**Unit 22 System Analysis and Design Task 1 Doc 2**

**Mohammed Mahin Ibnay Mamun / 346584**

**Software Development tools and techniques**

**What is CASE?**

CASE stands for Computer Aided Software Engineering. CASE is used to speed up the building process of the software system. CASE is a category of tools. Computer Aided software engineering was established in the 70s. Software tools are used to help the maintenance and development of software. The reason for using the case is to develop software.

Case can be suitable for various stages; therefore, it is separated into two dissimilar parts. These parts are UPPER and LOWER. Lower is appropriate for coding, design, and development. On the other hand, upper is acceptable for analysis, documentation, and requirements.

Automatic methods are created to save time Aswell as avoiding human errors. This technique is highly effective when it comes to repetitive tasks.

The central reason for CASE tools is to lower costs and develop time and expand software quality.

**Why is CASE important**

Software development teams rely on CASE because it increases productivity by saving time.

The reason CASE tools are developed is for:

* Fast installation.
* Saves time by reducing the time of the testing.
* Creates documents.
* Increases speed of the system development.

**Why is CASE used**

**Advantages**

* Quality is improved by case tools
* Makes systems which closely reach the user’s needs.
* Put together systems with documentation
* The tools are more successful for greater scale systems
* Time is reduced and errors by using the CASE tools

**Disadvantages**

* Highly confusing
* Is not simply maintainable
* High quality
* Trained staff are needed
* It might be harder to use with existing systems

**SSADM**

**What is SSADM?**

SSADM stands for Structured System Analysis and Design Method. This massively had a focus on the overall design. SSADM is also described as the waterfall model. This comes handy for the systems analysts who are part of smaller groups. In contrast to the waterfall method, SSADM has 6 stages of design and analysis.

**Why is it important?**

This method represents an extreme example of document system design, in contrast with more contemporary agile techniques such as DSDM and Scrum.

**Why and when can it be used?**

**What are the stages of SSADM**

* Feasibility stage
* Requirement analysis stage
* Requirement’s specification stage
* Technical system specification stage
* Logical system specification stage
* Physical design stage

**Feasibility stage:**

At the first stage, the logical requirement must be followed against the company's model and how they can meet their targets. During the requirement, investigations will take place. This is done to check for any logical problems.

**Requirement analysis stage:**

At the Requirements analysis stage, the study of logical requirements takes place of the development in the system. Business system options are evaluated at this stage.

**Requirement's specification stage:**

At this stage, A presentation of logical paths is produced, and it is put using analysis tools. For example, flow charts and data flow diagrams. The implementation will also be focused on. This stage is massively targeted around prototyping.

**Technical systems specification stage:**

During this stage, the development environments are implemented. These are decided bason on choices which were previously made.

**Logical system specification stage**

At this point, designs are updated with their specifications which are being developed further so it is ready for future production. If you have any other updates which are required, then it will be defined at this stage.

**Physical design stage:**

At this physical design stage, the project vision is ready. It is ready to be put in a stage of designs which are physical and processed from any choices and updates which have been made previously. Protocols and functions which are included in the project are finalized.

**Advantages**

* Does not need specific skills.
* Less errors by humans.
* SSADM lets one plan.
* Analysis of user needs is done.
* Documentation is taken very seriously.

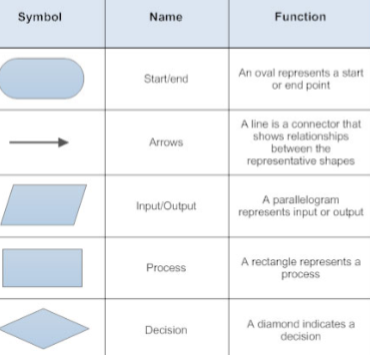
**Disadvantages**

* Over analyzing.
* The outline diagram can be unclear.
* Consistence checks cannot be carried out.

**Flowcharts**

**What is a Flowchart?**

The sequence of movement in a diagram is a Flowchart. Flowcharts are visual representations of a moving step and decision sequence which must be followed to perform a process. During the sequence, each step takes a shape followed by an arrow either to the side, up or down.



in this diagram, we can see all the symbols which are used in flowcharts. The diagram also tells us the name of the symbol and the function of what it does.

**Why do we need one?**

We can use flowcharts for several separate occasions in our daily lives. We also use flowcharts without even knowing. For example, when we make tea or coffee, we use the steps of a sequence which is also known as a flowchart.

We can also use flowcharts when: programming, workflow management, guides for troubleshooting, algorithms, and pseudocode.

**Advantages of flowcharts**

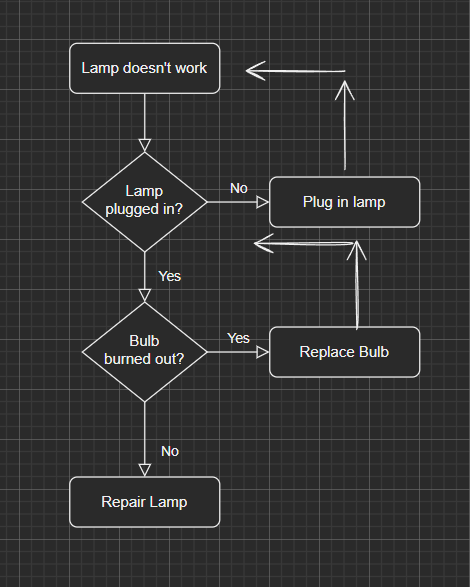
* Communication of a logic system
* More effective analysis
* Makes your task more efficient
* Helps in debugging
* Visual clarity
* Effective coordination

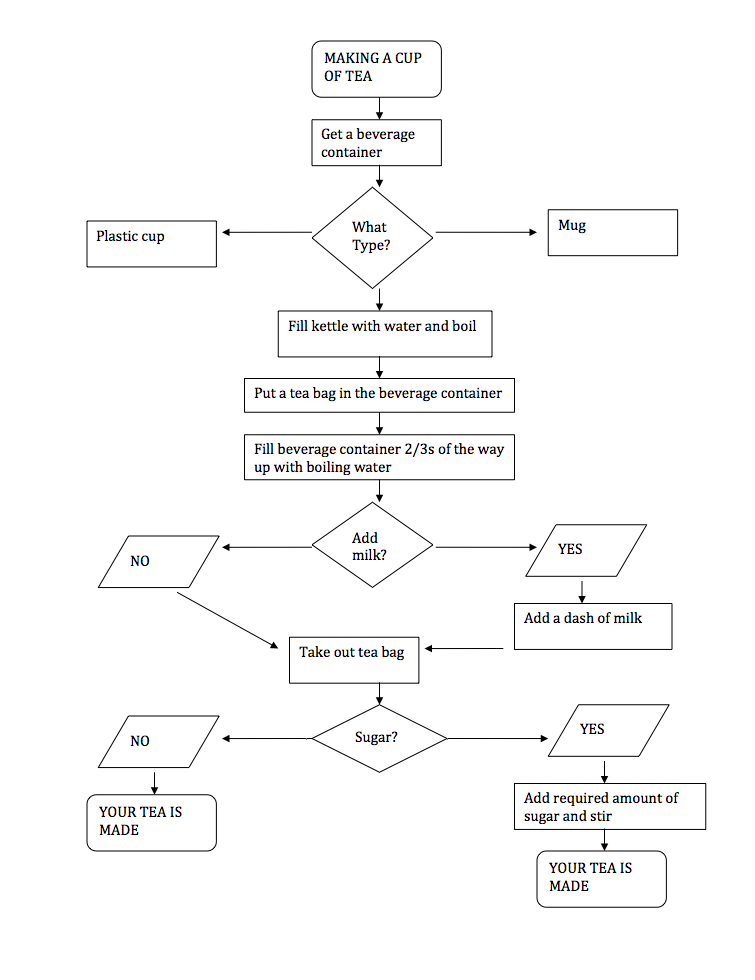
**Disadvantages of flowcharts**

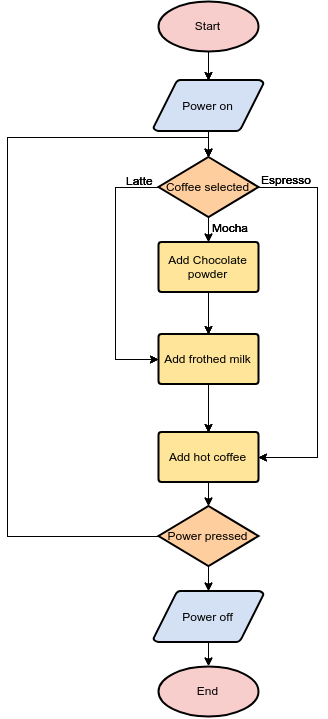
* The program can be complicated
* If changes are made you must redraw the flowchart
* Flowchart symbols cannot be typed you have to draw them
* Not made for man to computer communication

**Creating a flowchart**

I used [www.draw.io](http://www.draw.io) to make this flowchart.





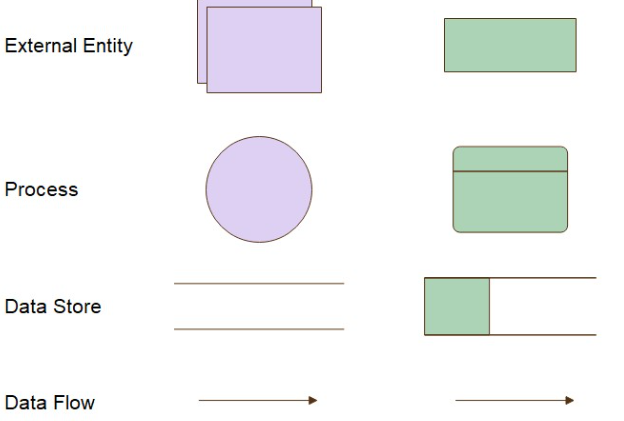
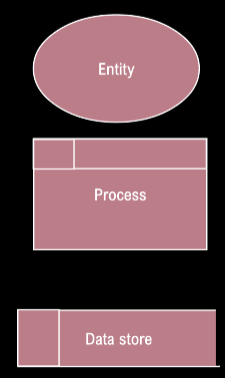


**DFDs**

**What are DFDs / Data flow diagrams?**

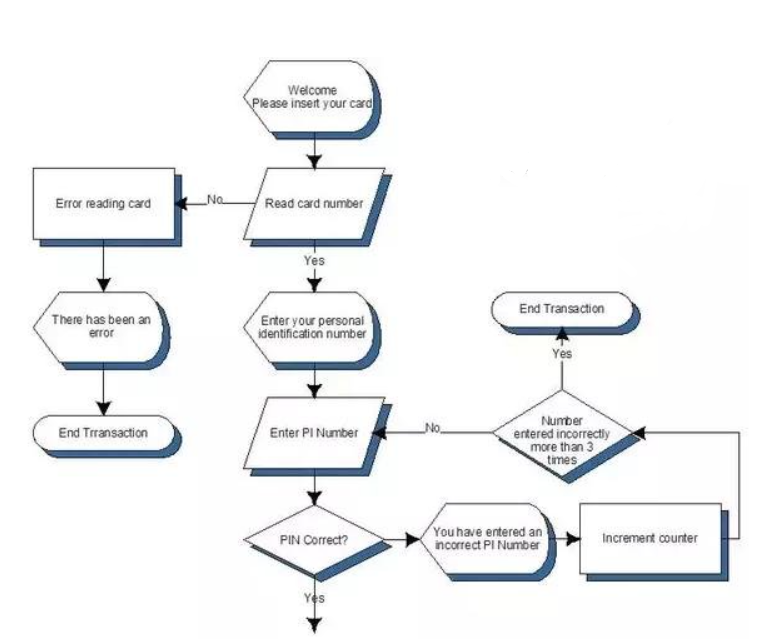
DFD is short for Data flow diagram. Data flow diagram is one way we can represent the flow of data for a process, we can also use data flow diagrams for a system. Data flow diagrams track where the data is coming from, where it gets stored and where it goes. If we compare a Data flow diagram to a flowchart, the data flow diagram is easier to understand. The reason for this is because with data flow diagrams there are no loops and decisions.

**What are the basic DFD symbols? Explain at least one.**

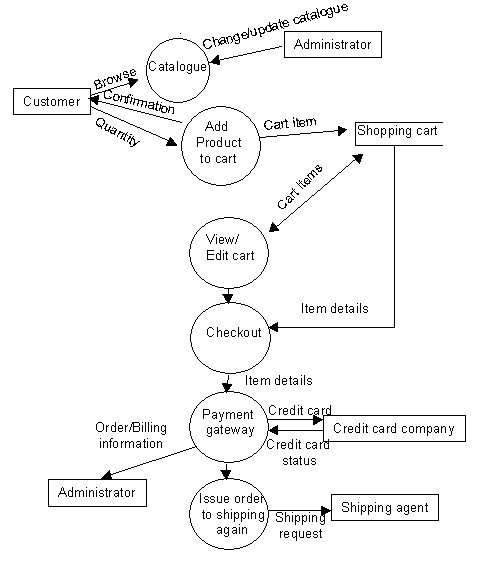


* Entities are shown by squares as the destination of data.
* Processes are represented by rectangles with rounded corners.
* Data Flows are demonstrated by arrows.
* Data Stores are open-ended rectangles.

**Creating a Data Flow Diagram**



Here is an example of a Data flow diagram for an ATM machine.



Here is a more complex data flow diagram which shows online shopping.

**Advantages of Data Flow Diagrams**

* Straight forward, making it easier to read
* Each section is broken down in detail
* Great for communicating
* Can be used for system documents
* Supports the logic behind the diagram

**Disadvantages of Data flow diagrams**

* Time consuming to make
* Physical requirements are not included
* Can be seen as a slow process
* Different symbols: gane and Sarson

**ULM / Unified Modelling Language**

UML is commonly mistaken for as a programming language, however it is not. It is a general modelling language. Rather than a programming language, it is more of a visual language. UML also has diagrams. These are different to the other types which are DFD and Flowcharts. ULM is used for portraying the structure of a system Aswell as the behavior.

This is highly beneficial for businesspeople, architects and software engineers who must model, Analyse, and design.

Structure diagram – Class diagram and component diagram

Behavior diagram – Activity diagram, use case diagram and interaction diagram

**What is a Class Diagram**

Class diagrams are a type of UML. UML stands for unified modelling language as I mentioned before. Class diagrams are a type of structure diagram which is used to describe the structure of a system. The classes which are in the class diagram define the main elements, interactions in the system and the classes which need to be programmed.

**Why are they used? / Why would we need one?**

Class diagrams are used for general conceptual modelling Aswell as detailing. One reason for using class diagrams is for data modelling. What is data modelling? Data modelling is the procedure of developing data model for data which is getting stored in a database.

**What are the Characteristics of class diagrams?**

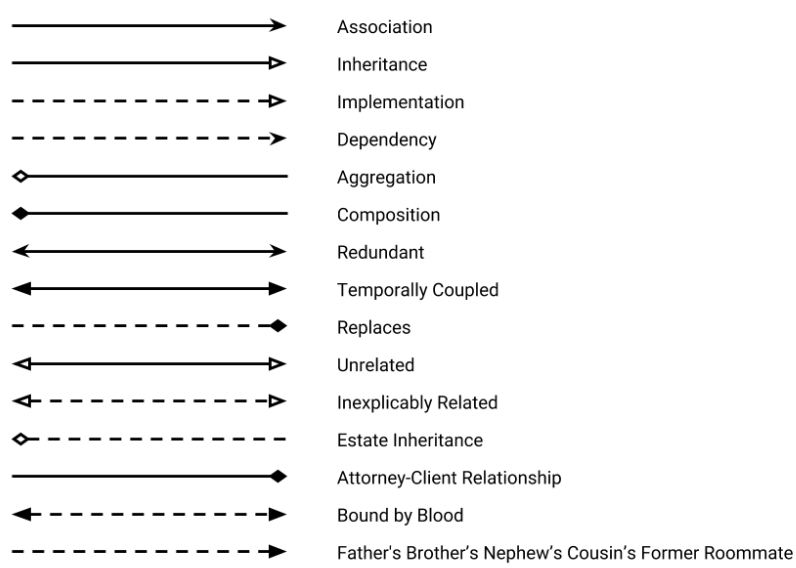
Methods – Lets you itemize behavioral features of a class. Methods are also referred to as operations or functions.

Attributes – An incredibly important region of data which consists of values to describe each class. In programming, Attributes are known as a variable. Others may also call it fields or properties.

Class – Shows what is in the system.

**What are the Class diagram Relationships?**

Below is an image which consists of a few relationship lines which are used in ULM.



In class diagrams, we use 4 from the image above. These 4 are: inheritance, Association, Aggregation, and composition.

Inheritance – Is capable of inheriting all the attributes from the class.

Association - Does not depend on any other class.

Aggregation – Can leave whenever and can be part of a subclass.

Composition – cannot exist without the parent object.

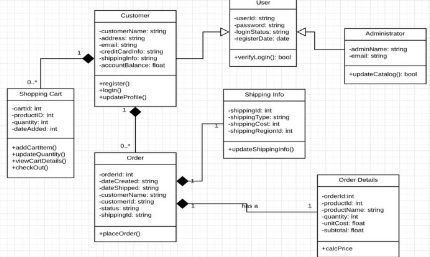
Classes in the diagram are shown with boxes which contain three components. These three components are the top, middle and bottom.

Top – consists of the name of the class. It is centered and typed in bold with an upper case for the first letter.

Middle – has the attributes of the class. This is aligned to the left and the first letter in contrast to the top component is in lowercase.

Bottom – This component contains the operation that the class can run / execute. Also left aligned with first letter lowercase same as the middle component.

**Using the example, explain the class diagram and its draw flow?**



The box in this class diagram on the top right-hand side is classed as administrator. The class in this situation is labelled administrator. Directly below that is the attribute. In this case there is – adminname = string so we know it is going to be a string and another attribution called – email which is also data typed as string. This attribute section in programming is referred to as the variable. Directly below the attribute is the method section. This is also called the function. The minus sign (-) which is placed in the boxes clearly signifies to us that they are private visibility. Having a plus sign (+) is contrast to the minus meaning that it will be public visibility. # Means it is protected and ~ means it is package or set by default.

To the left, (top middle box) which is classed as user in this image is the subclass. The administrator box in the diagram is connected to the user box with an inheritance line which means that the subclass, in this case is administrator, is inheritance to users. In this box the word ‘User’ is the superclass, and the name of the class is user. The Attributes are –userid , -password, - loginStatus and –registerData. Only the last one is data type as date and the rest are stored as string. This region is the Variable in coding.

**Advantages**

* Produces an overview of what the application will be like before having to code.
* Great for walking through and testing your design.
* Produces a blueprint for the programmer.
* Makes the programmer think.

**Disadvantages**

* Users need to learn about UML to create the class diagram.
* Making a complicated class diagram can lead to a difficult code.
* Creating the class diagram is time consuming. Leading you to having less time after.

**Activity Diagram**

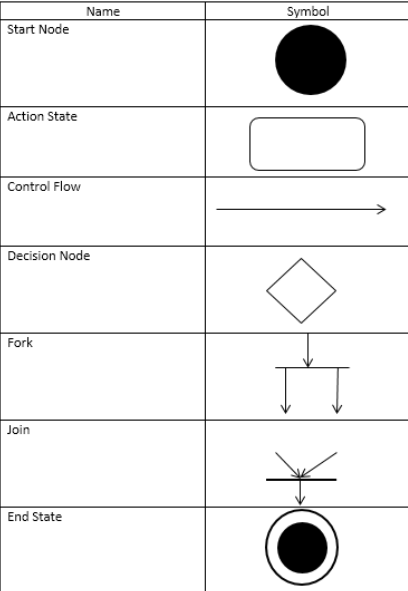
**What is an Activity Diagram?**

Data representation can be shown through activity diagrams. These can be split into more phases by an additional note which defines where the procedures will fall in the task. This can help with analysis of systems which are tasked with studying the flow of data.

**Why would we need an activity diagram?**

The reason for having an activity diagram is to capture the behavior of the system or application. This is the main reason people choose to use them. Compared to the other 4 diagrams, they are used to show to represent the flow from one item to another.

**What is the activity diagram symbol? Explain each one.**



Start node – This is a black circle it is used for the notation for the beginning of an activity. It can be on its own or you can use a note with it.

Action state / activity – This symbol is the basic block for building on the activity diagram. It usually has a description in it of the activity which is being represented.

Control flow - this is an arrow which simply represents the flow of the activity diagram. Like arrows in a flowchart.

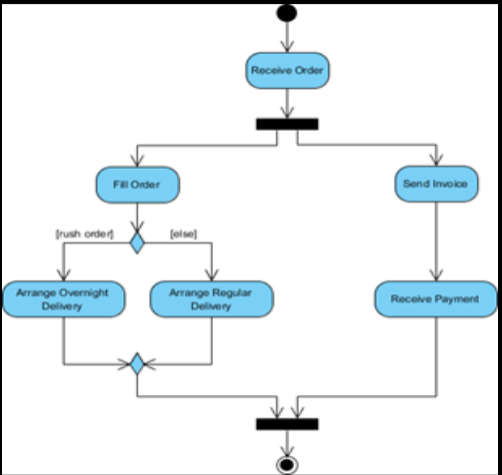
Decision node / branch – this takes the shape of a diamond, and it is used for making decisions. It can branch out 2 ways with one as (yes / true) and the other as (no/false) most of the time.

Fork – a fork in activity diagrams is used to split one activity in two.

Join - this symbol is used to join two actives back into a flow.

End node / Complete Flow – This is a black circle with a circle ring around it. This is used to express the end of the flow in the diagram.

**Using the example provided, explain the activity diagram and its data flow.**



In this example, we can see at the very top there is a start node. This is used to show the user or the viewers where to start from. After this we can see an arrow which is referred to as control flow. It is joined to an activity which is labelled receive order. Once again, we have a control flow to show us where to go next. Then we have a fork symbol. This tells us there is going to be a split in two directions. Both ways have activity one side is send invoice the other side has the activity fill order. From the left side we then still see control flows which take us to the branch. This gives us two options. We can go either way. After that there is a join symbol which shows both control flows which were previously forked out are now back in one flow. And finally at the very bottom we have the complete activity flow symbol. Showing it is the end.

**Advantages**

* Highly like flowcharts
* Easy to read and understand
* Used a small range of symbols not too many
* Diagrams are user friendly
* The diagram shows us situations

**Disadvantages**

* Can be complex if you add a lot
* Might not be used for theory
* People may just use flowcharts thinking it is same

**Business Process Reengineering (bpr)**

**What is business process reengineering?**

This is a method which is used by organizations to change their business aim. This is the measure which is taken after the decisions. This is done to restructure a process. BPR is an approach to render management when tasks required are radically redistributed.

**What organizations will use it?**

One example which is also one of the best when it comes to using bpr is ford. During 1980 the car manufactory when in recession. The business had to be analyzed by the managers at ford.

**Advantages of BPR**

* Creates an appropriate focus
* Abstracts unessential
* It coordinates many functions immediately
* Improved viality is provided

**Disadvantages of BPR**

* Not for every business
* Can replace humans when getting jobs
* Requires a lot of IT
* Not an immediate fix

**M2 - Justify the design decisions, explaining how they will meet the user's needs and be fit for purpose.**

The choices of designs which you can use are:

* Flowchart
* DFD / Data flow diagram
* Class Diagram
* Activity Diagram

**Flowchart**

How will they meet the needs of the user and be fit for purpose? Flowcharts are useful as they provide graphical representation. Flowcharts are the sequence of movement in a diagram. Flowcharts are representations of a moving step and decision sequence which must be followed to perform a process. During the sequence, each step takes a shape followed by an arrow either to the side, up or down.

Why is it fit for purpose?

* Communication of a logic system
* More effective analysis
* Makes your task more efficient
* Help in debugging
* Visual clarity
* Effective coordination

**Data flow diagram**

How will they meet the needs of the user and be fit for purpose? Using DFD makes it more consist of portraying the business requirements of the system or application. This is successfully done by representing the flow of data for a process, we can also use data flow diagrams for a system. Data flow diagrams track where the data is coming from, where it gets stored and where it goes.

Why is it fit for purpose?

* Straight forward, making it easier to read
* Each section is broken down in detail
* Great for communicating
* Can be used for system documents
* Supports the logic behind the diagram

**Class Diagram**

How will they meet the needs of the user and be fit for purpose? Class diagrams are a type of UML. Class diagrams are a type of structure diagram which is used to describe the structure of a system. The classes which are in the class diagram define the main elements, interactions in the system and the classes which need to be programmed.

Why is it fit for purpose?

* Produces an overview of what the application will be like before having to code.
* Great for walking through and testing your design.
* Produces a blueprint for the programmer.
* Makes the programmer think.

**Activity Diagram**

How will they meet the needs of the user and be fit for purpose? Data representation can be shown through activity diagrams. These can be split into more phases by an additional note which defines where the procedures will fall in the task. This can help with analysis of systems which are tasked with studying the flow of data. The reason for having an activity diagram is to capture the behavior of the system or application. This is the main reason people choose to use them. Compared to the other 4 diagrams, they are used to show to represent the flow from one item to another.

Why is it fit for purpose?

* Highly like flowcharts
* Easy to read and understand
* Used a small range of symbols not too many
* Diagrams are user friendly
* The diagram shows us situations

**D2 - Evaluate the requirements specification and design against the organization's requirements.**

**Client brief:**

You have been approached by Tameside General Hospital to analyze and design a new outpatient booking system. The current system is old, slow, and outputting many errors such as medical staff being double booked, patients being double booked, patients not knowing they have an appointment, the system generating ghost appointments etc.

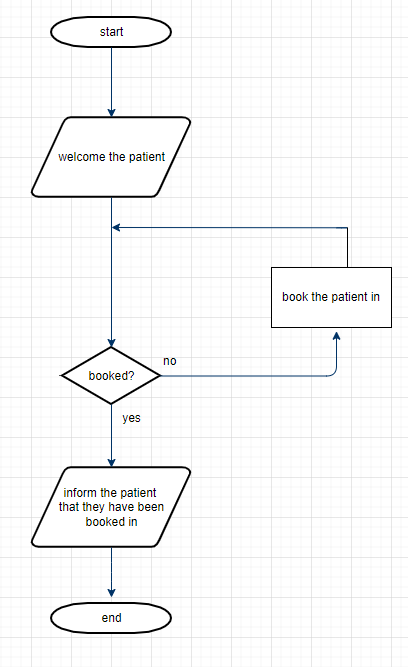
The design which is wanted by the organization is a new outpatient booking system. To create a new booking system for the organization we can use any of the following to help:

* Flowchart
* DFD / Data flow diagram
* Class Diagram
* Activity Diagram

Each of these diagrams have their own benefits which I have already mentioned before.

The main issue with their system is that the booking is being doubled Aswell as patients not being told they have an appointment.

For example, we could create a simple flowchart for their organization to fix this.



Here is a draft which I have made for this scenario. In this flowchart we can know make sure that the patient has been booked and we can also make sure that they are aware they have been booked in. To expand on this, we can also add a counter.

**ADVANTAGES & DISADVANTAGES**

SDLC:

Waterfall:

**Advantages**

* Easy to plan for the manager.
* It is the go-to approach for small projects.
* The entire process is well-documented and well scripted.
* Each phase is separate and complete within a given timeframe; there is less need for reworking.

**Disadvantages**

* Once started, changes cannot be made
* Does not work well if you have a lot of frequent changes.
* Once you agree you cannot make changes.
* It takes a huge amount of risk.
* It can lead to late detections.

RAD:

**Advantages**

* Reduced development time.
* Increases reusability of components
* Quick initial reviews occur
* Encourages customer feedback

**disadvantages**

* Depends on the teams and individual performances for identifying requirements.
* Does not work with all systems
* Requires highly skilled developers/designers.
* High dependency on modeling skills
* Quite expensive

Scrum:

**Advantages**:

* Helps teams quickly complete tasks.
* Big goals are cut down to smaller tasks to make it more manageable.
* Great for fast moving development projects.
* The team gets clear visibility.
* Short sprints enable changes.

**Disadvantages**:

* Scrum often leads to scope creep, due to lack of definite end date.
* The chances of project failures are high if individuals are not very committed.
* Needs experienced people.
* Daily meetings
* If anyone leaves during the project, it can have a significant impact.

DSDM

**Advantages**

* User involvement is higher
* Basic functionalities are delivered faster
* Projects are delivered on time
* Provides access by developers

**Disadvantages**

* Not suitable for small projects
* As it is newer than old models it is not as common
* DSDM can be restrictive and hard to work

ASD

**Advantages**

* Involved in the final users
* Succeed in early deliveries
* Creates more transparency between the two (client and developers)

**Disadvantages**

* Creates testing for each stage
* Scope creep can be created by the lack of iteration
* It is difficult to facilitate the involvement of users