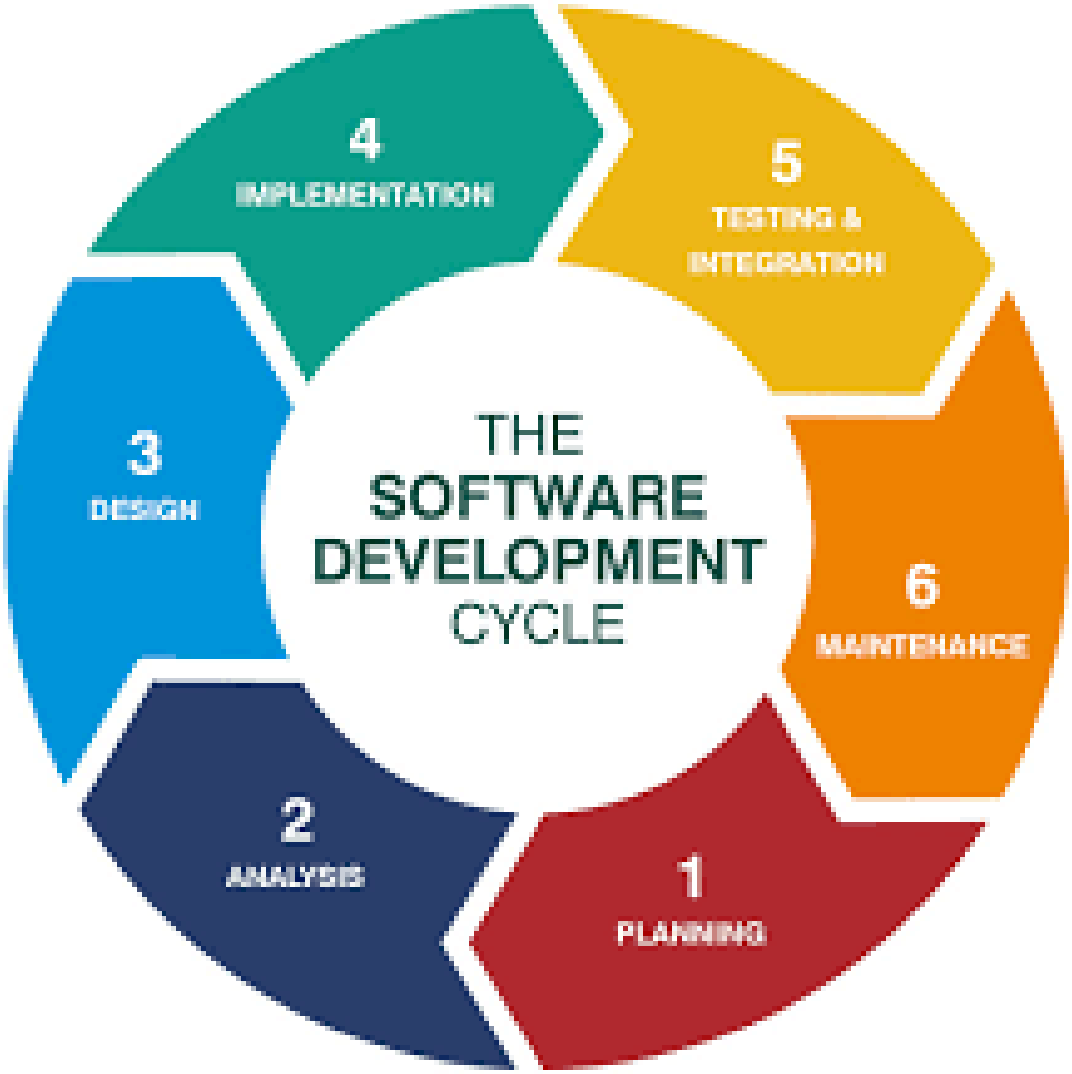
CSE 317
System Analysis and Design

Here's how the **University Course Registration System** can be divided and organized according to each phase of the System Development Life Cycle (**SDLC**):



---

1. Planning Phase

Objective: Establish the project's **goals**, **feasibility**, **resources**, and **schedule**.

Tasks:

- **Define Project Goals:** Create a system to streamline course registration, schedule management, and enrollment tracking for students, faculty, and administrators.
- **Identify Stakeholders:** Key stakeholders include students, faculty, university administrators, and IT staff.
- **Feasibility Analysis:** Assess the technical, financial, and operational feasibility of the project.
- **Resource Allocation:** Identify necessary resources such as developers, hardware, and software tools.
- **Timeline and Budget:** Create a timeline and budget plan based on estimated tasks and required resources.

Outcome: A project plan detailing objectives, timelines, resources, and stakeholders, forming a roadmap for the project.

---

2. Analysis Phase

Objective: Gather detailed requirements to understand what the system should do.

Tasks:

- **Requirements Gathering:** Conduct interviews and surveys with stakeholders to identify system needs.
    - *Students* need an easy way to view course options, register, and manage schedules.

    - *Faculty* need to manage course details, monitor enrollments, and view their schedules.
    - *Administrators* require full control over course offerings, user accounts, and reporting.
- **Functional Requirements:**
    - User management with role-based access (students, faculty, administrators).
    - Course catalog and schedule creation.
    - Enrollment management, including waitlists and prerequisites checks.
    - Reporting tools for academic progress and course enrollment data.
- **Non-Functional Requirements:**
    - Security protocols for data privacy.
    - Scalability to handle high loads during registration periods.
    - Usability for a smooth user experience across different devices.
- **Requirements Document:** Summarize and document all gathered requirements.

Outcome: A requirements specification document outlining the functional and non-functional needs of the system.

---

3. Design Phase

Objective: Define the architecture, data structures, user interface, and components of the system.

Tasks:

- **System Architecture Design:** Outline the overall structure, including the database, application servers, and user interface.
- **Database Design:** Create a relational database model to store data about students, courses, enrollments, schedules, and user roles.
- **User Interface (UI) Design:** Design wireframes and mockups for student, faculty, and administrator portals.
- **Data Flow Diagrams (DFD):** Create diagrams showing data movement between modules.
- **System Security Design:** Design security protocols for user authentication and data access control.
- **Technical Specifications:** Define technology stack, such as programming languages, frameworks, and database management systems.

Outcome: A design document including system architecture, database schema, UI mockups, and data flow diagrams.

---

4. Implementation (Development) Phase

Objective: Build the system according to the design specifications.

Tasks:

- **Front-End Development:** Develop the user interface based on the UI mockups, creating responsive pages for different user roles (students, faculty, and administrators).
- **Back-End Development:** Implement the server-side logic, including user authentication, course enrollment, waitlisting, and reporting.
- **Database Implementation:** Set up the database schema and develop queries for data retrieval and manipulation.
- **Integration:** Link the front-end and back-end components, ensuring that data flows seamlessly between the database and user interface.
- **Security Implementation:** Integrate secure authentication and access control for different user levels.

Outcome: A functioning University Course Registration System with core features implemented and ready for testing.

Testing Phase

Objective: Ensure the system functions correctly and meets all requirements.

- **Unit Testing:** Test individual components.
- **Integration Testing:** Ensure components work together.
- **System Testing:** Test the complete system functionality.
- **User Acceptance Testing (UAT):** Get feedback from actual users.
- **Performance Testing:** Ensure the system performs well under load.
- **Security Testing:** Check for vulnerabilities and data protection.
- **Regression Testing:** Ensure no features are broken after updates.

Outcome: A bug-free, functional system ready for deployment.

---

6. Implementation (Deployment) Phase

Objective: Deploy the system to the production environment.

- **Preparation for Deployment:** Set up servers and ensure dependencies.
- **Data Migration:** Move data from legacy systems.
- **Deployment:** Launch the system live.
- **User Training:** Provide training and guides for end-users.

Outcome: A live system with trained users.

---

7. Maintenance Phase

Objective: Keep the system running smoothly post-deployment.

- **Bug Fixes:** Address any reported issues.
- **System Updates:** Add new features and enhancements.
- **Performance Monitoring:** Ensure the system handles peak loads.
- **Security Updates:** Apply patches for vulnerabilities.
- **User Support:** Provide ongoing assistance to users.

Outcome: A stable, secure system with continuous updates.

Project Proposal

Project Title

- Example: ***"Online Hospital Appointment System"*** or ***"Inventory Management System for Small Retailers"***

Project Introduction

- Briefly describe the project's purpose.
- Example: *"This project aims to develop an online system for booking and managing hospital appointments, enhancing patient convenience and reducing administrative workload."*

Problem Statement

- Define the problem the project will address.
- Example: *"Current hospital appointment processes are time-consuming and require in-person visits, leading to patient dissatisfaction and delays."*

Objectives

- Outline the main objectives of the project. Be specific.
    - Example:
        - Develop an easy-to-use online booking interface for patients.
        - Provide real-time updates on appointment availability.
        - Reduce administrative processing time.

Scope of the Project

- Define what the project will and will not include.
    - Example:

    - *In Scope*: User registration, appointment scheduling, notifications.
    - *Out of Scope*: Payments, patient record management.

System Requirements

- List the initial functional and non-functional requirements.
    - **Functional Requirements**: e.g., "User login and authentication," "Appointment scheduling."
    - **Non-Functional Requirements**: e.g., "Responsive interface," "Data security."

Project Methodology

- Briefly explain the approach (e.g., waterfall, agile).
- Describe each phase: requirement analysis, system design, implementation, testing, etc.

Proposed Tools and Technologies

- List programming languages, frameworks, and software tools you plan to use.
- Example: "HTML, CSS, JavaScript for the frontend; Python and Django for the backend; MySQL for the database."

Expected Outcomes

- Describe the benefits or impact of the project.
- Example: "By implementing this system, hospitals can streamline appointments, improve patient satisfaction, and reduce wait times."

Conclusion

- Summarize why this project is valuable and feasible.

Lab Assignment - 1

1. Brainstroming about two project ideas.
2. Make two project ideas and submit it to me.
3. Deadline: 08/11/2024