Vanier College
Continuing Education

01 Introduction to System Development

System Development

420-940-VA Sect 87414

Teacher: Jay Patel

What is a System?

A system is a set of interrelated components working together toward a common goal.
Example: A car engine is a system of parts that work together to move the vehicle.

Key Characteristics of a System:

- **Components**: Separate parts of the system.
- **Interrelationships**: How the parts work together.
- **Boundary**: Defines what is inside and outside the system.
- **Purpose**: Systems exist for a specific reason.
- **Input & Output**: Every system has inputs and produces outputs.

What is Information System ?

An **Information System (IS)** is a structured system that collects, processes, stores, and distributes information to support decision-making, coordination, control, analysis, and visualization within an organization or for individual users

It combines **technology**, **people**, and **processes** to manage data and transform it into meaningful information that can be used to achieve business goals.

Components of an Information System

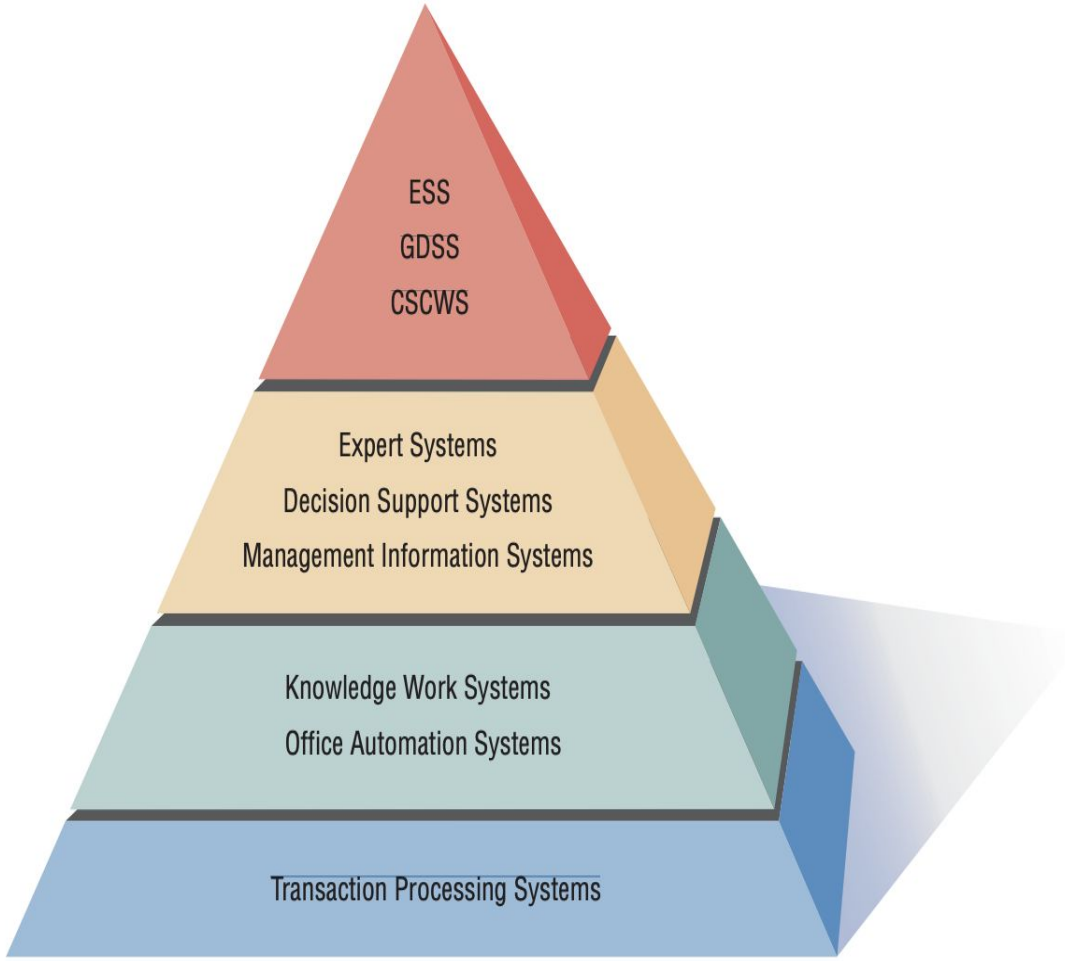
- **Hardware**: Physical devices like computers, servers, networking equipment, and storage devices that are used to collect, store, and process data.
- **Software**: Programs and applications that run on the hardware, including operating systems, databases, and specific applications that help manage and manipulate data.
- **Data**: Raw facts and figures that the system processes into useful information. This can include anything from customer details to sales figures, inventory levels, or financial records.
- **People**: Users who interact with the information system, including IT staff, management, and end-users. They define what data is needed, what processes the system should follow, and how the system is utilized.
- **Processes/Procedures**: The workflows and methods that dictate how data is collected, processed, and analyzed. This can include everything from data entry to reporting, and procedures for using the system to solve problems or make decisions.
- **Networks**: Communication systems that link the components together and allow for the exchange of data across different systems and geographic locations.

Functions of Information Systems

- **Data Collection and Storage**: Systems gather data from various sources and store it in databases.
- **Data Processing**: The raw data is processed and transformed into meaningful information. For example, sales data may be used to generate reports on product performance.
- **Information Distribution**: The system shares the processed information with users, helping them make smart choices. This information can be shown as reports, dashboards, or notifications.
- **Support for Decision-Making**: Information systems provide tools for analyzing data, supporting managers and executives in making strategic and operational decisions.
- **Automation of Business Processes**: IS can automate repetitive tasks, such as payroll processing, inventory tracking, or customer support, improving efficiency and accuracy.

Types of Information Systems

- Transaction Processing Systems (TPS)
- Office Automation Systems (OAS)
- Knowledge Work Systems (KWS)
- Management Information Systems (MIS)
- Decision Support Systems (DSS)
- Expert Systems
- Executive Support Systems (ESS)
- Group Decision Support Systems (GDSS)
- Computer-Supported Collaborative Work Systems (CSCWS)



Transaction Processing Systems (TPS)

Purpose: Automate routine business transactions (e.g., payroll, inventory).

- **Key Role**: Operates at the **operational level** of an organization.
- **Characteristics**:
  - Boundary-spanning, interacting with external environments.
  - Essential for day-to-day operations.

Examples: Transaction Processing Systems (TPS):

- **Point of Sale (POS) System**: Used in retail stores to process sales transactions. It records customer purchases, updates inventory, and prints receipts.
  - Example: **Square POS** for small businesses or **Walmart's POS** system that handles millions of transactions daily.
- **Online Banking Systems**: Used by banks to process transactions such as deposits, withdrawals, and fund transfers.
  - Example: **Bank of America** or **Chase Bank** online banking systems.
- **Payroll Systems**: Automates the processing of employee salaries, deductions, and taxes.
  - Example: **ADP Payroll System** or **Workday** for managing employee payroll and benefits.

Office Automation Systems (OAS) and Knowledge Work Systems (KWS)

- **Office Automation Systems (OAS)**:
  - Supports **data workers** (e.g., word processing, spreadsheets, emails).
  - Analyzes and manipulates data before sharing within or outside the organization.
- **Knowledge Work Systems (KWS)**:
  - Supports **professional workers** (e.g., scientists, engineers).
  - Aids in **creating new knowledge** and sharing it with the organization.

Examples: Office Automation Systems (OAS)

- **Microsoft Office Suite**: A collection of productivity software for word processing, spreadsheets, and presentations.
- **Google Workspace**: Cloud-based tools for creating and collaborating on documents, spreadsheets, and presentations.
- **Microsoft Outlook / Google Mail**: Email and scheduling systems for communication and managing calendars.
- **Zoom / Microsoft Teams (Video Conferencing)**: Tools for virtual meetings, video calls, and online collaboration.
- **Google Calendar / Microsoft Outlook Calendar**: Electronic scheduling tools for managing appointments and meetings.
- **DocuSign**: A system for managing electronic signatures and secure document exchange.
- **SharePoint**: A platform for document management and collaboration within organizations.

Examples: Knowledge Work Systems (KWS)

- **AutoCAD**: Software for creating 2D and 3D designs used by engineers and architects.
- **MATLAB**: A programming environment used for data analysis and algorithm development.
- **SAS**: A system for advanced analytics, data management, and business intelligence.
- **SPSS**: A software tool for statistical analysis in research and business applications.
- **SolidWorks**: A 3D CAD software used for designing mechanical products and simulations.
- **IBM Watson**: An AI-powered system for data analysis and knowledge discovery.
- **Epic Systems (EHR Systems)**: Software for managing electronic health records in healthcare.
- **Google Patents**: A tool for searching patents and intellectual property databases.

Management Information Systems (MIS)

- Supports broader organizational tasks at Management Level
- MIS integrate data from various business functions (e.g., sales, finance, human resources) into a centralized database, making it easier to track performance, monitor trends, and generate reports for informed decision-making.
- Provides routine reports and information to help manage operations
- Typically used by middle and lower-level managers.

Examples: Management Information Systems (MIS)

- **Inventory Management System**: Helps businesses track their inventory levels, orders, and sales to ensure efficient stock control.
  - Example: **Oracle NetSuite** or **SAP ERP** for inventory management.
- **Human Resource Management System (HRMS)**: Used for managing employee data, benefits, and performance reviews.
  - Example: **BambooHR** or **SAP SuccessFactors** for HR management.
- **Sales Reporting System**: Generates sales reports for managers, helping them make informed decisions based on trends, customer behavior, and sales performance.
  - Example: **Salesforce CRM** or **Zoho CRM** for sales management and reporting

Decision Support Systems (DSS)

- Helps in **decision-making** phases but leaves the final decision to the user.
- Assists in making complex, unstructured, or semi-structured decisions.
- **Focus**: Business intelligence and supporting decisions.
- Typically used by higher-level managers and decision-makers.
- Output: Interactive tools like simulations, what-if analysis, and data visualizations

Decision Support Systems (DSS):

- **Business Intelligence (BI) System**: Provides data analysis and visualization tools to help managers make strategic decisions by identifying trends, patterns, and insights from large datasets.
  - Example: **Microsoft Power BI**, **Tableau**, or **QlikView** for business intelligence and decision support.
- **Supply Chain Management (SCM) System**: Assists in optimizing the supply chain by analyzing inventory levels, supplier performance, and logistics to improve efficiency.
  - Example: **SAP SCM** or **Oracle SCM** for supply chain planning and optimization.

Expert Systems

- **Purpose**: Solves **structured problems** by applying expert knowledge.
- **Components**:
  - **Knowledge base**: Stores expert information.
  - **Inference engine**: Processes queries and provides solutions.
  - **User Interface**: Connects the user with the system.
- **Key Feature**: Chooses the best solution, unlike DSS, which leaves decisions to users.

Examples: Expert Systems

- **MYCIN**: An early expert system used in medical diagnosis to identify bacterial infections and recommend treatments.
- **DENDRAL**: Developed for chemists, it helps in identifying molecular structures of chemical compounds based on mass spectrometry data.

Group Decision Support Systems (GDSS) and Computer-Supported Collaborative Work Systems (CSCWS):

- **Group Decision Support Systems (GDSS)**:
  - Helps groups make **semi structured** or **unstructured decisions**.
  - Encourages group collaboration with tools like **brainstorming**, polling, and scenario creation.
- **Computer-Supported Collaborative Work Systems (CSCWS)**:
  - Broader term for systems that support group collaboration

Examples: GDSS and CSCWS

Group Decision Support Systems (GDSS) Examples:

1. **GroupSystems**: A GDSS platform for brainstorming, voting, and decision-making, often used in organizational meetings.
2. **ThinkTank**: A cloud-based GDSS that facilitates collaboration, idea generation, and decision-making for teams.
3. **MeetingWorks**: A GDSS designed to support collaborative decision-making processes, particularly in workshops and large group meetings.

Computer-Supported Collaborative Work Systems (CSCWS) Examples:

1. **Microsoft Teams**: A collaboration platform that supports team communication, file sharing, and virtual meetings to facilitate collaborative work.
2. **Slack**: A messaging platform that enables teams to communicate and collaborate on projects in real time, supporting remote work and team coordination.
3. **Google Workspace**: A suite of productivity tools (Docs, Sheets, Slides, etc.) that allow multiple users to collaborate on documents simultaneously.
4. **Trello**: A project management tool that helps teams collaborate, organize tasks, and track project progress through boards and lists.

Executive Support Systems (ESS)

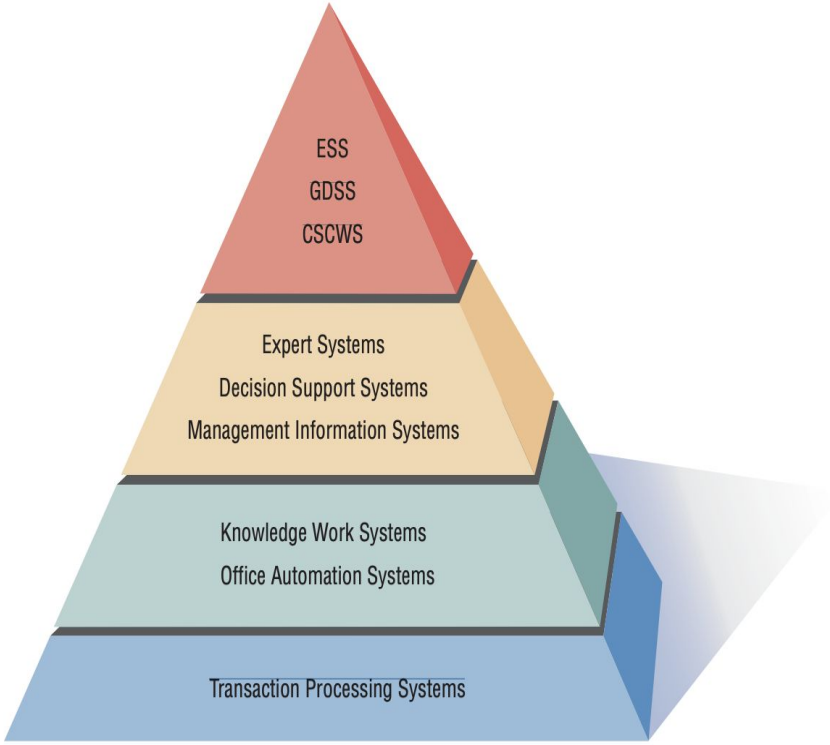
- **Purpose**: Supports **strategic-level** decision-making.
- **Features**: Provides **graphics** and **communication tools** to assist executives.
- **Role**: Helps executives understand their environment and make informed decisions.

Executive Support Systems (ESS)

- **BOARD**: A decision-making platform that combines business intelligence and performance management tools for executive use.
- **SAP BusinessObjects**: A business intelligence platform that provides executives with dashboards, reports, and data visualization tools for strategic decision-making.

Hierarchy of Information Systems

- **TPS**: Operational level – Routine transactions.
- **OAS and KWS**: Knowledge level – Data workers and professionals.
- **MIS and DSS**: Management level – Decision-making support.
- **ESS, GDSS, CSCWS**: Strategic level – Group and executive decisions.



System Analysis

- System analysis is the process of examining and understanding the needs of an organization or system. It involves identifying the inputs, outputs, data flows, processes, and storage requirements in order to understand the existing system or business processes.
- The goal of system analysis is to identify the problem areas, evaluate current practices, and determine user needs to ensure that the new system addresses these issues effectively.

Who is a System Analyst and Their Roles?

System Analyst: A professional responsible for analyzing, designing, and implementing information systems to meet business and user needs.

Key Roles:

1. **Requirement Gathering**: Works with stakeholders to identify and document system requirements.
2. **System Design**: Develops system architecture, interfaces, and data models.
3. **Feasibility Analysis**: Evaluates technical, economic, and operational feasibility.
4. **System Implementation**: Supervises development and ensures proper system testing.
5. **Problem Solving**: Addresses technical issues and suggests system improvements.
6. **Project Management**: Coordinates project timelines and collaborates with teams.
7. **Training & Documentation**: Creates user manuals and trains end-users on the system.

A **System Analyst** acts as a middle man between business and IT teams, ensuring smooth communication and efficient system development.

Introduction to System Development

What is System Development?

- The process of creating or improving systems to meet user needs or organizational goals.

Goal: To deliver an efficient, functional, and scalable solution.

Key Steps:

- Planning, analyzing, designing, implementing, and maintaining the system.

Importance of System Development

- **Why Develop Systems?**
  - Automates processes and improves efficiency.
  - Reduces human error and operational costs.
  - Enhances user satisfaction by fulfilling requirements.
- **Real-World Examples**:
  - Hospital management systems, banking systems, and inventory management.

SDLC (System Development Life Cycle)

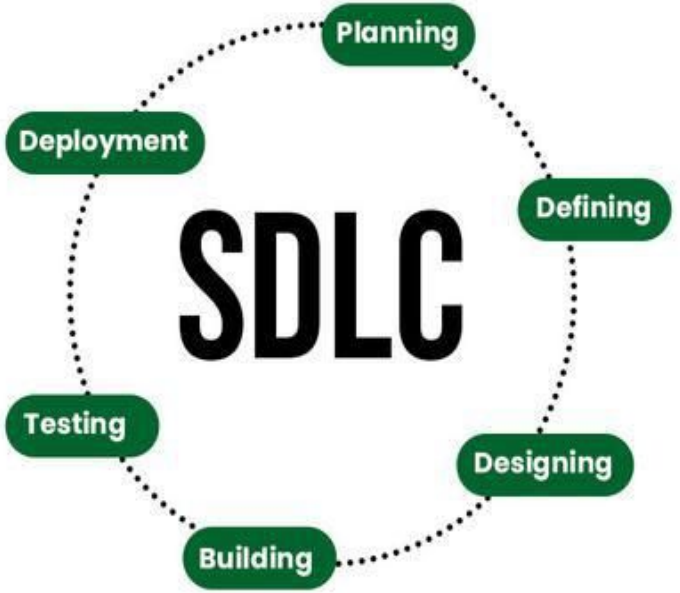
- SDLC is a structured approach used to design, develop, and maintain information systems.
- **Goal**: To ensure the system meets requirements, is cost-effective, and is delivered on time.
- **Common Phases**:
  - Planning, Analysis, Design, Implementation, Testing, Deployment, Maintenance.

Some Benefits of Using SDLC

- **Structured Process**: Ensures that development follows a clear, logical sequence.
- **Risk Management**: Identifies and mitigates risks early in the process.
- **Improved Quality**: Results in high-quality systems with fewer bugs.
- **Stakeholder Involvement**: Regular feedback from users and clients.

SDLC Phases

- **Phase 1**: Planning
- **Phase 2**: Analysis
- **Phase 3**: Design
- **Phase 4**: Implementation (Coding)
- **Phase 5**: Testing
  - Deployment
- **Phase 6**: Maintenance



Phase 1: Planning

- In the **Planning Phase** of the **System Development Life Cycle (SDLC)**, several key activities occur that lay the foundation for the entire project.
- This phase is critical because it sets the direction, scope, and objectives for the system.

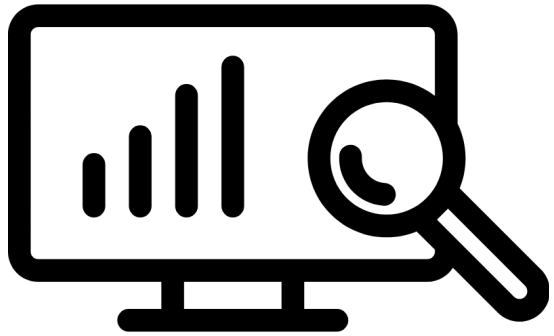
Phase 1: Planning

Key Activities

- **Feasibility Study**: Assess technical, economic, operational, and legal viability.
- **Stakeholder Identification**: Identify and involve key stakeholders.
- **Requirement Gathering**: Conduct interviews with clients and understand requirements.
- **PESTLE and SWOT Analysis**: Examine external and internal factors, including strengths, weaknesses, opportunities, and threats.
- **Project Scope Definition**: Outline the project's goals and boundaries.
- **Risk Management**: Identify risks and develop mitigation strategies.
- **Resource Allocation**: Plan the required resources (team, budget, technology, time).
- **Project Plan Creation**: Develop a detailed timeline, budget, and work structure.
- **Approval**: Get formal sign-off from stakeholders to proceed.

Phase 2: Analysis

- The **Requirement Analysis** phase in the **System Development Life Cycle (SDLC)** is one of the most crucial stages because it ensures that the system being developed meets the needs of users and stakeholders.
- The primary goal is to gather detailed information about what the system should do (functional requirements) and how it should perform (non-functional requirements), ensuring the system aligns with the business objectives.

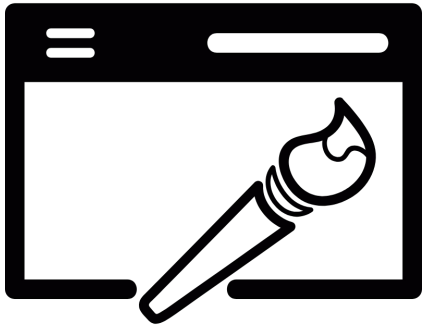


Phase 2: Analysis

- **Detailed Requirement Analysis**: Requirements are examined in depth to ensure clarity, consistency, and completeness.
- **Creating Use Cases and User Stories**: Use cases describe user interactions; user stories outline features from the user's perspective.
- **Data Modeling**: Logical data models define data entities, attributes, and relationships, often using ER diagrams.
- **Process Modeling**: Data Flow Diagrams (DFDs) and flowcharts map how data and processes move through the system.
- **Requirement Validation and Verification**: Stakeholders review and approve requirements to ensure they meet business needs.
- **Developing a Requirement Specification Document**: The final SRS document is created, detailing all requirements, models, and structures.

Phase 3: Design

- In the **Design Phase** of the **System Development Life Cycle (SDLC)**, the focus shifts to defining how the system will operate to meet the requirements gathered during the Requirement Analysis phase.
- This is where the system's architecture, components, interfaces, and data flows are designed in detail.
- The goal is to provide a blueprint for developers to follow during the implementation phase.



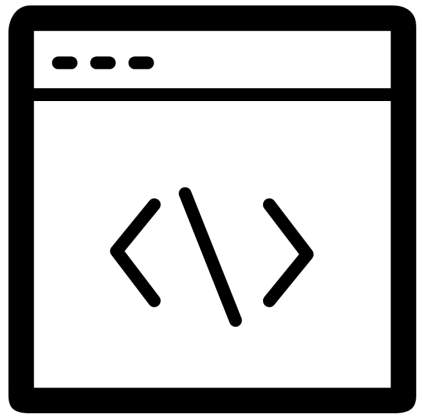
Phase 3: Design

Key Deliverables of the Design Phase:

1. **High-Level Design (HLD) Document**: Outlines the overall system architecture, major components, and technology stack.
2. **Low-Level Design (LLD) Document**: Provides detailed component-level designs, data structures, and interaction details.
3. **Database Schema**: Fully defined schema for the system's database.
4. **User Interface (UI) Design**: Wireframes, prototypes, and UI specifications for the system's user interface.
5. **Security Design**: A security blueprint detailing how the system will handle authentication, authorization, encryption, and data protection.
6. **Interface Specifications**: Documentation for any external system interfaces or APIs.
7. **Prototypes**: Working models or prototypes used to validate the design and gather stakeholder feedback.

Phase 4: Implementation

- The **Implementation Phase** is where the actual system is built based on the design specifications created during the Design Phase.
- In this phase, the system's components are coded, integrated, tested, and made operational. It’s where the design is transformed into a working system or application.
- This phase is critical because any issues or errors here will affect the functionality, performance, and usability of the final system.



Phase 5: Testing and Deployment

Testing Phase

- In the **Testing Phase** of the SDLC, the entire system is rigorously tested to ensure that it meets the defined requirements, is free of bugs, and functions as expected.
- This phase focuses on identifying defects, verifying system functionality, and validating that the system meets user expectations.

Deployment Phase

- In the **Deployment Phase** of the SDLC, the system is moved from the testing environment to the production environment, where it becomes operational and accessible to end-users.
- This phase also includes activities related to user training and system monitoring.

Phase 6: Maintenance

- The **Maintenance Phase** is the final phase of the **System Development Life Cycle (SDLC)** and begins once the system is deployed to the production environment.
- During this phase, the system is monitored, maintained, and updated as necessary to ensure it continues to function correctly and meets any evolving user or business requirements.
- Maintenance involves bug fixes, system updates, enhancements, and performance optimization to keep the system running smoothly over time.

Phases of **SDLC** with respect to **Course Outline**

Week	SDLC Phase	Steps Required to Finish the phase
Week 1	Planning Phase	Define project scope, objectives, and deliverables; Develop a project plan with milestones and timelines.
Week 2	Planning Phase	Understand and document the business domain and environment; Conduct structured interviews for information.
Week 3	Planning Phase	Conduct client interviews; Analyze interview data and finalize system requirements.
Week 4	Analysis Phase	Analyze the existing system or analogous systems; Model the system with data flow diagrams (DFDs) and ERDs.
Week 5	Analysis Phase	Identify business problems; Document and validate functional and non-functional requirements with the client.
Week 6	Design Phase	Create wireframes and prototypes for the user interface; Obtain client feedback and refine UI designs.
Week 7	Design Phase	Refine and finalize UI design based on feedback; Prepare for the next phase of database and system design.
Week 8	Design Phase	Design the database schema; Validate database design to ensure alignment with project requirements.
Week 9	Implementation Phase	Plan implementation of selected features; Set up the development environment and tools.
Week 10	Implementation Phase	Begin coding core system features; Integrate the database and gather initial client reviews and feedback.
Week 11	Implementation Phase	Address client feedback; Refine system, implement additional features, and prepare for the final presentation.
Week 12	Testing & Deployment	Conduct final testing and user acceptance testing (UAT); Prepare documentation, deploy the system, and review.

References

- Azal University. (2020). *Software Development Life Cycle* [PDF file]. Retrieved from https://www.auhd.edu.ye/upfiles/elibrary/Azal2020-01-22-12-35-12-90529.pdf
- Image from: Devox Software. (n.d.). *Software Development Life Cycle* [Blog post]. Retrieved from https://devoxsoftware.com/blog/software-development-lifecycle/
- SDLC Content contributions: ChatGPT-4 by OpenAI.

Thank You!