Summary of CMPG223 2023 work

System Analysis and Design (North-West University)

1. **System Design Approaches:**
    - *Model-Driven:* Emphasizes system models for documentation.
    - *Modern Structured Design:* Decomposes system processes into manageable components.
    - *Information Engineering:* Process-sensitive technique for planning and designing information systems.
    - *Prototyping:* Involves an iterative process with close collaboration between designers and users.
    - *Object-Oriented:* Refines object requirements, eliminating concerns about data and process separation.
    - *RAD (Rapid Application Development):* Merges structured, prototyping, and JAD techniques for quick system development.
    - *JAD (Joint Application Development):* Emphasizes participative development, complementing other analysis and design techniques.
2. **Tasks for In-house Development Project:**
    - *Application Architecture:*
        - Define technologies for information systems.
        - Revise models as physical representations.
    - *System Databases:*
        - Develop a database schema optimized for the implementation DBMS.
    - *System Interface:*
        - Specify Input, Output, and Dialogue requirements.
    - *Design Specifications:*
        - Develop prototypes.
    - *Update Project Plan:*
        - Revise the project plan as needed.

Commercial Software Procurement Overview:

1. **Research Technical Criteria and Options:**
    - Utilize magazines, journals, and internal standards for hard/software selection.
    - Information services constantly survey the marketplace for new products.
    - Trade newspapers offer insights on various hardware and software experiences.
2. **Solicit Proposals or Quotes from Vendors:**
    - *Request for Proposals (RFP):* Communicate requirements and desired features to prospective vendors.
    - *Request for Quotations (RFQ):* Used when a specific product is already decided; vendors respond with price quotations.
3. **Validate Vendor Claims and Performances:**
    - Review proposals, eliminating those not meeting mandatory requirements.
    - Validate vendor claims against criteria using user references, technical manuals, and demonstrations.
4. **Evaluate and Rank Vendor Proposals:**
    - Conduct feasibility assessment.
    - Use a scoring system considering hard-dollar costs (payment to selected vendor) and soft-dollar costs (additional costs if selected).
5. **Award Contract and Debrief Vendors:**
    - Negotiate the contract terms.
    - Debrief vendors who lost, providing information on weaknesses in their proposals.
        - No second chances for vendors.
        - Inform vendors of weaknesses in their proposals.

Information System Architecture Components:

1. **Application Architecture:**
    - *Definition:* Specifies technologies for information system implementation.
    - *Considerations:*
        - Degree of centralization or distribution.
        - Distribution of stored data.

        - Implementation of in-house software.
        - Integration of commercial off-the-shelf software.
2. **Physical Data Flow Diagram:**
    - *Definition:* Process model communicating technical implementation characteristics.
    - *Purpose:*
        - Communicate technical choices and design decisions.
        - Guide system construction.
3. **Physical Processes:**
    - *Components:*
        - Logical processes assigned to physical processors (e.g., PCs, servers).
        - Logical processes split into multiple physical processes.
    - *Considerations:*
        - Define aspects performed by people/computers.
        - Implement different technologies.
        - Show multiple implementations of the same process.
        - Incorporate processes for exceptions and security.
4. **Physical Data Flows:**
    - *Definition:* Planned implementation of input/output from a physical process.
    - *Types:*
        - Database commands/actions.
        - Import/export of data.
        - Flow of data between modules or subroutines.
5. **Physical External Agents:**
    - *Transition:* Carried over from logical DFD models.
    - *Note:* Logical models should change before drawing physical models.
6. **Physical Data Stores:**
    - *Representation:* Planned implementation of:
        - Database.
        - Table in a database.
        - Computer file.
        - Media backup.
        - Temporary file.
        - Non-computerized file.

Centralized vs. Distributed Computing:

Distributed System:

- *Definition:* Components distributed across multiple locations and networks.
- *Processing Workload:* Distributed across multiple computers.
- *LAN (Local Area Network):* Set of client computers connected over a short distance.
- *File Server System:* Server hosts data; other layers implemented on clients.
    - Excessive network traffic.
    - Robust client.
    - Database integrity maintained.

Centralized Systems:

- *Definition:* All components hosted by a central computer.
- *User Interaction:* Via terminals; almost all processing on the host.

Alternative Information System Design:

Computing Layers:

- *Presentation:* UI.

- *Presentation Logic:* Processing for generating presentation.
- *Application Logic:* Supports business rules.
- *Data Manipulation:* Store & retrieve data.
- *Data:* Business data.

Client/Server Architecture:

- *Thin Client:* Low-power PC for UI.
- *Fat Client:* Powerful PC.
    - **Types of Servers:**
        - Database: Hosts DBs, executes all data.
        - Transaction: Ensures all DB updates for a transaction.
        - Application: Hosts application logic.
        - Msg/Groupware: Hosts services for calendaring, email, etc.
        - Web: Hosts websites.
    - **Distributed Presentation:** Presentation & logic layers shift from server to client.
    - **Distributed Data:** Data & manipulation layers shift to the client.
    - **Distributed Data & Application:**
        - Data, Manipulation & Application Logic layers on own servers.
        - Presentation & Logic on clients.
    - **Partitioning:** Determines how to distribute components on the network.

Intra- & Internet Technologies:

- *Network Computing System:* Presentation & logic layers in client-side web browsers.
- *Intranet:* Secure network using internet tech to integrate desktop & workgroup computing.
    - **Technologies:**
        - Java: Programming servlets/applets.
        - HTML: Presentation layer programming.
        - XML: Data content transportation.
        - SQL: Database manipulation.
        - Web Browsers.

Strategies for Information System Architecture Development:

Data Architectures:

1. **Relational Database:**
    - *Definition:* Stores data in tabular form; each file is a table; each field is a column.
    - *Characteristic:* Related records are duplicated in two tables.
2. **Distributed Relational Database:**
    - *Definition:* Duplicates tables to multiple servers in different geographic locations.
3. **Distributed Relational Database Management System (DRDBMS):**
    - *Definition:* Controls access to and maintenance of stored data in relational format.
    - *Strategies:*
        - **Data Partitioning:**
            - Vertical (columns) and horizontal (rows) distribution to different servers with no duplication.
        - **Data Replication:**
            - Duplicates some or all tables, updating all servers with duplicated data.
4. **Electronic Data Interchange (EDI):**
    - *Definition:* Standardized electronic flow of business transactions between businesses.
5. **Middleware:**
    - *Definition:* Utility software enabling communication between processors in the system.
    - *Types:*
        - Presentation, application, and database middleware.

Process Architectures:

6. **Software Development Environment (SDE):**
    - *Definition:* Language toolkit for developing apps.
    - *Existence for:*
        - Centralized computing.
        - Distributed presentation.
        - Two-tiered client/server.
        - Multi-tiered client/server.
        - Internet and intranet client/server.
7. **Clean Layering:**
    - *Design Strategy:* Physically separates presentation, application, and data layers.
8. **Design Units:**
    - *Definition:* Self-contained collection of processes, data stores, and data flows sharing similar characteristics.

Distinguishing Outputs:

Internal Outputs:

- *Detailed Reports:* Presents info with little or no filtering (e.g., listing all customers).
- *Summary Reports:* Categorizes info for managers with less detail, often using charts.
- *Exception Reports:* Filters data to report exceptions (e.g., past-due accounts).

External Outputs:

- *Purpose:* Intended for recipients outside the organization (customers, suppliers, etc.).

Turnaround Documents:

- *Definition:* External output that may re-enter the system for further processing.

Output Implementation Methods:

- **Printed:**
    - *Tabular:* Presents in columns.
    - *Zoned:* Places text & numbers in designated areas.
- **Screen:**
    - *Graphic Output.*
- **Others:**
    - *POS Terminals.*
    - *Multimedia.*
    - *E-mail.*
    - *Hyperlinks.*
    - *Microfilm (historically used).*

Charts and Their Uses:

- **Line:**
    - *Use:* Represents one or more series of data over a period.
- **Area:**
    - *Use:* Summarizes and shows change in data.
- **Bar:**
    - *Use:* Compares series or categories of data.
- **Column:**
    - *Use:* Compares same categories at different times.

- **Pie:**
  - *Use:* Illustrates the relationship of parts to a whole.
- **Donut:**
  - *Use:* Like a pie chart but supports multiple series.
- **Radar:**
  - *Use:* Compares different aspects of multiple series.
- **Scatter:**
  - *Use:* Shows the relationship between two or more series.

General Principles for Output Design:

- *Readability:* Outputs should be simple to read.
- *Timing:* Timing of outputs is important.
- *Distribution:* Outputs must be distributed sufficiently to assist all relevant stakeholders.
- *Acceptability:* Outputs must be acceptable to the system users who will receive them.

Output Design Process:

1. Identify system output requirements.
2. Specify physical output requirements.
3. Design preprinted forms.
4. Design, validate, and test outputs using layout tools, prototyping tools, and code generating tools.

Prototyping and Design of Computer Outputs:

1. **Tabular:**
   - *Format:* Rows and columns presenting data in a table.
   - *Example:* Spreadsheet-style output.
2. **Graphical Report:**
   - *Format:* Visual representation of data using graphs or charts.
   - *Example:* Bar charts, pie charts.
3. **Record-at-a-Time:**
   - *Format:* Displays one record at a time.
   - *Example:* Customer record with details.
4. **Web Database:**
   - *Format:* Online presentation of a database.
   - *Example:* Interactive web page displaying database content.
5. **Windows/Web Media Player:**
   - *Format:* Multimedia player interface.
   - *Example:* Video/audio playback controls on a media player.

Appropriate Format and Media for Computer Input:

- *Format:* Structured layout that facilitates efficient input.
- *Media:* Electronic forms, online data entry interfaces.

Difference between Data Capture, Entry, and Input:

- *Data Capture:* Identification and acquisition of new data at its source.
- *Data Entry:* Translation of source data into a readable format.
- *Data Processing:* Occurs on the data after input, including batch, online, and remote batch processing.

Automatic Data Collection Technologies:

- *Optical Mark Recognition (OMR):* Captures marked fields on forms.
- *Bar Codes:* Alphanumeric values entered by scanning.

- *Optical Character Recognition (OCR):* Converts printed or handwritten text into machine-encoded text.

Human Factors in Computer Input Design:

- *Instructions:* Include clear instructions for completing forms.
- *Handwriting:* Minimize reliance on handwriting.
- *Data Sequencing:* Design for top-to-bottom and left-to-right flow.
- *Metaphors:* Use designs based on known metaphors.

Internal Controls for Computer Inputs:

- *Monitoring Inputs:* Monitor the number of inputs, use control slips for batch processing, log each transaction for online processing.
- *Validation Checks:* Perform checks for existence, datatype, domain, combination, self-checking, and format.

Screen-Based Controls for GUI Inputs:

- *Text Boxes:* Unlimited scope data values.
- *Radio Buttons:* Predefined set of mutually exclusive values.
- *Check Boxes:* Yes or No values.
- *List Boxes:* Large number of possible values.
- *Drop Down Lists:* Large number of possible values with small screen space.
- *Combination Boxes:* Selecting from a list or typing a new value.
- *Spin Boxes:* Navigate through a small set of choices or type directly.
- *Buttons:* User presses them.

Advanced Controls:

- *Drop-down Calendars.*
- *Slider Edit Controls.*
- *Masked Edit Controls.*
- *Ellipsis Controls.*
- *Alternate Numerical Spinners.*
- *Check List Boxes.*
- *Check Tree Boxes.*

Designing Web-Based Input Interface:

- Refer to SU5 slides for practical application examples.

Types of Computer Users and Design Considerations:

1. **Expert User:**
    - *Characteristics:*
        - Experienced and spends a lot of time with specific apps.
        - Use of the computer is non-discretionary.
        - Dedicated user.
    - *Design Considerations:*
        - Advanced features readily accessible.
        - Shortcuts and hotkeys for efficiency.
        - Streamlined interfaces.
2. **Novice User:**
    - *Characteristics:*
        - Less experienced and less frequent use of the computer.
        - Use of the computer is viewed as discretionary.
        - Casual user.

- *Design Considerations:*
  - Intuitive and user-friendly interfaces.
  - On-screen guidance and tooltips.
  - Clear instructions for basic functions.

Human Engineering Factors and Guidelines:

- User should be aware of what to do next.
- Instructions and messages display in the same area.
- Display messages long enough to be readable.
- Use display attributes sparingly.
- Specify default values.
- Anticipate errors and prevent users from proceeding with errors.
- Lock the keyboard when users could make catastrophic errors, providing instructions to seek assistance.

Integrating Output and Input Design into User Interface:

- **Dialogue:**
  - *Flow of Screens and Messages:*
    - Consistent tone.
      - Use simple, grammatically correct sentences.
      - Avoid being funny or condescending.
    - Clear and simple terminology.
      - Avoid computer jargon and abbreviations.
      - Be consistent in terminology.
      - Carefully phrase instructions.

Role of Operating Systems, Web Browsers, and Technologies:

- **User Interface Technology:**
  - *Operating Systems & Web Browsers:*
    - GUI interfaces for Windows, Macintosh, UNIX.
    - Growing importance of platform independence.
  - *Display Monitors:*
    - Regular PC monitors.
    - Non-GUI terminals.
    - Growing importance of devices like handhelds.
  - *Paging and Scrolling:*
    - Paging – complete screen of characters.
    - Scrolling – display info up or down a screen.
  - *Keyboards and Printers:*
    - Mouse, Pens.
  - *Graphical User Interfaces Styles and Considerations:*
    - Windows and frames.
    - Menu-driven interfaces (pull-down, cascading, tear-off, pop-up, toolbar, iconic, hypertext, hyperlink menus).
    - Instruction-driven interfaces (language-based syntax, mnemonic syntax, natural language syntax).
    - Question-answer dialogue.

User Interface Strategies and State Transition Diagram:

Special Considerations for UI Design:

- **Internal Controls:**

    - Authentication and authorization (User ID & Password, Privileges assigned to roles, Web certificates).
- **Online Help:**
    - HTML Help systems, Help authoring packages, Tooltips, Help Wizards, Agents (reusable software).

User Interface Design Process:

1. **Chart the User Interface Dialogue:**
    - State Transition Diagram (depicts the sequence and variation of screens during a user session).
2. **Prototype Dialogue & UI:**
3. **Obtain User Feedback:**
4. **Iterate (Return to Step 1 or 2 if needed).**

Prototyping for User Interface Design:

- **Purpose:**
    - Quick creation of a working model for user evaluation.
    - Allows users to interact with the system.
    - Identifies issues and improvements early in the design process.

Construction and Implementation Phases:

- **Purpose:**
    - Develop, install, and test system components.
    - Ensure the system is ready for production.

Construction Implementation Phases - Major Tasks, Roles, Inputs, and Outputs:

1. **Build and Test Networks:**
    - **Roles:**
        - Network Designer (designs LAN & WAN connectivity).
        - Network Admin (builds and tests network architecture standards, security).
        - System Analyst (facilitates, ensures business requirements).
    - **Inputs:** Existing networks.
    - **Outputs:** Tested and built networks.
2. **Build and Test Databases:**
    - **Roles:**
        - System Users (provide and/or approve test data).
        - DB Designer (builds tables, views, stored procedures).
        - DB Admin (tunes database for performance, security, backup, and recovery).
        - System Analyst (builds non-corporate, applications-oriented database, ensures compliance).
    - **Inputs:** Sample data.
    - **Outputs:** Unpopulated database structure.
3. **Install and Test New Software Packages:**
    - **Roles:**
        - System Analyst (clarifies business requirements).
        - System Designer (clarifies program design and integration requirements).
        - Application Programmer (writes/tests software).
    - **Inputs:** In-house and purchased/leased software.
    - **Outputs:** Installed and tested software.
4. **Write and Test New Programs:**
    - **Roles:**
        - System Analyst (clarifies business requirements).
        - System Designer (clarifies program design and integration requirements).
        - Application Programmer (writes/tests software).
    - **Inputs:** Business requirements.

- **Outputs:** Developed, tested, and documented programs.

Systems Implementation:

- **Conduct System Test:**
  - **Roles:**
    - System Analyst (develops system test data, communicates problems).
    - System Builders (resolve problems revealed during testing).
    - System Owners and Users (verify if the system operates correctly).

System Conversion Strategies:

- **Prepare Conversion Plan:**
  - Develop detailed conversion plan.
  - **Roles:**
    - System Analyst (develop detailed conversion plan).
    - Steering Committee (approves plan and timetable).
- **Installation Strategies:**
  - Abrupt cutover, Parallel conversion, Location conversion, Staged Conversion.
- **Install Databases:**
  - **Roles:**
    - Application Programmers (write special programs).
    - System Analyst (calculate database sizes, estimate time for install).
- **Train Users:**
  - **Roles:**
    - System Analyst (plan and conduct training, write documentation, help users).
    - System Owners (approve release time for training).
    - System Users (attend training, accept the system).

Convert New System:

- Ownership transfers from analysts and builders to end users.
  - **Roles:**
    - **System Analyst:**
      - Carries out conversion.
      - Corrects shortcomings.
      - Measures system acceptance.
    - **System Owners & Users:**
      - Provide feedback.

Explanation of Application Program and System Tests:

1. **System Acceptance Test:**
   - Performed on the final system.
   - Users conduct verification, validation, and audit tests.
2. **Verification Testing:**
   - Runs the system in a simulated environment.
   - Uses simulated data.
3. **Validation Testing:**
   - Runs the system live using real data.
4. **Audit Testing:**
   - Certifies that the system is free of errors and ready.

Application Program and System Tests:

1. **Stub Test:**

    - Test performed on a subset of a program.
2. **Unit or Program Test:**
    - Test performed on the whole program.
3. **Systems Test:**
    - Test performed on the entire system.

Definition of Systems Operation and Support:

- **Systems Support:**
    - Ongoing technical support for users and required maintenance.
- **System Operation:**
    - Periodic execution of an information system's processes.
- **Operational System:**
    - System that has been placed in operation.

Roles of Repository, Program Library, and Database:

- **Repository:**
    - Data stores of system knowledge and documentation during system development.
- **Program Library:**
    - Data stores of application programs.
- **Business Data:**
    - Data stores with business data.

Differentiation of System Support Activities:

- **System Maintenance:**
    - Corrects bugs and errors.
- **System Recovery:**
    - Restoration of a system after system failure.
- **Technical Support:**
    - Assistance provided to users in response to unforeseen situations.
- **System Enhancement:**
    - Improvement of the system to handle new business problems.

Tasks Required for Program Maintenance in Response to Bugs:

1. Validate the problem.
2. Benchmark the program.
3. Study and debug.
4. Test the program.

Tasks in System Enhancement and Relationship with Original Systems Development:

1. Analyse enhancement request.
2. Make a quick fix if appropriate (changes without restructuring or updating stored data).
3. Recover the existing physical system:
    - Update repository and documentation.
    - Database recovery and restructuring.
    - Program analysis, recovery, and restructuring.

Role of Re-engineering in System Enhancement:

- **Re-engineering:**
    - Process of restructuring existing computer code or system architecture.

Three Types of Re-engineering:

1. **Code Reorganization:**
    - Modularity and/or logic.
2. **Code Conversion:**
    - From one language to another.
3. **Code Slicing:**
    - Create reusable software components or objects out of existing code.