­­­­­­­Table of Contents

[Analysis Section 2](#_Toc368343391)

[1. Feasibility Study (background to - and identification of - the problem) 2](#_Toc368343392)

[2. Formal Investigation 3](#_Toc368343393)

[**Interview** 3](#_Toc368343394)

[**OBSERVATION RECORD** 3](#_Toc368343395)

[**DOCUMENT ANALYSIS** 4](#_Toc368343396)

[3. Description of the Current System 5](#_Toc368343397)

[4. System Requirements 5](#_Toc368343398)

[5. Identification of the Prospective User(s) and their skills 5](#_Toc368343399)

[6. Constraints/Limitations 6](#_Toc368343400)

[7. Data Source(s) and Destinations(s) 6](#_Toc368343401)

[8. Data Volumes 6](#_Toc368343402)

[9. Analysis Data Dictionary (from the perspective of the end user) 6](#_Toc368343403)

[10. Entity Relationship Model 7](#_Toc368343404)

[11. Object Analysis Diagrams 7](#_Toc368343405)

[12. DfDs (existing and proposed system) to level 1 7](#_Toc368343406)

[13. Objectives for the proposed system 8](#_Toc368343407)

[14. Realistic appraisal of the feasibility of potential solutions 8](#_Toc368343408)

[15. Justification of chosen solution 9](#_Toc368343409)

# Analysis Section

## Feasibility Study (background to - and identification of - the problem)

Name of Client: Matthew Wilson

Working Title of System: Saviour Backup System

**What is the purpose of the new system?**

To allow users to backup removable media easily, quickly and automatically, so that in the event of something happening to the media, the backup would still be intact, and the software can restore it to another device. This can also be done through the software.

**What are the anticipated benefits of the new system?**

Reduces the chance of data loss on removable media, because the media can be backed up in the background automatically with little affect to the user.

**What useful outputs / information will the new system produce?**

Backup files of the removable media and config files (which are tiny) stored on the memory stick.

**Who will enter data into the new system?**

The data will be entered by no one. There is no data to enter. The configuration for the USB manager for backups is generated automatically, from a template. The settings from that are set up by the user or the owner of the media.

**Who will make use of the outputs / information?**

Any business, big or small, even home users that want to keep their data backed up seamlessly. If someone has important data that they carry around on a USB drive, the software will stop them losing the data in the event of something happening to the drive.

**Will the new system need to permanently store data?**

Yes, it will store the backed up data on the client computer, and maybe on an FTP server they own. There will also be configuration data stored on the USB device.

**Is there an existing system that will be replaced?**

No, after research there is no other piece of software like this in existence on the global market, however there may be bespoke software solutions in place. This system isn’t designed to replace any other system, therefore there is no consideration needed for this product to meet any prior requirements or interfaces.

**Is there an existing system that the new system will need to interface with?**

Other than the windows operating system, no. Everything will be done by the program with no need for any other systems, other than the software needed for the program to run (.Net framework etc.)

**Does the new system need to run on any particular hardware?**

It must run on a windows based machine, in the desktop environment. It will be compiled for 32-bit to maintain support for different computers, and can be compiled for any different version of windows (8, 7 etc.).

**Outline a possible solution**

Programming Language: The program will be written in C#, with XML for the config files. Small pieces of other languages may be used (possible SQL and Batch, and **MAYBE** small amounts of C++ for file management)

File System: The file system will be windows based, and will be installed onto the client’s machine. Visual studio will make the installer, to make sure there are no problems with it there.

User Interface: The user interface will be using windows forms mixed with DotNetBar, which allows it to look nicer than stock. There will be some sections where console readout will be used for status.

## Formal Investigation

### **Interview**

***What is needed***

1. Can you tell me more about what you want the system to do?
   * Backup on ejection (ejection through program that is)
   * Simple user interface
   * Resource friendly
   * Backup restore in case of loss or replacement of drive.
   * Only backup drives that have been set up.
2. Does it need to run on any particular hardware?
   * No
3. Does it need to use any particular software?
   * No, all run inside windows
4. Should it be able to interface with any other systems?
   * No, run silently in the background

***Problems***

1. What are the benefits of the new system to companies and home users?
   * Piece of mind that if I lost my important data, then there would be a copy of the data still. However, I would still question the need for it because I use cloud, but loads of people still use USB devices
2. What would be an indication that this system is beneficial over other products?
   * That I have piece of mind that it does it itself.

***The users***

1. Who will use the system?
   * Small networks and home users
2. Are they likely to be competent IT users?
   * No
3. What level of user documentation and/or training will they need?
   * Basic overview of how to turn it on and make sure its running

***Security/Compliance***

1. Should the system store user information for analytics etc?
   * No, just make it work, simple as.
2. How secure should the software be (Password protected etc)?
   * No, it doesn’t worry me. The computers are likely to be password protected.
3. How should the data be stored (settings for backup, drive identity etc)?
   * Just as a basic folder, no compression.

### **OBSERVATION RECORD**

**You are looking to understand how the system works at present. Observe the main processes of the current system, what you noticed and any problem areas IN AS MUCH DETAIL AS POSSIBLE. Ask questions while you watch. Make sure that what you see is typical; find out if anything different ever happens!**

**State where you went and who you watched.**

**Summarise what you saw and any problems in table form, e.g.**

|  |  |  |
| --- | --- | --- |
| **Process** | **What I observed** | **Any problems** |
| 1. Take order | A customer phoned to place the order. Sam wrote down the order details on a note pad and calculated the total on a calculator. The note was then put into a tray marked “orders”.  Apparently a few orders do come in by email. When this happens, the customer is still phoned for the details to be checked. | The details of the order were hard to decipher as Sam had the phone in one hand while he wrote. |
| 1. Check that the item stock | Sam went to the stock room to check that the item was in stock. The item is then retrieved and put in a dispatch area. | The stock room was a mess and it was hard to find anything.  Apparently if there is no stock they have to call the customer back to say there will be a delay and the item must be ordered from supplier. |

### **DOCUMENT ANALYSIS**

There are no documents from any past system that needs to be analysed.

## Description of the Current System

There is currently no system that needs to be updated or replaced. And there are no other systems that do anything like what I am designing the system to do. There are other systems that do roughly the same thing (backup data), but they aren’t appropriate really to this because they aren’t designed for removable devices.

## System Requirements

**What the client and/or users want the new system to do in as much detail as possible.**

## Identification of the Prospective User(s) and their skills

**Identify the person(s) that will be the end-user(s) of the system. This is not necessarily the client and should come from the “answers” to your interview questions. You could identify them by name and role, or by group. If different users will use different parts of the system you need to explain this too.**

**e.g. Mrs Dowling, the IT teacher will enter students grades into the grade book..... or .......all of the IT Teachers will use the system to enter grades and produce progress reports.**

**Describe the competency of each end-user, or group of users, in terms of their IT skills. Do they use computers on a regular basis? Are they used to using the type of system that you envisage? Will any of them have any special requirements – e.g. for poor eyesight ? What type of user documentation will they need? What type (if any) of training will they need?**

## Constraints/Limitations

The software needs to run on windows, and be very lightweight when running. It needs to run a background service when the computer starts up, so it can scan any device from start up.

## Data Source(s) and Destinations(s)

**Summarise where the data for the system will come from. Is it provided by customers? Does any of it come from another system?**

**Where will data and/or information that is produced from the system go to? Customers? Head Office? Other departments? Inland Revenue?**

## Data Volumes

The system will be run every time the computer is started, and used heavily whenever the back. The customer base and user base is anyone, it will be released on the internet, I just have a client that wants to sell the software and needs someone to make it. Events will be triggered when the user starts up the computer, or when the user inserts a removable device into the computer. Data will be kept for all the time that the computer keeps a backup record of the drive.

## Analysis Data Dictionary (from the perspective of the end user)

**This flows from the analysis of existing documents/systems or from notes you took during interview/observation. You need to identify all of the data that your system needs to store.**

**This should not be technical - use terminology that your client/user would use. The data names can (and should) have spaces in. The validation is how the client/user would specify any restrictions/rules for the values that are allowed.**

|  |  |  |  |
| --- | --- | --- | --- |
| **Data Item** | **Description** | **Type (and length)** | **Validation** |
| Product Code | A Code made up of two letters and 4 numbers that uniquely identifies a product | Two letters and four numbers (6 digits in total) | Must conform to correct code pattern.  Every item must have a valid product code. |
| Price | Used to store current price of each product | Currency | In range £1 to £500 |
| Quantity | Used to store the quantity of each product in stock | Numeric | In range 0 - 100 |

## Entity Relationship Model

**You need only do this if you are anticipating a database storage solution. If you can’t do it yet, leave half a blank page**

## Object Analysis Diagrams

**You need only do this if you are anticipating use of Object Oriented Programming (OOP). If you are not sure how to do it, leave one blank page**

## DfDs (existing and proposed system) to level 1

**You can leave this for now – leave 2 blank pages**

## Objectives for the proposed system

**This is the most important part of the analysis – you are going to state the objectives of a system that looks sufficiently COMPLEX – i.e. HAS LOTS OF PROCESSING REQUIREMENTS. The system needs to be “open ended” – i.e. you should be able to add/update and delete things so that the system can be used for the forseeable future.**

**NUMBER the objectives – the exam board says you should have around 20!**

**You must produce clear and measurable objectives that are specific to your project. Generalised objectives such as ‘user friendly’ are not SMART:**

**S**pecific – State precisely what you will achieve?  
**M**easurable – Will you be able to measure your success?  
**A**chievable – Are you capable of achieving your objectives?  
**R**ealistic – Can you achieve them in time with your available resources?  
**T**ime – When do you want to achieve the objectives by?

* It may help to group your objectives into
* Inputs. For example: the system must provide a means to input order details making use of drop down lists of options wherever possible, to minimize input errors.
* Outputs. For example: the system must produce a report of **last year’s sales**, grouped and sub-totalled by month.
* Processing. For example: the system must calculate monthly sales totals
* Storage. For example: the system must store customer details in a way that they can be retrieved by specifying surname or first name.
* Security/Backup. For example: the system must be password protected. Only managers should be able to change staff rotas.
* Technical. For example: the system must run on a Windows 7 computer with 2 Mb RAM and an Intel Core Duo processor.

## Realistic appraisal of the feasibility of potential solutions

**The proposed solution needs to meet the stated objectives. Consider three alternative solutions for each area. This is one area in which complexity will be assessed so be detailed (especially for your preferred solution)!**

**Explain briefly how each solution would work in view of its ability to meet the most important of the requirements/objectives.**

**Consider the negative points of each solution too (making sure your preferred option has a lot more positives than negatives!!!).**

|  |  |  |  |
| --- | --- | --- | --- |
|  | Option 1 | Option 2 | Option 3 |
| Programming Language |  |  |  |
| File Storage/database management system |  |  |  |
| GUI |  |  |  |

**You may have other options to consider depending on the nature of the system.**

## Justification of chosen solution

**State what your recommended solution is and why. Make sure you focus on meeting the system objectives. There should be some convincing reasons why this is the best solution.**