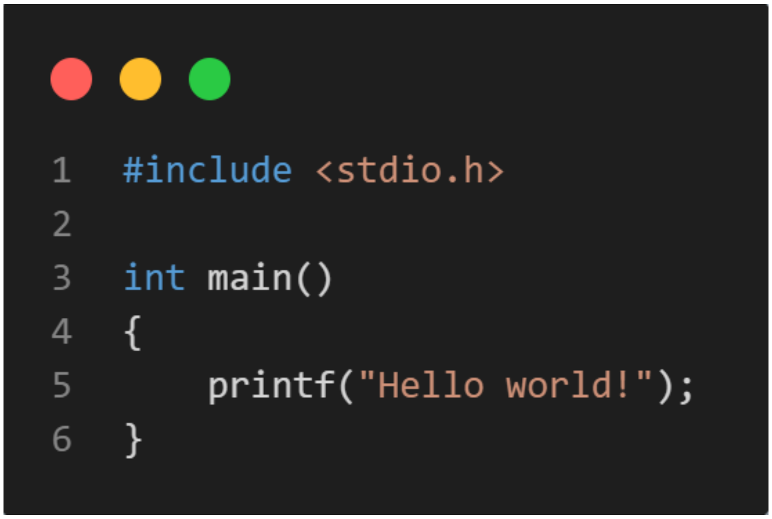
Software Engineering Module 1 – Overview of IT Assignment

1. What is a Program?

Lab Exercise:



```
#include <stdio.h>

int main()
{
    printf("Hello world!");
}
```

Theory Exercise: A program is list of instructions to get a desired output

2. What is programming?

Theory Exercise: Steps involved in programming are
Planning -> Analysis -> Design -> Implementation -> Testing -> Maintenance
And this process is called software development life cycle (SDLC)

3. Types of programming languages

Theory Exercise: The main difference between High-level and low-level programming languages is that in low-level programming languages the code is closer to machine code and is not human-readable whereas in high-level programming languages the code is close to human language and can be read and understood by a human.

4. World Wide Web and How Internet Works

Lab Exercise:

Step 1 -> The client requests information.
Step 2 -> TCP/IP protocol creates a connection between the client and the server
Step 3 -> The server responds and the data is broken into packets.
Step 4 -> The packets reach at the destination and are reassembled (the client has received information).

Theory Exercise: In web communication the client is the one who sends a request to the server to get some information and the server is the place where all the data of the particular program in stored

5. Network Layers on Client and Server

Lab Exercise:

Step 1 -> import a socket library
Step 2 -> create a TCP socket using the function from the library
Step 3 -> use a function to find connections
Step 4 -> accept one of the connections
Step 5 -> Read and process the request from the client
Step 6 -> Send a response
Step 7 -> close the connection

Theory Exercise: TCP/IP is a protocol which is used when a connection is to be made between two or more devices. Layers of TCP/IP protocol are: -

- Application layer:-
  Provide network services and help build the connection between software and applications
- Transport Layer: -
  Ensures the transfer of data between devices and manages the communication
- Internet Layer : -
  Looks after the sending of packets (the data that is broken into pieces)
- Network Layer: -
  Manages data transmission of the physical devices on the same network

6. Client and Servers

Theory Exercise: In a client-Servers communication the client initiates a request to the server and server responds with a message.

7. Type of Internet Connections

Lab Exercise:

Broadband – Is easily accessible (almost everyone has a broadband at home) and it also affordable to other options. But it has slower speed compared to other options and the strength of connection depends on your distance from the nearer network tower.

Satellite – It can be accessible from anywhere it doesn't have the limitation of distance. But it is dependent on weather conditions as the satellite is in space so clouds can obstruct its signals and it is very expense.

Theory Exercise: Broadband is a wireless way of getting internet access, and has slower internet speeds and higher latency compared to fiber-optic connections. Fiber-optic connection uses fiber-optic cables to transmit the data so it is faster and has lesser latency compared to broadband.

8. Protocols

Lab Exercise:

Steps to simulate HTTP requests using the command line with Curl: -
1 – GET request
2 – POST request with data
Steps to simulate FTP requests with Curl: -
1 – Download a file (GET)
2- Upload a file (PUT)

Theory Exercise: The HTTPS protocol in a secured connection whereas HTTP is not secure.

9. Application Security

Lab Exercise:

SQL Injection – If a login takes input and directly puts it into a query the attacker can enter Username as admin' – and anything in for the password.
The '—'is used make comments so using that in username will comment the rest of the query.

Solution: - Use parameterized queries

Cross-site Scripting- When a website displays user comments without sanitizations (it is a step to modify the user input in case the user provides a harmful input).The attacker can run an alert script .

Solution: - Escape the output to prevent scripts from executing.

Broken Authentication- If the website stores passwords in plain texts format, the attackers can access their database and can immediately see user passwords.

Solution: - Store passwords using hashing.

Theory Exercise: Encryption helps in converting the data in an unreadable format so while transferring the data no one other than the receiver can understand it.

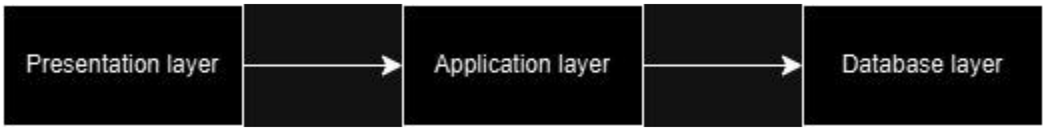
10. Software Applications and its Types

Lab Exercise: Identifying either the applications I daily use are System software or Application software
i) Instagram – Application software
ii) Calculator – System software
iii) Camera – System software
iv) Spotify – Application software
v) Calendar – System software

Theory Exercise: The difference between system software and Application software is that System software are software which help in the functioning of the device and coordination with hardware e.g. the Operating System, whereas application software are software which are used to perform a specific function e.g. MS word to type documents, MS powerpoint to make presentations, etc.

11. Software Architecture

Lab Exercise:



Theory Exercise: Modularity in software architecture helps to break complex problems into smaller and simpler problems which makes the development process more efficient.

12. Layers in Software Architecture

Lab Exercise: Let's take an example of online bookstore. Where the customers can browse books, add them to cart and make purchases.

1) Presentation Layer: - This layer is the user interface that interacts with the customer. It provides the components like webpages like – Login page, Book collection list, shopping cart, etc.

2) Business logic Layer: - This layer has the main functionality of the system. This layer is responsible to process the data received. It also creates a link between the presentation layer and database layer.
Some examples of its functions are – Authentication (checking user info provide by comparing from the stored data), Searching books, processing orders.

3) Database Layer: - This layer interacts with the database. It provides the create, read, update and delete functionality for the business logic layer.
Examples of its functionalities are – Fetch books (showing books according to the parameters given by the user), maintaining the record of inventory, Store information related to users and their orders.

Theory Exercise: The layers in software architecture are important as because of this every layer specific functions which reduces the complexity; also layers can be reusability some other application; it makes maintaining the application easier as changes in one layer generally doesn't affect other layer because of this upgrading the application is easier too as it technology in one layer is outdated it can be changed without affecting other layers.

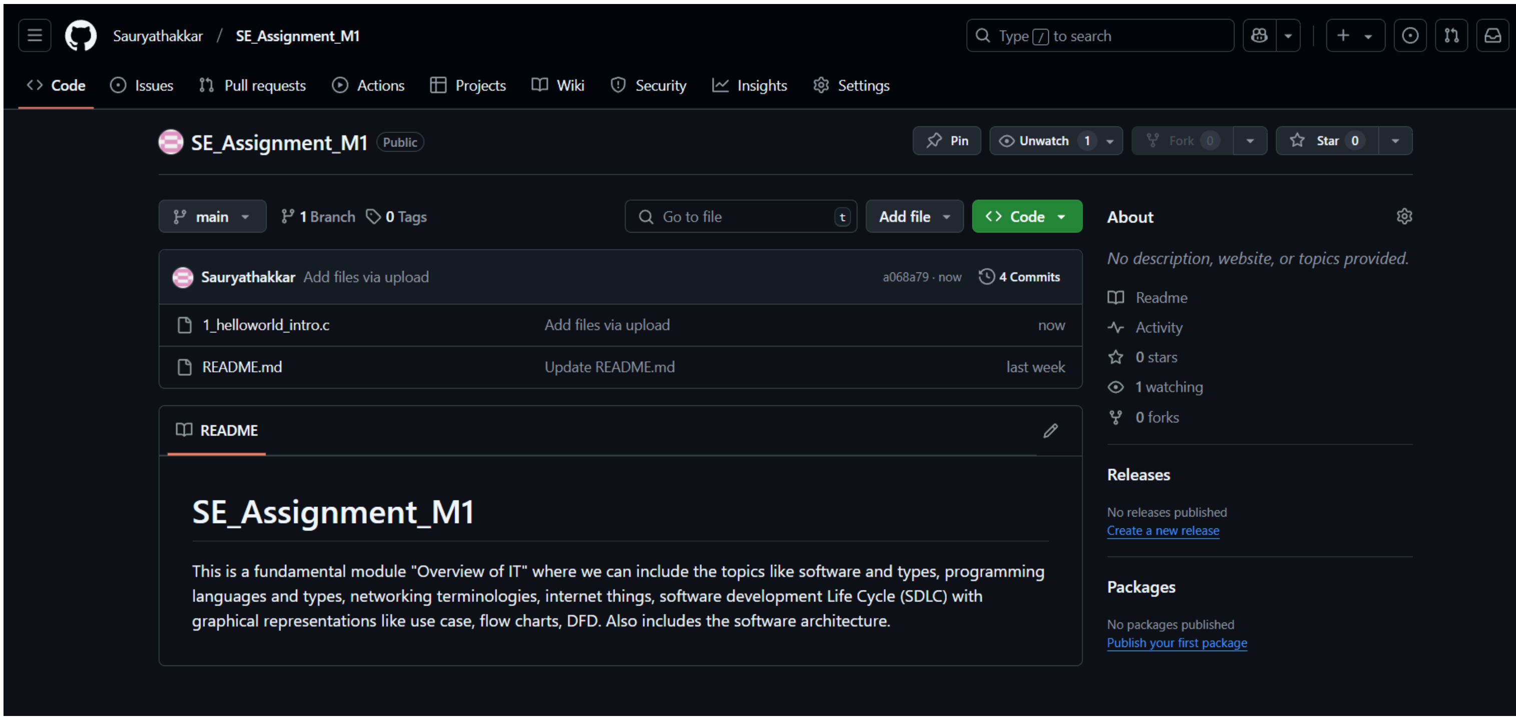
13. Software Environments

Lab Exercise:
Steps to set-up a basic software environment in a virtual machine: -
1 – Choose your virtual machine
2 – Select an operating system
3 – Install necessary software
4 – Create separate virtual environments for different stages

Theory Exercise: Development environment is important as it gives an accessible place for the programmer to write code

14. Source Code

Lab Exercise:



Sauryathakkar Add files via upload a068a79 · now 4 Commits

1_helloworld_intro.c	Add files via upload	now
README.md	Update README.md	last week

No description, website, or topics provided.

No releases published

No packages published

SE_Assignment_M1

This is a fundamental module "Overview of IT" where we can include the topics like software and types, programming languages and types, networking terminologies, internet things, software development Life Cycle (SDLC) with graphical representations like use case, flow charts, DFD. Also includes the software architecture.

Theory Exercise: Source code the code written by human in a programming language whereas machine code is the binary code which is made my converting source code to machine code using interpreter or compiler.

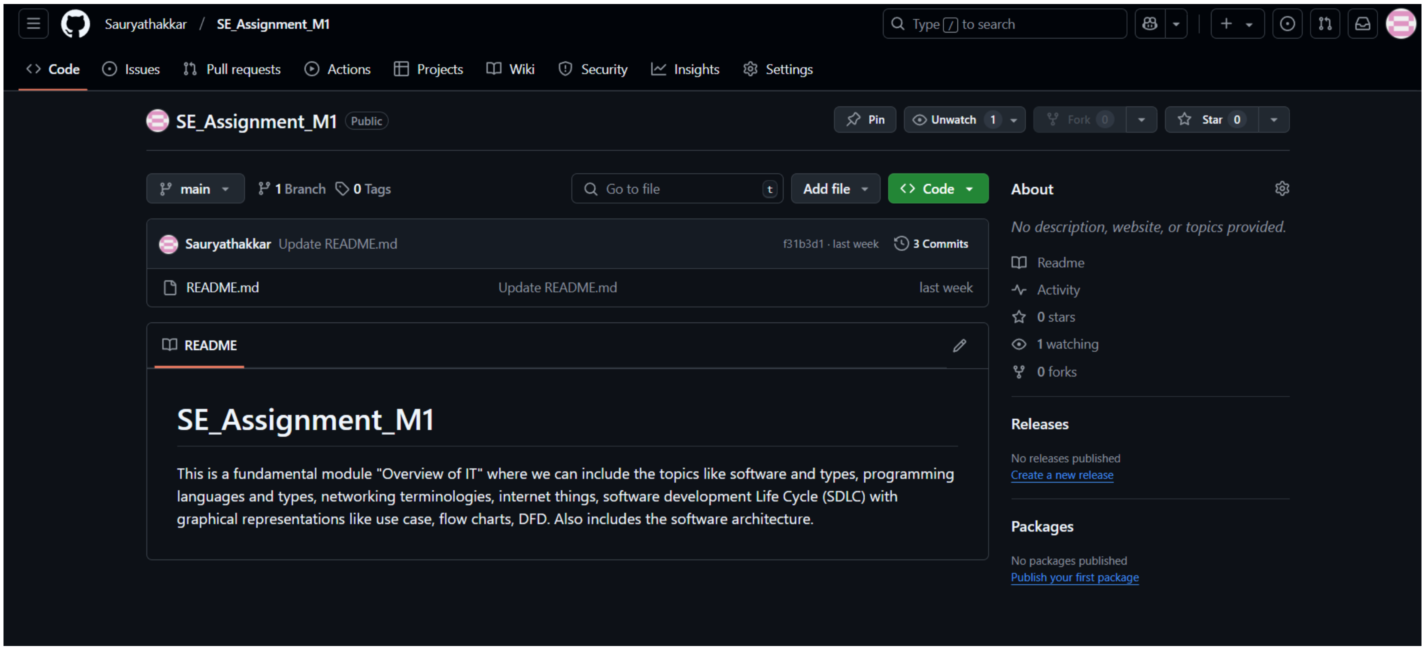
15. Github and introductions

Lab Exercise: To commit and push changes in a GitHub repository we use 'git commit <file_name>' and git push <file_name>' commands.

Theory Exercise: Version control is important as it helps to manage the project when a whole team is working on it.

16. Students Account in Github

Lab Exercise:



Sauryathakkar Update README.md f31b3d1 · last week 3 Commits

README.md	Update README.md	last week

SE_Assignment_M1

This is a fundamental module "Overview of IT" where we can include the topics like software and types, programming languages and types, networking terminologies, internet things, software development Life Cycle (SDLC) with graphical representations like use case, flow charts, DFD. Also includes the software architecture.

Theory Exercise: Benefits of using GitHub for students is that GitHub helps in managing a project and also show it other people. It provides tools that help in learning. It also makes collaboration easier.

17. Types of software

Lab Exercise:

Name of software	Type of software
WinRAR	Utility software
McAfee (antivirus)	Utility software
Notepad	System software
Adobe Acrobat	Application software
VS code	Application software

Theory Exercise: Open-source software’s code is available for anyone to change, view and distribute their modifications, whereas proprietary software is owned by a company or person and cannot be accessed by anyone.

18. GIT and GITHUB training

Lab Exercise: Command used for cloning, branching and merging repositories in GitHub are: -
Cloning – git clone <URL of repository>
Branching – git branch <branch_name>

Merging – git merge <branch_name>

Theory Exercise: Git improves collaboration in a software development by making is possible for many programmers to work on a project together in the same codespace and keep track of the changes made by all the programmers and merge all the parts developed by all programmers.

19. Application software

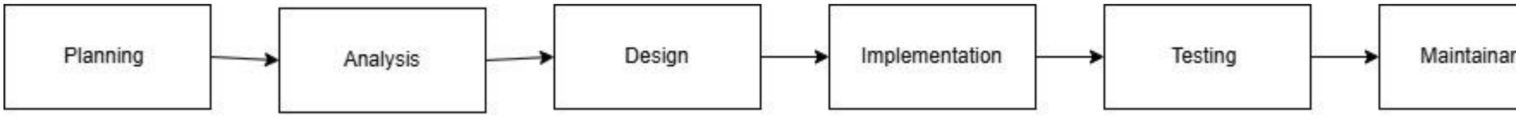
Lab Exercise: Application softwares help to improve productivity by making hard and time taking tasks easier and complete in lesser time for example: -

- MS word makes it is easy to create documents
- MS excel helps to organize data in table, visualize it using different types of graphs and analyze it using formulas built into the app
- MS powerpoint helps to create presentations which makes explaining things to other people easier.

Theory Exercise: Application software helps businesses to perform complex tasks and be more efficient.
Application software can help to store, manage, analyze and share information more easily.

20. Software Development process

Lab Exercise:



Theory Exercise: The stages in software development life cycle are: -
Planning, Analysis, Design, Implementation, Testing, Maintenance

21. Software requirement

Lab Exercise: Requirement specification for a simple library management system will be: -

- Add Book
- Edit Book
- Delete Book
- Search Book

Theory Exercise: Requirement analysis phase is critical in software development because it clearly defines the needs and expectations of the users and stakeholders for the project. Which ensures that the final project is made according to those needs, reducing the need on reworking on part of the project.

22. Software Analysis

Lab Exercise: An online shopping system would have following features: -

- User Account management – Registration, login, profile management and password reset.
- Product catalog navigation – Product listing, product details page, search functionality and sorting options
- Shopping cart management – Add to cart, view cart and cart updates.
- Checkout Process – Billing address, shipping options, payment gateway integration and order summary.
- Order management – Order confirmation, order tracking and order history
- Customer support – Contact Us form, FAQs and Live chat

23. System Design

Lab Exercise: Basic system architecture of a food delivery app: -

1 - Frontend

- Platforms:
  - Mobile Apps (iOS, Android)
  - Web Application
- Key Features:
  - User Registration/Login
  - Restaurant Search and Browsing
  - Menu Display and Customization
  - Cart and Order Placement
  - Real-time Order Tracking
  - Payment Integration
  - Notifications

2. Backend

- Core Components:
  - Authentication and Authorization
  - Restaurant Management
  - Order Management

- Payment Processing
- Recommendations

3. Database layer

Theory Exercise: The elements of system design include: -

- Architecture – The design of phases which define major components, their relations and the overall flow of information
- Scalability – The possibility to expand the current project into something bigger without losing efficiency
- Security – The protection of sensitive data from unwanted access, use or disclosure.
- Documentation – Documentation of the process, style guides, principles and components.

24. Software Testing

Lab Exercise: The test case for a simple calculator may include testing if addition, subtraction, multiplication and division of two given provides correct outputs or not.

Theory Exercise: Software testing is important as it makes sure that the program doesn't given out incorrect outputs.

25. Maintenance

Lab Exercise: Case Study: Boeing 737 MAX crisis
Boeing 737 MAX's usage was stopped worldwide in March 2019 after two fatal crashes. The reason of the crashes was identified as a flaw in their Maneuvering Characteristics Augmentation System (MCAS), a software which is designed to increase aircraft's stability.

The MCAS relied on data from a single Angle of Attack (AoA) sensor. If this sensor provides wrong data, MCAS could push the plane's nose down unnecessarily. The Pilots were not trained to recognize this issue and also the MCAS activation in this situation wasn't documented in the flight manual. And the FAA (Federal Aviation Administration) relied on outdated certification processes which allowed this issued to be unrecognized.

Theory Exercise: Types of software maintenance are:

- Corrective maintenance – Fix bugs and errors after the software's been released.
- Preventive maintenance – Actively finds and fixes issues before they cause the need of downtime
- Perfective maintenance – Improves the software's functionality, performance and reliability.

- Adaptive maintenance – Modifies the software to adapt to changes in its environment.

26. Development

Theory Exercise: Differences between web apps and desktop apps are as follows: -

Desktop applications	Web applications
These are needed to be downloaded to the user's device to be used	No installation is needed, anyone with an internet connection can use these
These can be accessed only from the device they are installed on	Anyone with an internet connection can access these.
These are faster as they don't have to communicate with a server over the net (i.e.no latency)	They are slower as they need to communicate with a server with create lag between the input and output.

27. Web Applications

Theory Exercise: Advantages of web applications is that they don't need to be installed in the device, it can be used any device with internet connection because of this it doesn't use too much of the device's resources.

28. Designing

Theory exercise: UX/UI design ensures that the application's design is visually appealing and also easily understandable to the user

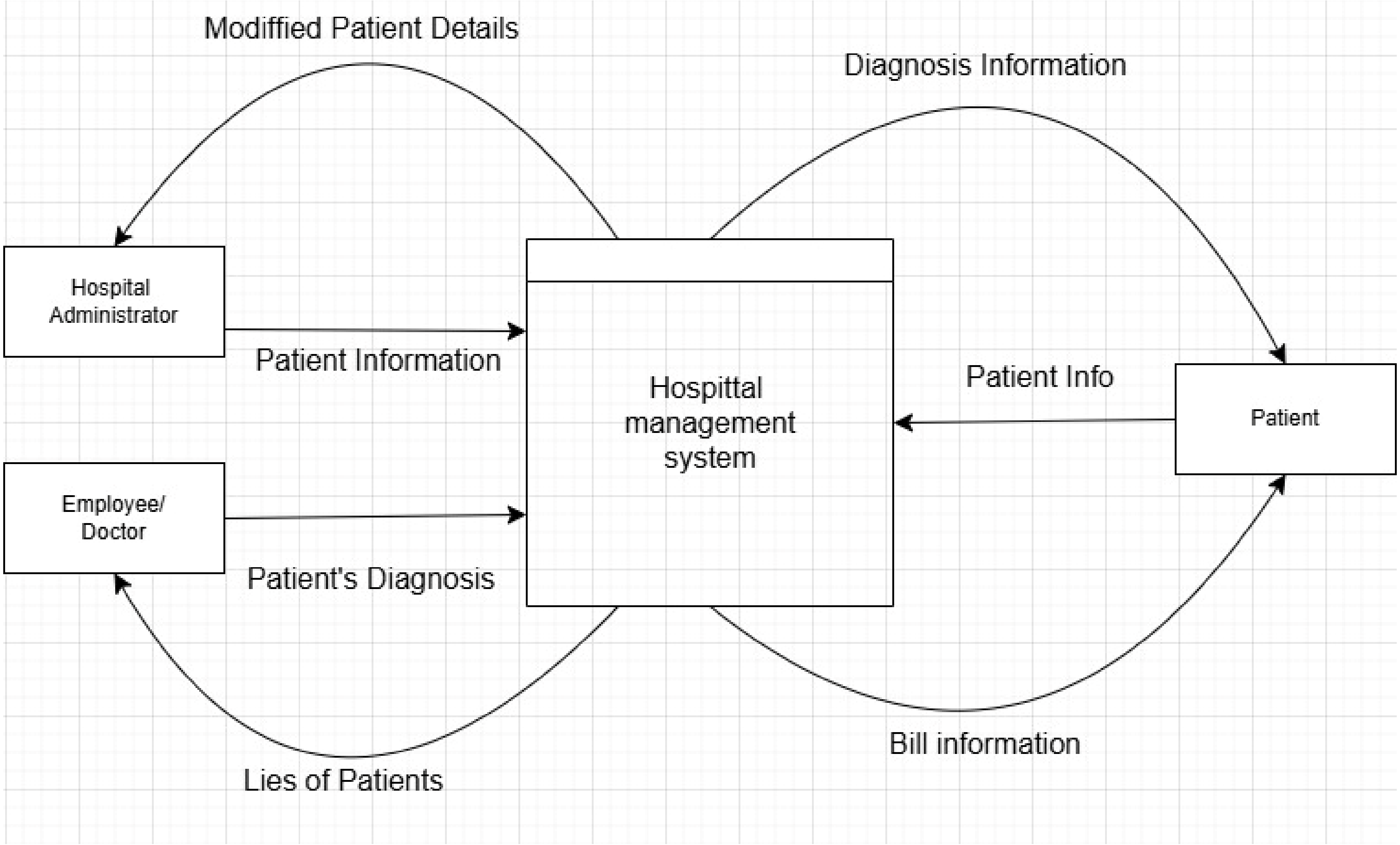
29. Mobile Applications

Theory Exercise: Difference between native and hybrid mobile apps:

Native apps	Hybrid apps
These are built using a specific programming language for the platform they're being developed for. Ex – Swift for IOS	These are built using a single codebase. Ex- HTML, CSS and JavaScript.
These are only available from platform specific app stores.	These are available on many app stores and can also be accessed through a web link.
These tend to perform better as they can utilize the hardware and software capabilities of the device fully.	These can't do the same as these are built for multiple platforms which make them weaker in performance.

30. DFD

Lab Exercise:



Theory Exercise: As DFD shows the flow of data in system, it makes it easy to visualize and understand the flow of data in that system before the actual system is ready.

31. Desktop Application

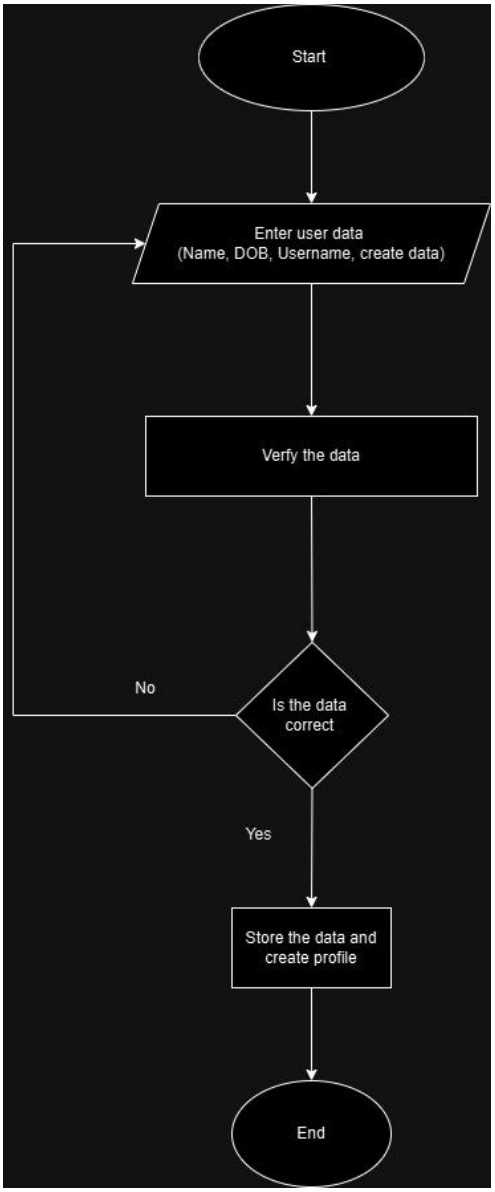
Lab Exercise: Steps to build a simple calculator application using a GUI library: -

- Create a skeleton for your calculator app
- Complete the App’s view
- Implementing the calculator’s model
- Creating the controller class for your calculator
- Running the calculator.

Theory Exercise: Desktop applications offer faster performance compared to web applications as the desktop apps run locally on the device, because of which it is able to use the computer's resources.

32. Flow Chart

Lab Exercise:


Start
Enter user data
(Name, DOB, Username, create data)
Verfy the data
Is the data correct
No
Yes
Store the data and create profile
End

Theory Exercise: Flowcharts helps to visualize the flow of a program