



RQF LEVEL 4



SWDBS401
SOFTWARE
DEVELOPMENT

Backend System Design

TRAINER'S MANUAL

October, 2024



BACKEND SYSTEM DESIGN



AUTHOR'S NOTE PAGE (COPYRIGHT)

The competent development body of this manual is Rwanda TVET Board ©, reproduce with permission.

All rights reserved.

- This work has been produced initially with the Rwanda TVET Board with the support from KOICA through TQUM Project.
- This work has copyright, but permission is given to all the Administrative and Academic Staff of the RTB and TVET Schools to make copies by photocopying or other duplicating processes for use at their own workplaces.
- This permission does not extend to making of copies for use outside the immediate environment for which they are made, nor making copies for hire or resale to third parties.
- The views expressed in this version of the work do not necessarily represent the views of RTB. The competent body does not give warranty nor accept any liability
- RTB owns the copyright to the trainee and trainer's manuals. Training providers may reproduce these training manuals in part or in full for training purposes only. Acknowledgment of RTB copyright must be included on any reproductions. Any other use of the manuals must be referred to the RTB.

© **Rwanda TVET Board**

Copies available from:

- *HQs: Rwanda TVET Board-RTB*
- *Web: www.rtb.gov.rw*
- **KIGALI-RWANDA**

Original published version: October 2024

ACKNOWLEDGEMENTS

The publisher would like to thank the following for their assistance in the elaboration of this training manual:

Rwanda TVET Board (RTB) extends its appreciation to all parties who contributed to the development of the trainer's and trainee's manuals for the TVET Certificate IV in Software Development, specifically for the module " **SWDBS401: Backend System Design**".

We extend our gratitude to KOICA Rwanda for its contribution to the development of these training manuals and for its ongoing support of the TVET system in Rwanda.

We extend our gratitude to the TQUM Project for its financial and technical support in the development of these training manuals.

We would also like to acknowledge the valuable contributions of all TVET trainers and industry practitioners in the development of this training manual.

The management of Rwanda TVET Board extends its appreciation to both its staff and the staff of the TQUM Project for their efforts in coordinating these activities.

This training manual was developed:

Under Rwanda TVET Board (RTB) guiding policies and directives



Under Financial and Technical support of



COORDINATION TEAM

RWAMASIRABO Aimable

MARIA Bernadette M. Ramos

MUTIJIMA Asher Emmanuel

Production Team

Authoring and Review

TUYIZERE Emmanuel

HATANGIMANA John

Validation

KWIZERA Israel

MBONYISIMBI Dieudonne

Conception, Adaptation and Editorial works

HATEGEKIMANA Olivier

GANZA Jean Francois Regis

HARELIMANA Wilson

NZABIRINDA Aimable

DUKUZIMANA Therese

NIYONKURU Sylvestre

MANIRAKIZA Jean de Dieu

Formatting, Graphics, Illustrations, and infographics

YEONWOO Choe

SUA Lim

SAEM Lee

SOYEON Kim

WONYEONG Jeong

KANANI Vincent

Financial and Technical support

KOICA through TQUM Project

TABLE OF CONTENT

AUTHOR'S NOTE PAGE (COPYRIGHT)-----	iii
ACKNOWLEDGEMENTS-----	iv
TABLE OF CONTENT -----	vii
INTRODUCTION -----	1
MODULE CODE AND TITLE: SWDBS401 BACKEND SYSTEM DESIGN-----	2
Learning Outcome 1: Analyse System Backend-----	3
Key Competencies for Learning Outcome 1: Analyze System Backend-----	4
Indicative content 1.1: Gathering FURPS Requirements -----	6
Indicative content 1.2: Identification of Main Objects of Backend System -----	12
Indicative content 1.3: Description of System Interaction-----	14
Indicative content 1.4: Report of the System Backend Requirements-----	16
Learning outcome 1 end assessment -----	18
Further information to the trainer-----	23
Learning Outcome 2: Develop System Structure -----	24
Key Competencies for Learning Outcome 2: Develop System Structure -----	25
Indicative content 2.1: Identification of System Design Tools -----	27
Indicative content 2.2: Identification of Hardware and Software Technology-----	31
Indicative content 2.3: Application of SSADM (Structured System Analysis and Design Methods) -----	33
Indicative content 2.4: Application of Object-Oriented Analysis and Design -----	36
Learning outcome 2 end assessment -----	38
Further information to the trainer-----	41
Learning Outcome 3: Build System Design-----	42
Key Competencies for Learning Outcome 3: Build System Design-----	43
Indicative content 3.1: Development of Data Flow Diagram-----	45
Indicative content 3.2: Application of Physical Data Model -----	48
Indicative content 3.3: Documentation of System Design -----	51
Learning outcome 3 end assessment -----	56
Further information to the trainer-----	59

ACRONYMS

API: Application programming Interface

CPU: Central processing Unit

DB: Database

DBaaS: Database as a service

DBMS: Database Management System

DDL: Data Definition language

DFD: Data Flow Diagram

DML: Data Manipulation Language

ERD: Entity Relational Diagram

FSD: Functional Specification Document

FURPS: Functionality, Usability, Reliability, Performance, and Security.

GWT: Google Web Toolkit

ITC: Insert Time Clustering

JSON: JavaScript Object Notation

KOICA: Korean international corporation Agency

KPIS: Key performance Indicators

MDC: multitenant database containers

MVC: Model-View-Controller

MYSQL: My Structured Query Language

NICs: Network Interface Cards

OOAD: Object Oriented Analysis and Design

ORM: Object Relational Mapper

OS: Operating System

PHP: Hypertext Preprocessor

RAD: Rapid Application Development

RAM: Random Access Memory

RTB: Rwanda TVET Board

SDD: System Design Document

SDLC: System development life cycle

SPA: Single Page Application

SQL: Structure query Language

SRIS: Student Result Management System

SRS: Software Requirement Specification

SSADM: Structured System Analysis Design Methods

SSDs: Solid State Drives

SSO: Sing sign on

TQUM Project: TVET Quality Management Project

TSD: Technical Specification Document

UML: Unified Modeling language

XML: Extensible Markup Language

INTRODUCTION

This trainer's manual includes all the methodologies required to effectively deliver the module titled "**Backend system design**". Trainees enrolled in this module will engage in practical activities designed to develop and enhance their competencies.

The development of this training manual followed the Competency-Based Training and Assessment (CBT/A) approach, offering ample practical opportunities that mirror real-life situations.

The trainer's manual is organized into Learning Outcomes, which is broken down into indicative content that includes both theoretical and practical activities. It provides detailed information on the key competencies required for each learning outcome, along with the objectives to be achieved.

As a trainer, you will begin by asking questions related to the activities to encourage critical thinking and guide trainees toward real-world applications in the labor market. The manual also outlines essential information such as learning hours, didactic materials, and suggested methodologies.

This manual outlines the procedures and methodologies for guiding trainees through various activities as detailed in their respective trainee manuals. The activities included in this training manual are designed to offer students opportunities for both individual and group work. Upon completing all activities, you will assist trainees in conducting a formative assessment known as the end learning outcome assessment. Ensure that trainees review the key reading and the points to remember section.

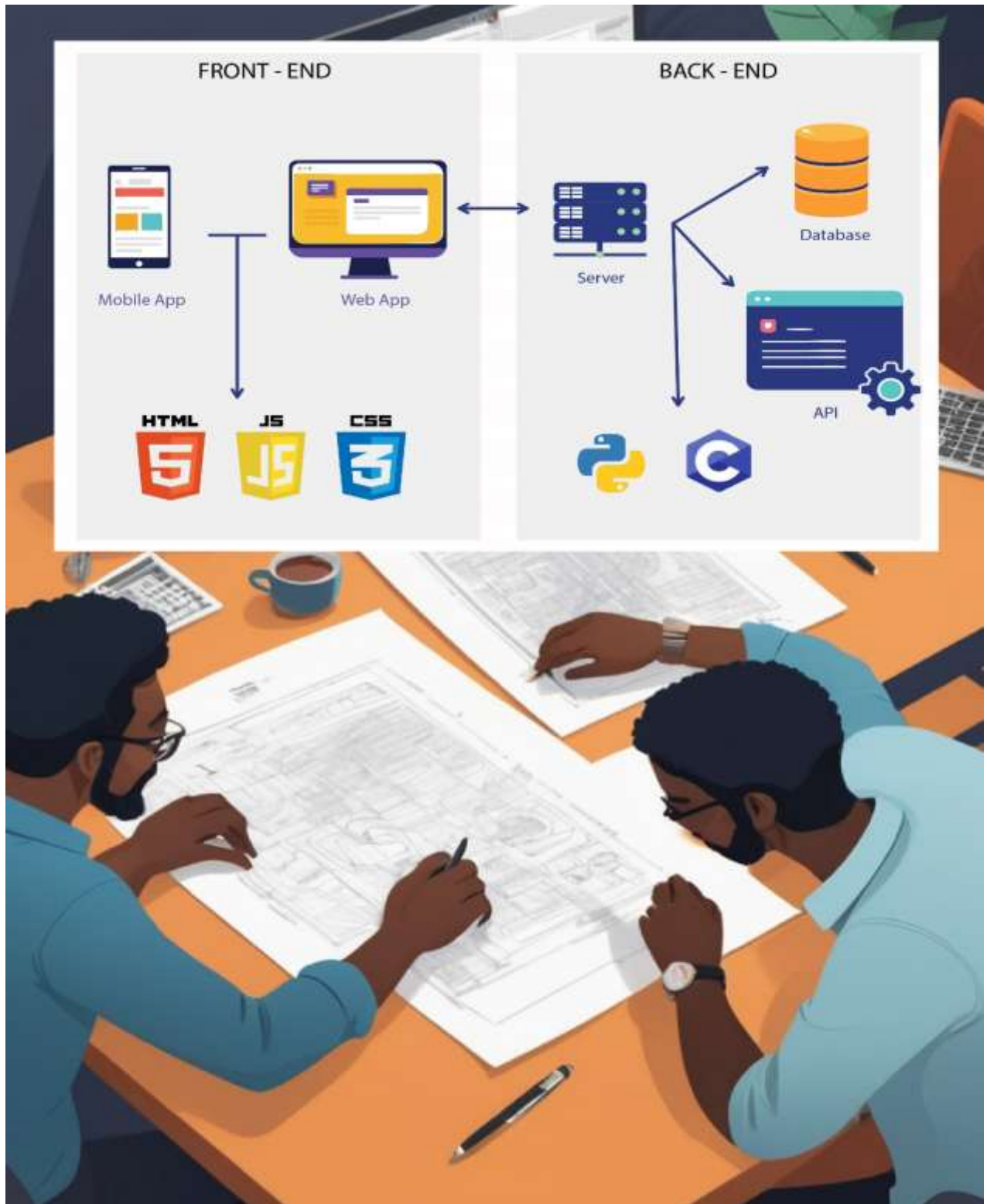
MODULE CODE AND TITLE: SWDBS401 BACKEND SYSTEM DESIGN

Learning Outcome 1: Analyze System Backend

Learning Outcome 2: Develop System Structure

Learning Outcome 3: Build System Design

Learning Outcome 1: Analyse System Backend



Indicative contents

1.1 Gathering FURPS Requirements

1.2 Identification of Main objects of Backend System

1.3 Description of System Interaction

1.4 Report of the System Backend Requirements

Key Competencies for Learning Outcome 1: Analyze System Backend

Knowledge	Skills	Attitudes
● Description of FURPS key terms in Backend System ● Description of backend development Technologies and tools. ● Description of System development life cycle (SDLC) ● Identification of system Analysis tools ● Identification of FURPS Requirements ● Identification of main objects of Backend System ● Description of system Interaction	● Conducting data gathering ● Documenting report of system requirements	● Having Team work ● Being critical thinker ● Being Innovative ● Being attentive. ● Being creative ● Having Curiosity and Open Mindedness



Duration: 17 hrs

Learning outcome 1 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Describe properly key terms of backend system according to backend system design.
2. Describe correctly backend development Technologies according to backend system design.
3. Describe properly system development life cycle (SDLC) according to backend system design
4. Identify properly system analysis tools based on backend system design
5. Identify properly main objects based on the system analysis methodology
6. Describe clearly system interactions based on identified objects
7. Report appropriately system backend requirements in line with software requirements document



Resources

Equipment	Tools	Materials
• Computer	• Microsoft Office • Browser • E- Draw • Visual paradigm	• Internet • Electricity



Advance Preparation:

- Before delivering this learning outcome, you are recommended to:
- Avail computer with Draw-Max installed



Indicative content 1.1: Gathering FURPS Requirements



Duration: 5 hrs



Theoretical Activity 1.1.1: Description of FURPS key terms in Backend System



Notes to the trainer:

- Trainer may use small group for describing key terms in Backend System
- Avail images related to backend system to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- Define the following terms as applied in backend system
 - Backend
 - System
 - Database
 - Operating system
 - System development lifecycle (SDLC)
 - Framework
 - JSON
 - UML
 - FURPS
- What are the five categories of software requirements defined by FURPS?
- Differentiate Server from API

Step 2: Ask trainees to write the findings on papers

Step 3: Ask Trainees to present their findings to the trainer and colleagues.

Step 4: Provide expert view and address any questions or concerns

Step 5: Ask trainees to read the key readings 1.1.1 in their trainee manual



Points to Remember

- The backend refers to the server-side of an application where data is processed, and business logic is executed.
- A system is a set of interconnected components working together to achieve a common goal.
- A database is an organized collection of data that is stored and accessed electronically.
- The operating system is software that manages hardware and other software on a computer.
- The systems development life cycle (SDLC) is the overall process for developing information systems from planning and analysis through implementation and maintenance.
- A framework is a pre-built, reusable set of tools, libraries, and conventions that provide a structured way to develop, organize, and deploy backend applications.
- JSON (JavaScript Object Notation) is a lightweight data interchange format that is widely used in backend system design.
- UML (Unified Modeling Language) is a standardized modeling language used in software engineering to visualize, specify, construct, and document the components of a system
- FURPS stands for Functionality, Usability, Reliability, Performance, and Supportability.
- categories of software requirements defined by FURPS are: Functionality, Usability, Performance, Reliability and Supportability
- An API is a set of rules and protocols that allows one software application to interact with another while a server is a computer or software that provides services or resources to other computers or clients in a network.



Theoretical Activity 1.1.2: Description of System development life cycle (SDLC) models



Notes to the trainer:

- Trainer may use small group for describing system development life cycle (SDLC) models
- Avail images of SDLC Models to be used as didactical materials.



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** Introduce the activity and ask trainees to answer the following questions:
- Explain briefly the Phases of System development life cycle (SDLC)
 - Describe SDLCs models
- Step 2:** Ask trainees to write the findings on papers
- Step 3:** Ask trainees to present their findings
- Step 4:** Provide expert view and address any questions or concerns
- Step 5:** Ask trainees to read the Key readings 1.1.2 in their manuals



Points to Remember

- System Development Life Cycle (SDLC) Phases are : Planning, Design ,Implementation (Coding) ,Testing, Deployment and Maintenance
- SDLC Models are: Waterfall Model, RAD Model, Spiral Model ,V-Model ,Incremental Model, Agile Model , Iterative Model and Big bang model



Theoretical Activity 1.1.3: Description of backend development technologies



Notes to the trainer:

- Trainer may use small group for describing system backend development technologies and System Analysis tools.
- Avail videos to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- i. Describe backend development technologies
- ii. List and explain system analysis tools

Step 2: Ask trainees to write the findings on papers

Step 3: Ask Trainees to present their findings to the trainer and colleagues.

Step 4: Provide expert view and address any questions or concerns

Step 5: Ask trainees to read the key readings 1.1.3 in trainee's manual



Points to Remember

- Backend development Technologies are Python and its Frameworks, PHP and its Frameworks, JAVA and its Frameworks and Ruby and its Framework
- **System Analysis Tools** are Grid Charts, System Flow Chart, Decision Tree, Simulation and Decision Tables



Theoretical Activity 1.1.4: Description of data gathering and identification FURPS

Requirements



Notes to the trainer:

- Trainer may use small group for identification of FURPS requirements.
- Avail videos to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- i. Explain data gathering process
- ii. Identify FURPS requirements

Step 2: Ask trainees to write the findings on papers

Step 3: Ask Trainees to present their findings to the trainer or whole class.

Step 4: Provide expert view and address any questions or concerns

Step 5: Ask trainees to read the key readings 1.1.4 in trainee's manual



Points to Remember

- Data gathering is the first and most important step in the research process, regardless of the type of research being conducted.
- Data gathering process are:
 - ✓ Define Objectives
 - ✓ Identify Data Sources
 - ✓ Choose Data Collection Methods
 - ✓ Design Data Collection Instruments
 - ✓ Pilot Testing
 - ✓ Collect Data
 - ✓ Data Validation
 - ✓ Store and Organize Data
 - ✓ Analyze Data
 - ✓ Report Findings
- Requirements are: Functionality, Usability, Reliability and Performance



Practical Activity 1.1.5: Conducting data gathering



Notes to the trainer

- This activity must take place in computer lab where trainees have to perform data gathering.
- The trainer prepares Case study
- Have a well-prepared computer lab installed with MS Office



Key steps:

While delivering this activity, pass through the following steps:

- Step 1:** By referring to the activity 1.1.4 perform the following task:
Collect all data that are needed to develop a trainee registration system of your school to enrol L3 trainees.
- Step 2:** Explain the task and provide clear instructions
- Step 3:** Demonstrate step by step how to conduct data gathering.

Step 4: Ask trainees to perform the task and monitor the activity.

Step 5: Verify whether the activity have been performed.

Step 6: Ask trainee to read key reading 1.1.5 in trainee manual



Points to Remember

- **Performing data gathering you follow the following steps:**
 - ✓ Define Objectives
 - ✓ Determine Data Requirements
 - ✓ Choose Data Collection Methods
 - ✓ Develop Data Collection Instruments
 - ✓ Collect Data
 - ✓ Data Verification
 - ✓ Analyze Data
 - ✓ Report Findings
 - ✓ Review and Reflect



Application of learning 1.1

XYZ school has a problem of registering new trainees In level 3, normally it use Analog registration form (use paper) then want to change current system to digital registration system as trainee who learn data gathering you are requested to collect all data needed develop digital registration system to enroll new Level 3 trainees

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Trainees' information are well collected	1.1 Trainees profile are collected		
		1.2 School profile are collected		
		1.3 Previous Home profile are collected		
		1.4 Family background are collected		



Indicative content 1.2: Identification of Main Objects of Backend System



Duration : 4hrs



Theoretical Activity 1.2.1: Description main objects of Back end System



Notes to the trainer:

- Trainer may use small group for describing main objects of backend system.
- Avail videos to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- Define the Scope of Backend System
- What do you understand by the followings:
 - Database
 - APIs
 - Servers
 - Frameworks

Step 2: Ask trainees to write their findings on papers

Step 3: Ask Trainees to present their findings to the trainer and colleagues.

Step 4: Provide expert view and address any questions or concerns

Step 5: Ask trainees to read the key readings 1.2.1 in trainee's manual



Points to Remember

- Defining the scope of a backend system involves determining the specific functionalities and features that the system will encompass.
- A database is an organized collection of structured information, or data, typically stored electronically in a computer system
- API is the acronym for application programming interface a software intermediary that allows two applications to talk to each other.
- A server is a computer program or device that provides a service to another computer program and its user, also known as the client

- Backend frameworks are server-side frameworks designed to make tasks easier for developers.



Application of learning 1.2

XYZ school has a problem of registering new trainees In level 3, normally it use Analog registration form (use paper) then want to change current system to digital registration system as trainee who learn main object of backend System design you are requested to help your school to identify the main object of backend system of digital registration system .

Checklist:

Main point to check	Yes	No
1.Scope of Backend System is identified		
2.Database is identified		
3.APIs is identified		
4.Servers is identified		
4.Frameworks is identified		



Indicative content 1.3: Description of System Interaction



Duration: 3hrs



Theoretical Activity 1.3.1: Description of System Interaction



Notes to the trainer:

- Trainer may use small group for describing system interaction.
- Avail videos to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce to trainees by asking following questions:

- i. What is Purpose of System Interaction
- ii. Describe the Main components of System Interaction

Step 2: Ask trainees to write the findings on papers

Step 3: Ask Trainees to present their findings to the trainer and colleagues.

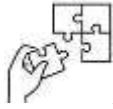
Step 4: Provide expert view and address any questions or concerns

Step 5: Ask trainees to read the key readings 1.3.1 in trainee's manual



Points to Remember

- Interactions between systems are a necessity, a source of opportunity, and a source of difficulty and complication in building, implementing, and maintaining IT-reliant systems in organizations.
- Main components of System Interaction: Web server, Application Server, Database Server External Services and API and Message queues or event Streams.



Application of learning 1.3

XYZ school has a problem of registering new trainees In level 3, normally it use Analog registration form (use paper) then want to change current system to digital registration system as trainee who learn main object of backend System design you are requested to help your school to identify the system interaction and main components of System Interaction of digital registration system.

Checklist:

Main point to check	Yes	No
1.Interactions between systems is identified		
2.Main components of System Interaction are identified		



Indicative content 1.4: Report of the System Backend Requirements



Duration: 5 hrs



Practical Activity 1.4.1: Reporting the system backend requirements



Notes to the trainer

- This activity should take place in computer lab where trainees have to report the system backend requirements.



Key steps:

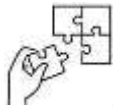
While delivering this activity, pass through the following steps:

- Step 1:** Introduce the activity and ask trainees to perform the following task:
Your school need to develop system that will help them in management of students' results you are requested to report the system backend requirements
- Step 2:** Explain the task and provide clear work instructions.
- Step 3:** Demonstrate how to report the system backend requirements, while demonstrating explain the steps to be followed.
- Step 4:** Ask trainees to report the system backend requirements and monitor the activity.
- Step 5:** Verify whether the task have been performed address any questions or concerns
- Step 6:** Ask trainees to read the key readings 1.4.1 in trainee's manual



Points to Remember

- **While Reporting the System Backend Requirements you have to follow the following steps:**
 - ✓ Develop executive summary
 - ✓ Detailed analysis of the Current State
 - ✓ Findings on Gaps and issues
 - ✓ Write recommendations



Application of learning 1.4.

The xyz district in Rwanda has been experiencing rapid growth in its student population. The existing student information system (SIS) is outdated and unable to meet the increasing demands of the district's schools. This has led to inefficiencies in various administrative tasks, such as student enrolment, attendance tracking, and grade management. The district has decided to implement a new SIS to address the challenges faced by its schools. The district has commissioned a report on the system's backend requirements,

You are asked to make the report of Backend system requirements that can include Executive Summary, Detailed Analysis of the Current State, Findings on Gaps and Issues, Recommendations.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Executive Summary are clearly constructed	1.1 Executive Summary content are defined		
		1.2 Executive Summary content are understandable		
2	Detailed Analysis of the Current State are properly provided	2.1 current state are analysed		
		2.2 current state are documented		
3	Findings on Gaps and Issues are well identified	3.1 Gaps and Issues is identified		
4	Recommendations are well constructed	4.1 Recommendations are constructed		



Learning outcome 1 end assessment

Theoretical assessment

I. Choose correct answer by circling the corresponding letter in the following Questions

1. Which of the following is NOT a phase in the System Development Life Cycle (SDLC)?

- A) Implementation
- B) Maintenance
- C) Abstraction
- D) Testing

Answer: C) Abstraction

2. Which framework is commonly used with Python for backend development?

- A) Django
- B) Laravel
- C) Ruby on Rails
- D) Spring

Answer: A) Django

3. What does FURPS stand for in requirements gathering?

- A) Functionality, Usability, Reliability, Performance, Supportability
- B) Functionality, Usability, Responsiveness, Performance, Scalability
- C) Functionality, Universality, Reliability, Performance, Supportability
- D) Flexibility, Usability, Reliability, Performance, Supportability

Answer: A) Functionality, Usability, Reliability, Performance, Supportability

4. Which of the following is an example of a system analysis tool?

- A) JSON
- B) Decision tree
- C) API
- D) UML

Answer: B) Decision tree

5.Which of the following is a key component of system interaction?

- A) Web server
- B) Functionality requirements
- C) FURPS
- D) SDLC

Answer: A) Web server

II. Answers by using True or False in the following statements

- a) FURPS is an acronym used to describe the phases of the System Development Life Cycle (SDLC).

Answer: False

- b) PHP is a backend development technology that has its own frameworks.

Answer: True

- c) JSON is a data interchange format that is often used in APIs.

Answer: True

- d) A Decision table is a system analysis tool used to represent and analyse complex decision-making scenarios.

Answer: True

- b) The purpose of system interaction is to define the roles and functions of various components within a system.

Answer: True

III. The following table in column A it has the key terms of backend and their definition in column B, Match Key Term with the corresponding definition in the table below in column C

Column (C)	Column (A)	Column (B)
... I.....	1.Backend	A) A structured collection of data stored and accessed electronically.
...H....	2.FURPS	B) A software environment that provides foundational services for other software.
...K....	3.System	C) A set of guidelines, libraries, and tools used to develop software applications.
...D....	4.Server	D) The physical or virtual machine that provides services to other computers over a network.
...A....	5.Database	E) The process of planning, creating, testing, and deploying an information system.
...B....	6.Operating System	F) A method of representing complex systems visually through diagrams.

...E....	7.System Development Life Cycle (SDLC)	G) A software component that allows two applications to communicate with each other.
...G....	8.API	H) A concept that outlines the requirements for software, focusing on different quality attributes.
...J....	9.JSON	I) The part of a system that handles the logic, database interactions, and server-side processing.
...C....	10.Framework	J) A data format often used in web APIs for transmitting structured data.
...F....	11.UML	K) An organized set of components working together to achieve a common goal.

IV. Open questions

a) What does the "Technology Stack" mean in the report?

Ans:

The "Technology Stack" refers to the set of tools, programming languages, frameworks, libraries, and technologies used to develop and operate a software application. In a report, it outlines the backend, frontend, database, and other infrastructure components that form the foundation of the system. For example, it might include technologies like Node.js for the backend, React for the frontend, MongoDB for the database, and AWS for cloud hosting.

b) How does the report look at backend security?

Ans:

The report examines backend security by assessing measures taken to protect data, prevent unauthorized access, and ensure system integrity. It may include:

- Authentication and authorization mechanisms (e.g., OAuth, JWT).
- Encryption of data in transit (SSL/TLS) and at rest.
- Secure APIs with input validation and rate limiting.
- Regular updates and patches to address vulnerabilities.
- Auditing and monitoring for suspicious activities.

c) Why is it important to measure system performance?

Ans:

Measuring system performance is crucial to ensure:

1. **User Satisfaction:** Fast response times and low latency enhance the user experience.
2. **Reliability:** Identifying and addressing performance bottlenecks improves stability.

3. **Scalability:** Performance metrics help determine if the system can handle increased loads.
4. **Cost Efficiency:** Optimizing performance reduces resource usage and operational costs.
5. **Troubleshooting:** Metrics help pinpoint and resolve issues quickly.

d) What problems can happen in managing backend systems?

Ans:

Managing backend systems can present challenges such as:

- **Downtime:** Unexpected failures or server crashes can disrupt services.
- **Scalability Issues:** Difficulty handling increased traffic or load.
- **Security Threats:** Data breaches, hacking, or malware attacks.
- **Complexity:** Managing multiple services, APIs, and dependencies.
- **Integration Problems:** Incompatibility with third-party systems.
- **Maintenance Overload:** Keeping all components updated and operational.

e) What are some examples of problems found in backend systems?

Ans:

Examples include:

1. **Database Deadlocks:** Multiple processes competing for the same resources.
2. **Slow Query Performance:** Inefficient database queries slowing down response times.
3. **Memory Leaks:** Leading to server crashes or degraded performance.
4. **API Overload:** Excessive requests causing throttling or downtime.
5. **Authentication Errors:** Misconfigured or broken authentication flows.
6. **Latency Issues:** Slow response times due to network or server bottlenecks.

f) Why is scalability important for a backend system?

Ans:

Scalability ensures that a backend system can handle increased demands without degrading performance. It is important because:

- **Supports Growth:** Accommodates more users and data as the application grows.
- **Improves Reliability:** Maintains performance under peak loads or spikes.
- **Enhances User Experience:** Consistently delivers fast and responsive services.
- **Optimizes Costs:** Efficiently uses resources to scale up or down as needed.

g) What role does training the team play in backend system improvements?

Ans:

Training the team plays a critical role in improving backend systems by:

- **Enhancing Skills:** Equipping team members with the latest knowledge and tools.
- **Reducing Errors:** Improving coding practices and system management.
- **Fostering Innovation:** Encouraging the adoption of new, more efficient technologies.
- **Strengthening Security:** Educating on secure coding practices and threat mitigation.
- **Improving Collaboration:** Aligning team understanding for smoother operations.
- **Ensuring Continuity:** Preparing for effective system updates and troubleshooting.

Practical assessment

A school management system (SMS) is a complex application that manages various aspects of a school's operations, including student information, attendance, grading, and communication. By applying the FURPS (Functionality, Usability, Reliability, Performance, and Supportability) framework, we can ensure that the SMS meets the specific needs of the school and provides a high-quality user experience. You are hired to make analysis, identification of FURPS requirements and make the report of FURPS requirements of SMS

SN	Criteria	Indicators	Observation	
			Yes	No
1	SMS are well analysed	1.1 SMS are analysed		
		1.2 Analysis is documented		
2	FURPS requirements are well Gathered	2.1 FURPS requirements are identified		
		2.2 FURPS requirements are gathered		
3	Report of the System Backend Requirements are well reported	3.1 Executive summary is constructed		
		3.2 Detailed analysis of the Current State are identified		
		3.3 Gaps and Issues is identified		
		3.4 Recommendations are given		

END



Further information to the trainer

Books:

- Bird, S. K. (2009). Natural language processing with Python. O'Reilly Media.
- Elgendy. (2015). Business analysis for beginners. Outskirts Press.
- Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.
- Hohpe, G. &. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley.
- Kendall, K. E. (2013). Systems analysis and design. Pearson.
- Kotonya, G. &. (1998). Requirements engineering: Processes and techniques. Wiley.
- McConnell, S. (2004). A practical handbook of software construction. Microsoft Press.
- McConnell, S. (2004). Documenting system backend requirements. Microsoft Press.
- Pressman, R. S. (2014). Software engineering. McGraw-Hill Education.
- Satzinger, J. W. (2012). Systems analysis and design. Course Technology, Cengage Learning.
- Sommerville. (2015). Software engineering. Pearson.

Web links:

- Ambler, S. W. (n.d.). Agile requirements change management. Retrieved from <http://www.agilemodeling.com/essays/requirementsChangeManagement.htm>
- Backend developer. (n.d.). Retrieved from Guru99: <https://www.guru99.com/backend-developer.html>
- How to gather requirements. (n.d.). Retrieved from Lucidchart: <https://www.lucidchart.com/blog/how-to-gather-requirements>
- Mozilla Developer Network. (n.d.). Retrieved from JSON: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- Stackify. (n.d.). Retrieved from What is backend: <https://stackify.com/what-is-backend/>
- Techopedia. (n.d.). Retrieved from FURPS: <https://www.techopedia.com/definition/25949/furps>
- Tutorials Point. (n.d.). Retrieved from SDLC : https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

Learning Outcome 2: Develop System Structure



Indicative contents

2.1 Identification of system design tools

2.2 Identification of Hardware and software technology

2.3 Application of SSADM (Structured System Analysis and Design Methods)

2.4 Application of Object-Oriented Analysis and Design

Key Competencies for Learning Outcome 2: Develop System Structure

Knowledge	Skills	Attitudes
● Identification of system design tools ● Identification of hardware and software technologies ● Identification of application of SSADM (Structured System Analysis and Design Methods) ● Description of application of Object-Oriented Analysis and Design ● Identification of stages for drawing ● Description of application Phases in Object-Oriented Software development	● Implementing stages for drawing SSADM	● Having Team work ● Being critical thinker ● Being Innovative ● Being attentive. ● Being creative ● Having Curiosity and Open Mindedness



Duration: 31 hrs

Learning outcome 2 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Identify properly system design tools based on analysis report and design methodology
2. Identify properly hardware and software technologies accurately based on system requirements.
3. Identify the application of SSADM (Structured System Analysis and Design Methods) based on the backend system based on system requirements
4. Apply correctly object-Oriented Analysis and Design based on system requirements
5. Implement correctly stages for drawing SSADM based on system requirements
6. Apply correctly Phases in Object-Oriented Software development report and design methodology



Resources

Equipment	Tools		Materials
Computer	- Microsoft Office - E- Draw - Visual paradigm	- Draw io - Browser	- Internet - Electricity



Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Prepare Learning Materials: Develop or gather resources such as diagrams, case studies, or software tools that will support the lesson.
- Provide Context: Relate the topic to real-world applications or examples, making it relevant and engaging for students.
- Design Assessments: Create formative assessments to gauge understanding during and after the lesson, such as quizzes or project proposals.



Indicative content 2.1: Identification of System Design Tools



Duration: 5 hrs



Theoretical Activity 2.1.1: Identification of system design tools



Notes to the trainer:

- While delivering this content, a small group can be used to Identification of system design tools
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1:Introduce the activity and ask trainees to answer to the following questions:

- What is the purpose of system design in software development?
 - What are the key components involved in the system design process
 - What role do system design tools play in the development process, and what are some examples
- iv. Define the following terms:
- UML (Unified Modeling Language)
 - Algorithm
 - Flowchart
 - Data flow Diagram (DFD)
 - Entity Relation diagram (ERD)
 - Context diagram
 - Decision table
 - Use case diagram
 - Class Diagram
 - Decision tree

Step 2:Involve trainees in presentation of their findings.

Step 3:Provide expert view and clarifies ideas

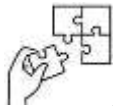
Step 4:Address any questions or concerns.

Step 5:Ask trainees to read the key reading 2.1.1 in the trainee manual.



Points to Remember

- **System Design:** Is the full process of thinking, developing, and creating the architecture, structure, and components of a system to meet specified criteria, functions, or purposes is known as system design.
- **UML (Unified Modeling Language):** is a standardized modeling language used to visualize the design of a system. It includes various diagrams like class diagrams, sequence diagrams, and use case diagrams, providing a comprehensive way to model software systems from different perspectives.
- **An algorithm:** is a set of defined steps designed to perform a specific objective. This can be a simple process, such as a recipe to bake a cake, or a complex series of operations used in machine learning to analyze large datasets and make predictions.
- **A flowchart** is a diagram that represents a process, showing the steps as boxes of various kinds, and their order by connecting them with arrows.
- **A Data Flow Diagram (DFD)** is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both.
- It shows how data enters and leaves the system, what changes the information, and where data is stored.
- **Entity-Relationship Diagrams**
- **ER-modeling** is a data modeling method used in software engineering to produce a conceptual data model of an information system. Diagrams created using this ER-modeling method are called Entity-Relationship Diagrams or ER diagrams or ERDs.
- **Decision Table** is a tabular representation that captures complex decision logic by mapping conditions to corresponding actions.
- **A use case diagram** is used to represent the dynamic behavior of a system. It encapsulates the system's functionality by incorporating use cases, actors, and their relationships.
- **Decision Tree** is a decision-making tool represented as a tree-like structure in Unified Modeling Language (UML). It shows various decision points (nodes), possible actions (branches), and the outcomes (leaves) based on specific conditions or rules. Each path from the root to a leaf represents a possible decision path.



Application of learning 2.1.

The Student Result Management System (SRIS) is a software application designed to manage and automate the student result processing for a school. The system allows for the entry, storage, processing, and reporting of student grades for various courses. Based on this project scenario, you can create the UML diagrams using standard symbols and notations.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	UML is well created (Unified Modeling Language)	1.1. Identify and create different UML diagrams (Class, Use Case, and Sequence).		
		1.2. Understand the relationships between Different UML diagrams.		
2	Algorithm is well created	2.1. Write step-by-step algorithms for solving simple problems.		
		2.2. Understand the importance of algorithm efficiency (time and space).		
3	Flowchart is well created	3.1 Create flowcharts to represent the logic of algorithms.		
		3.2. Flowchart symbols are correctly used (e.g., decision, process, and start/end).		
4	Data Flow Diagram (DFD)	4.1. Draw Level 0 (Context) and Level 1 DFDs accurately.		
		4.2 Understand the data flows, processes, data stores, and external entities.		
5	Entity-Relationship Diagram (ERD)	5.1 Create an ERD that correctly identifies entities, attributes, and relationships.		
		5.2 Differentiate between a context diagram and other DFD levels.		
6	Context Diagram	6.1. Differentiate between a context diagram and other DFD levels.		
		6.2 Identify system boundaries and external entities in a context diagram.		
7	Decision Table	7.1 Construct a decision table for a given set of conditions and actions.		

		7.2 Understand how to use decision tables for complex decision-making.		
8	Use Case Diagram	8.1 Create a use case diagram identifying actors and use cases.		
		8.2 Understand the relationships between use cases (e.g., include, extend).		
9	Class Diagram	9.1 Draw a class diagram with classes, attributes, methods, and relationships.		
		9.2 Understand concepts like inheritance, association, and multiplicity.		
10	Decision Tree	10.1 Create a decision tree for a given problem scenario		
		10.2 Understand the use of decision trees in making structured decisions.		



Indicative content 2.2: Identification of Hardware and Software Technology



Duration : 8 hrs



Theoretical Activity 2.2.1: Identification of hardware and software technology



Notes to the trainer:

- While delivering this content, a small group can be used to identify hardware and software technology in system design.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- i. Discuss on computer hardware as used in system design
- ii. Describe system software as used in system design
- iii. Explain application software needed to design backend

Step 2: Ask trainees to write their findings on papers

Step 3: Ask trainees to present their findings to whole class and trainer

Step 4: Provides expert view using didactic material.

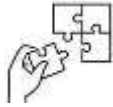
Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.2.1 in the trainee manual.



Points to Remember

- **CPU and Memory:** The CPU processes instructions, while RAM provides quick access to data for faster performance.
- **Storage and Networking:** Hard drives/SSDs store data long-term, and networking equipment enables connectivity.
- **Input/Output Devices:** Facilitate user interaction and data exchange with the system.
- **System Software:** The OS manages hardware and software; drivers and utilities optimize performance.
- **Application Software and Browsers:** Tools like Microsoft Office and web browsers support productivity and access to online resources.



Application of learning 2.2.

APC Ltd want to develop a system that records information of clients in their shop that sells products including delivery and payment, you are hired as backend developer responsible for identifying hardware and software to be used while developing that web application.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Hardware technology is well identified	1.1 Central Processing Unit is identified		
		1.2 Memory (RAM) is identified		
		1.3 Networking Equipment are identified		
		1.4 Input/Output Devices is identified		
		1.5 Storage Devices		
2	Software technology is well identified	2.1 Microsoft Office is identified		
		2.2 Visual Paradigm is identified		
		2.3 E-Draw Max is identified		
		2.4 Browser is identified		



Indicative content 2.3: Application of SSADM (Structured System Analysis and Design Methods)



Duration : 10 hrs



Theoretical Activity 2.3.1: Description of SSADM



Notes to the trainer:

- While delivering this content, a small group can be used to describe SSADM
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- Explain the objective of SSADM
- Describe SSADM Techniques

Step 2: Ask trainees to write their findings on papers

Step 3: Ask trainees to present their findings to whole class and trainer

Step 4: Provides expert view using didactic material.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.3.1 in the trainee manual.



Points to Remember

- SSADM provides a comprehensive framework for analysing and designing information systems.
- By utilizing techniques like logical data modeling, data flow modeling, and entity behavior modelling, developers can ensure that the resulting system is well structured, meets user needs, and is of high quality.



Practical Activity 2.3.2: Implementing stages for drawing SSADM



Notes to the trainer

- This activity should take place in computer lab where trainees must implement stages for drawing SSADM.
- Avail sample videos to be used as didactic material.



Key steps:

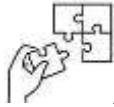
While delivering this activity, pass through the following steps:

- Step 1:** Introduce the activity and ask trainees to implement the stages of drawing SSADM.
- Step 2:** Explain clear work instructions
- Step 3:** Demonstrate how to implement the stages of drawing SSADM, while demonstrating explain the steps to design SSADM.
- Step 4:** Ask trainees to draw SSADM and monitor the activity.
- Step 5:** Verify whether the SSADM have been drawn
- Step 6:** Ask trainees to read key reading 2.3.2 in trainee manual.



Points to Remember

- By following these stages, organizations can effectively implement SSADM, ensuring that their systems are well-analyzed, designed, and maintained to meet the evolving needs of users and the business:
 - ✚ Feasibility Study
 - ✚ Requirements Analysis
 - ✚ Logical System Specification
 - ✚ System Design
 - ✚ Implementation
 - ✚ Testing and Validation
 - ✚ Deployment
 - ✚ Maintenance



Application of learning 2.3.

APC Ltd want to develop a system that records information of clients in their shop that sells products including delivery and payment, you are hired as backend developer responsible for implementing stages for drawing SSADM while developing that web application.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Stages for drawing SSADM are well implemented	1.1 Feasibility Study is performed		
		1.2 Requirements Analysis is performed		
		1.3 Logical System Specification is performed		
		1.4 System Design is performed		
		1.5 Implementation is performed		
		1.6 Testing and Validation is performed		
		1.7 Deployment is performed		
		1.8 Maintenance is performed		



Indicative content 2.4: Application of Object-Oriented Analysis and Design



Duration: 8 hrs



Theoretical Activity 2.4.1: Description of Object-Oriented Analysis and Design



Notes to the trainer:

- While delivering this content, a small group can be used for describing OOAD.
- Avail sample videos or images to be used as didactic materials.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- Explain the term OOAD
- Describe the Advantages and disadvantages of OOAD
- Describe the phases in object-oriented software development

Step 2: Ask trainees to write their findings on papers

Step 3: Ask trainees to present their findings to whole class and trainer

Step 4: Provides expert view using didactic material.

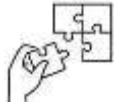
Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 2.4.1 in the trainee manual.



Points to Remember

- Object-Oriented Principles: Focus on encapsulation, inheritance, polymorphism, and abstraction.
- OOAD in SDLC: Iteratively analyze, design, and refine using object-oriented concepts throughout the software life cycle.
- System Modeling: Use UML diagrams to represent different perspectives—external, interaction, structural, and behavioral.
- Key Phases: Move from requirements gathering to analysis, design, implementation, testing, deployment, and maintenance.
- Modularity & Reusability: Emphasize designing modular, scalable, and reusable components for complex software systems.



Application of learning 2.4.

As a trainee, you have been assigned to develop a Digital Library Management System for a school. The school's existing system for managing books, borrowing records, and member information is manual, making it challenging for students and staff to keep track of available books, issue/return dates, and overdue notices.

Your task is to design and develop a digital system that will allow:

- Students and staff to search, borrow, and return books digitally.
- Library staff to manage book inventories and user records efficiently.
- The system to send automatic notifications about due dates and new arrivals.

You are expected to apply your knowledge of Object-Oriented Analysis and Design (OOAD) to complete this project.

Checklist

SN	Criteria	Indicators	Observation	
			Yes	No
1	Object-Oriented Analysis	1.1 Identify the Key Objects (e.g., User, Book, BorrowingRecord, LibraryStaff, Notification)		
		1.2 Define Object Attributes and Behaviors		
		1.3 Establish Relationships Between Objects		
		1.4 Develop Use Cases (e.g., "Search for a Book," "Borrow a Book," "Return a Book," "Add New Books to Inventory")		
		1.5 A comprehensive analysis model that outlines objects, attributes, relationships, and use cases		
2	Object-Oriented Design	2.1 Create Class Diagrams representing the structure of the system		
		2.2 Design Object Relationships		
		2.3 Select Design Patterns (e.g., Singleton for database connection, Observer for notifications)		
		2.4 Define System Architecture		
		2.5 A detailed design model specifying classes, relationships, and system architecture		



Learning outcome 2 end assessment

Theoretical assessment

1. Match the following terms with their definitions:

Answer	Term	Definition
....4.....	A) Data Flow Diagram	1) A diagram representing a sequence of operations
....3....	B) Class Diagram	2) A diagram depicting object interactions over time
.....2...	C) Sequence Diagram	3) A diagram showing the static structure of a system
.....1..	D) Activity Diagram	4) A diagram representing data flow within a system

2. Match the following UML diagrams with their purpose:

Answer	UML Diagram	Purpose
....2.....	A) State Machine Diagram	1) Represents system interactions and message exchange
....3.....	B) Use Case Diagram	2) Shows dynamic behavior using state transitions
.....1....	C) Communication Diagram	3) Shows high-level system functionality with actors
.....4....	D) Component Diagram	4) Describes the physical components of a system

3. Evaluate the Following Statements as True or False:

- a) A sequence diagram shows the interactions between objects in terms of the messages exchanged over time.

Answer: True

- b) An activity diagram portrays the static structure of the system by showing its classes and methods.

Answer: False

- c) A use case diagram encapsulates the functional requirements of a system by utilizing actors and use cases.

Answer: True

- d) The composite structure diagram displays the relationships between the parts within a class, focusing on its internal structure.

Answer: True

e) A package diagram is used to describe the dynamic behavior of a system and how objects interact over time.

Answer: False

4. What are the main advantages and disadvantages of using UML diagrams in system design

Answer:

Advantages:

1. **Standardization:** UML provides a standardized way to visualize system architecture, which helps in maintaining consistency across different projects and teams.
2. **Visual Communication:** UML diagrams offer a graphical representation of systems, making it easier for stakeholders, including non-technical ones, to understand the system's structure and behavior.
3. **Documentation:** UML diagrams serve as a form of documentation that can be referred to throughout the development lifecycle. They can help new team members understand the project quickly.
4. **Enhanced Design:** By modeling the system before coding, UML helps identify potential design flaws early in the process, leading to a more robust system architecture.
5. **Support for Various Perspectives:** UML includes multiple diagram types (like class diagrams, sequence diagrams, and use case diagrams) that address different aspects of the system, facilitating a comprehensive view of the system.
6. **Facilitates Object-Oriented Design:** UML is particularly well-suited for object-oriented programming and can help in defining the relationships between classes and objects.
7. **Tool Support:** Numerous tools support UML diagram creation and can automate some parts of the design process, improving productivity.

Disadvantages:

1. **Complexity:** UML can become complex and overwhelming, especially for larger systems. Managing and interpreting many diagrams can be difficult.
2. **Learning Curve:** For teams not familiar with UML, there may be a significant learning curve. Understanding the various types of diagrams and their proper use requires training.
3. **Overhead:** The process of creating and maintaining UML diagrams can be time-consuming, potentially diverting resources from actual development work.
4. **Ambiguity:** Different interpretations of UML diagrams can lead to misunderstandings among team members. Clear communication is essential to mitigate this.

5. **Not Always Necessary:** For smaller projects or those with a limited scope, UML may be overkill. Simpler documentation methods may suffice.
6. **Focus on Design Over Implementation:** UML emphasizes design aspects, which may lead teams to overlook practical implementation concerns.
7. **Static Nature:** UML diagrams often represent static views of a system, which might not capture dynamic behaviour adequately unless complemented by other documentation.

Practical assessment

Imagine you are a software system designer for a student-owned project. Each student should manage their own project tasks and progress. Begin by creating a use case diagram to identify the primary actors (such as students, project mentors, and system administrators) and their interactions with the system. Next, develop a class diagram that outlines the system's structure, including the classes (like Student, Task, Project, and Mentor), attributes (such as task description, deadline, and status), methods (such as addTask and updateProgress), and relationships (like Student works on Task).

Using this foundation, write an algorithm to manage the project tasks and progress tracking, ensuring it efficiently handles various scenarios, including task assignment and status updates. Then, create a flowchart to represent this algorithm visually, considering the algorithm's efficiency in terms of time complexity.

Following this, draw a Level 0 Data Flow Diagram (DFD) to illustrate the main processes (such as task management and progress tracking), data flows (like task details and student updates), and external entities (such as students and mentors) involved in the system. Develop an Entity-Relationship Diagram (ERD) that captures the data requirements, detailing the entities (like Tasks, Projects, and Students), attributes, and relationships. Explain how the DFD and ERD complement each other regarding data flow and storage.

Finally, for a task prioritization process within this system (for students managing multiple tasks), create a decision table that includes conditions such as task urgency, student workload, and deadlines. Construct a decision tree from this table to visualize the decision-making process. Discuss how the decision table and decision tree help in making structured decisions for this scenario.

END



Further information to the trainer

Books:

Bird, S. K. (2009). Natural language processing with Python. O'Reilly Media.

Elgendy. (2015). Business analysis for beginners. Outskirts Press.

Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.

Hohpe, G. &. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley.

Kendall, K. E. (2013). Systems analysis and design. Pearson.

Kotonya, G. &. (1998). Requirements engineering: Processes and techniques. Wiley.

McConnell, S. (2004). A practical handbook of software construction. Microsoft Press.

McConnell, S. (2004). Documenting system backend requirements. Microsoft Press.

Pressman, R. S. (2014). Software engineering. McGraw-Hill Education.

Satzinger, J. W. (2012). Systems analysis and design. Course Technology, Cengage Learning.

Whitten, J. L. (2004). Systems analysis and design methods. McGraw-Hill.

Web links:

Ambler, S. W. (n.d.). Agile requirements change management. Retrieved from <http://www.agilemodeling.com/essays/requirementsChangeManagement.htm>

Backend developer. (n.d.). Retrieved from Guru99: <https://www.guru99.com/backend-developer.html>

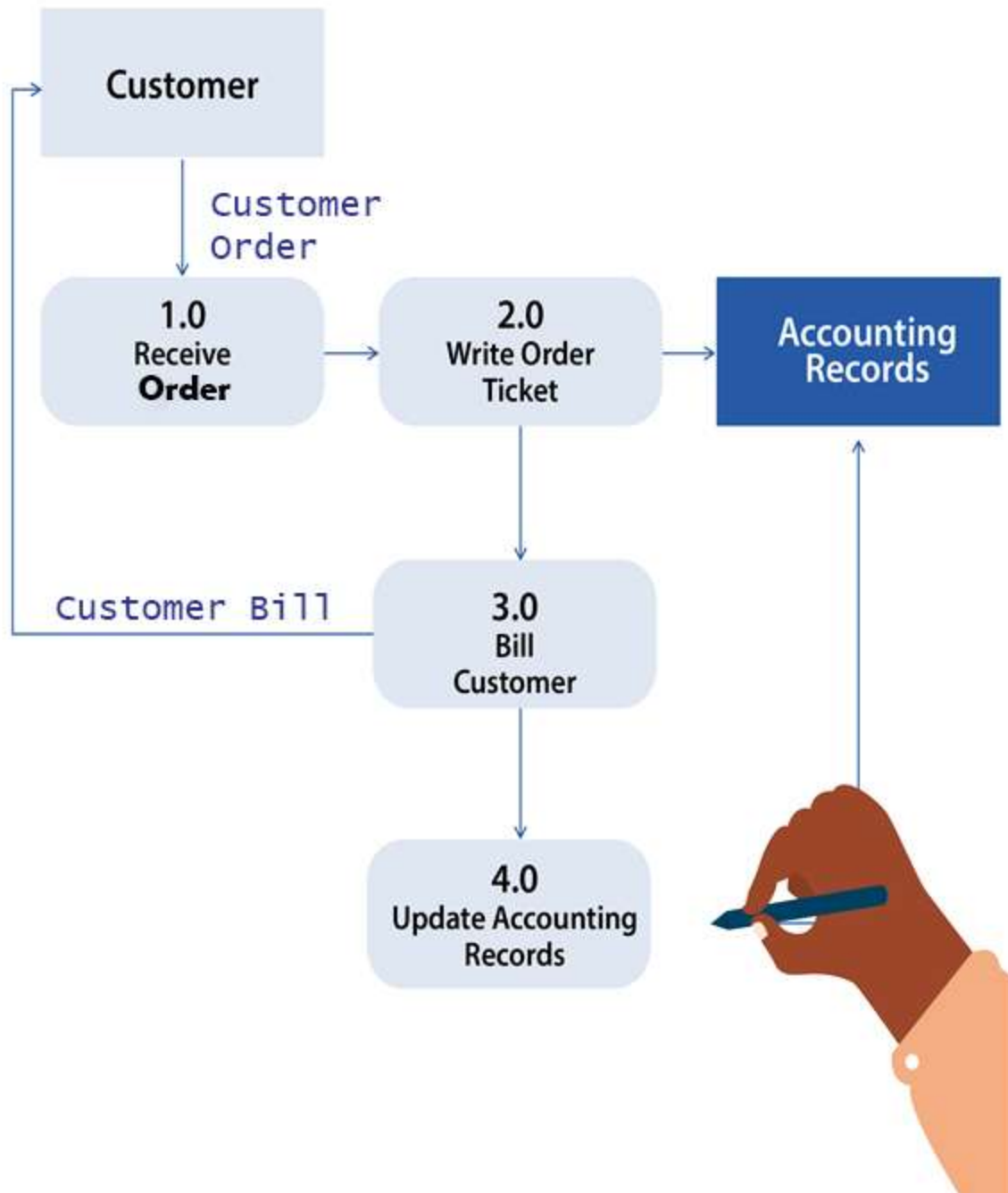
How to gather requirements. (n.d.). Retrieved from Lucidchart: <https://www.lucidchart.com/blog/how-to-gather-requirements>

Stackify. (n.d.). Retrieved from what is backend: <https://stackify.com/what-is-backend/>

Techopedia. (n.d.). Retrieved from FURPS: <https://www.techopedia.com/definition/25949/furps>

Tutorials Point. (n.d.). Retrieved from SDLC overview: https://www.tutorialspoint.com/sdlc/sdlc_overview.htm

Learning Outcome 3: Build System Design



Indicative contents

3.1 Development of Data Flow

3.2 Application of Physical Data Model

3.3 Documentation of system design

Key Competencies for Learning Outcome 3: Build System Design

Knowledge	Skills	Attitudes
● Identification of database Objects ● Identification of Software tools ● Description of Types of documentation of system design	● Developing Data flow diagram of system ● Designing Database ● Developing System documentation ● Developing User documentation	● Having Team work ● Being critical thinker ● Being Innovative ● Being attentive. ● Being creative ● Having Curiosity and Open Mindedness



Duration: 32 hrs

Learning outcome 3 objectives:



By the end of the learning outcome, the trainees will be able to:

1. Identify correctly software design tools based on the system design Requirements
2. Identify clearly database Objects based on the system design Requirements
3. Describe correctly types of documentation of system design based on the system design Requirements
4. Develop properly data Flow diagram based on the system design Requirements
5. Design correctly Database based on the system design Requirements
6. Develop correctly System documentation based on system analysis and architecture
7. Prepare correctly user documentation based on system analysis and architecture



Resources

Equipment	Tools	Materials
• Computer	• Microsoft office • Visual paradigm • E- Draw max • Browser	• Internet



Advance Preparation:

Before delivering this learning outcome, you are recommended to:

- Have a well-prepared Computer installed Application Software such as Microsoft office, Visual paradigm, E- Draw max and Browser



Indicative content 3.1: Development of Data Flow Diagram



Duration: 12hrs



Theoretical Activity 3.1.1: Description of Data Flow Diagram



Notes to the trainer:

- While delivering this activity small groups can be used for describing data flow diagram.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

- i. Elements of DFD
- ii. Rules of Drawing a DFD
- iii. Describe types of DFD
- iv. Explain software tools used to design DFD

Step 2: Ask trainees to write their findings on papers

Step 3: Ask trainees to present their findings to whole class and trainer

Step 4: Provides expert view using didactic material.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 3.1.1 in the trainee manual.



Points to Remember

- **A Data Flow Diagram (DFD)** is a traditional visual representation of the information flows within a system. A neat and clear DFD can depict the right amount of the system requirement graphically. It can be manual, automated, or a combination of both
- **Elements of DFD**
 - Elements that may be included in a flowchart are a sequence of actions, materials or services entering or leaving the process (inputs and outputs), decisions that must be made, people who become involved, time involved at each step, and/or process measurements.
- **DFD rules and tips**
 - Each process should have at least one input and an output.
 - Each data store should have at least one data flow in and one data flow out.

- Data stored in a system must go through a process.
- All processes in a DFD go to another process or a data store
- **Data Flow Diagram types**
 - 🔗 Level 0 DFD
 - 🔗 Level 1 DFD
 - 🔗 Level 2 DFD



Practical Activity 3.1.2: Develop Data Flow diagram



Notes to the trainer

- This activity should take place in computer lab where trainees have to develop data flow diagram.
- Avail computer installed with Microsoft office
- Avail computers connected to the internet.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activities and ask trainees to do the task below:

Your school need to develop a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the Data flow diagram level 1 that will be used while developing the backend of that system.

Step 2: Explain the task and provide clear work instructions.

Step 3: Demonstrate how to develop data flow diagram, while demonstrating explain the steps to be followed.

Step 4: Ask trainees to develop DFD level 1 and monitor the activity.

Step 5: Verify whether the activity have been performed and provide feedback if necessary.

Step 6: Ask trainees to read key readings 3.1.2 in trainee manuals



Points to Remember

- **10 simple steps to draw a data flow diagram online with Lucidchart**

Now that you know what makes up a data flow diagram, let's see how easy it is to make one using our powerful, online tool. We provide a ton of templates to use as a starting point. In this how-to, we're going to create a Level 0 DFD for an online shopping experience. Log in to your account (if you don't have one, sign up to try Lucid chart free for a week) and follow the steps below to make a DFD.

1. Select a data flow diagram template
2. Name the data flow diagram
3. Add an external entity that starts the process
4. Add a Process to the DFD
5. Add a data store to the diagram
6. Continue to add items to the DFD
7. Add data flow to the DFD
8. Name the data flow
9. Customize the DFD with colors and fonts
10. Add a title and share your data flow diagram



Application of learning 3.1.

Your school need to develop a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the Data flow diagram level 1 that will be used while developing the backend of that system.

Checklist :

SN	Criteria	Indicators	Yes	No
1	Data flow diagram is well designed	1.1 symbols are identified		
		1. 2 Symbols are applied		
		1.3 Dataflow system is designed		



Indicative content 3.2: Application of Physical Data Model



Duration : 8hrs



Theoretical Activity 3.2.1: Description of database Objects



Notes to the trainer:

- While delivering this activity small groups can be used for describing database objects.



Key steps:

While delivering this activity, pass through the following steps:

Step 1:Introduce the activity and ask trainees to answer the following questions:

Describe the following database objects:

- Tables
- Constraints
- Indexes
- Triggers
- Sequences
- Views
- Schemas
- Synonyms

Step 2:Ask trainees to write their findings on papers

Step 3:Ask trainees to present their findings to whole class and trainer

Step 4:Provides expert view using didactic material.

Step 5:Address any questions or concerns.

Step 6:Ask trainees to read the key reading 3.2.1 in the trainee manual.



Points to Remember

You can create the following database objects in a DB2 database:

- ✓ Tables
- ✓ Constraints
- ✓ Indexes
- ✓ Triggers
- ✓ Sequences
- ✓ Views
- ✓ Usage lists



Practical Activity 3.2.2: Designing Database E Draw Max



Notes to the trainer

- This activity should take place in computer lab where trainees must design database.
- Avail computers installed with E- draw Max



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activities and ask trainees to do the task below:

Your school need to develop a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the database that will be used while developing the backend of that system.

Step 2: Explain the task and provide clear work instructions.

Step 3: Demonstrate how to design database, while demonstrating explain the steps to be followed.

Step 4: Ask trainees to design database and monitor the activity.

Step 5: Verify whether the activity have been performed and provide feedback if necessary.

Step 6: Ask trainees to read key readings 3.2.2 in trainee manuals



Points to Remember

- Relationships are meaningful associations between tables that contain related information — they're what make databases useful.
- Identifying the connections you'll need between tables is part of the data modeling and schema design process — that is, the process of figuring out how your data fits together, and how exactly you should configure your tables and their fields.



Application of learning 3.2.

Your school need to develop a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the database that will be used while developing the backend of that system.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	Database is well design	1.1 Entity is combined with their attributes		
		1. 2 arrow is placed according the given to tasks		
		1.3 Relationships between entities are organised		



Indicative content 3.3: Documentation of System Design



Duration : 12 hrs



Theoretical Activity 3.3.1: Description of Documentation of system design



Notes to the trainer:

- While delivering this activity small groups can be used for describing documentation of system design.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activity and ask trainees to answer the following questions:

Describe the following types of documentation of system design:

- ✚ System Design document (SDD)
- ✚ Functional Specification Document (FSD)
- ✚ Technical Specification Document (TSD)
- ✚ Database design Document
- ✚ Use case Document

Step 2: Ask trainees to write their findings on papers

Step 3: Ask trainees to present their findings to whole class and trainer

Step 4: Provides expert view using didactic material.

Step 5: Address any questions or concerns.

Step 6: Ask trainees to read the key reading 3.3.1 in the trainee manual.



Points to Remember

- **System Design Document (SDD):** Provides a detailed description of the system architecture and design, including components, interfaces, and data flow.
- **Functional Specification Document (FSD):** Defines the system's functionality, detailing how the system should behave and what it should do.
- **Technical Specification Document (TSD):** Outlines the technical details of the system, including hardware, software, and network requirements.
- **Database Design Document:** Documents the structure of the database, including tables, relationships, and constraints.

- **Use Case Document:** Describes specific scenarios of system interaction, detailing how users will interact with the system to achieve a goal.
- **User Documentation:** Offers instructions and guides for end-users on how to use the system's features and functionalities.



Practical Activity 3.3.2: Developing system documentation for a backend system



Notes to the trainer

- This activity should take place in computer lab where trainees have to develop system documentation.
- Avail GitHub accounts for students to be used.
- Avail computer connected to internet
- Avail Ruby set file on external driver



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activities and ask trainees to do the task below:

Your school developed a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the system documentation that will be used as guide including the following files

- ✓ /_config.yml
- ✓ /index.md
- ✓ /architecture.md
- ✓ /api.md
- ✓ /database.md
- ✓ /deployment.md
- ✓ /troubleshooting.md.

Step 2: Explain the task and provide clear work instructions.

Step 3: Demonstrate how to design database, while demonstrating explain the steps to be followed.

Step 4: Ask trainees to design database and monitor the activity.

Step 5: Verify whether the activity have been performed and provide feedback if necessary.

Step 6: Ask trainees to read key readings 3.3.2 in trainee manuals



Points to Remember

- While developing system documentation in GitHub pages you follow those steps:
 - ✚ Create a GitHub Repository
 - ✚ Enable GitHub Pages
 - ✚ Choose a Documentation Framework
 - ✚ Set Up Your Local Environment
 - ✚ Organize Your Documentation Structure
 - ✚ Write Documentation
 - ✚ Add Visuals
 - ✚ Customize Your Site
 - ✚ Preview Locally
 - ✚ Push Changes to GitHub
 - ✚ Access Your Documentation
 - ✚ Maintain and Update
 - ✚ Promote Documentation Usage



Practical Activity 3.3.3: Developing User documentation



Notes to the trainer

- This activity should take place in computer lab where trainees must develop user documentation.
- Avail GitHub accounts for students to be used.



Key steps:

While delivering this activity, pass through the following steps:

Step 1: Introduce the activities and ask trainees to do the task below:

Your school developed a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop User documentation that will be used as guide including the following files

- ✓ /_config.yml
- ✓ /index.md

- ✓ /architecture.md
- ✓ /api.md
- ✓ /database.md
- ✓ /deployment.md
- ✓ /troubleshooting.md.

Step 2: Explain the task and provide clear work instructions.

Step 3: Ask trainees to develop the system documentation and monitor the activity.

Step 4: Verify whether the activity have been performed and provide feedback if necessary.

Step 5: Ask trainees to read key readings 3.3.2 in trainee manuals



Points to Remember

- While developing user documentation in GitHub pages you follow those steps:
 - ✚ Create a GitHub Repository
 - ✚ Enable GitHub Pages
 - ✚ Choose a Documentation Framework
 - ✚ Set Up Your Local Environment
 - ✚ Organize Your Documentation Structure
 - ✚ Write Documentation Content
 - ✚ Add Visuals
 - ✚ Customize Your Site
 - ✚ Preview Locally
 - ✚ Push Changes to GitHub
 - ✚ Access Your Documentation Online
 - ✚ Maintain and Update Regularly
 - ✚ Promote Documentation Usage
 - ✚ Gather Feedback



Application of learning 3.3.

Your school developed a library system that will be used while managing the process of renting books and returning them in library, as backend developer you are requested to develop the system and user documentation that will be used as guide.

Checklist:

SN	Criteria	Indicators	Observation	
			Yes	No
1	System documentation is well created	1.1. GitHub Repository is created		
		1.2. GitHub Pages is enabled		
		1.3. Documentation Framework is selected		
		1.4. Set Up Your Local Environment		
		1.5. Documentation Structure is organised		
		1.6. Documentation is written		
		1.7. Customize Your Site		
		1.8. Preview Locally		
		1.9. Push Changes to GitHub		
		1.9. Access Your Documentation		
2	User documentation is well developed	2.1. Organize Your Documentation Structure		
		2.2. Write Documentation Content		
		2.3. Add Visuals		
		2.4. Customize Your Site		
		2.5. Preview Locally		
		2.6. Access Your Documentation Online		



Learning outcome 3 end assessment

Theoretical assessment

I. Multiple Choice Questions

1. What is the purpose of a Data Flow Diagram (DFD)?

- A) To display the physical layout of hardware components
- B) To represent the flow of data within a system
- C) To document user manuals
- D) To visualize software code structure

Answer: B) To represent the flow of data within a system

2. Which of the following is NOT an element of a Data Flow Diagram (DFD)?

- A) Data Store
- B) Process
- C) Entity
- D) Network

Answer: D) Network

3. What is the main difference between Level 0 and Level 1 DFD?

- A) Level 0 DFD is more detailed than Level 1 DFD
- B) Level 1 DFD provides a high-level overview, while Level 0 DFD is detailed
- C) Level 0 DFD is the context diagram, while Level 1 DFD breaks down processes into more detail
- D) There is no difference between Level 0 and Level 1 DFD

Answer: C) Level 0 DFD is the context diagram, while Level 1 DFD breaks down processes into more detail

4. Which software tool is commonly used for creating Data Flow Diagrams?

- A) Adobe Photoshop
- B) Microsoft Office
- C) Visual Paradigm
- D) Oracle Database

Answer: C) Visual Paradigm

5. In the context of database design, what is a 'relationship'?
- A) A link between two tables that defines how data in one table relates to data in another
 - B) A connection between two software applications
 - C) A feature in Microsoft Word
 - D) A data storage unit

Answer: A) A link between two tables that defines how data in one table relates to data in another

6. Which document details the functionality of a system?
- A) Technical Specification Document (TSD)
 - B) Functional Specification Document (FSD)
 - C) System Design Document (SDD)
 - D) Database Design Document

Answer: B) Functional Specification Document (FSD)

II. Answer by True or False for the following Statements

1. A Data Flow Diagram (DFD) is used to represent the flow of data within a system.

Answer: True

2. Level 2 DFD is typically the highest level, providing a very broad overview of the system.

Answer: False (Level 2 DFD involves further decomposition of processes for more detail.)

3. Microsoft Office is a preferred tool for creating Data Flow Diagrams.

Answer: False (While it can be used, tools like Visual Paradigm or E-draw are more suited for creating DFDs.)

4. In database design, identifying database objects includes defining tables and their relationships.

Answer: True

5. System Design Document (SDD) primarily focuses on the technical aspects of system implementation.

Answer: False (SDD focuses on the overall system design, not just the technical aspects.)

6. Use Case Documents are a type of system design documentation.

Answer: True

7. The Physical Data Model represents the logical structure of the database.

Answer: False (It represents the actual implementation of the database on storage media.)

8. Technical Specification Document (TSD) includes detailed information about the system's hardware requirements.

Answer: True

Practical assessment

To address the development and design of the Estg school management system located in Rubavu, Rwanda, we will go through several key aspects, including the development of a Data Flow Diagram (DFD), application of a Physical Data Model, identification and design of database objects, and documentation of the system design. You are hired to develop dataflow diagram, physical data model and design database object and make documentation of the system design.

SN	Criteria	Indicators	Observation	
			Yes	No
1	Data flow diagram is well design	1.1 symbols are identified according the given to tasks		
		1. 2 Symbols is well applied according the given to tasks		
		1.3 Dataflow system is clearly design according the given to tasks		
2	Database is well design	2.1 Entity is well combined with their attributes according the given to tasks		
		2. 2 Arrow is well placed according the given to tasks		
		2.3 Relationships between entities are well organised according the given to tasks		
3	system design is properly documented	3.1 system design is well documented according to the given task		

END



Further information to the trainer

Books:

- Bird, S. K. (2009). Natural language processing with Python. O'Reilly Media.
- Elgendy. (2015). Business analysis for beginners. Outskirts Press.
- Fowler, M. (2002). Patterns of enterprise application architecture. Addison-Wesley.
- Hohpe, G. &. (2004). Enterprise integration patterns: Designing, building, and deploying messaging solutions. Addison-Wesley.
- Kendall, K. E. (2013). Systems analysis and design. Pearson.
- Kotonya, G. &. (1998). Requirements engineering Processes and techniques. Wiley.
- McConnell, S. (2004). A practical handbook of software construction. Microsoft Press.
- McConnell, S. (2004). Documenting system backend requirements. Microsoft Press.
- Pressman, R. S. (2014). Software engineering. McGraw-Hill Education.
- Satzinger, J. W. (2012). Systems analysis and design. Course Technology, Cengage Learning.
- Sommerville. (2015). Software engineering. Pearson.

Web links:

- Ambler, S. W. (n.d.). Agile requirements change management. Retrieved from <http://www.agilemodeling.com/essays/requirementsChangeManagement.htm>
- Backend developer. (n.d.). Retrieved from Guru99: <https://www.guru99.com/backend-developer.html>
- How to gather requirements. (n.d.). Retrieved from Lucidchart: <https://www.lucidchart.com/blog/how-to-gather-requirements>
- Mozilla Developer Network. (n.d.). Retrieved from JSON: <https://developer.mozilla.org/en-US/docs/Learn/JavaScript/Objects/JSON>
- Stackify. (n.d.). Retrieved from What is backend: <https://stackify.com/what-is-backend/>
- Techopedia. (n.d.). Retrieved from FURPS: <https://www.techopedia.com/definition/25949/furps>
- Tutorials Point. (n.d.). Retrieved from SDLC overview: https://www.tutorialspoint.com/sdlc/sdlc_overview.htm



October 2024