Infective Endocarditis Database

Group No: B6

Mentor: Dr Shelan Jeawak

Group Members:

Sandra Sahnoune 21039366

Aisha Alsayed 20044865

Ahmed Amr Mohamed Abdelaziz Bayomi 20051210

Alaeddine Ballaou 21003026

Hamza Addad

Table Of Contents

1. Introduction:

The human body has millions of functions and is an extremely complex and sophisticated system. Man has pieced together all these complex functions via his studies and trials. Medicine became an important aspect of the research as science and technology advanced. Medical science evolved become a distinct field of study over time, and involved by building various institutions such as hospitals, currently make up the Health Sector. The heart known as the life's organ is facing a terrible disease called "Infective Endocarditis." Infective endocarditis (IE) is an infection of one or more heart valves, the mural endocardium, or a septal defect on the endocardial surfaces of the heart. Severe valve insufficiency, persistent congestive heart failure, and myocardial abscesses are among the intracardiac consequences. If left untreated, IE will eventually be fatal. Endocarditis can appear gradually or quickly, depending on the bacteria that are causing the illness and if you have any underlying heart issues. Endocarditis signs and symptoms vary between people. When germs, bacteria, enter your circulation, travel to your heart, and adhere to faulty heart valves or damaged heart tissue, it's called a heart attack. Endocarditis can also be caused by fungi or other pathogens. As a result, scientists are working tirelessly to provide a helpful remedy and effective drug to combat this horrible sickness. However, the number of various episodes that IE may make is in the vast number, and unique between people, also the work performance of the record takes a long time and is prone to mistakes. The main idea of this project is to automate and build a high-level software system capable of storing and displaying all data related to patients, hospitals, and

diseases in a clear and efficient manner, as well as making it simple and flexible to use at any time and from any location to aid doctors and scientists in their research. This project involved a team worker of five developers, with different tasks to implement over a limited period. The programme has been designed in such a way that the user will have no trouble using it and that subsequent growth will be achievable without much work. The major goal of this challenge is to automate rather than manually complete each IE activity which is time consuming and produce an accessible software to those without any programming background.

2. Aims and Objectives:

Hospitals are an important part of our life, giving the greatest medical care to those suffering from a variety of diseases, which can be caused by a variety of factors such as changing climatic conditions, increasing workload, mental trauma stress, and so on. It is vital for hospitals to maintain track of their daily operations, especially one redoubted service “Cardio” which supports Infective Endocarditis disease.

The main aim of this project “IE system database” is to design a paperless database, where all the records are saved and display clearly in one and unique software which interacts easily and efficiently with the user as well as providing excellent data security and robustness.to build a database using the given csv files that enables the audit to enter and manage data both individually and bulk data , in addition to that the software will have the ability to produce

basic and complex reports through a clean and simple interface.

The proposed software product will be used in any Cardio Sector and in long term.

The challenges of this concept are highly leveled and specified as follow:

A. Should Write the report along the implementation
B. Must display data in a clear, easy and efficient way such as Tree View table.
C. Must have multiple buttons in order to manage data; read, write, update, delete.
D. Won't use extra framework and make it simple except the ones included in python packages in order to minimize the costs of the project and pandas for reading csv files.
E. Must include options as import files for future purposes.
F. Should finish the project within the 3rd Friday of April

The expectation is to overcome all these challenges, the purpose is to create a new concept of managing data in both simple and efficient way by avoiding high anticipations in order to honor pledges.

3. Literature Review:

The first stage is to do extensive research on the examined topic to have a thorough understanding of the type of data we are working with and to visualise the design to be implemented later.

A. Infective Endocarditis disease:

- **Pathology:**

Infection of the heart's endocardial surface, including valves, chordae tendinea, septal defect locations, and the mural endocardium.

Infection of the endocardium, frequently involving a valve, the mitral or aortic valve being the most common. It is possible that prosthetic or natural valves will be affected. Usually occurs after transitory bacteraemia and turbulent flow through the valve.

Induces the growth of vegetations on valves that include bacteria, fibrin, and platelets. Individuals with prosthetic or abnormal native valves are at a higher risk, but individuals with normal valves account for half of the occurrences.

Septicaemia causes systemic and multi-system disease, which is exacerbated by septic or sterile emboli and immune-complex deposition. (Mylonakis, E., & Calderwood, S. B. (2001))

- **Causes:**

Strep Viridians is the most prevalent bacterial infection in those who have valve dysfunction (Sick Valves). IV drug users (Substance Abusers) and those with artificial valves are the most susceptible to Staph Aureus. Another common cause is enterococci.
Candida, Aspergillus
SLE, cancer are examples of non-infectious diseases.

- **Classification:**
  - Acute IE (50 percent of cases): develops over days to weeks, usually in normal valves, and is caused by Staph.
  - Subacute IE (also known as subacute bacterial endocarditis, or SBE) is a endocarditis that develops over weeks to months and is caused by Strep viridians.

  The distinction between these two can be complex

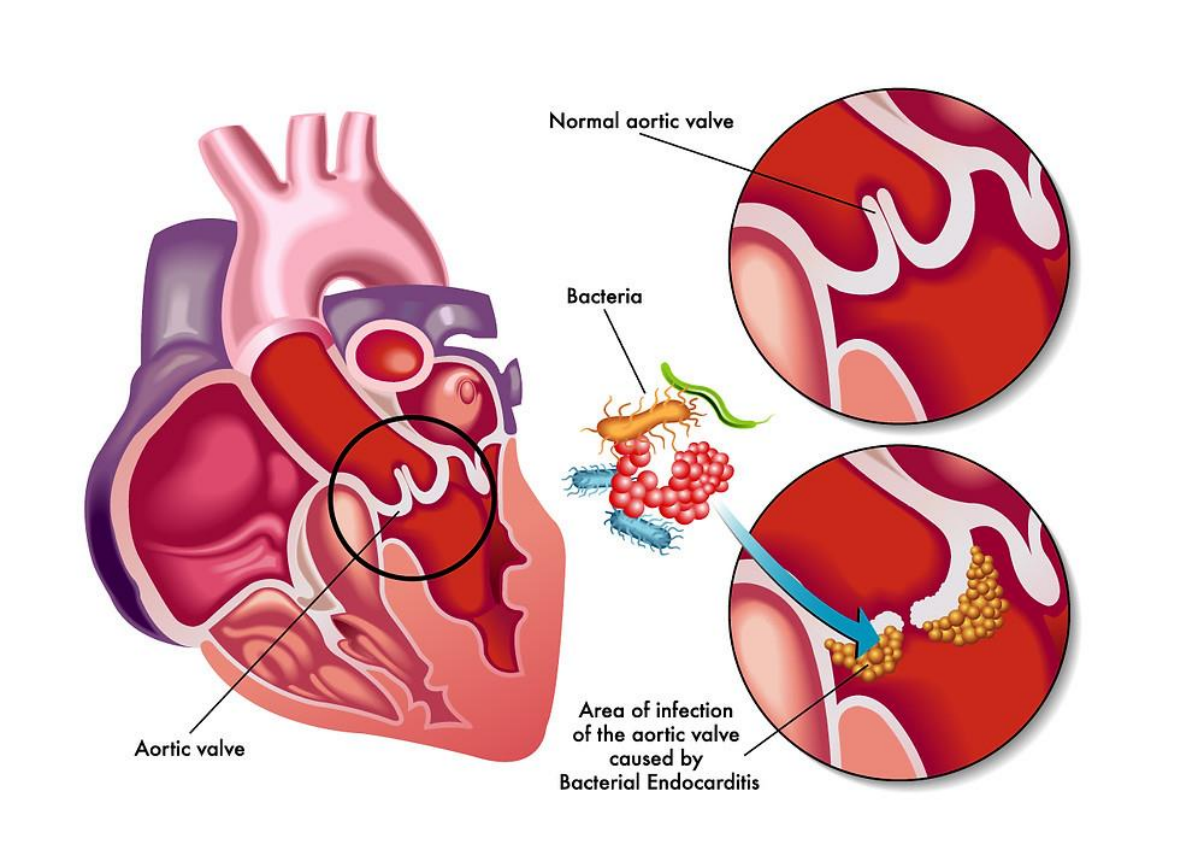


Figure 1

- **Signs and symptoms:**
  - Fever, rigours, and night sweats are all indications and symptoms of septic infection.

- Malaise
- Loss of weight.
- Splenomegaly, clubbing, implying subacute IE since it takes time to manifest.

Infarcts and abscesses can be caused by emboli, which most usually damage the brain (stroke), lung, and spleen.

Subacute illness is frequently indicated by immune-complex deposition and vasculitis. Among the warning signs are:

- Petechiae and splinter haemorrhages are the most prevalent symptoms.

Janeway lesions are plantar/palmar lesions that are painless.

Osler's nodes are infarcts in the distal phalanges of fingers and toes that are painful ('Ouch').

Retinal haemorrhages with a pale centre are known as Roth spots.

Glomerulonephritis

- **Risk factors:**

Increased turbulent flow:

- Valve disease.
- Prosthetic valves.
- Structural disease: unrepaired PDA, VSD, HCM.
- Rheumatic heart disease.

Increased pathogen entry and bacteraemia:

- IV drug use.
- Haemodialysis
- Dermatitis

Chronic disease:

- Diabetes
- Kidney disease

- **Treatment and prevention**

Acute management:

Antibiotics as soon as blood cultures taken:

- 4-6 weeks, including at least 2 weeks IV initially.
- Empiric therapy and for streptococci: benzylpenicillin (or amoxicillin) +Gentamicin.
- Staph. aureus: flucloxacillin if native valve, add rifampicin and gentamicin if prosthetic valve

Surgery:

- Debridement, repair, or replacement required in 20%.
- Indications: refractory HF, persistent sepsis or emboli, or fungal IE.

Prevention:

Prevention Prophylactic antibiotics for high-risk dental operations in individuals at risk of infection (e.g., prosthetic valves) are debatable: It is recommended by European and American rules, yet it is not considered good. (Mylonakis, E., & Calderwood, S. B. (2001))

B. The database management system:

The first major decision was to select a management system database for the medical data. This section begins with a comprehensive overview of the current state of the art before moving on to the chosen choice. This state of the art is an inventory of the approaches and software now in use in the field of massive database management.

A database is a vast collection of structured data that is as devoid of duplicates as feasible. These are connected

and stored on either centralised or distributed digital media. The same database can be accessed by multiple users and programmes working in parallel. A Database Management System (DBMS), software that is responsible for structuring, storing, updating, and managing a database, makes this feasible.

It is the link between computer scientists and data, allowing them to build schemas and write programmes. On the other hand, it lets people and data to communicate directly for the purposes of viewing and changing data.

DBMSs can be hierarchical, networked, object-oriented, and so on. The relational model (SGBDR), a mathematical model based on set theory, is the subject of particular interest here. Fields (columns) and entries are used to organise the data in tables (rows). The fields reflect distinct types of qualities that are distinguished by their names. A primary key is assigned to each entry and serves as a unique identifier. Other non-unique foreign keys represent the value of a connected table's primary key, allowing associations to be established across tables. MySQL, Oracle, PostgreSQL, Ingres, SQLite, and Access are among the most well-known database management systems. MySQL and PostgreSQL are both open-source databases that are free to use. MySQL is also secure.

Furthermore, MySQL is easier to use and more suitable

to small databases than PostgreSQL, which is oriented towards database management

massive data (containing millions of billions of entries).

The SQL (Structured Query Language) programming language is used by most RDBMSs to allow users to interface with the system.

management of the data base It is a declarative language that allows you to express yourself in a variety of ways.

queries, to quickly code the required result without having to explain how to get it.

These searches are made up of keywords that enable for easy manipulation of data.

DELETE, INSERT, SELECT, and UPDATE are data operations. Other SQL procedures, such as GRANT and REVOKE, allow you to adjust table access privileges, whereas CREATE, ALTER and DROP allow you to build database structures.

The installation of a database management system relational data is required to administer a database. There are two sorts of RDBMS physical implementations: some, like SQLite and Retrieve, employ a file sharing service to access the data, while others, like MySQL and Oracle, are installed on a data server that acts as a centralised application5. This implies that SQLite and Read are simple to use in practise: the database is represented by a Sqlite or.mdb file (SQLite and Access, respectively), and the SQLite or Microsoft Access programme is used to access and alter the data. (Wiederhold, G. (1983). *Database design* (Vol. 1077))

Primarily, the database management systems SQLite and Access have been explicitly excluded because their implementation in files makes information interchange

difficult. Other software, such as Oracle, to name just a few well-known examples, is chargeable and hence cannot be used outside of the ULB, which makes it available to students. Their use in a medical context regarding deep research in such a critical field would result in legal ramifications, and no hospital can afford to face a licencing issue. As a result, this software has also been removed. The options were then limited to MySQL or PostgreSQL, and CSV files both of which were open source and free. MySQL, on the other hand, is more popular and easier to use for the same characteristics, as indicated in the preceding section. Consequently, the use of CSV file is more suitable for this task as data are updated daily and changeable, also the data is independent, huge and easier to manage over a csv file than database as every new record comes in form of csv files In order to avoid losing current data on cervical cancer patients and hence continue to monitor these patients.(Effendi, Y. A., & Retrialisca, F. (2021, June)

C. The Programming language:

The number of programming languages available is enormous. It was important to do a market inventory before deciding which to employ throughout the project. The following section follows this one.

Python, C++, Java, Perl, Ruby, PHP, and other programming languages can be used to create application graphs that access a database. Among these alternatives, those related to the languages taught during the bachelor's degree course in engineering, as well as PHP, were evaluated. This section goes over these options briefly.

Python is an intuitive language that is simple to learn and use, thanks to its imperative, functional programming style and memory management system. However, unlike several other languages, it is an interpreted language rather than a compiled one. This means that the programme processes and analyses data at every execution rather than simply once at compile time. Because the syntax, the names of the variables, the names of the functions, and the types are all checked at runtime, the biggest disadvantage of Python is that we lose a lot of time starting the programme and only seeing the problems at that point.

A compiled language has the advantage of discovering all flaws in one pass during compilation, and none can elude the programmer because programme execution cannot take place if there are any remaining problems. Both C++ and Java, unlike Python, are compiled languages.

Their hold, though, is stronger. The subtleties and specificities of C++ make it exceedingly challenging to work with in the construction of interfaces when compared to other languages. Java, on the other hand, is easier to master than C++, the language from which it is derived. It is also popular in the worlds of databases and graphical software.

The PHP (Hypertext Pre-processor) programming language is both imperative and object-oriented. It is, however, a programming language responsible for developing dynamic web pages using a http server, unlike Python. PHP, like Python, can run locally, but it also has the benefit of being able to run in online applications.

The application was ultimately coded using the Python programming language. This has the advantage of being relatively easy to learn and has, moreover, been taught during
of the course within the university of the west of England. Java may have also been used, as it is an ideal compromise between the advantages and disadvantages of Python and C++. However, it was covered in more depth during the course. Although the online solution is valid and as simple to learn as Python, none of the members of the group were familiar with the PHP language or web technology in general. This word encompasses a wide range of concepts that are not related to PHP, such as web pages, HTML, CSS, JavaScript, sessions, cookies, and so on. There was so much fundamental knowledge that it would have taken months to learn.

In a project of this size, using a language that had already been studied saved a lot of time. The introduction of a new programming language has advocated knowledge of database concepts and database management systems. As a result, Python version 3.10 was used. (Grayson, J. E. (two thousand))

D. Graphical Interfaces:

The windows interface establishes a connection between the user and the database. The latter must be intuitive for medical personnel to learn more quickly. Indeed, well-made windows can save a significant amount of time in learning how to use the system.

The interfaces, as previously said, ensure human-machine communication. The purpose of a graphical

interface is to provide easy-to-understand information to the user.
understand, but also allowing him to change it using multiple text fields, lists, buttons, and menus. The tools needed to set up these interfaces are provided by GUIs (Graphical User Interfaces). There are many, but Tkinter is one of the most well-known and simple to utilise with a Python programme.
Although the Qt software allows for the building of complete and tidy interfaces and has the benefit of being portable and providing a native look, Tkinter is much easier to use and has already been researched, and its interface adjusts quickly. This means that on Windows, the interface windows will adapt to a "Windows" design, whereas on Mac, the graphics will be Mac-specific, and so on. Tkinter also comes with a large and easy-to-use widget library. These various arguments have made it a powerful opponent when contrasted to other GUIs that are both more efficient and more sophisticated.
Other GUIs, on the other hand, are Python-compatible. GTK+ is one of the most well-known, yet it is less feature-rich than Qt and more difficult to master. Although it is compatible with a variety of operating platforms, it is only very unsound on both Mac and Windows. Tkinter, on the other hand, is exclusively compatible with Python and comes with it.
The chosen GUI must be able to work with the Python programming language, be portable across Windows, Mac OS X, and Linux platforms, and have a good set of libraries. This is the case with Qt, GTK+, and Tkinter, therefore a decision was required.

The goal was to create simple and intuitive interfaces that would make educating medical personnel easier. Tkinter is the programme that best matches our requirements since it is simple to use in Python, has a wide variety of widgets, and is portable across all major operating systems. The Tkinter software allowed for the creation of similar windows. The goal of this application is to manage the design of the windows in a purely graphical manner before converting the Gui file and obtaining the Python code for this interface. In comparison to traditional programming, this technique provides for a simple and quick stylistic approach.

The information maintained by the system produced as part of this project must be always accessible and in a variety of locations, and it must be secure and built to assist medical researchers in gaining a better understanding of the various IE sector cases. (Moore, A. D. (2018))

4. Requirements:

The goal of the project is to assist auditors and researchers by making data easier to use to treat a deadly and rare condition (Infectious Endocarditis), which is responsible for a number of deaths. The goal is to create a database that allows the auditor to enter data both individually and in bulk, make notes, and add new files to the system.

The software is built using python as main programming languages, Tkinter as Gui for the front-end to interact with the user and csv files provided by the Hospital as a data holder.

The basic goal is to provide a clear and simple display of all the cases' bulk data, which is saved in five distinct files, as well as the ability to update and keep fresh records of changes over time.
The following table shows the functional and non-functional features planned for the project:

Functional	Non-Functional
• Must Insert button for cases	• Should have Login page
• Could Display report	• Won’t Generate a PDF report
• Must Delete button for each data	• Should Date and time
• Must Display data in Treeview table	• Should Organize the five data tables in notebook
• Should Extract data from tables	• Each Patient won’t has self-data page
• Could Use csv file as data holder	• Could Import csv file button
• Use Pandas to enable managing data	• Won't have a pdf downloader.
• Must include import csv file in order to include new files to the system	• Won’t use a database

This project is a novel concept in the field of information technology; most other projects are focused on hospital management systems. However, this is a stand-alone and unique project that allows the user to manage data as needed and import new files while maintaining complete data protection. Because the software is accessible at any time and from any location, and because the data is saved as csv files, it is unlikely to be lost.

Furthermore, the program is available to all IE researchers who have enrolled in the database. The user must first log in with their credentials before being able to view the main window, where he will see a notebook with records and data from five separate files:

- GBS AndIEDiagnosisforUWE.csv
- NBT IE data for UWE.csv
- SA AndIEDiagnosis for UWE.csv
- StrepAndIEDiagnosisfor UWE.xlsx
- UHB IE data for UWE.csv

The software depends on importing python internal libraries:

- Tkinter: it is used for the Gui
- Csv: to import the data
- Asyncore: to display the data

An external framework Pandas is used to have efficiency to manage data stored in files.

Pandas manages missing data easily and provides an efficient and flexible way to slice, merge, concatenate and reshape the data.

One of the most significant benefits of medical software is the confidentiality of sensitive data. As a result, a login

page was built for unique medical staff and flat files were generated to save records and minimise data loss.

Obviously, the project given can be improved over time and with more knowledge, but it is still strong, safe, and useful for medical data processing. (Amaechi James, C., Agbasonu Valerian, C., & Nwawudu Sixtus, E. (2018).)

5. Project Planning and Team Role:

Before a project can begin, the development team must first determine what they need to achieve and how they will do it. On a task-by-task basis, we should examine the resources needed and the time it will take to complete the work. The following are the primary actions in the Project Planning phase:

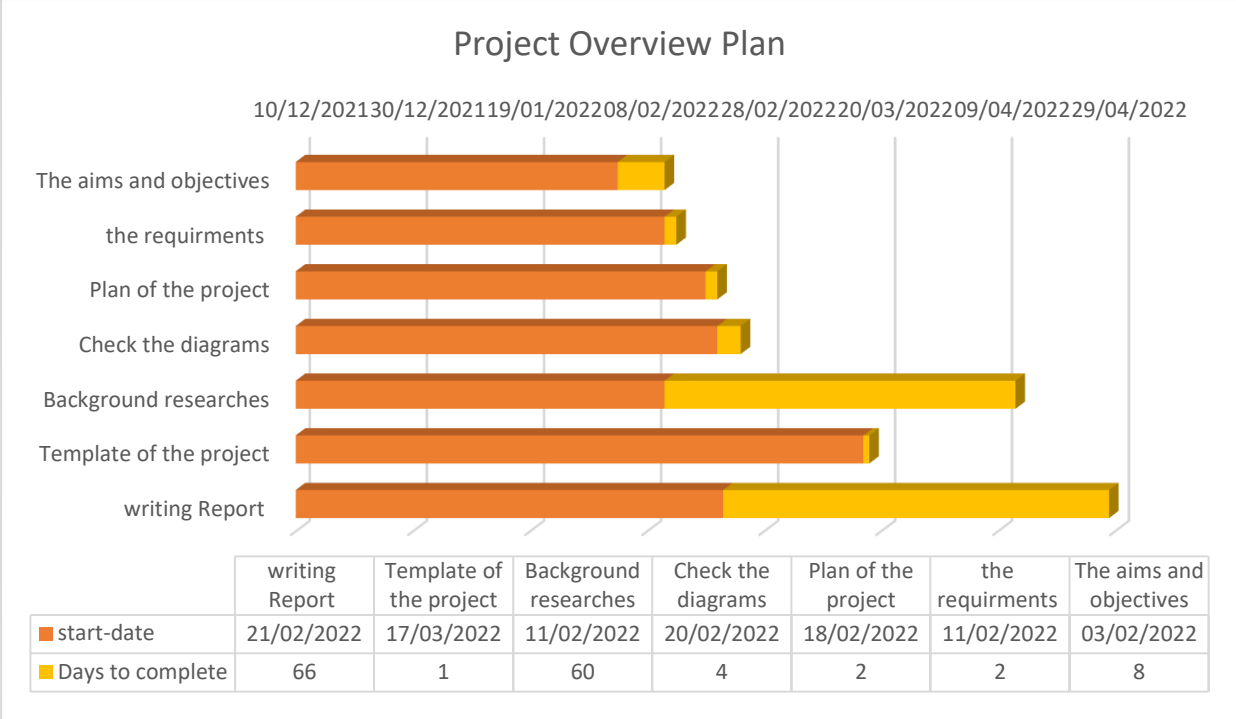
• Identifying available project resources

• Providing a summary of the tasks necessary to accomplish the project

• Identifying potential project hazards.

• to develop a realistic timeline for completing those activities in the most efficient way possible.

To keep the code clean and the process in check, software projects should always be run inside a certain framework of standards and procedures established by the team leader. A well-organized and structured setup eliminates turmoil and allows new team members to be onboarded swiftly and smoothly. And, of course, it improves the end product's quality.



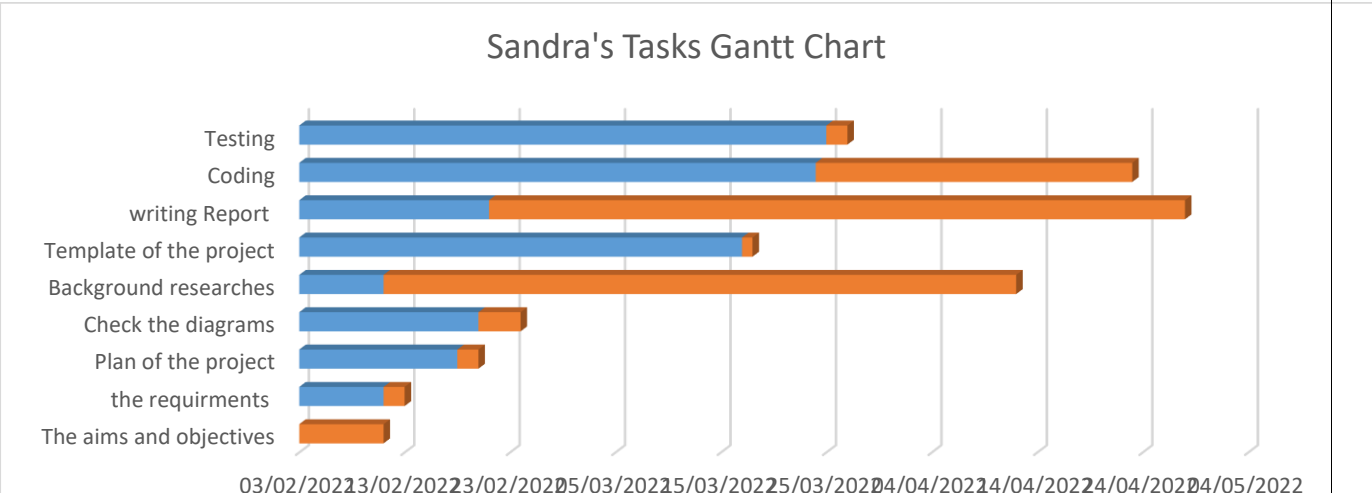
	writing Report	Template of the project	Background researches	Check the diagrams	Plan of the project	the requirments	The aims and objectives
start-date	21/02/2022	17/03/2022	11/02/2022	20/02/2022	18/02/2022	11/02/2022	03/02/2022
Days to complete	66	1	60	4	2	2	8

Figure 2: Gantt Chart

Ms Sandra Sahnoune was nominated as Team Leader, on a technical level, she has the responsibility to decide the direction for development. Giving instructions — and doing so well – is a significant duty. When necessary, instructions should be unambiguous and delivered in a respectful tone, followed by a constructive explanation. A project she must also ensure that the project's business context, as well as the client's requirements, are recognised and understood by the entire team, and that these understandings are then converted into individual expectations for each team member.

She takes sure to communicate the task requirements and divide the work among the team members, as well as maintain a synchronisation effort by incorporating multiple ideas.

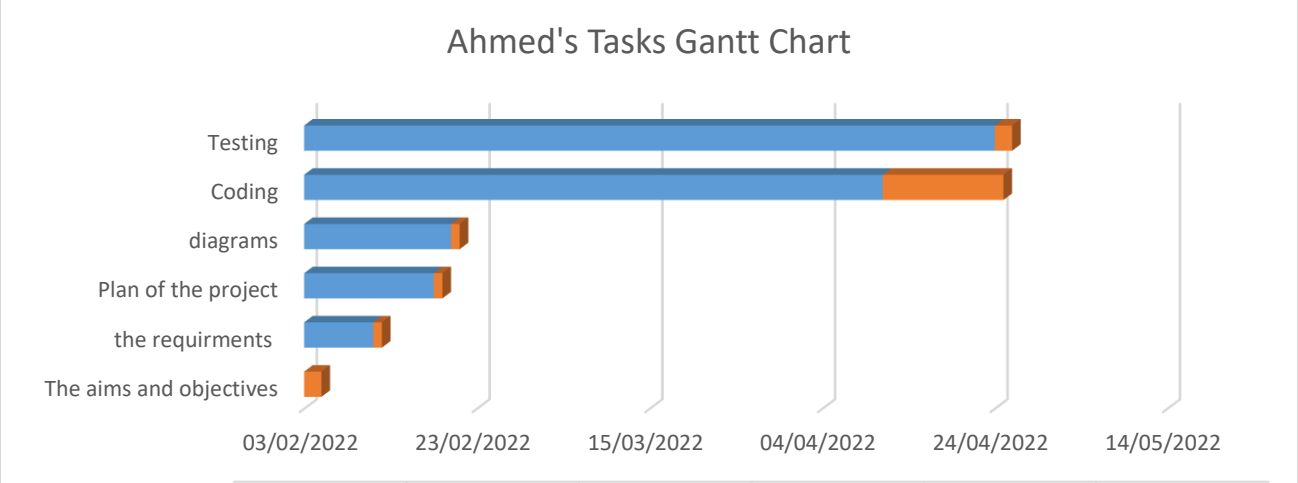
In addition, she contributed to Start Coding the Interface of the software as well as authoring the report along the process.



	The aims and objectives	the requirments	Plan of the project	Check the diagrams	Background researches	Template of the project	writing Report	Coding	Testing
start-date	03/02/2022	11/02/2022	18/02/2022	20/02/2022	11/02/2022	17/03/2022	21/02/2022	24/03/2022	25/03/2022
Days to complete	8	2	2	4	60	1	66	30	2

Figure 3

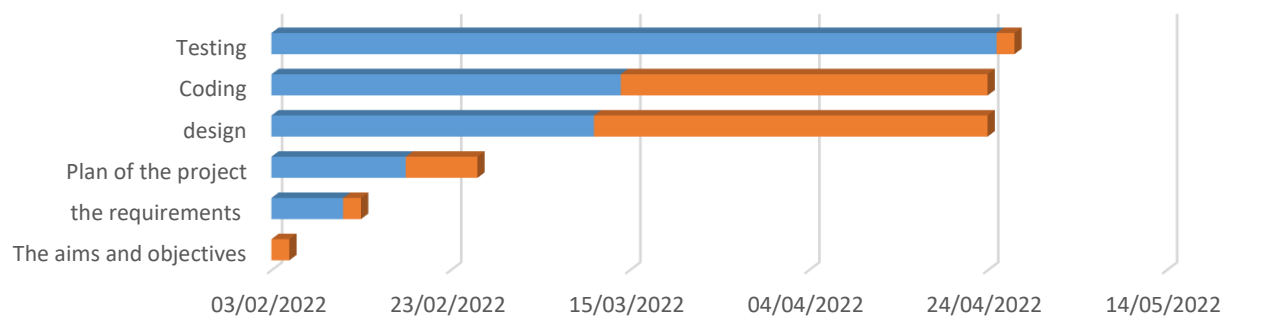
Mr Ahmad Bayomi is responsible of connecting the database to the front-end structure, importing Panda's library, and displaying the data of the csv files.



	The aims and objectives	the requirments	Plan of the project	diagrams	Coding	Testing
start-date	03/02/2022	11/02/2022	18/02/2022	20/02/2022	11/04/2022	24/04/2022
Days to complete	2	1	1	1	14	2

Figure 4

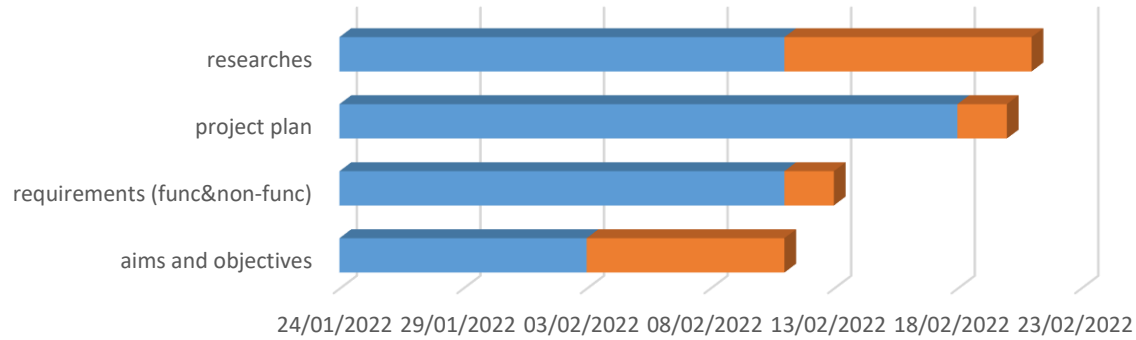
Ms Aisha Alsayed is responsible of the design; she designed a first wireframe template which was used as a primary idea for the final design. She linked the pages and gave dynamism to the options.



	The aims and objectives	the requirements	Plan of the project	design	Coding	Testing
start-date	03/02/2022	11/02/2022	18/02/2022	11/03/2022	14/03/2022	25/04/2022
Days to complete	2	2	8	44	41	2

Figure 5

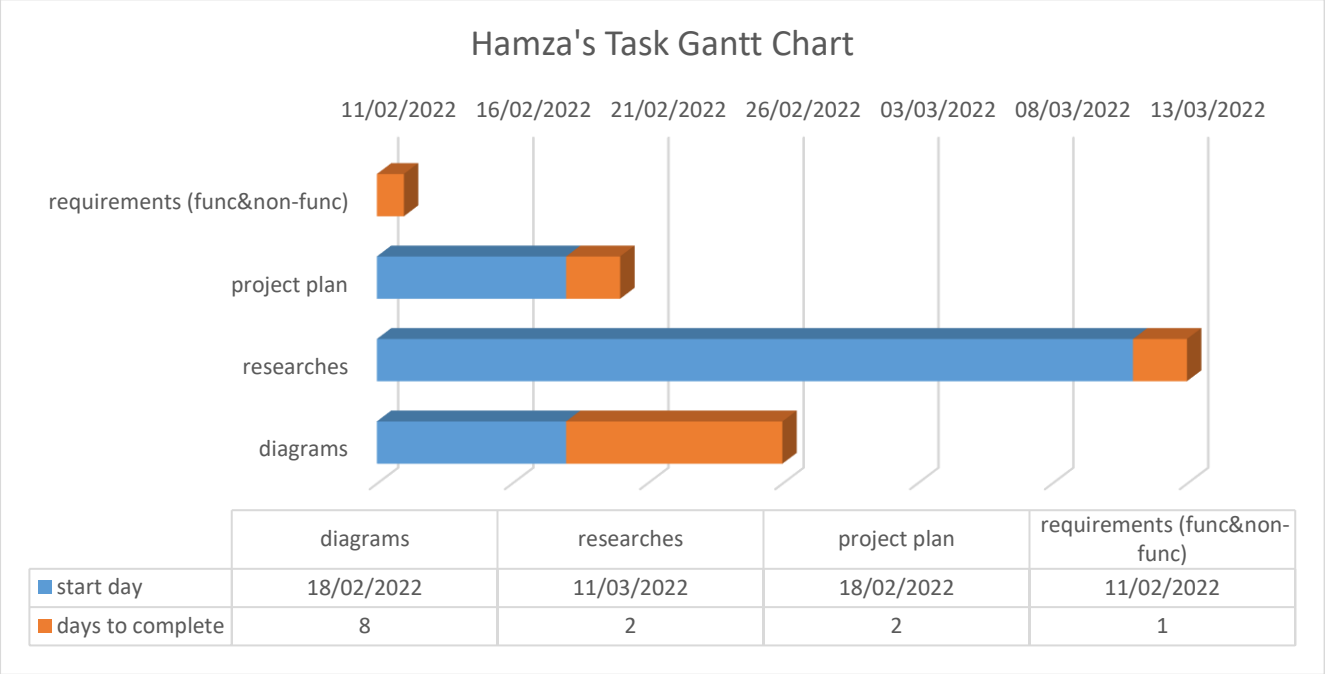
Mr Ala Eddine Ballalou has the duty to keep track of all the records and updates as well as he contributed on some research for the report.



	aims and objectives	requirements (func&non-func)	project plan	researches
start day	03/02/2022	11/02/2022	18/02/2022	11/02/2022
days to complete	8	2	2	10

Figure 6

Mr Hamza Addad designed the UML for the software. He has the duty to design the Class diagram and model-view-controller as well as proposing ideas of concepts.



	diagrams	researches	project plan	requirements (func&non-func)
start day	18/02/2022	11/03/2022	18/02/2022	11/02/2022
days to complete	8	2	2	1

Figure 7

The following table define the role of each member regarding the project:

Team members	Roles
Sandra Sahnoune	Team leader + Developer: responsibilities: - divide the tasks for each member – keep records of efforts done – organize meetings – code the structure of the project (login and window page) -author the report –check every progression made –

	synchronize the work done – explain the project plan and how it should be implemented.
Aisha Alsayed	Developer Responsibilities: design the front-end and wireframe as well as linking the pages.
Ala Eddine Ballalou	Developer Responsibilities: keep records of all the updates and author the report
Ahmed Bayomi	Developer Responsibilities: Displaying the data and linking the csv files
Hamza Addad	Developer Responsibilities: Designing the Class and Sequence diagrams

- **The Challenges and risks:**

The key difficulty on this project was to make medical data as clear and efficient as possible to make it easier for medical personnel to read and analyse it.

On the other hand, a feature was to be implemented that would allow users to efficiently read data from csv and xls files for long-term usage.

In terms of strategic risks, it will be a matter of ensuring that the project does not jeopardise the company's overall operations, such as requiring an excessive budget or human resources. It will also be a matter of determining whether rival ventures in the same field exist, the project's idea was minimalistic in this sense, we tried to use cheap and efficient features to not exceed the imposed budget.

The second risk was the lack of experience and availability of some members of the team, the solution was to give each member the task that is suitable for them.

The last risk was a technical one; the library and framework chosen were critical; the notion of constructing a desktop application utilising Tkinter and pandas to read csv files, especially if the data file grows over time, will slow down the system thus creating separate classes for each function may facilitate the process.(Koolmanojwong, S., & Boehm, B. (2013, May))

The following Matrix shows Risk Register:

RISK DESCRIPTION	**IMPACT DESCRIPTION**	**IMPACT**	**PROBABILITY LEVEL**	**PRIORITY LEVEL**	**MITIGATION NOTES**	**OWNER**
		LEVEL				
Give a summary of the risk.	What will happen if the risk is not mitigated or eliminated?	Rate	Rate	(IMPACT X PROBABILITY)	What can be done to lower or eliminate the impact or probability?	Who is responsible?
		1 (LOW) to	1 (LOW) to	Address the highest first.		
		5 (HIGH)	5 (HIGH)			
The project data set was not released on time	Delay on on project plan	5	4	20	the responsibilities have been shared to the module leader and the tutor	The project releaser
Bulk and individual	Cannot provide a	4	3	12	Create a large Tree	Back-end Team

data hard to understand	Clear and useful software				view interface to display data	
Data store in multiple CSV files hard to manage	Time consuming for the doctors to find a solution	5	4	20	Create a Notebook to store the CSV files	Front-end Team
exceed the actual Budget	Jeoparadise the company's overall operations	4	2	8	Use cheap and efficient features, only one internal library is used for the front end: Tkinter, and one external Pandas for the back end	Team Leader

See figure 9 for table keys.

6. Design and Test Plan:

The database was created in such a manner that it is, on the one hand, the most succinct in terms of code and, on the other hand, as adaptable and modular as feasible in terms of the project's long-term viability logic. It is certainly conceivable for a laboratory's list of exams to be updated in response to scientific and technical advancements. He also believes that, overall, new laboratories will be added to the list dedicated to them, that the names of some laboratories will change, and

that others will close. Similarly, the patient list is meant to alter on a regular basis when new ladies arrive.

Finally, changes to the list of doctors occur when a new user is added to the database data or when a given user is granted new rights. After some thought, it was discovered that the operations to be performed on these several lists (class information (), class buttons1(), class frames ()) were the same; It is a matter of adding, altering, or removing an entry from a list each time. The idea arose to design the most generic framework feasible that would allow the same operations to be performed on these four types of lists.

The Scenario:

The user can log in using the authentication details that have already been saved in the database. Once logged in, the user can see all the data and records that have been displayed, update and delete data, and import a new csv file into the database. He can also write and save notes as he goes through the analysis. Using the Tree view function, he may access several data sets.

The data is structured as follows:

- Two actors:
- Multiple use cases shown as follow:

Use Case diagram:

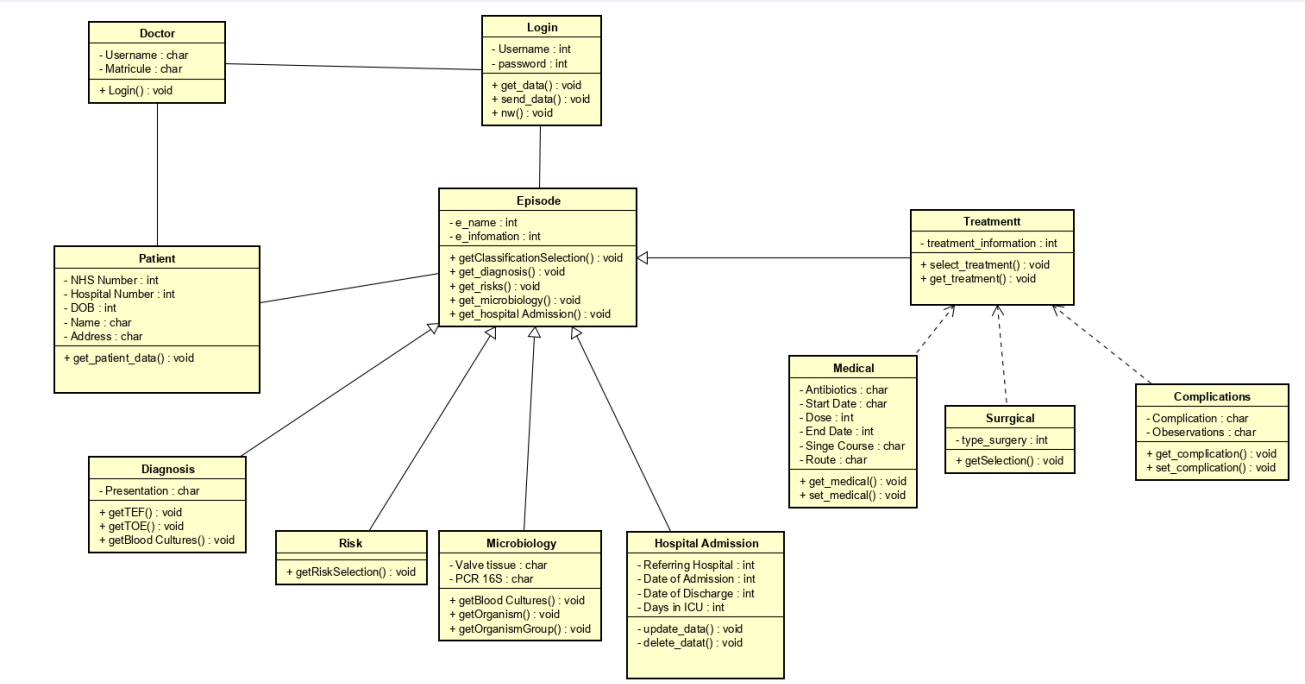
Furthermore, two different classes were implemented:

- Patient class have: NHS Number, Hospital Number, DoB, Name, Date of death, Addresse
- IE class: Classification/Nature (Native valve, Prosthetic Valve, CDI, Root abscess, Valves effected A, M, P, T), Hospital Admission, Diagnosis, Risk Factors, Microbiology, Treatment

  Each IE methods have sub-classes:
- Hospital_admission ()
- Diagnosis ()
- Risk factors ()
- Treatments ()

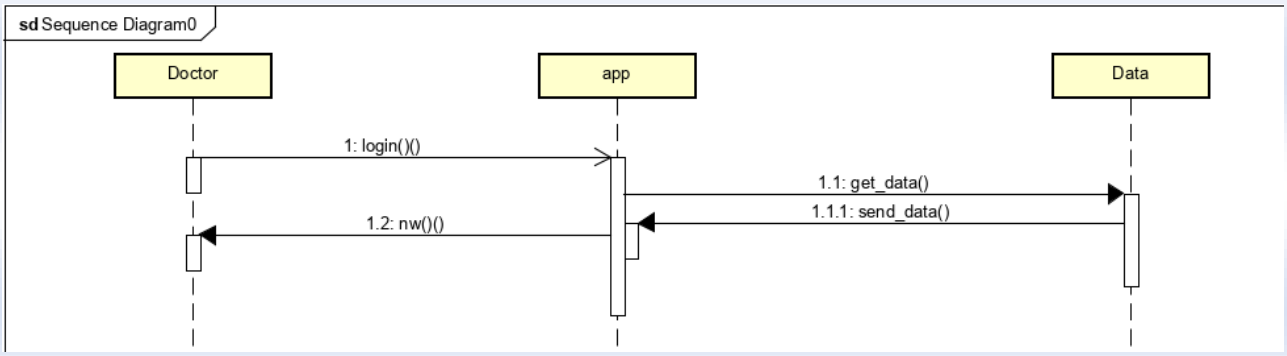
  The following diagram represents a class diagram, it shows 12 Classes linked to each other

  Class Diagrams:



A model view was implemented to explain how the process was implemented; the general structure described previously entails some redundancy in the file creation process. nw is the name of the main window. When the programme is launched, the user is presented with a login screen where he must enter his authentication information (username and password) to access the data.

The view establishes a connection between the user and the database. This last one's job is to show the data collected from the model on the one hand, and to take orders from the user on the other.



Sequence diagram for login scene

In addition, a window appears with three sections: a notebook with six chapters displayed in a Tree view table, an option section where the user can manage the data flexibly using the update, delete, and import csv files buttons, and a notes or information section where the user can write, take, and save additional notes about the records and multiple cases.

7. Implementation and Testing:

The database was created with the goal of being as user-friendly as possible, and the interfaces were created with this in mind. The windows have been streamlined to focus on the fundamentals, and their design is redundant for clarity.

Figure 7.1 shows the identification window after the Code has been run.

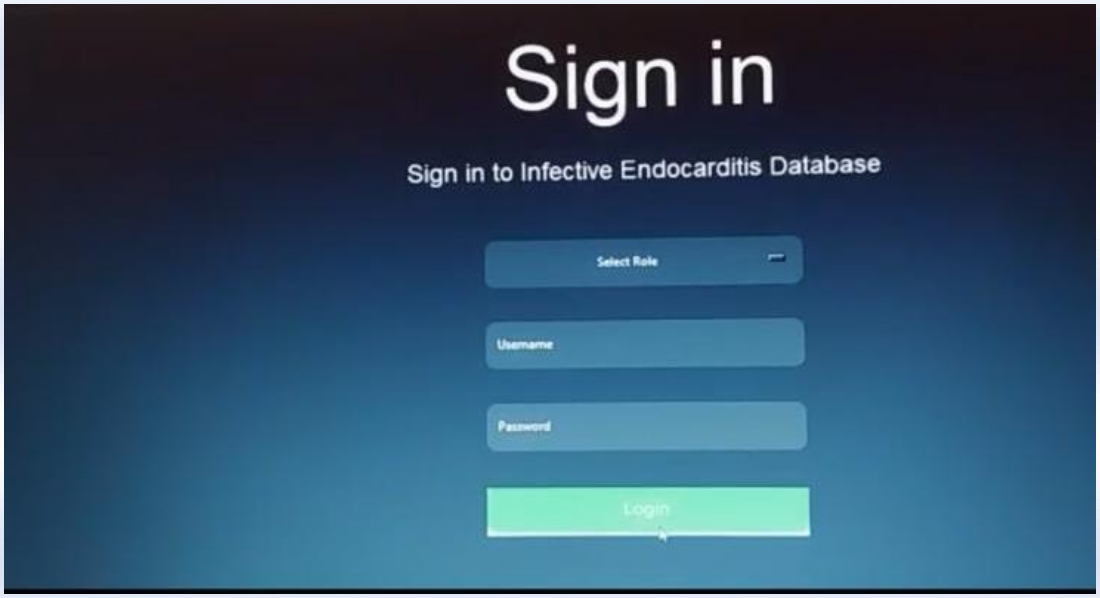


Figure 7.1

To connect to the database of patients and cases, enter a username and password in this box and click the "Ok" button. The user may then view patient data or add information about them in the main menu window 7.2. It is vital to remember that a user's identification is nothing more than his ID and password, which are originally sent by the database administrator a verification message box appears.

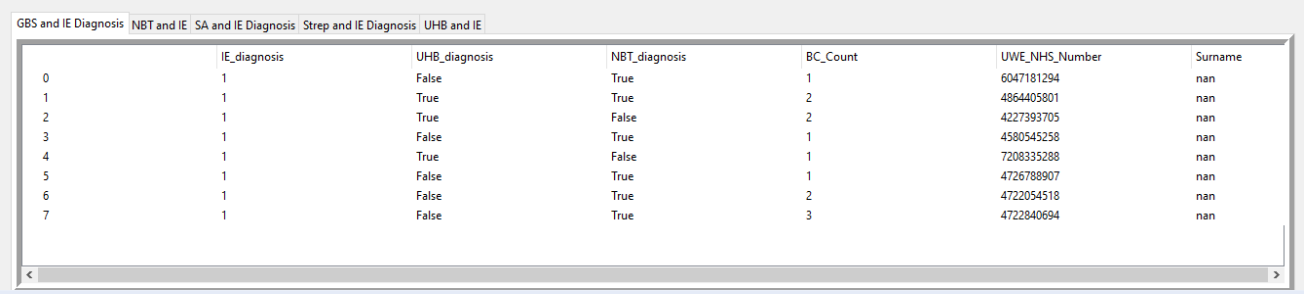
Figure 7.2

If a user mistakenly creates a new password by dragging an error, the user will be unable to login to data in the future. Therefore, there is a verification system in place. The user must encrypt his password twice, and the change will only be made if the two fields are equivalent.

The primary window, for starters, is divided into three sections:

The notebook is separated into five chapters, each of which displays data from an individual IE analysis as well as information from many cases.

It displays the information in a Tree View table, which is a simple and effective technique, figure7.3


GBS and IE Diagnosis | NBT and IE | SA and IE Diagnosis | Strep and IE Diagnosis | UHB and IE

	IE_diagnosis	UHB_diagnosis	NBT_diagnosis	BC_Count	UWE_NHS_Number	Surname
0	1	False	True	1	6047181294	nan
1	1	True	True	2	4864405801	nan
2	1	True	False	2	4227393705	nan
3	1	False	True	1	4580545258	nan
4	1	True	False	1	7208335288	nan
5	1	False	True	1	4726788907	nan
6	1	False	True	2	4722054518	nan
7	1	False	True	3	4722840694	nan

Figure 7.3

Administrators can change this list of users using the three buttons at the bottom of the tab.

The first ("Delete") allows you to remove a user from the list that you have already selected. The "add" button is used to create a new user, but it is also possible to change the data or rights of an existing user. To do so, just choose the line that corresponds to this individual before pressing the "update" button.

The option section provides access to three alternative data management options; the update button allows the user to

update new records in the event of modifications or add a new one to the end of the list in the event of a new case.

Any recorded data may be deleted and removed from the table using the delete button.

The import csv files button allows doctors to import existing recorder data contained in csv files into the database, resulting in the creation of a new widget that shows the data and allows for flexible management.

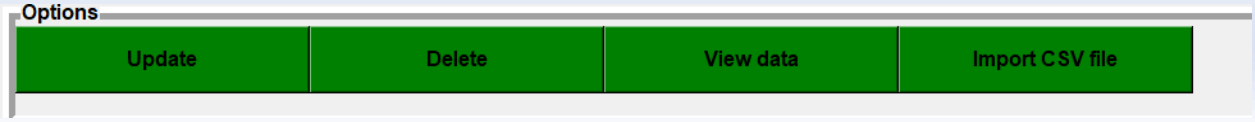


Figure 7.4

This feature is extremely beneficial since it allows physicians and researchers to save time and analyse data fast throughout time to having many files that must be analysed separately. Because Internet Explorer is updated and involved, it needs medical attention to diagnose the problem.

Note that the imported files must be in csv format and will only be used to populate the database and notebook.

The information section allows the user to read and add notes or remark on any instances or records that have been completed throughout the analysis process; these data may be saved in the database by clicking the save button.

Figure 7.5

Many obstacles were encountered throughout the creation of the programme, for example, the implementation of the buttons

was difficult and time-consuming owing to the lack of understanding and involvement of some individuals.

The programme reads and displays data as expected; however, the outward design should have been done more professionally.

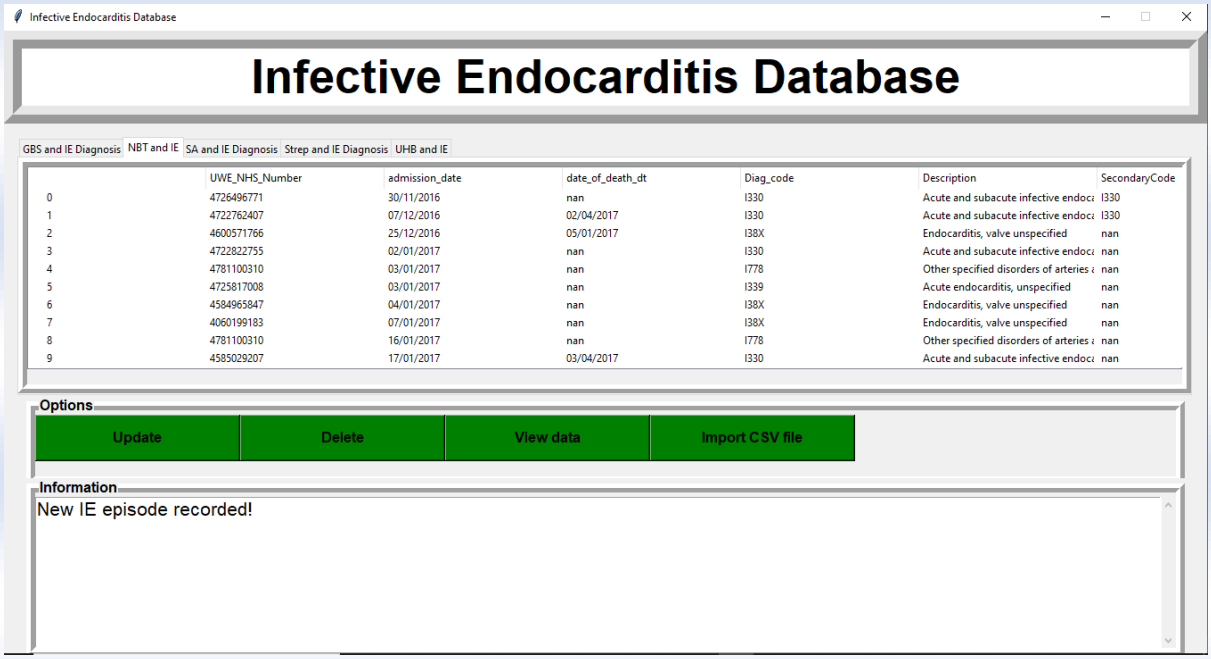


Figure 7.6s

Although the general idea is satisfactory, the database implementation might have been more transparent and efficient.

Despite not having time to design a button generator for displaying or downloading reports, not showing date and time, and creating an individual area for each patient's information, most of the requirements were successfully completed.

8. Evaluation and Lesson Learned:

The group's biggest problem was a lack of input and involvement in the project; nonetheless, Ms Sandra and Ms Aisha worked hard to keep the project moving forward and finish on time; motivation was the first criterion for starting the project and encouraging other members to do so. The second obstacle was the members' insufficient expertise; such a large project for a hospital needed a broad understanding of programming and object-oriented and software design, which was difficult for some members.

When reviewing the project, there were several notable achievements, such as the implementation of multiple features. Since we started with the goal of including only one library which is Tkinter as Gui to be processed, it was a huge accomplishment to create a notebook frame to display the data in a Tree View Table. In addition, as we advanced, we discovered how to add new framework Pandas to the application, allowing for more useful features such as reading the data and display it efficiently.

We the idea to create multiple option to the system was to enable the user to manage the data as needed in a simple and flexible way.

To begin, we create a system sequence diagram for the module in question, based on the use case and Class diagram based on the Project topic specification, to identify all the functionality. We completed this work in a group to obtain all the group members' perspectives in order to produce the best possible analysis.

However, after receiving the data set and the 5 csv and xls files, the plan changed, and the purpose was to find an efficient way to display the bulk and individual data as clear as possible.

Then, when the UML was built, we described the various graphic aspects of the module so that a member could begin building the module's interface using Tkinter as GUI. However, we also specify the various technical parts by identifying the many processes (functionalities) that must be completed for the module to work properly.

We build the various aspects of the module in one or two, independently of each other, depending on the modules, their complexity, and the diverse talents of each member.

Subsequently, we check that the product meets expectations, that the module corresponds

to the planned scenario, from a graphical point of view (effect visible to the user) and from a technical point of view

by the results of the database (which the user does not see)

Our working method is built on the separation of development and integration, which allows us to concentrate on development rather than integration and so be more productive.

Future work can be completed, and the system can be improved in instance having a cleaner and professional interface corresponding hospital demands, and a limited login credentials for more safety, in addition a graph generator to produce stats of the cases to help researchers to find solutions quickly. However, these various features are costly and take longer time to be achieved.

To keep the project moving forward, the idea was to divide the work among the team members who have strong understanding of them with the help of the team leader if needed (See Figure 11 for divided tasks).

Despite the difficulties encountered during the project, it was completed and released on time. The group overcame these obstacles by setting a deadline and a target for the next week and inspiring members to complete a good task.

We as a group worked cohesively throughout the project, which helped us, even with a lack of contribution and late task achievement however we were able to communicate and help and motivate each other's to achieve a phenomenally successful completed project. There were few problems during development, but one that was a major issue for us was working with uwe GitLab; some of our group members had no prior experience with uwe GitLab, so we spent some time learning how it works over practical. Also, during development, there was a problem where two people were working on the same branch and one of them submitted an update, causing the other to lose sync with the remote branch.

This caused some progress to be delayed, but it was readily fixed with no serious issues. In the end, Uwe Gitlab was an especially useful tool that greatly aided development because of the ease with which other users can contribute. Users can work on various parts of the project without affecting the other modules that other members are working on, and the parts can be merged to create a working system.

Working on a medical system for the IE branch was challenging for all members because the data was large and confusing, and it took a long time for everyone to understand it. However, doing research and putting ourselves in the shoes of a user was beneficial and helpful in building a simple and clear system that anyone without IT knowledge could use.

Every week for the past three months, there has been a new job with a new seven-day deadline and a Thursday meeting. This programme played a critical role in the project's planning.

Teamwork was a valuable and enlightening learning experience for all members of the team; we learned how to support and assist one another, and the team leader allocated tasks effectively and adhered to realistic timeframes. We also learnt how to communicate properly and exchange ideas, such as the csv file import button.

The most essential element was to offer credit where credit was due. For example, we always let members know that the first tests and templates they created were well done, and we explained how things might be improved and updated in a friendly manner. (Koolmanojwong, S., & Boehm, B. (2013, May))

Obviously, online and book research were beneficial to this project; samples were chosen from various project documents found on GitHub and the Uwe Library, allowing participants to identify good, efficient, and off-topic sources.

Looking back on the project, it was an enormous success; we had in overall a nice team spirit, which is crucial for the growth of a successful project. This contributed to our getting along well with one another, aiding one another with issues, and attending all planned meetings especially during Holidays period. The project's success is demonstrated by the completed project's achievement of most of its goals and objectives.

Figure 8 shows a pie chart of the working days contribution of each member regarding the project.

These statistics was calculated upon the working days of each member since the start of the project.

This project was our first time working on a similar industrial real-world situation, and the primary lessons learnt were time management, separate tasks, and synchronisation.

Understanding and expanding information for coding from many sources of study is also important. In addition, each team member tries to contribute to the three main aspects of software development: front-end, back-end, and database (see Figure 10).

We attempted to apply the same technique to the IE database project utilising experience of Agile Software projects since it is fantastic, but one must first determine whether the qualities of the project they are working on make agile the best match. Therefore, agile project software can be customised by combining elements of its procedures with traditional project management methodologies to develop a methodology that best suits a project.

Many more features would have been incorporated if time and experience had been available, but we are glad that we have gained experience with the creation of real-world hospital software.

9. References:

- Amaechi James, C., Agbasonu Valerian, C., & Nwawudu Sixtus, E. (2018). Design and Implementation of a Hospital Database Management System (HDMS) for Medical Doctors. *International Journal of Computer Theory and Engineering*, *10*(1).
- Mylonakis, E., & Calderwood, S. B. (2001). Infective endocarditis in adults. *New England Journal of Medicine*, *345*(18), 1318-1330.
- Effendi, Y. A., & Retrialisca, F. (2021, June). Transforming Timestamp of a CSV Database to an Event Log. In *2021 8th*

International Conference on Computer and Communication Engineering (ICCCE) (pp. 367-372). IEEE.

- Wiederhold, G. (1983). *Database design* (Vol. 1077). New York: McGraw-Hill.
- Grayson, J. E. (2000). *Python and Tkinter programming*. Manning Publications Co. Greenwich.
- Moore, A. D. (2018). *Python GUI Programming with Tkinter: Develop responsive and powerful GUI applications with Tkinter*. Packt Publishing Ltd.
- Koolmanojwong, S., & Boehm, B. (2013, May). A look at software engineering risks in a team project course. In *2013 26th International Conference on Software Engineering Education and Training (CSEE&T)* (pp. 21-30). IEEE.

10. Appendices:

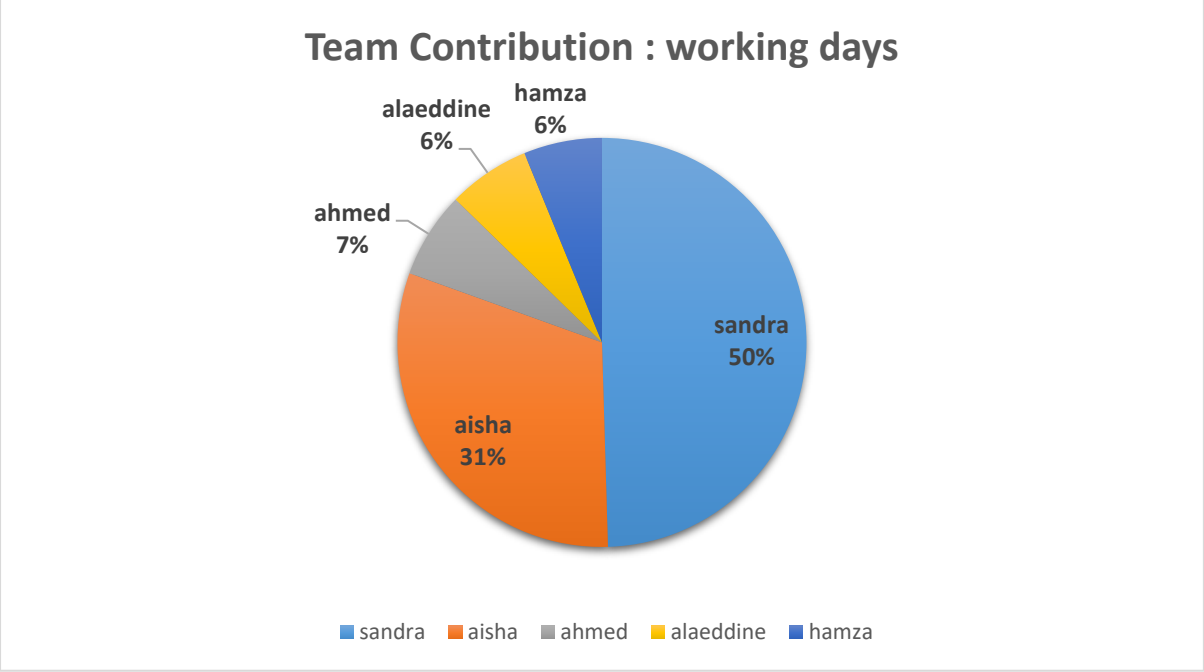


Figure 8

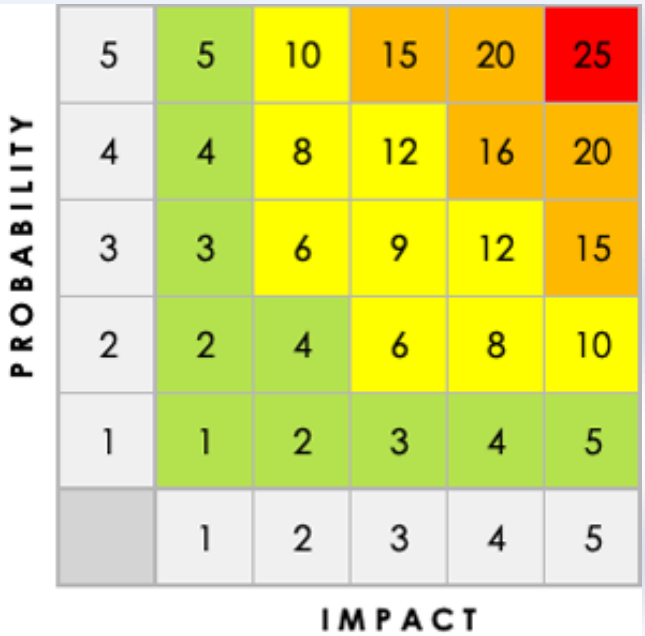


Figure 9

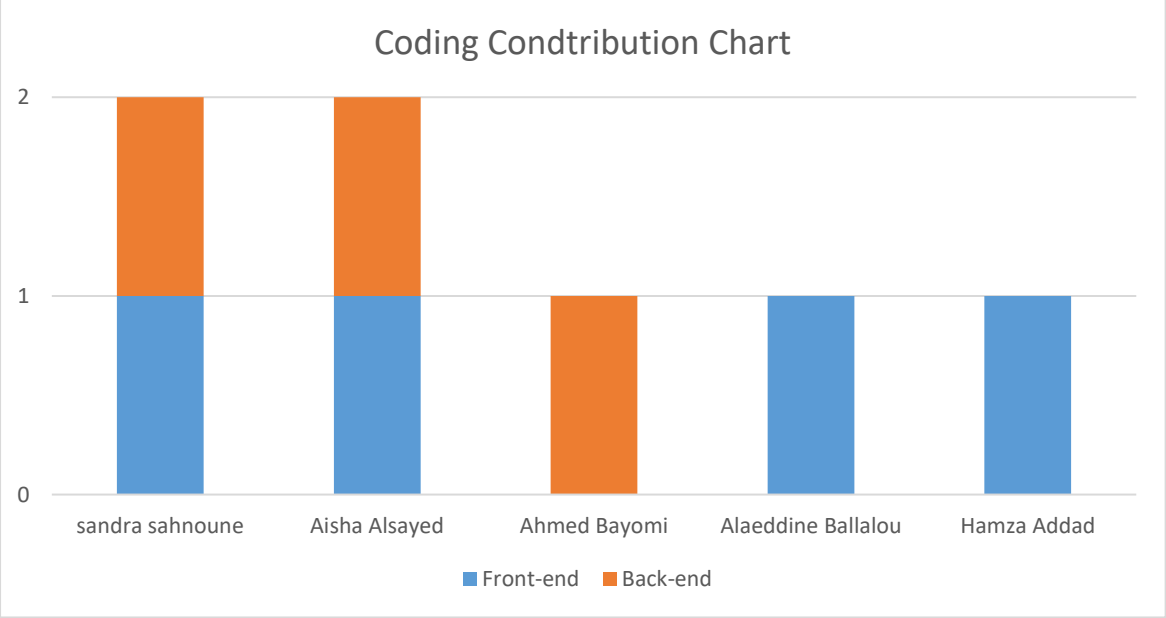


Figure 10

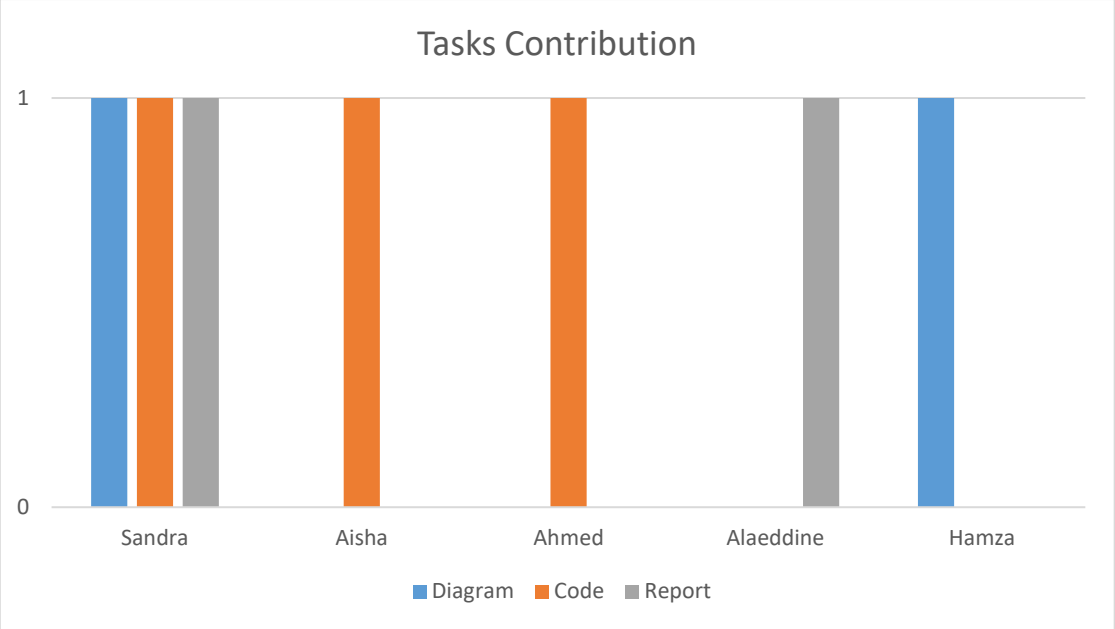

Tasks Contribution
1
0
Sandra
Aisha
Ahmed
Alaeddine
Hamza
Diagram
Code
Report

Figure 11