ALT Revision

ALT 1 - Interactive Information Systems

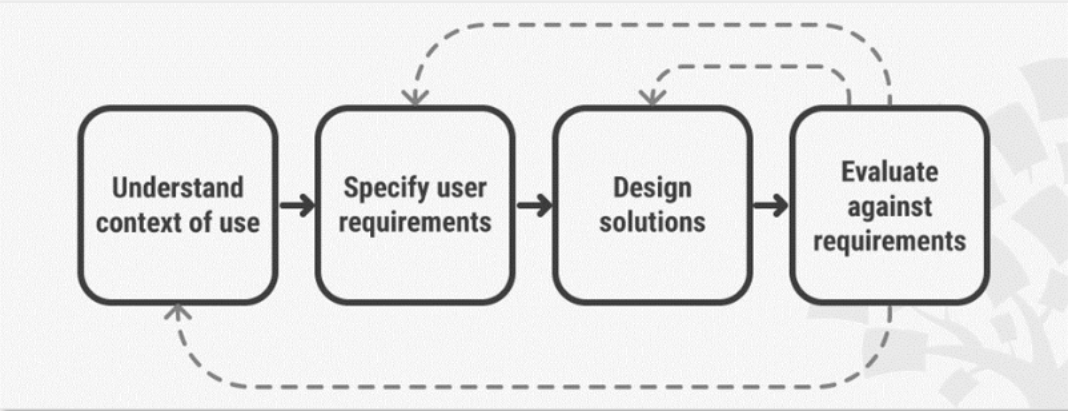
3.1 Understand and list user needs/requirements before defining a solution.

This relates to the software development section. You would carry out an **investigation** prior to developing your software where you would identify things like what kind of software users are actually looking for and what they would like the software to be able to do before you begin your coding. You would have to consider the principles of **Universal Design**, https://universaldesign.ie/what-is-universal-design/the-7-principles/the-7-principles.html, during the **design** stage when developing prototypes for the user.

User Centred Design

User experience (UX) design is the process of creating products that provide meaningful and relevant experiences to users. This involves the design of the entire process of the product, including aspects of branding, design, usability, and how a user interacts with and experiences the product, system or service.

UX design is a user-centred process and often include user research, creating personas, designing wireframes and interactive prototypes as well as testing designs. User-Centred Design is an iterative process that takes an understanding of the users and their context as a starting point for all design and development.

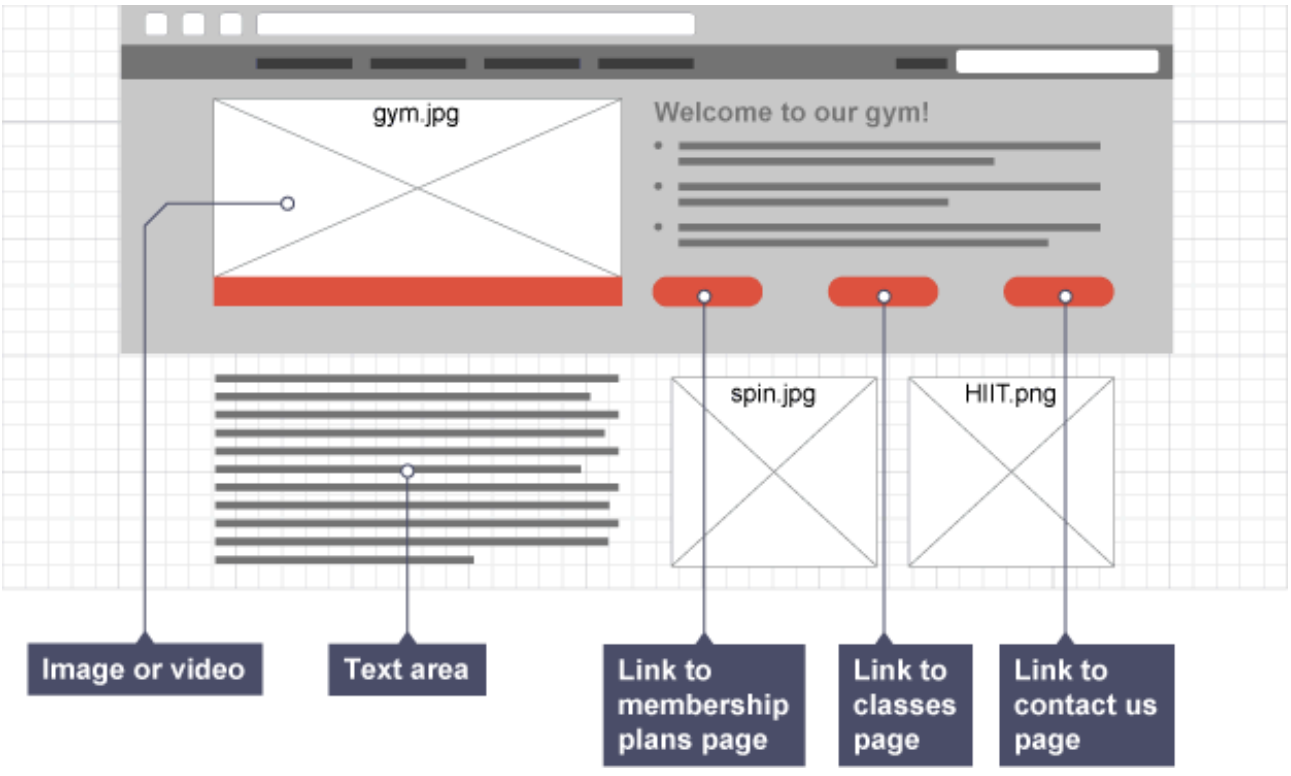


User-centred design is a different style of designing from **iterative design**, as it bases the design of a product around the needs of the **target market** rather than the continual development of **iterations**. User-centred design involves putting the end user at the centre of every decision made during the design process and, as a result, the end product is more likely to fit their needs

Web Design

Wireframes are used to design the visual layout of a webpage. You can think of them as a similar process to pseudocode for writing code, except you draw what your webpage will look like rather than code.

- Wireframes would include things like:
    - navigational bars/links
    - all text elements on the page
    - media elements (including file format)
    - elements that allow a user to interact
    - form inputs
    - hyperlinks (including types)



A wireframe can be done in anyway that suits your own design preference. You can literally draw it on a piece of paper or in Powerpoint and then hand it on to a graphical designer for them to develop. There are also a range of free tools available such as https://wireframe.cc/ .

3.2 Create a basic **relational** database to store and retrieve a variety of forms of data types

A **database** is an organised collection of data or information typically stored electronically in a computer system for storage, rapid search and retrieval of data.

A **database management system** (DBMS) extracts information from a database in response to queries from a user.

Database entity is a thing, person, place, unit, object or any item about which the data should be captured and stored in the form of properties, and tables.

Flat File database (CSV or Excel)

A flat file database is a type of database that stores data in a single table, think of an excel spreadsheet.This is unlike a relational database, which makes use of multiple tables and relations.

- Flat file databases hold all data in one table only.
- Suitable for very simple databases only.

Patient Id	Name	D.o.B	Gender	Phone	Doctor Id	Doctor	Room
134	Jeff	4-Jul-1993	Male	7876453	01	Dr Hyde	03
178	David	8-Feb-1987	Male	8635467	02	Dr Jekyll	06
198	Lisa	18-Dec-1979	Female	7498735	01	Dr Hyde	03
210	Frank	29-Apr-1983	Male	7943521	01	Dr Hyde	03
258	Rachel	8-Feb-1987	Female	8367242	02	Dr Jekyll	06

Flat File Database, all info on one database

Limitations

- Duplicated data which takes up memory and space
- Wasted space/memory because of duplicate info.
- Duplicated data takes a long time to enter and update. (For example if in the table above Dr Hyde leaves, you have to change every entry that has Dr Hyde in it)

Relational Database

To avoid duplicate data and the problems associated with it we can use a relational database.

- Relational Databases use two or more tables linked together (to form a relationship).
- Relational Databases do not store all the data in the same table.

Patient Id	Name	D.o.B	Gender	Phone	Doctor Id
134	Jeff	4-Jul-1993	Male	7876453	01
178	David	8-Feb-1987	Male	8635467	02
198	Lisa	18-Dec-1979	Female	7498735	01
210	Frank	29-Apr-1983	Male	7943521	01
258	Rachel	8-Feb-1987	Female	8367242	02

Doctor Id	Doctor	Room
01	Dr Hyde	03
02	Dr Jekyll	06

Relational Databases, patient database(left) and doctor database (right)

Repeated data is moved into it's own table as shown in the image above:

Now if doctor Hyde leaves, I only have one place to change it, in the green doctor table, rather than in each doctor id column in the flat file type.

What is a relationship?

A relationship is formed when our two tables are joined together.

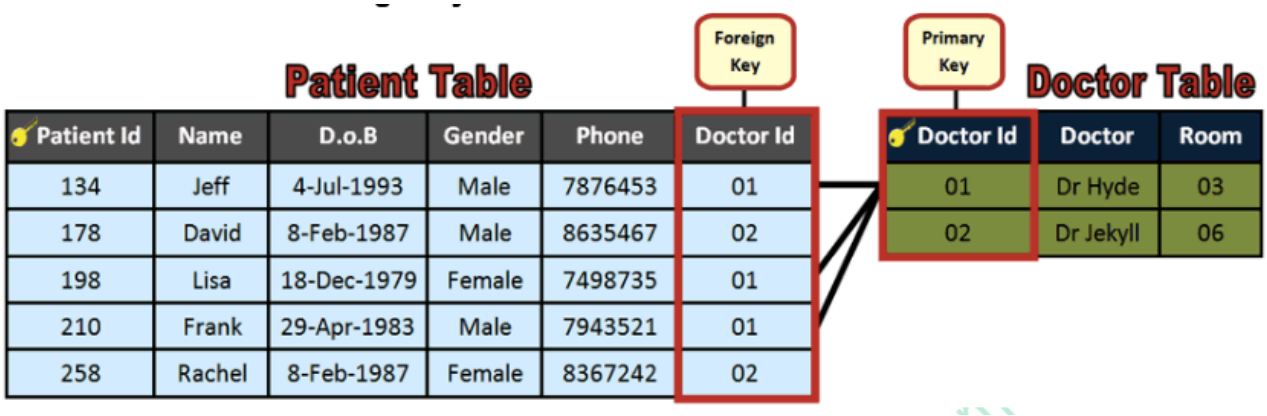
Relationships make use of **foreign keys** and **primary keys** to allow the two tables to communicate with each other and share their data.

Primary Key.

The primary key of a table within a relational database is a field which **uniquely identifies each record** in the table. It can be pre-existing data, for example PPSN or it can be generated specifically for use in the database, eg admission number for a school student.

What is a foreign key?

A foreign key is a regular field in one table which is being used as the primary key in another table. Foreign keys are used to provide the link (relationship) between the tables.



Foreign key and primary key relationship

Entity relationship diagrams

The relationship between two linked tables can be described in one of three ways:

- One to one
- One to many
- Many to many

One-to-one relationships

Occur when there is a direct **one to one link between data held on two different tables**. For example, you have your own exam number for the leaving cert. It is not possible for pupils in the leaving cert database to have more than one exam number and it is not possible for the same number to be allocated to tow or more students. It is a one to one relationship

Candidate number	First name	Surname	Date of Birth
SCN2312345	Harry	Wilson	04/03/2000
SCN4565432	Jenny	McMillan	17/11/1998
SCN5565532	Hamish	Moore	30/07/1999

1:1 Student entity database

Candidate number	Date of registration
SCN2312345	15/08/2012
SCN4565432	13/08/2010
SCN5565532	19/08/2011

1:1 Student number entity database

One-to-one relationships are usually unnecessary, as combining the data held in both entities is often possible without resulting in any duplication of data.

In the leaving cert number example, the data could be combined into one entity without duplication so a relational database would not be necessary.

One-to-many

One-to-many relationships exist where one instance of an entity can exist lots of times in another database. A good example relates to the doctor patient relational database. One doctor can relate to multiple patients.

Many-to-many

Most schools offer a variety of extracurricular clubs. Clubs can accommodate many pupils and pupils can attend many clubs. This is an example of a many-to-many relationship.

Many-to-many relationships are not good when designing a relational database as they can lead to unnecessary storage of the same data more than once (duplicated) and can make it difficult to avoid errors such as update, deletion or insertion anomalies.

ALT 2 - Analytics

3.4 Develop algorithms that can find the frequency, mean, median and mode of a data set

This outcome specifically says to **develop** so you should know how to write code to carry out all these functions.

```
mylist = ["a", "a", "b", "c", "c", "c", "c", "d", "d"] #<-- List to carry out analysis

# These are not the most efficient way to carry out these analysis
# but you don't have access to modules like statistics in the exam


# Frequency: How often each element occurs
unique_list = []

for i in mylist:
    if i not in unique_list:
        unique_list.append(i)

for i in unique_list:
    x = mylist.count(i)
    print(f'There are {x} occurences of {i} in the list.')


# Mode: Which occurs the most
max_count = 0
mode_element =''
for i in mylist:
    x = mylist.count(i)
    if x > max_count:
        max_count = x
        mode_element = i

print(f'The mode of this list is {mode_element} and it occurs {max_count}')
# Median: Sort the list and its the middle value
mylist.sort()
length_of_list = len(mylist)

if length_of_list%2 == 0:  #<-- If the length of list is even there is no middle number
    median1 = mylist[length_of_list//2]
    median2 = mylist[length_of_list//2 - 1]
    median = (median1 + median2)/2
else:
    median = mylist[length_of_list//2]
print(f'The median is {median}')


# Mean of a list of numbers
mylist = [1,2,3,4,5,6,7,8]
mean = sum(mylist)/len(mylist)
print(f'The mean of the list is {mean}')

```

3.6 Structure and transform raw data to prepare it for analysis

Data types and examples

Programmers use specific data types that exactly match a program's requirements. IN Python there are a number of data types, some of which you have used regularly as per image below and some that you haven't such as dictionaries and sets.

```
# Data Types
x = 'martin' #<--- String str
y = '9'      #<--- String str
z = 9        #<--- Integer int
a = 1.23     #<--- Float float
b = True     #<--- Boolean bool
c = [1,2,3]  #<--- List list
```

Depending on your data you may need to change between data types, for example in this ALT you may be required to read data from a CSV or txt file. This data will be stored as a 'string' in Python. If the data is numerical you will have to change the data from a string to an integer or float before carrying out numerical analysis on it.

Processing the data may also mean having to take into account blank spaces or invalid data. For example, the 2020 paper has this data set:

Surname	Gender	Age	Time
Murphy	M	17	13,12
Ogene	M	16	12.14
Ogene	M	16	12.14
Mc Intyre	F.	17	12.87
Lopez	F	-18	14.01
	F	17	1 329
McCarthy	M	77	13.65
Ó Brádaigh	f	16	13.09

This table is considered the **raw** data as it has not been prepared for analysis. There are a number of issues with the data ranging from blank spaces were there should be names and a negative age. Processing the data should remove or resolve all these issues.

ALT 3 - Modelling

3.10 Explain the benefits of using agent-based modelling and how it can be used to demonstrate emergent behaviour.

A ‘model’ is a representation of a real-world situation. Using processes like decomposition and abstraction complex systems can be simplified and studied using computer systems. For example, a government will trying plan how it will spend its budget and develop the country, it will using data about birth rates, death rates, migration rates etc to build a model of population growth in the country and use this model to plan development of housing and infrastructure.

Simulation is the process of implementing the model in order to understand the behaviour of the system, and to evaluate various strategies for the systems operation.For example, in our population example, after running a simulation of population growth the government might notice a growth in population in West of Ireland and take action to invest more money in that area. Common examples of simulations include:

- Financial risk analysis
- Population predictions
- Queueing problems
- Climate change predictions
- Engineering design problems

Simulation is also used to test a theory before it is turned into a physical or practical form. Again, in the population model above, the government will run a model to find out the effects of investing different amounts of money in the West, allowing them to plan for the future. Being able to test a theory before putting it into practice is useful because:

- Physically dangerous situations can be prevented
- Different simulations can give options that can help decision making
- Unnecessary costs can be avoided

Population growth is an example of mathematical based modelling. There is a number of proven formulas that can be used to plan for population growth, but what about a more complex system such as traffic flow? How can a formula predict how a driver will react to a yellow light? Some will speed up and others will slow down, similarly pedestrians may cross the road when they shouldn’t which will affect the flow of traffic. How could all the possible scenarios here be boiled down to a simple formula?

Agent based modelling

A system becomes "complex" when it has so many variables and interacting forces that it can't be understood simply or mathematically, for example think of all the variables that are in play when trying to create a climate change model and how each of those variables can interact with each other.

Agent-based modelling provides an approach, which focusses on modelling individual agents in the system. We use abstraction to identify and describe the behaviours of all the main components within a system, what they do, and how they are related. For example, we cannot describe the traffic in Cork City as a single function, but we know that cars – in particular those used in school runs – and delivery trucks play a major role in city centre traffic jams, as well as constraints imposed by traffic lights, one way streets, and pedestrians. Instead of trying to come up with a complex mathematical solution, each taxi, car, truck and potential pedestrian is modelled as an agent. And instead of describing the system, we describe the components, what they do, and how they interact with each other. Then we execute the simulation and observe how the overall system behaves

Nice link here to how agent based modelling was using to fight Covid and identifies some of the agents. https://www.ucd.ie/research/covid19response/news/computationalmodelling/

Benefits of ABM

- Ability to see how micro-scale behavior impacts an overall system e.g how is traffic effected if the time taken to change between red and green lights is changed?
- Ability to model individual decision making entities and their interactions.
- Ability of agents following simple rules to simulate a complex, nonlinear, and dynamic system
- Particularly useful when too little is known about a system to construct an operating equation-based mathematical simulation.

ALT 4 - Embedded Systems

All these concepts are covered in computer systems really.

Key things to be aware of are:

- Difference between digital and analogue signals and give an example.
- What is an embedded system, how is this different to a general system.