System Analysis and Design (SAD)

Module 1: Overview system analysis/design concepts

Understand Data processing system

Data processing system

- Definition
- Function
- Importance

Definition:

A data processing system is a set of software, hardware, and procedures used to process and manipulate data in order to convert it into useful information. These systems are used to automate repetitive tasks and to extract meaningful insights from large volumes of data.

Function:

The primary function of a data processing system is to transform raw data into meaningful information that can be used for decision-making, reporting, or analysis. This involves a series of operations such as validation, sorting, filtering, transformation, and aggregation of data. The operations performed by data processing systems depend on the specific requirements of the organization or the project, but typically involve one or more of the following:

Data input: This involves collecting and entering data into the system, which can be done manually or through automated processes.

Data storage: This involves storing data in a structured manner, such as in databases or files, so that it can be easily retrieved and processed.

Data processing: This involves transforming the input data into a desired format or structure. This can include sorting, filtering, aggregating, and calculating data.

Data output: This involves presenting the processed data in a meaningful way, such as through reports, dashboards, or visualizations.

Data analysis: This involves examining the processed data to identify patterns, trends, and insights that can be used for decision-making or further analysis.

Importance:

Data processing systems are critical for organizations that rely on data to drive their business operations. The importance of data processing systems can be understood by the following points:

Efficiency: Data processing systems automate repetitive tasks, which reduces the time and effort required to process data manually. This leads to increased efficiency and productivity.

Accuracy: Data processing systems ensure that data is processed accurately and consistently, reducing the risk of errors and inconsistencies.

Timeliness: Data processing systems can process data in real-time or near real-time, enabling organizations to make timely decisions based on the latest data.

Insights: Data processing systems can extract meaningful insights from large volumes of data, which can be used to identify opportunities, mitigate risks, and optimize operations.

Competitive advantage: Data processing systems enable organizations to leverage their data assets to gain a competitive advantage, by improving their products and services, reducing costs, and identifying new revenue streams.

In conclusion, data processing systems are critical tools for organizations that rely on data to drive their business operations. These systems transform raw data into meaningful information that can be used for decision-making, reporting, or analysis. By automating repetitive tasks, ensuring accuracy, providing timely insights, and enabling competitive advantage, data processing systems play a crucial role in modern business operations.

Explain Management Information System

Management information system

- Definition
- Function
- Importance

Definition:

A Management Information System (MIS) is a computerized system that collects, processes, and reports information to support the decision-making process of an organization's management. The system uses data from various sources, such as internal and external databases, to provide timely and accurate information to decision-makers.

Function:

The primary function of an MIS is to provide relevant and timely information to managers to support their decision-making process. This involves collecting data from various sources, processing it to generate meaningful insights, and presenting the insights in a user-friendly format. The specific functions of an MIS include:

Data collection: An MIS collects data from various internal and external sources, such as transactional databases, sensors, and social media.

Data processing: The collected data is processed to generate meaningful insights, which can be used to support decision-making. This involves various techniques such as data analysis, data mining, and data visualization.

Information dissemination: The processed data is presented to the users in a user-friendly format, such as reports, dashboards, or visualizations. This helps managers to understand the insights and make informed decisions quickly and easily.

Decision support: An MIS provides decision support tools to help managers make informed decisions. This can include tools such as simulations, predictive analytics, and optimization models.

Importance:

The importance of an MIS can be understood by the following points:

Strategic planning: An MIS provides the information required for strategic planning by providing insights into market trends, customer behaviour, and competitors' activities.

Decision-making: An MIS supports decision-making by providing timely and accurate information to managers, which enables them to make informed decisions.

Efficiency: An MIS improves the efficiency of an organization by automating repetitive tasks and providing decision support tools to managers.

Performance monitoring: An MIS provides information on the performance of an organization, which enables managers to monitor and control operations effectively.

Competitive advantage: An MIS can provide a competitive advantage by enabling organizations to make informed decisions based on real-time data.

In summary, a Management Information System is a computerized system that collects, processes, and reports information to support the decision-making process of an organization's management. Its functions include data collection, data processing, information dissemination, and decision support. The importance of an MIS lies in its ability to support strategic planning, decision-making, efficiency, performance monitoring, and competitive advantage.

Describe Data support system

Decision support systems

- Definition
- Function
- Importance

Definition:

A Decision Support System (DSS) is a computer-based information system that supports decision-making activities of individuals, groups, or organizations. It is designed to help decision-makers make effective decisions by providing access to relevant data and tools to analyse it.

Function:

The primary function of a DSS is to provide decision-makers with the information and tools they need to make effective decisions. This involves collecting and analysing data from various sources, such as databases, spreadsheets, and external sources. The specific functions of a DSS include:

Data collection: A DSS collects data from various sources, such as databases, spreadsheets, and external sources.

Data analysis: The collected data is analysed using various techniques, such as data mining, statistical analysis, and artificial intelligence.

Decision modelling: A DSS provides decision modelling tools, such as what-if analysis, optimization, and simulation, to help decision-makers explore different scenarios and outcomes.

Information presentation: The analysed data and decision models are presented to the decision-makers in a user-friendly format, such as reports, graphs, and charts.

Importance:

The importance of a DSS can be understood by the following points:

Improved decision-making: A DSS provides decision-makers with access to relevant and timely information, enabling them to make more informed decisions.

Increased efficiency: A DSS can automate repetitive tasks, such as data collection and analysis, thereby increasing efficiency and productivity.

Enhanced problem-solving: A DSS provides tools to help decision-makers explore different scenarios and outcomes, thereby enhancing problem-solving capabilities.

Competitive advantage: A DSS can provide a competitive advantage by enabling organizations to make faster, more informed decisions based on real-time data.

In summary, a Decision Support System is a computer-based information system that supports decision-making activities of individuals, groups, or organizations. Its functions include data collection, data analysis, decision modelling, and information presentation. The importance of a DSS lies in its ability to improve decision-making, increase efficiency, enhance problem-solving, and provide a competitive advantage.

Understand artificial intelligence

Artificial intelligence

- Definition
- Function
- Importance

Definition:

Artificial Intelligence (AI) is a field of computer science and engineering that focuses on the development of intelligent machines that can perform tasks that would typically require human intelligence. AI systems are designed to learn from data and experience, make predictions or recommendations, and take actions based on those predictions.

Function:

The primary function of AI is to perform tasks that would typically require human intelligence. This involves developing algorithms and models that can learn from data and experience and use that learning to make predictions or recommendations. The specific functions of AI include:

Machine learning: AI systems use machine learning algorithms to learn from data and experience, and use that learning to make predictions or recommendations.

Natural language processing: AI systems use natural language processing algorithms to understand and interpret human language.

Computer vision: AI systems use computer vision algorithms to interpret and understand visual data, such as images and videos.

Robotics: AI systems are used in robotics to enable machines to perform tasks that would typically require human intelligence.

Importance:

The importance of AI can be understood by the following points:

Automation: AI enables automation of tasks that were previously performed by humans, thereby increasing efficiency and productivity.

Decision-making: AI can provide decision support by analyzing data and making predictions or recommendations, thereby improving decision-making.

Personalization: AI can personalize user experiences by analyzing user data and providing personalized recommendations and content.

Innovation: AI enables the development of new products and services that were not previously possible, such as self-driving cars and virtual assistants.

Competitive advantage: AI can provide a competitive advantage by enabling organizations to make faster and more informed decisions, automate tasks, and innovate.

In summary, Artificial Intelligence is a field of computer science and engineering that focuses on the development of intelligent machines that can perform tasks that would typically require human intelligence. Its functions include machine learning, natural language processing, computer vision, and robotics. The importance of AI lies in its ability to automate tasks, improve decision-making, personalize user experiences, drive innovation, and provide a competitive advantage.

Identify system analysis and design

System analysis/design identify

- Definition
- Function
- Importance

Definition:

System Analysis and Design is a process of developing information systems that involves examining, analyzing, and designing the system to meet the specific needs and requirements of an organization. It involves understanding the current system, identifying areas of improvement, and designing a new system that meets the needs of the organization.

Function:

The primary function of System Analysis and Design is to develop information systems that meet the needs and requirements of an organization. This involves understanding the current system, identifying areas of improvement, and designing a new system. The specific functions of System Analysis and Design include:

Requirement gathering: System Analysis and Design involves gathering requirements from stakeholders to understand the needs of the organization.

System analysis: The current system is analyzed to identify areas of improvement, inefficiencies, and potential areas of automation.

System design: Based on the requirements and analysis, a new system is designed to meet the specific needs and requirements of the organization.

Implementation: The new system is developed and implemented based on the design.

Testing: The new system is tested to ensure that it meets the requirements and functions properly.

Maintenance: Once the new system is implemented, it requires ongoing maintenance and support to ensure that it continues to function properly.

Importance:

The importance of System Analysis and Design can be understood by the following points:

Improved efficiency: System Analysis and Design can identify areas of improvement in the current system and design a new system that is more efficient and effective.

Cost savings: System Analysis and Design can identify areas of automation that can reduce costs and improve productivity.

Better decision-making: System Analysis and Design can provide better data and information to support decision-making.

Improved customer satisfaction: System Analysis and Design can design a new system that meets the specific needs and requirements of the organization, thereby improving customer satisfaction.

Competitive advantage: System Analysis and Design can provide a competitive advantage by developing a new system that is more efficient, effective, and meets the specific needs of the organization.

In summary, System Analysis and Design is a process of developing information systems that involves examining, analyzing, and designing the system to meet the specific needs and requirements of an organization. Its functions include requirement gathering, system analysis, system design, implementation, testing, and maintenance. The importance of System Analysis and Design lies in its ability to improve efficiency, reduce costs, support decision-making, improve customer satisfaction, and provide a competitive advantage.

Familiarize with system development and flow

- System analysis
- System design
- Program design
- Programming
- Program test
- System test
- Running test

System analysis:

System analysis is the process of studying the current system to identify areas of improvement and determine the requirements of the new system. It involves gathering information about the current system, analyzing its strengths and weaknesses, and identifying areas where the new system can improve efficiency and effectiveness.

System design:

System design is the process of creating a plan or blueprint for the new system based on the requirements gathered during system analysis. This plan includes system architecture, system components, interfaces, and data structures.

Program design:

Program design involves breaking down the system design into smaller components and designing algorithms and data structures for each component. The design must be detailed enough to provide a roadmap for programmers to follow when writing the code.

Programming:

Programming involves writing code in a programming language based on the program design. Programmers use tools like integrated development environments (IDEs) and text editors to write and test the code.

Program test:

Program testing is the process of testing individual components or modules of the program to ensure that they work correctly. This testing is usually done using automated testing tools or by writing test scripts.

System test:

System testing involves testing the entire system to ensure that it meets the requirements specified in the system design. This testing is usually done after the program testing is complete.

Running test:

Running tests involve testing the system in a live environment to ensure that it works as expected. This testing is usually done after the system testing is complete and involves testing the system with real users and data.

Overall, system development and flow involve a series of steps that build upon one another, from system analysis to running tests. Each step is critical to ensure that the final system meets the needs and requirements of the organization.

Prepare major system development models

- Water fall model
- Spiral model
- Others

Waterfall model:

The Waterfall model is a linear, sequential approach to system development. In this model, each phase of the development process must be completed before moving on to the next phase. The phases include requirements gathering and analysis, system design, implementation, testing, and maintenance. The Waterfall model is best suited for projects with well-defined requirements and a predictable development process.

Spiral model:

The Spiral model is a more flexible approach to system development that is based on the Waterfall model. The Spiral model includes four phases: planning, risk analysis, engineering, and evaluation. Each phase is followed by a review process that determines whether to move on to the next phase or return to a previous phase. The Spiral model is best suited for projects with high levels of complexity or uncertainty.

Others: Prototype model,

Agile model:

The Agile model is an iterative and incremental approach to system development that emphasizes collaboration, flexibility, and customer satisfaction. Agile development is based on the principles outlined in the Agile Manifesto, which values individuals and interactions, working software, customer collaboration, and responding to change. Agile development involves working in short iterations or sprints, with a focus on delivering working software quickly and responding to feedback from customers.

Rapid Application Development (RAD) model:

The Rapid Application Development (RAD) model is a fast-paced, iterative approach to system development that emphasizes prototyping and user involvement. The RAD model includes four phases: requirements planning, user design, construction, and cutover. The RAD model is best suited for projects with high user involvement and a need for rapid development.

DevOps model:

The DevOps model is an approach to system development that emphasizes collaboration and communication between development teams and operations teams. DevOps aims to streamline the development and deployment process by breaking down silos between teams and automating repetitive tasks. The DevOps model is best suited for projects with a focus on continuous delivery and deployment.

Overall, choosing the right system development model depends on the specific needs and requirements of the project. Each model has its advantages and disadvantages, and it's important to carefully evaluate each option before deciding which one to use.

Module 2: Manage Analysis/Design activities

Identify problem

- Project initiation, Specific sign of problem

During the project initiation phase, it is important to identify potential problems and risks that could impact the success of the project.

Some specific signs of problems during project initiation include:

- **Undefined goals and objectives:** If the project goals and objectives are unclear or undefined, it can lead to confusion and misalignment among project stakeholders.
- **Lack of stakeholder buy-in:** If key stakeholders are not committed to the project, it can lead to resistance and opposition that can delay or derail the project.
- **Inadequate resource allocation**: If resources such as budget, personnel, and technology are not allocated appropriately, it can lead to delays and compromises in the quality of the project deliverables.
- **Unclear project scope**: If the project scope is not clearly defined, it can lead to scope creep, which is the addition of new requirements or features that were not originally planned for.
- **Lack of project management skills:** If the project team does not have the necessary skills and experience to manage the project, it can lead to ineffective project management and poor decision-making.

Identifying these signs of problems during project initiation is important because it allows project managers to take corrective action early in the project lifecycle. This can help to mitigate risks and ensure that the project is completed successfully within the allotted timeframe and budget.

Identify opportunity for improvement

- Possibilities for improvement, criteria for selection of system projects

Identifying opportunities for improvement involves assessing the current systems and processes used within an organization and identifying areas where improvements can be made. This can be done through a variety of methods, including process mapping, surveys, and benchmarking against industry standards.

Criteria for selecting system projects typically include factors such as:

Alignment with organizational goals: Projects should align with the overall strategic goals of the organization and support its mission and values.

Potential benefits: Projects should provide tangible benefits to the organization, such as increased efficiency, improved customer satisfaction, or increased revenue.

Feasibility: Projects should be technically feasible and should not require excessive resources or time to complete.

Cost-effectiveness: Projects should be cost-effective and should provide a positive return on investment.

Risk management: Projects should be evaluated for potential risks and steps should be taken to mitigate those risks.

Once potential opportunities for improvement have been identified and projects have been selected, a detailed project plan should be developed that includes timelines, resource requirements, and a budget. Effective project management is critical to ensuring that the project is completed successfully and that the expected benefits are realized.

Determine feasibility

- Objectives, Resources
- Types of feasibility; Technical, Economical, Operational
- Making judgement on feasibility

Determining feasibility is a critical step in the system analysis and design process. It involves evaluating the proposed project to determine whether it is technically, economically, and operationally feasible. Here are some details on the different types of feasibility and the process of making a judgment on feasibility:

Objectives: The objectives of the proposed system should be clearly defined and aligned with the organization's strategic goals.

Resources: The resources required to develop and implement the proposed system should be identified, including personnel, hardware, software, and budget.

Types of feasibility: The following types of feasibility should be considered:

> **Technical feasibility:** This involves evaluating whether the proposed system can be developed and implemented using the available technology and resources.
>
> **Economic feasibility:** This involves evaluating the costs and benefits of the proposed system to determine whether it is financially viable and whether the expected benefits outweigh the costs.
>
> **Operational feasibility:** This involves evaluating whether the proposed system is practical and feasible to implement in the organization's existing environment, including the organizational structure, culture, and processes.

Making a judgment on feasibility: Based on the analysis of the objectives, resources, and types of feasibility, a judgment can be made on the feasibility of the proposed system. This judgment should consider the risks, benefits, and costs of the proposed system and should be based on a thorough analysis of the proposed system's potential impact on the organization. If the proposed system is deemed feasible, the system analysis and design process can proceed to the next phase. If the proposed system is deemed not feasible, alternatives should be explored or the project should be terminated.

Plan / Control activities

- Time estimation, Gantt charts, CPM and PERT

Planning and controlling activities is an essential part of the system analysis and design process. It involves developing a project plan that outlines the tasks, timelines, resources, and budget required to complete the project successfully. Here are some key techniques used for planning and controlling activities:

Time estimation: Time estimation involves determining the time required to complete each task in the project plan. This is typically done using historical data, expert opinions, and other methods to estimate how long each task will take.

Gantt charts: Gantt charts are graphical representations of the project plan that show the timeline for each task and how they are dependent on each other. They are useful for visualizing the project schedule, identifying critical paths, and tracking progress.

CPM (Critical Path Method): CPM is a project management technique that identifies the critical path through the project plan. The critical path is the sequence of tasks that must be completed on time to ensure the project is completed on schedule. CPM can be used to determine the earliest possible completion time for a project and to identify tasks that can be delayed without affecting the overall project timeline.

PERT (Program Evaluation and Review Technique): PERT is a project management technique that is like CPM but takes into account the uncertainties and risks associated with each task. PERT uses statistical methods to estimate the time required for each task, and it can be used to identify the probability of completing the project within a given timeframe.

Overall, effective planning and control are crucial to ensuring that the project is completed on time and within budget. Techniques such as time estimation, Gantt charts, CPM, and PERT can help project managers to develop a realistic project plan, identify potential risks, and track progress to ensure that the project is completed successfully.

Manage analysis / design activities

- Communication strategies for team management,
- Productivity goal, motivation
- Project failures

Managing analysis and design activities involves effective communication strategies, setting productivity goals, motivating team members, and addressing project failures. Here are some key points to consider:

Communication strategies for team management: Effective communication is critical to the success of any project. Team members should be clear on their roles and responsibilities, project goals, timelines, and deliverables. Communication tools such as project management software, email, and regular team meetings can help to keep everyone on the same page.

Productivity goals and motivation: Setting clear productivity goals and incentives can help to motivate team members and improve their performance. This can include setting deadlines, providing rewards for completing tasks on time, and recognizing and rewarding outstanding performance.

Project failures: Project failures can occur due to a variety of reasons, including poor planning, miscommunication, lack of resources, and unforeseen obstacles. It's important to identify potential risks and plan for contingencies to avoid project failures. If a project does fail, it's important to conduct a post-mortem analysis to determine what went wrong and how to avoid similar failures in the future.

Overall, managing analysis and design activities requires effective communication, clear productivity goals, motivation, and a plan for addressing potential risks and failures. By implementing these strategies, project managers can ensure that their teams are productive, motivated, and successful in delivering high-quality systems.

Module 3: Collect data

Design sample

- Need of sampling
- Types of sampling; Convenience, Purposive, Random
- Sample size

Designing a sample is an important aspect of data collection in system analysis and design. Here are some key points to consider:

Need of sampling: Sampling is a method of selecting a subset of the population to be studied. This is done when it is not feasible or practical to study the entire population. Sampling can help to save time and resources while still providing accurate information.

Types of sampling: There are several types of sampling methods including convenience sampling, purposive sampling, and random sampling. ***Convenience sampling*** involves selecting individuals who are easily accessible. ***Purposive sampling*** involves selecting individuals based on specific criteria. ***Random sampling*** involves selecting individuals at random from the population.

Sample size: The sample size is an important consideration when designing a sample. A larger sample size can provide more accurate results, but it can also be more time-consuming and expensive to collect. The sample size should be large enough to provide a representative sample of the population, but small enough to be practical.

Overall, designing a sample involves selecting a sampling method that is appropriate for the research question and population of interest, determining the sample size, and selecting individuals to participate in the study. By designing a representative sample, system analysts and designers can obtain accurate and reliable data to inform their work.

Conduct interview

- Planning interview, Objectives of interview
- Types of question; Close & Open ended, Probe
- Question pitfalls

Conducting interviews is a common method of data collection in system analysis and design. Here are some key points to consider:

Planning interview: Planning an interview involves determining the objectives of the interview, identifying the individuals to be interviewed, and deciding on the format and structure of the interview.

Objectives of interview: The objectives of the interview should be clear and well-defined. These may include gathering information about the system being analyzed, understanding user needs and requirements, or evaluating the effectiveness of a system.

Types of questions: There are two main types of questions that can be used in an interview: closed-ended and open-ended. Closed-ended questions require a specific response and are useful for gathering quantitative data. Open-ended questions allow the interviewee to provide a more detailed response and are useful for gathering qualitative data. Probing questions can also be used to encourage the interviewee to provide more detailed responses.

Question pitfalls: It is important to be aware of potential pitfalls in interview questions, such as leading questions that bias the response, double-barrelled questions that ask about two things at once, or complex questions that may confuse the interviewee.

Overall, conducting interviews can be a valuable method of data collection in system analysis and design. By carefully planning the interview, using appropriate types of questions, and avoiding question pitfalls, system analysts and designers can gather accurate and reliable data to inform their work.

Administer questionnaires

- Planning questionnaire administration
- Scales in questionnaire
- Validity, Reliability
- Clustering of questions of similar content

Administering questionnaires is another common method of data collection in system analysis and design. Here are some key points to consider:

Planning questionnaire administration: Planning questionnaire administration involves identifying the target population, determining the purpose of the questionnaire, and designing the questionnaire itself. It is important to ensure that the questions are clear and unambiguous and that they address the research objectives.

Scales in questionnaire: Scales can be used in questionnaires to measure variables such as attitudes, beliefs, or opinions. Common types of scales include Likert scales, semantic differential scales, and visual analogue scales. It is important to select the appropriate type of scale for the research objectives and to ensure that the scale is used consistently throughout the questionnaire.

Validity and reliability: Validity refers to the extent to which a questionnaire measures what it is intended to measure. Reliability refers to the consistency of the questionnaire over time and across different groups of respondents. To ensure validity and reliability, it is important to pilot test the questionnaire and to use established measures whenever possible.

Clustering of questions of similar content: Clustering questions of similar content can help to ensure that the questionnaire is organized and easy to complete. Clustering can also help to identify patterns in the data and to analyze the results more effectively.

Overall, administering questionnaires can be an effective method of data collection in system analysis and design. By carefully planning the questionnaire administration, using appropriate scales, ensuring validity and reliability, and clustering questions of similar content, system analysts and designers can gather accurate and reliable data to inform their work.

Observe decision making activity

- Time event sampling, body language of decision maker

Observing decision-making activities is another method of data collection in system analysis and design. Here are some key points to consider:

Time event sampling: Time event sampling involves observing the decision-making process and recording what is happening at predetermined intervals. For example, an observer might record what the decision maker is doing every 5 minutes. Time event sampling can help to identify patterns in the decision-making process and to identify areas where improvements can be made.

Body language of decision maker: Observing the body language of the decision maker can provide important insights into their thought processes and emotional state. For example, a decision maker who is leaning forward and making eye contact may be more engaged in the decision-making process than one who is slouching and avoiding eye contact. Observing body language can help to identify the decision maker's level of interest, confidence, and engagement in the decision-making process.

Overall, observing decision-making activities can be an effective method of data collection in system analysis and design. By using time event sampling and observing the body language of the decision maker, system analysts and designers can gather valuable insights into the decision-making process and use this information to inform their work.

Observe office environment

- Filmic & Organisational elements, STROBE (STRuctured Observation of the Environment)

Observing the office environment can be an important method of data collection in system analysis and design. Here are some key points to consider:

Filmic and organizational elements: Observing the office environment involves paying attention to both the physical and organizational elements of the workplace. Physical elements may include the layout of the space, the furniture and equipment used, and the lighting and sound quality. Organizational elements may include the communication and collaboration processes used, the level of employee engagement and satisfaction, and the overall culture and values of the organization. By observing these elements, system analysts and designers can gain a better understanding of how the office environment supports or hinders the work being done.

STROBE: STROBE (Structured Observation of the Environment) is a method of observing the office environment that involves using a structured framework to collect data. This framework includes a

set of predefined categories and subcategories that cover different aspects of the office environment, such as the physical layout, the communication processes, and the level of collaboration among employees. Using the STROBE framework can help to ensure that data is collected consistently and systematically, making it easier to analyze and interpret.

Overall, observing the office environment can be a valuable method of data collection in system analysis and design. By paying attention to both the physical and organizational elements of the workplace and using a structured framework such as STROBE, system analysts and designers can gather important insights into how the office environment supports or hinders the work being done, and use this information to inform their work.

Develop prototype / model

- Types of Prototypes; Patched up, Non-operational, First -of-a-series, selected featured
- Guidelines for developing a prototypes
- Modifications on user's interface
- Advantage and disadvantage of prototypes

A prototype is a preliminary model or sample of a system or product that is being developed. Its main purpose is to allow users and developers to explore and test the system's functionality, usability, and design before the final version is completed. The following are some key points related to developing a prototype/model:

Types of prototypes: There are several types of prototypes, including patched-up, non-operational, first-of-a-series, and selected featured. **A patched-up** prototype is a quick and simple model that can be easily put together. **A non-operational** prototype is one that has the look and feel of the final product but does not function. A **first-of-a-series** prototype is a more refined version of a non-operational prototype, and it is used to test and refine the system's functionality. A **selected featured** prototype is a combination of the above prototypes and is used to test specific features of the system.

Guidelines for developing a prototype: The following are some guidelines for developing a prototype: identify the purpose and scope of the prototype, select the appropriate type of prototype, involve users in the design and testing process, use existing software and hardware where possible, and document the prototype development process.

Modifications on user's interface: One of the key benefits of a prototype is that it allows developers to make modifications to the user interface based on feedback from users. This can help to ensure that the final product is user-friendly and meets the needs of the intended audience.

Advantage and disadvantage of prototypes: The main advantage of a prototype is that it allows developers to test and refine the system's functionality and design before the final version is completed. This can help to identify and address potential issues early in the development process, which can save time and resources. However, prototypes can be time-consuming and expensive to develop, and there is a risk that users may become attached to certain features of the prototype that cannot be included in the final product.

Module 4: Analyze /Process data

Develop data flow diagram

- Basic symbols of data flow, conventions

- Context diagram
- Approaches in data flow diagram
- Labelling

A **data flow diagram (DFD)** is a graphical representation of the flow of data through a system. The development of a DFD is an important step in system analysis and design. The following are some of the key topics that need to be covered while developing a DFD:

Basic symbols of data flow: A DFD consists of four basic symbols: circles, squares, arrows, and lines. Circles represent entities or processes that are involved in the system, squares represent data stores where the data is stored, arrows represent data flows, and lines represent the boundaries of the system.

Conventions: There are certain conventions that need to be followed while developing a DFD, such as each process should have at least one input and one output, a data store should have at least one input or output, and data should flow from higher-level processes to lower-level processes.

Context diagram: A context diagram is a high-level view of the system that shows the system as a single process with inputs and outputs. It is used to identify the system boundaries and the external entities that interact with the system.

Approaches in data flow diagram: There are two approaches to developing a DFD: top-down and bottom-up. In the top-down approach, the system is divided into subsystems and each subsystem is further analyzed until the required level of detail is reached. In the bottom-up approach, the subsystems are analyzed first and then combined to form the overall system.

Labelling: Each data flow, process, and data store should be labelled appropriately to clearly identify its purpose and function within the system.

Overall, a well-designed DFD can help in understanding the flow of data through the system, identifying bottlenecks, and improving the overall efficiency of the system.

Develop data dictionary

- Data in data dictionary,
- Cataloguing,
- Data processes, data flow, data store, data structure, data elements
- Steps in compiling data dictionaries

A data dictionary is a central repository of information about data that is used within an organization. It provides a description of the data elements, data structures, data flows, and data stores in a system. The following are the steps involved in compiling a data dictionary:

Data in data dictionary: The data dictionary should include information about all data elements used in the system. This information should include data element names, descriptions, data types, lengths, formats, and other properties.

Cataloguing: Cataloguing is the process of organizing and grouping data elements in a logical way to make it easier to manage and maintain the data dictionary. This involves creating categories and subcategories of data elements.

Data processes: Data processes are the steps involved in creating, updating, retrieving, or deleting data in the system. The data dictionary should include information about all data processes and their associated data elements.

Data flow: Data flow describes the movement of data through the system. The data dictionary should include information about all data flows and their associated data elements.

Data store: Data store is where data is stored in the system. The data dictionary should include information about all data stores and their associated data elements.

Data structure: Data structure refers to the way data is organized within a data store. The data dictionary should include information about all data structures used in the system.

Data elements: Data elements are the basic units of information used in the system. The data dictionary should include detailed information about all data elements used in the system.

Steps in compiling data dictionaries: The process of compiling a data dictionary involves the following steps:

- Identify all data elements used in the system
- Catalogue data elements into logical categories and subcategories
- Define data elements and their properties, such as data type, length, format, and other attributes
- Document data flows, data stores, and data structures
- Validate and verify the accuracy of the data dictionary
- Update the data dictionary as changes are made to the system.

By following these steps, a comprehensive data dictionary can be created that provides a detailed understanding of the data used in a system, which can be useful for system analysis, design, and maintenance.

Analyze structured decision system

- Information required for structured decision,
- Structured English
- Decision table and trees
- Completeness and accuracy

Information required for structured decision:

Structured decision-making involves using a defined process to make decisions based on clear and well-defined criteria. To analyze a structured decision system, it is essential to first identify the information that is required for the decision-making process. This information can be in the form of data, reports, documents, or any other relevant material. Understanding the information requirements is crucial to ensure that the decision-making process is efficient and effective.

Structured English:

Structured English is a method for specifying the steps required to solve a problem or perform a task in a structured decision system. It is a type of structured programming language that is used to

describe the logic of the decision process in a clear and concise way. Structured English is designed to be easy to read and understand, making it a useful tool for communicating complex decision-making processes to stakeholders.

Decision table and trees:

Decision tables and trees are two common tools used in structured decision systems. A decision table is a matrix that outlines the criteria for a decision and the corresponding actions to be taken. It is useful for analyzing complex decision-making processes with multiple variables. Decision trees, on the other hand, are graphical representations of a decision-making process that map out all possible outcomes and decisions. Decision trees are useful for analyzing simple decision-making processes with a few variables.

Completeness and accuracy:

Analyzing the completeness and accuracy of a structured decision system is crucial to ensure that the system is reliable and effective. Completeness refers to the extent to which all possible scenarios have been considered in the decision-making process. Accuracy refers to the correctness of the decision-making process and the results it produces. Ensuring that the structured decision system is both complete and accurate is essential to ensure that the system meets the requirements of the stakeholders and produces reliable and trustworthy results.

Analyze semi-structured decision system

- Risk in decision making,
- Types of decision; Analytic, Heuristic and Intelligence
- Design and choice
- Multiple criteria for decision making; Tradeoff, Weighting, Sequential elimination, and Goal programming

Analyzing semi-structured decision systems involves understanding the different types of decisions, designing decision-making processes, and considering multiple criteria for decision making. Here are some explanations of the topics:

Risk in decision making: This involves identifying the possible risks or uncertainties associated with a decision and evaluating their potential impact. Decision makers can use risk management techniques to assess and mitigate these risks.

Types of decision: There are three main types of decision-making processes: analytic, heuristic, and intelligence. Analytic decision making is based on a systematic approach that involves identifying and evaluating all possible alternatives. Heuristic decision making relies on rules of thumb, intuition, and past experiences. Intelligence decision making involves using creativity and insight to arrive at a solution.

Design and choice: This involve designing a decision-making process that is appropriate for the specific decision at hand. This can include identifying decision criteria, developing alternatives, and evaluating the alternatives against the criteria.

Multiple criteria for decision making: In some cases, there may be multiple criteria that need to be considered when making a decision. **Tradeoff** analysis involves evaluating the different criteria and making decisions that optimize the tradeoff between them. **Weighting** involves assigning weights to

different criteria to reflect their relative importance. **Sequential elimination** involves evaluating the criteria one at a time and eliminating alternatives that do not meet each criterion. **Goal programming** involves identifying a set of goals and constraints and developing a decision-making process that achieves the goals while satisfying the constraints.

Prepare/Present system proposal

- Inventorying and evaluation of computer hardware
- Workload estimation
- Acquisition of computer equipment, Vendor support
- Software evaluation
- Benefit/Cost, Payback, Cash flow analysis, trends
- Organising system proposal and its content

Preparing and presenting a system proposal involves several steps that help in evaluating the feasibility of the proposed system and presenting its benefits to stakeholders. These steps include:

Inventorying and evaluation of computer hardware: This involves taking stock of the existing computer hardware and evaluating its capacity and suitability for the proposed system.

Workload estimation: This involves estimating the expected workload on the proposed system to determine the appropriate hardware and software requirements.

Acquisition of computer equipment, Vendor support: This involves selecting vendors for the required computer equipment and negotiating support agreements with them.

Software evaluation: This involves evaluating the available software options and selecting the most appropriate one for the proposed system.

Benefit/Cost, Payback, Cash flow analysis, trends: This involves analyzing the financial feasibility of the proposed system by estimating its benefits and costs, calculating payback period and cash flow, and considering market trends.

Organizing system proposal and its content: This involves organizing the proposal into sections that cover the objectives, requirements, scope, methodology, timelines, resources, benefits, and costs of the proposed system.

Overall, the system proposal should provide a clear understanding of the proposed system and its benefits, including its impact on the organization's operations, customers, and stakeholders.

Module 5: Design essential

Design output

- Objectives of effective output
- Output technology
- Avoiding biases
- Functional and stylistic/Aesthetic attributes

Designing output is an essential part of any system design process. Here are some details on the different aspects of designing output:

Objectives of effective output: The main objective of designing effective output is to provide useful information to the users in a clear, concise, and understandable way. The output should be accurate, timely, relevant, and complete. The users should be able to easily interpret and use the information provided by the output.

Output technology: There are different output technologies available, such as paper-based output, electronic output, audio output, video output, etc. The technology used for output should be selected based on the user requirements, type of information to be presented, and the environment in which it will be used.

Avoiding biases: The output should be designed in a way that avoids biases, such as cultural, gender, age, or disability biases. The output should be designed to be accessible to all users, regardless of their background or abilities.

Functional and stylistic/aesthetic attributes: The output should have both functional and stylistic/aesthetic attributes. Functional attributes include the clarity and accuracy of the information provided, while stylistic/aesthetic attributes refer to the visual appeal of the output. The output should be designed in a way that is visually appealing and easy to read, while also conveying the necessary information.

Overall, designing effective output requires careful consideration of user requirements, output technology, and the functional and aesthetic attributes of the output. The output should be designed to provide useful information in a clear, concise, and understandable way, while also being accessible to all users and avoiding biases.

Design input

- Guidelines for form design,
- Screen design; Icons and Colours

Design input refers to the process of designing the input methods for a computer system. This includes designing forms, screens, and other user interfaces that users interact with to input data into the system. The goal of design input is to create interfaces that are intuitive, efficient, and effective for users.

Guidelines for form design include creating forms that are easy to understand and use, with clear labels and instructions. Forms should be designed to minimize errors, with validation checks and error messages when necessary. The layout of the form should be organized logically, with related fields grouped together.

Screen design involves designing the visual layout of computer screens and the elements on those screens, such as buttons, menus, and text fields. **Icons and colours** can be used to help users navigate the system more easily and understand the meaning of different buttons and elements. For example, a red button might indicate that an action is critical or irreversible, while a green button might indicate that an action is safe and can be undone.

Overall, the goal of design input is to create interfaces that are easy to use and that help users input data accurately and efficiently. By following best practices and design guidelines, designers can create interfaces that are effective and meet the needs of users.

Design file/database

- Conventional files and databases
- Data concepts
- File and database organisation
- Normalisation

Designing a file/database involves creating an organized structure to store and retrieve data efficiently. The following are the key aspects of designing a file/database:

Conventional files and databases: Conventional files are traditional storage systems used to store and retrieve data. They are usually created using tools like Microsoft Excel or Google Sheets. In contrast, databases are more advanced and use a structured approach to store and manage data. They are designed to efficiently store, retrieve, and manage large amounts of data. Databases can be relational, hierarchical, or object-oriented.

Data concepts: Data concepts are the building blocks of a database. They include entities, attributes, and relationships. Entities are objects or concepts that are represented in a database, while attributes are the characteristics of those entities. Relationships describe how entities are related to one another.

File and database organisation: File and database organisation refers to the way data is stored and retrieved from a file or database. It includes file structures, indexing methods, and access methods. File structures determine how data is organised within a file, while indexing methods and access methods provide efficient ways of locating and retrieving data.

Normalisation: Normalisation is the process of organising data in a database to eliminate redundancy and improve data integrity. It involves breaking down a database into smaller, more manageable tables and establishing relationships between those tables. Normalisation ensures that data is stored in a consistent and efficient manner, reducing the risk of errors and inconsistencies in the data.

Design user interface

- Types of user interface; Natural language, Question-and-Answer, Menus, input/output, Command language, Direct manipulation and the Mouse

User interface design refers to the process of designing interfaces in software, applications, or devices that facilitate interaction between users and the system. It involves the arrangement of graphical elements, input controls, navigation mechanisms, and other features to enable efficient and intuitive communication with the system.

Here are the different types of user interface:

Natural language interface: This type of interface allows users to interact with the system using natural language, such as spoken or written commands. The system interprets the input and provides responses in a human-like manner.

Question-and-answer interface: In this type of interface, the system asks users a series of questions to elicit the information it needs to complete a task. The users respond with answers, and the system processes the responses to produce the desired output.

Menus interface: This interface provides users with a set of options or commands displayed in a hierarchical menu structure. Users select options from the menu to initiate actions or navigate through the system.

Input/output interface: This interface allows users to input data into the system and receive output from it. Examples include text boxes, drop-down menus, and radio buttons.

Command language interface: This interface requires users to type in commands in a specific syntax to initiate actions or manipulate data. It is commonly used in command-line interfaces for operating systems or programming languages.

Direct manipulation interface: This type of interface allows users to manipulate graphical objects directly using gestures or movements. Examples include drag-and-drop, pinch-to-zoom, and swipe gestures.

Mouse interface: This interface involves using a mouse or other pointing device to navigate and interact with the system. It is commonly used in graphical user interfaces (GUIs) for desktop applications.

Each type of user interface has its advantages and disadvantages depending on the context of use, user needs, and system requirements. Therefore, it is essential to consider these factors when designing a user interface to ensure its effectiveness and usability.

Design data entry procedures

- Objectives
- Purpose of coding
- Effective/Accurate coding
- Bottleneck in data entry

Design data entry procedures involves planning and designing the process of how data will be entered into a computer system. There are four important aspects of designing data entry procedures:

Objectives: The objectives are the goals that we want to achieve by entering the data into the system. For example, we might want to track inventory levels or customer information.

Purpose of coding: Coding is the process of assigning a code to each piece of data so that the computer can easily recognize and process it. The purpose of coding is to make data entry more efficient and accurate.

Effective/Accurate coding: Effective coding means that the codes we use accurately reflect the data we are entering. For example, if we are coding customer names, we would use a code that

is unique to each customer. Accurate coding means that the codes are entered correctly into the system.

Bottleneck in data entry: A bottleneck is a point in the data entry process where the speed or efficiency slows down. For example, if the computer system is slow, it could cause a bottleneck in the data entry process. Other bottlenecks could include errors in coding or data entry, or a lack of training for data entry staff.

By designing effective data entry procedures, we can ensure that the data entered the system is accurate, efficient, and useful for achieving our objectives.

Deploy project

- ♦ Implementation approaches

When it comes to deploying a project, there are several implementation approaches that can be taken, including:

Direct cutover: In this approach, the new system replaces the old one all at once. This can be risky, as there is no backup if the new system fails.

Phased implementation: In this approach, the new system is implemented in stages, one module or department at a time. This reduces risk and allows for easier testing and debugging.

Parallel implementation: In this approach, the new system is implemented alongside the old one for a period. This allows for testing and comparison of the two systems, but can be more costly and time-consuming.

Pilot implementation: In this approach, the new system is implemented in a small, controlled setting to test its effectiveness before being rolled out to the entire organization.

Once an implementation approach has been chosen, the deployment process can begin. This involves installing the new system, testing it, and training users on how to use it. It may also involve migrating data from the old system to the new one. After deployment, ongoing support and maintenance are important to ensure the system continues to function effectively.

Module 6: Control quality of system

- Explain total quality assurance approach
- Quality assurance approaches, Structured walkthrough
- Bottom-up and Topdown design

Total quality assurance approach:

The total quality assurance approach is a method used to ensure that a system meets the requirements and specifications of the stakeholders involved in the project. This approach involves using a range of techniques and tools to ensure that the system is tested thoroughly and meets the required quality standards.

Quality assurance approaches:

There are several quality assurance approaches that can be used to test a system, including structured walkthroughs. A structured walkthrough is a method of reviewing a system where a group of people examine the system to find any issues or potential problems. This is an effective way to ensure that the system meets the required standards and can identify any issues early in the development process.

Bottom-up and top-down design:

Bottom-up and top-down design are two approaches used to develop a system. Bottom-up design involves developing individual components of the system and then combining them to form the final product. Top-down design involves starting with the overall system design and then breaking it down into smaller components. Both approaches have their advantages and disadvantages, and the choice of which approach to use will depend on the specific project requirements and constraints.

Experiment system

- Testing processes
- White Box and Black Box test
- Bottom-up and Topdown test

Experimenting a system involves various testing processes to ensure that it meets the specified requirements and is functioning correctly.

Testing is an essential part of developing any system to ensure it is functioning correctly and meets the intended requirements. Here are some of the key aspects of testing processes:

Testing processes: There are several steps involved in the testing process, such as defining test cases, executing them, and reporting the results. It's important to create a comprehensive test plan and ensure that all aspects of the system are tested thoroughly to identify any potential issues.

White Box and Black Box test: These are two main approaches to testing a system. White box testing involves examining the internal structure of the system, including the code, to test its functionality. Black box testing, on the other hand, focuses on the external behavior of the system and tests its functionality without looking at the code.

Bottom-up and Top-down test: These are two main testing strategies. Bottom-up testing starts with testing individual components of the system and then testing their integration with each other to ensure that the whole system is functioning correctly. Top-down testing, on the other hand, starts with testing the overall system and then testing individual components to ensure that they are working correctly.

Maintain system

- Maintenance practices

System maintenance is an important process that involves keeping a computer system running efficiently and effectively. The goal of maintenance is to ensure that the system operates correctly, and that all components are updated, repaired, or replaced as necessary to prevent system downtime or failures.

Maintenance practices include a range of activities that are performed regularly to keep the system running smoothly. These may include:

Preventive Maintenance: This includes tasks such as checking for software and hardware updates, replacing worn-out components, cleaning the system, and conducting regular backups.

Corrective Maintenance: This includes repairing or replacing faulty hardware or software components and fixing any system bugs or errors.

Adaptive Maintenance: This involves modifying or enhancing the system to meet changing business requirements, user needs, or technology advancements.

Perfective Maintenance: This includes improving the system's performance, scalability, or usability by adding new features or functionality.

Emergency Maintenance: This is performed when a system failure occurs, and it is needed to restore the system to a functional state.

By following good maintenance practices, a system can remain in good working order for an extended period. This can help prevent system downtime, reduce repair costs, and extend the useful life of the system.

Audit system

- ♦ Internal/External auditing

An audit is a systematic and independent examination of a system, process, or organization to determine whether the established procedures and policies are being followed and to identify areas that require improvement. In the context of information systems, an audit is performed to evaluate the effectiveness and efficiency of an information system, to ensure compliance with laws and regulations, and to identify risks and vulnerabilities.

Internal auditing involves evaluating the information system by an internal audit department or unit within an organization. The primary objective of internal auditing is to provide assurance to the management that the information system is functioning effectively and efficiently and follows laws, regulations, and policies. Internal auditing also aims to identify risks and vulnerabilities and suggest recommendations for improvements.

External auditing involves the evaluation of the information system by an independent auditor who is not a part of the organization. The primary objective of external auditing is to provide an unbiased opinion about the effectiveness and efficiency of the information system. External auditors typically perform audits to ensure compliance with laws and regulations, identify risks and vulnerabilities, and suggest recommendations for improvement. The external audit report is usually submitted to the management and other stakeholders, including regulatory authorities, investors, and customers.

Both **internal and external auditing** are essential for ensuring the quality and integrity of the information system. The audit process typically involves reviewing documentation, interviewing personnel, and testing system controls and procedures. The audit findings and recommendations are then reported to the management, and corrective actions are taken to address any issues identified during the audit.

Module 7: Perform Project

Project Work