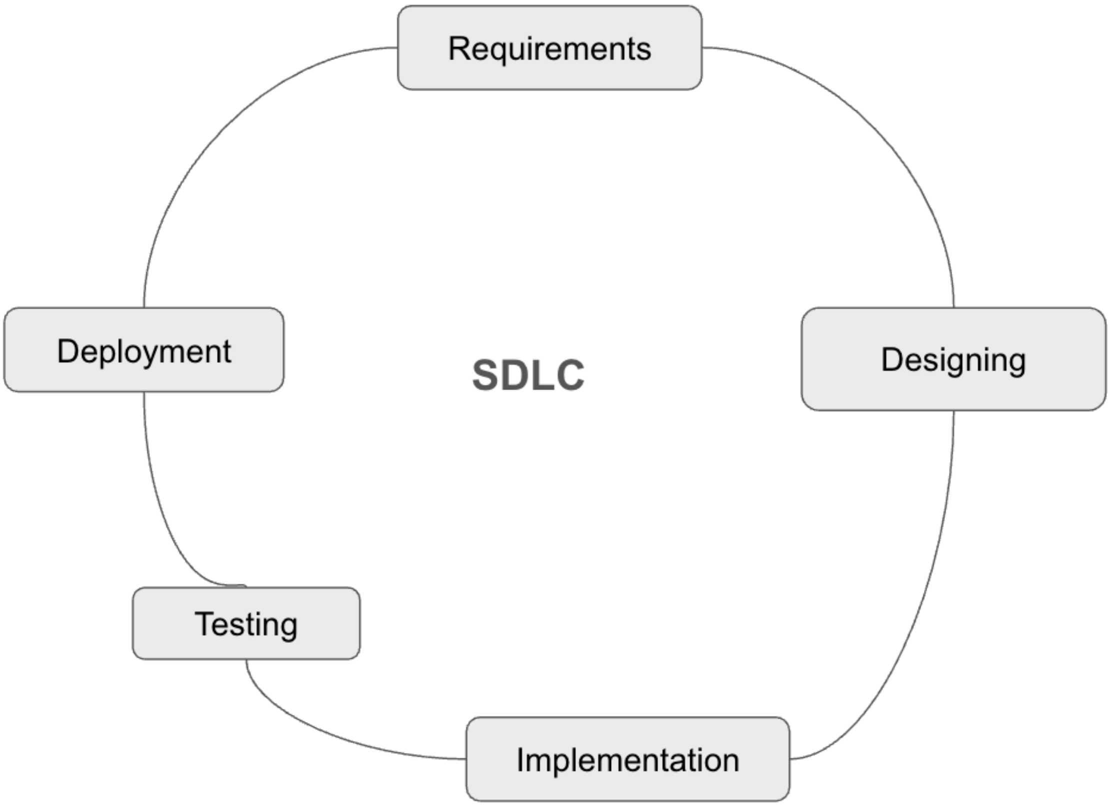
Q1. SDLC OverView - Create a one-page infographic that outlines the sdlc phases(Requirements,Design,Implementation,Testing,Deployment), highlighting the importance of each phases and how they interconnect.

Ans:



1. Requirements Phase
    - Importance: Defines the project scope, objectives, and features from the user's perspective.
    - Interconnection: Provides the foundation for the design phase by outlining what the software should accomplish.
2. Design Phase
    - Importance: Transforms the requirements into a detailed design that guides the development team.
    - Interconnection: Builds upon the requirements phase, ensuring that the software design aligns with user needs and project goals.
3. Implementation Phase
    - Importance: The actual coding and development of the software based on the design specifications.
    - Interconnection: Relies on the design phase to ensure that the implementation aligns with the planned architecture and features.
4. Testing Phase

    - Importance: Validates that the software meets the specified requirements and is free of defects.
    - Interconnection: Follows the implementation phase to verify that the software functions as intended and meets quality standards.
5. Deployment Phase
    - Importance: Involves releasing the software to users or clients for regular use.
    - Interconnection: Follows successful testing to ensure that the software is ready for use and meets user expectations.

Q2. Develop a case study analysing the implementation of SDLC phases in real world engineering projects. Evaluate how Requirement gathering ,Design Implementation,
Testing , Deployment, Maintenance contribute to the project outcome.

Ans:

Case Study: <u>Hospital Management System</u>

Introduction:

Hospital management systems (HMS) play a crucial role in modern healthcare facilities, improving patient care, streamlining administrative processes, and enhancing overall efficiency. This case study analyzes the implementation of an HMS in a large urban hospital to illustrate its impact on healthcare delivery.

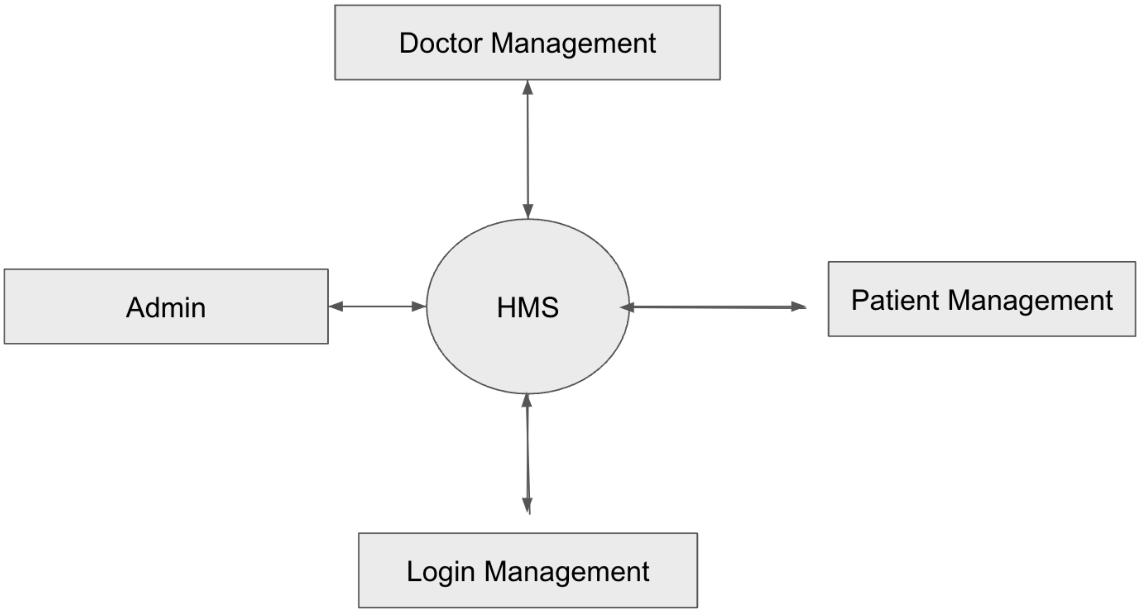
1. Requirement Gathering:

- Process: The hospital conducted extensive consultations with healthcare providers, administrators, and IT professionals to identify key requirements, and created SRS document for this
- Outcome: A comprehensive list of requirements was developed, including patient registration, appointment scheduling, electronic health records (EHRs), and billing functionalities

2. Design Implementation:

- Process: Based on the requirements, the HMS team designed a modular system architecture that integrated seamlessly with existing hospital systems.

- Outcome: Detailed design specifications, including database schemas, user interface mockups, and system flow diagrams, were created.



3. Implementation:

- Process: The HMS was developed using agile methodologies, with regular feedback loops to ensure alignment with user needs.
- Outcome: The system was implemented in phases, starting with core functionalities such as patient registration and gradually expanding to include Doctor Appointments and login.

4. Testing:

- Process: Rigorous testing was conducted at each stage of development, including unit testing, integration testing, and user acceptance testing.

5. Deployment:

- Process: The HMS was deployed in two environments initially as first in staging and second in production, in staging for testing and fixing bugs and later pushing that code into production.

6. Maintenance:

- Process: Post-deployment, the HMS underwent continuous monitoring and maintenance to ensure optimal performance and also for handling security risks.

.

Q3. Research and Compare SDLC models suitable for engineering projects. Present findings on Waterfall,Spiral,Agile and V model approaches, emphasise their advantages and disadvantages in different engineering contexts.

Ans:

WaterFall Model: Here in this model each phase must be completed before moving to the next stage.
Advantages:
1. It's very simple and easy to implement.
2. It's most suited for projects where requirements are stable.
Disadvantages:
1.Not suitable for projects which needs modifications in requirements,
2.Limited user involvement when the project is in its last phases(so that may create misunderstanding about some implementation part).

Spiral Model:
This is a combination of both Iterative and Waterfall models and used mainly for risk analysis and incremental releases.

Advantages:
1.Allows incremental releases which can reduce the risks.
2.here you can make changes in requirements during different phases.
Disadvantages:
1.More complex than waterfall model and requires thorough risk analysis.
2. Expensive due to its iterative nature.

Agile Model :
This is a combination of incremental and iterative models, here it delivers working softwares in short iterations .

Advantages:
1. Highly flexible and responsive to change.
2.Promotes developer and clients to work together.

Disadvantages:

1. Requires a high level of collaboration and communication between teams.
2. Predicting timeline and cost is challenging in initial phases.

V-Model:
This model is a variation of the Waterfall Model where each phase is associated with the testing phase.

Advantage:

1.Testing is there with every phase.
2.its better for projects which have strict requirements .

Disadvantage:
1.limited flexibility for changes, once a phase completed
2.Complex to implement especially in large projects.