needed to create a fully integrated working solution.

The sign-up functionality is missing and there is no mention of the integration that will be needed to support the Quiz system: the requirements to form a connection with the database used to store the quiz questions/answer then execute a SQL query to extract those details have been omitted. Although not listed explicitly, the end-user requirements can be determined from the details provided.

Project plan (2 marks)

The candidate was awarded **2 marks** because the complete and detailed project plan meets all the evidence requirements.

Total marks awarded – 8 (out of 10)

Candidate 2 Evidence

Description of the Problem

Project idea – Cosmetic Details

Website End users of the proposed site will be predominantly females ages 15 – 40 who have a keen interest in make-up. In the website, users will be able to view makeup products from selected brands and search for specific products.

The project meets the requirements of AH computing as:

- It will have an appropriate interface with all user inputs being validated
- Details of customers and products will be stored in a SQL database
- It will make use of HTML form processing using PHP
- SQL operations will be used to insert details and perform searches
- Web pages in the site will be styled using external CSS
- When customer login to the site, PHP session variables will be used to store their data across multiple pages of the site
- The website will adapt to different screen sizes using CSS media queries

Scope, Boundaries and Constraints

Scope

During the development of the website I will produce:

- Requirements specification
- Detailed design
- Fully working implementation
- Comprehensive test plan with a summary of the final testing

Boundaries

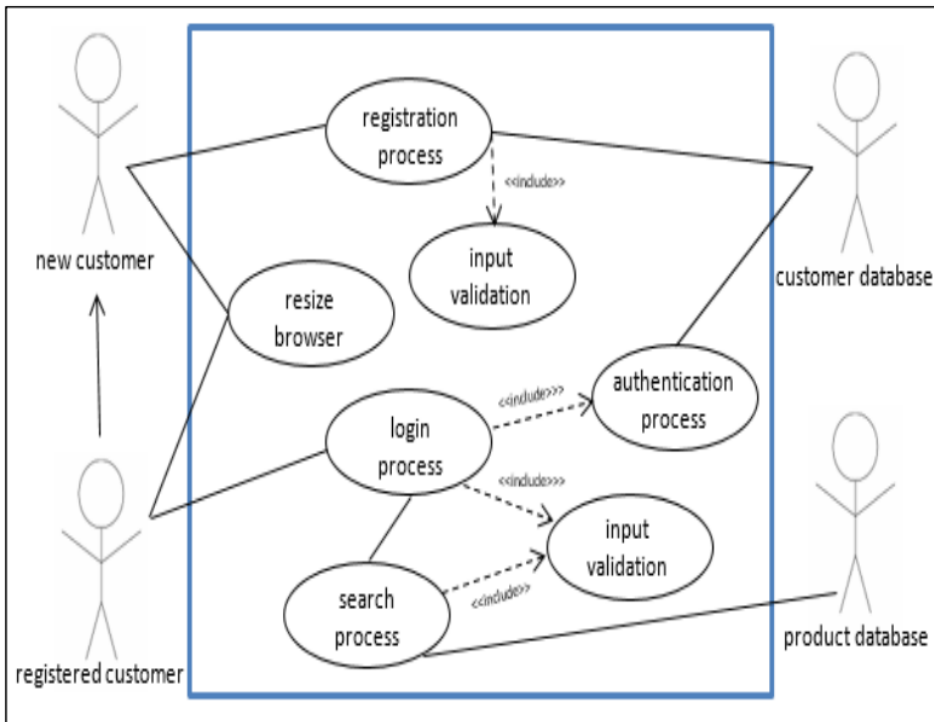
Due to the limited time available to develop this website, several restrictions are needed.

- The registration process will be used to create new customer accounts and input validation will be applied to all details entered to the website.
- The website will be made responsive to a different screen size by altering the size of the browser window. It doesn't need to be tested on different devices.
- Customers will navigate the site using a horizontal navigation bar.

Constraints

- Time: I have 6 months to complete my project.
- Technical: I will be learning PHP during the AH course.
- Costs: All the software I intend to use is open-source which means there is no cost implication for the school.
- Legal: I must comply with the Copyright, Design and Patents Act and ensure that the GDPR Regulations are met.

UML Case Diagram



Requirements Specification

Purpose of the site: The purpose of the website is for users to be able to view cosmetics from selected brands when they register and log into the site.

Functional Requirements

- The website will be able to store the details of customers and products in an external database
- The website will be able to validate all user inputs to the website
- It will be easy to navigate with a fully functioning navigation bar
- Check that users are registered when they login to the website
- Pages of the website will be formatted using an external style sheet
- Users will be able to search for products by brand or by product name by connecting to the database and executing SQL queries
- Display formatted search results (details of each product found, including an image of these products)
- Session variables will be used to store a customer's login details across pages of the website
- Responsive layout that changes depending on the screen size being used

End-User Requirements

Through the analysis of my user surveys, I have come up with a list of requirements that were the most common answers amongst the people answering my surveys:

- Palatable colour scheme (e.g. light, complimentary colours)
- Easy to navigate
- Clutter free and good use of space
- Easy to use search facility to search for products by name and/or by brand

Because of the user surveys, I will ensure to include these requirements when implementing the website to create the best user experience I can.

Inputs and Outputs

For users: Email Addresses
First Name
Last name
Home address
Telephone Number
Username
Password
Search criteria

For Website: Processes: Validation of all customer inputs
Open and close connection with database
Save customer details in database
Authenticate customer login
Execute SQL queries
Search for products and format query results
Use of session variables to preserve user login data

Outputs: Validation error messages
Personalised message using session variables on search page
Search results showing product details, including an image

Inputs: Populating the website with details of makeup products
Login query results
Registration query results
Search criteria and search query results
Size of the browser window

Project Plan

Resources Required

During the development, I will need access to a computer with the following software resources:

- Text editor
- Browser
- Web server tools

For clarity, I have repeated the tasks to be completed at each stage of the project's development.

Tasks	Est Duration (hours)	Target Date
Analysis Read requirements for AH project and create idea Write outline of project Scope, boundaries and constraints UML Use Case diagram Requirements specification - end-users Requirements specifications - functionality Description of inputs and output Project planning	6	6 th October 2018
Design Interface design - wireframes Structure diagram - website Pseudocode (server-side tasks) Database design	6	2 nd November 2018
Implementation Database structure HTML pages Navigation bar (CSS) PHP (registration page) PHP (login process) Session variables PHP (search process) Media query Validation (registration process) Ongoing testing Description of research/new skills acquired	24	15 th February 2019
Testing Test plan (requirements) Persona and test cases Final testing Test evidence	10	8 th March 2019
Evaluation Fitness for purpose Maintainability Robustness	2	15 th March 2019

Candidate 2 Commentary

The evidence for this candidate has achieved the following marks for each part of the analysis stage.

Description of the problem (2 marks)

The candidate was awarded **1 mark**.

Although the scope, boundaries and constraints have been listed, the details provided are not appropriate at Advanced Higher level. This meets the requirements of a WDD project that integrates with the DB content of the course.

UML use case diagram (2 marks)

The candidate was awarded **2 marks** because the complete and detailed use case diagram shows all necessary actors, use cases and relationships. Integration with the database is clearly indicated.

Requirements specification (4 marks)

The candidate was awarded **4 marks** because a complete and detailed requirements specification has been provided. Functionality needed to support integration with the database has been clearly indicated.

Project plan (2 marks)

The candidate was awarded **1 mark** because the project plan indicates the tasks that are required together with an estimate of the time needed for each task. Although the time estimates for each task are not clear in the Gantt chart, the additional table provides sufficient detail. Resources have been mentioned briefly but lack detail of specific software products.

Total marks awarded – 8 (out of 10)

