Republic of Rwanda
Ministry of Education

RTB | RWANDA TVET BOARD

Advanced Java

SPEAJ402

Develop Software using Java

Competence

RQF Level: 4

Learning Hours 120

Credits:12

Sector: ICT and Multimedia

Trade: Software Programming and Embedded Systems

Module Type: Specific Module

Curriculum: ICTSES4002 TVET Certificate IV in Software Programming and Embedded Systems

1200

Issue Date: August 2023

<table>
<tr><td>Purpose statement</td><td colspan="6">This module aims to equip learners with knowledge and skills necessary to develop sophisticated software solutions using advanced Java programming techniques.

By the end of this module, learners will be able to Apply Java Web Application Technologies, Develop Backend using Java, Build microservices, Manipulate file based data.

The learners will be able to carry out the above tasks with confidence and minimal supervision.</td></tr>
<tr><td>Learning assumed to be in place</td><td colspan="6">Fundamentals of OOP with Java, Basics of Databases Development, Discrete Mathematics</td></tr>
<tr><td rowspan="6">Delivery modality</td><td colspan="2">Training delivery</td><td>100 %</td><td colspan="2">Assessment</td><td>Total 100%</td></tr>
<tr><td colspan="2">Theoretical content</td><td>30%</td><td rowspan="4">Formative assessment</td><td>30%</td><td rowspan="4">50%</td></tr>
<tr><td colspan="2">Practical work:</td><td rowspan="3">70%</td><td rowspan="3">70%</td></tr>
<tr><td>Group project and presentation</td><td>30%</td></tr>
<tr><td>Individual project /Work</td><td>70%</td></tr>
<tr><td colspan="3"></td><td colspan="2">Summative Assessment</td><td>50%</td></tr>
</table>

Elements of Competence and Performance Criteria

Elements of competence	Performance criteria
1. Apply Java Web Application Technologies	1.1. The Java Servlets are appropriately developed based on java documentation
	1.2. Java server pages are properly developed based on Java EE Documentation
	1.3. The java expression language, JSP Standard Tag Library and user defined tags are efficiently applied based on Java EE documentation.
	1.4. Web application security is properly handled in accordance with the defined security measures.

2. Develop Backend using Java	2.1. Spring Boot concepts are properly applied. based on java documentation
	2.2. Hibernate is properly integrated with Spring Boot's JPA Support based on java documentation
	2.3. Reactive programming is applied to enhance Java application based on java documentation
	2.4. Web sockets are properly implemented using Spring Boot based on java documentation
	2.5. Reporting & analytics tools are swiftly integrated in java application based on java documentation
3. Build microservices	3.1. Cloud concepts are properly described based on Cloud Security Alliance (CSA) guidelines
	3.2. Microservices are efficiently designed based on Spring Boot documentation
	3.3. Service discovery and load balancing for microservices are properly implemented based Java documentation
	3.4. Microservices are properly dockerized and deployed according to docker documentation
4. Manipulate file based data	4.1. I/O files in Java are properly manipulated according to Java documentation
	4.2. Data is appropriately transferred with SFTP/FTP for file based systems based on Java documentation
	4.3. The system CRON schedule is efficiently built to automate data transfer schedules based on Java documentation.

Intended Knowledge, Skills and Attitude

Knowledge	Skills	Attitude
✓ Understand popular Java frameworks ✓ Explain data access object concepts ✓ Describe the concepts of microservice architecture ✓ Understand version control systems	✓ Set up development environment and install selected libraries ✓ Apply required framework ✓ Perform integration testing ✓ Configure deployment environment ✓ Configure selected database ✓ Develop servlets ✓ Design and create JSP layout ✓ Build microservices ✓ Apply Scriptlet and expression language (EL) ✓ Handle of Web Application Security ✓ Use version control systems ✓ Optimise system performance ✓ Develop backend ✓ Consume APIs ✓ Apply i18N	✓ Use creativity and innovation throughout the UI design ✓ Use creativity and innovation throughout the software architecture ✓ Familiarize with testing tools ✓ Have positive attitude towards problem-solving and a willingness to tackle complex problems ✓ Be eager to learn about the latest advancements in Java and related technologies ✓ Have willingness to adapt to new tools, libraries, and best practices ✓ Be carefully to details, especially when it comes to writing clean, efficient, and maintainable code ✓ Work effectively in a team, communicate ideas clearly, and collaborate with team members

Course content

Learning outcome 1: Apply Java Web Application Technologies	Learning hours: 25 Hours

Indicative content

- **Development of Java Servlets**
  - ✓ Java Servlets
  - ✓ Role of Java Servlets.
  - ✓ Servlet lifecycle and request/response handling.
    - Overview of HTTP Basics
    - ServletRequest and ServletResponse
    - Servlet Initialization and Configuration
    - URL Mapping
    - Form Data Processing
    - Asynchronous Processing
    - Servlet Collaboration
    - Servlet filters
  - ✓ servlets containers
  - ✓ Session Management.
  - ✓ Servlet Error & Exception handling
  - ✓ MVC Architecture with Servlets.
  - ✓ Servlet Security considerations:
    - Authentication & Authorization
    - Secure communications

Development of Java Server Pages (JSP)

- ✓ introduction to Java Server Pages
- ✓ JSP Architecture
- ✓ JSP scripting elements, directives, and page lifecycle.
  - EL (Expression Language)
  - JSTL (JavaServer Pages Standard Tag Library)
  - Custom tags
  - JSP Templates & Tag files
  - JSP implicit Objects
  - JSP Error pages
- ✓ MVC Architecture with JSP
- ✓ Internationalization with JSP (i18n)
- ✓ JSP security

- Cross site Scripting Prevention
- Secure Configuration

Application of Java Expression Language and JSP Standard Tag Library

- ✓ Java Expression Language (EL)
- ✓ EL integration with JSP…
  - JSP EL Basics
  - EL operators & expressions
  - Functions & Method call with EL
  - use of EL in attribute values
- ✓ JSP Standard Tag Library(JSTL)
  - Core tags
  - Formatting & localization tags
  - JSTL functions
  - use of JSTL in attribute values
- ✓ Internationalization using JSTL
- ✓ EL & JSTL Integration
- ✓ JSTL Performance considerations.
  - Tag Reuse
  - Caching

Handling of Web Application Security

- ✓ OWASP web application security threats and vulnerabilities
- ✓ Mitigation of web security threats & vulnerabilities:
  - Cross-Site Scripting (XSS)
  - Cross-Site Request Forgery (CSRF)
  - SQL Injection
  - Authentication and Authorization
- ✓ Input validation and Output encoding
- ✓ Data sanitization
- ✓ Security Headers
- ✓ Security testing
- ✓ Threat modelling
- ✓ HTTPS implementation

Equipment	Computers, projector,
Materials	slide_decks, code examples, Books & Online resources:
Tools	IDEs(IntelliJ, Eclipse, TextEditor (VS code), Database Management Tool, Web Server(Apache), Security testing tools (OWASP ZAP, Burp Suite, etc,....)
Facilitation techniques	Brainstorming Group discussion on ‘security threats’ Demonstration on ‘JSP’, ‘Servlets’’, ‘mitigation of security threats’’ Practical exercise on ‘JSP, Servlets, security’

Formative assessment methods /(CAT)	Oral assessment Written assessment Practical assessment

Learning outcome 2: Develop Backend using Java	Learning hours: 45 Hours

Indicative content

- **Application of Spring Boot concepts**
  - ✓ Introduction to Spring Boot
  - ✓ Auto-Configuration
  - ✓ Starter Dependencies
  - ✓ Spring Boot Initializr
  - ✓ Embedded Servers
  - ✓ Spring Boot Actuator
  - ✓ Application Properties and YAML Configuration
- **Hibernate with Spring Boot**
  - ✓ Introduction to Object-Relational Mapping (ORM).
  - ✓ Hibernate Architecture
  - ✓ Hibernate configuration
    - properties
    - YAML files
  - ✓ Annotations
  - ✓ Mapping Java classes to Database tables
    - common mapping
    - Inheritance mapping
    - Association mappings
    - Embedded objects
  - ✓ Hibernate Query Language (HQL)
  - ✓ CRUD operations with Hibernate
  - ✓ Criteria API
  - ✓ Lazy Loading & Eager Fetching
  - ✓ Caching Mechanisms
    - First level cache
    - Second level cache
  - ✓ Transactions in Hibernate
  - ✓ Concurrency Control
  - ✓ Hibernate integration with Spring Boot
  - ✓ Spring Data JPA with Hibernate

- ✓ NoSQL with Hibernate

- **Reactive programming**
  - ✓ Introduction to reactive programming.
  - ✓ Reactive libraries in java
    - Reactor
    - RxJava
  - ✓ Reactive APIs and endpoints with Spring WebFlux
  - ✓ Handling streams of data
    - Flux
    - Mono
  - ✓ Reactive UI components using Reactor and JavaFX
  - ✓ Reactive streams for complex data processing
  - ✓ Hot vs Cold Publishers
  - ✓ Integration with databases & external APIs
    - R2DBC
  - ✓ Backpressure
  - ✓ Errors & Exception Handling
- **Web sockets in Spring Boot**
  - ✓ Introduction to web sockets.
  - ✓ Web sockets configuration in Spring Boot
  - ✓ Web Sockets Endpoint handling
  - ✓ Real-time messaging with STOMP
  - ✓ Broadcasting & Sub-Pub
  - ✓ Security considerations for Websockets
- **Integrating Reporting & Analytics Tools**
  - ✓ Introduction reporting & analytics tools in Java ecosystem
  - ✓ JasperReports
  - ✓ BIRT
  - ✓ Reporting tools integration
  - ✓ Dynamic reports generation
  - ✓ Data visualization

Resources required for the Learning outcome

Equipment	Computers, projector,
Materials	slide_decks, code examples, Books & Online resources
Tools	IDEs(IntelliJ, Eclipse, TextEditor (VS code), Database Management Tool, Web Server(Apache), Security testing tools (OWASP ZAP, Burp Suite, etc,....)
Facilitation techniques	Brainstorming Group discussion Demonstration Practical exercise

Formative assessment methods /(CAT)	Oral assessment Written assessment Practical assessment

Learning outcome 3: Build Microservices	Learning hours: 35 Hours

Indicative content

Cloud Concepts

- ✓ Virtualization
- ✓ Cloud architecture
- ✓ Cloud Service models
- ✓ Cloud Deployment models
- ✓ Cloud providers
- ✓ Microservices architecture in cloud environment

Microservices design with Spring Boot

- ✓ Introduction to microservice architecture
- ✓ monoliths vs microservices
  - DDD principles
- ✓ Microservices communication patterns
  - synchronous
  - asynchronous
  - event-driven architectures
- ✓ Domain modeling in microservices
- ✓ Containerization
  - docker container
  - container orchestration
  - kubernetes
- ✓ API gateway
- ✓ Data management
- ✓ Fault Tolerance and Resilience
  - circuit breakers
  - retries
  - fallback mechanism
- ✓ monitoring, logging and tracing
  - Spring Cloud Sleuth
  - Zipkin

Service Discovery and Load Balancing for Microservices

- ✓ Service Discovery in Microservices Architecture
- ✓ Service Discovery with Spring Cloud
  - Netflix Eureka
  - HashiCorp Consul
- ✓ Service registry setup
- ✓ Load balancing
  - dynamic load balancing
  - client-side load balancing
  - server-side load balancing
  - load balancing algorithms
- ✓ dynamic scaling and elasticity
- ✓ Security considerations

Dockerization and Deployment of Microservice

- ✓ Introduction to Containerization and Docker
  - image
  - container
  - networking
  - volumes
- ✓ Docker images with Dockerfiles
- ✓ container orchestration
  - docker compose
  - docker swan
- ✓ Microservice deployment with kubernetes
  - introduction to kubernetes
  - components (pods, services, deployments,..)
  - kubernetes configuration files

Resources required for the Learning outcome

Equipment	Computers, projector,
Materials	slide_decks, code examples, Books & Online resources:
Tools	IDEs(IntelliJ, Eclipse, TextEditor (VS code), Database Management Tool, Web Server(Apache)
Facilitation techniques	Brainstorming Group discussion on 'spring best practices' Demonstration on 'configuring Hibernate', 'setting up Spring application' Practical exercise on 'crecsaate Hibernate mappings, implementation of Spring components'
Formative assessment	Oral assessment Written assessment Practical assessment

methods /(CAT)	

Learning outcome 4: Manipulate file based data	Learning hours:15 Hours

Indicative content

- **Manipulation of I/O Files in Java**
  - ✓ Introduction to Java I/O
  - ✓ File Class and Paths
  - ✓ Streams and Readers/Writers
  - ✓ Streams.
    - Stream Classes (InputStream, OutputStream, reader, writer)
    - Byte streams
    - Character streams
    - Buffered streams
  - ✓ Reading and Writing Text Files
  - ✓ Reading and Writing Binary Files
  - ✓ File Compression
  - ✓ Working with Directories
  - ✓ File I/O Exception Handling
  - ✓ Character Encoding
  - ✓ File Serialization
  - ✓ NIO (New I/O)
  - ✓ Java I/O best practices
- **Data Transfer with SFTP/FTP for File-Based Systems**
  - ✓ SFTP vs. FTP

- ✓ SFTP/FTP Libraries in Java
- ✓ Establishing SFTP/FTP Connection
- ✓ Uploading files using SFTP/FTP
- ✓ Downloading files using SFTP/FTP
- ✓ Directory Operations
- ✓ Progress Monitoring
- ✓ Handling Large Files
- ✓ SFTP Security Features
- ✓ SFTP/FTP Error Handling

- **Building of System CRON Schedule for Automated Data Transfer**
  - ✓ Introduction to CRON Scheduling
  - ✓ CRON Syntax
  - ✓ CRON Components
  - ✓ CRON Special characters
  - ✓ CRON Jobs.
    - CRON Job Setup
    - CRON Job output and logging
  - ✓ Advanced CRON features.
    - Environment variables
    - Email alerts
    - Management of multiple CRON schedules
  - ✓ Data Transfer Automation using CRON
  - ✓ CRON alternatives
  - ✓ CRON best practices

Resources required for the Learning outcome

Equipment	Computers, projector,
Materials	slide_decks, code examples, Books & Online resources
Tools	IDEs(IntelliJ, Eclipse, TextEditor (VS code), CRON job scheduler(Quartz Scheduler, Spring Scheduler,), SFTP/FTP libraries,
Facilitation techniques	• Brainstorming • Group discussion on 'spring best practices' • Demonstration on 'I/O classes, ', 'SFTP/FTP, setting up CRON Jobs'' • Practical exercise on 'automated data transfer with CRON',
Formative assessment methods /(CAT)	• Oral assessment • Written assessment • Practical assessment

Integrated/Summative assessment (For specific module)

Integrated situation

Rwanda Healthcare Management System

Develop a comprehensive Java Web application addressing challenges in rural healthcare management within Rwanda. The project aims to provide a technology-driven solution to improve healthcare access and quality for Rwanda.

Tasks

★ Database Design and Mapping
- Design a robust database schema encompassing patient records, medical facilities, providers, and appointments.
- Utilize Hibernate ORM to seamlessly map Java classes to corresponding database tables, efficiently handling relationships and associations.

★ User Authentication and Access Control
- Implement a secure user authentication mechanism catering to healthcare providers, administrators, and patients.
- Apply a role-based access control framework to ensure precise data access and maintain confidentiality.

★ Medical Facilities and Services
- Showcase a user-friendly interface displaying a comprehensive list of accessible medical facilities across various healthcare centres.
- Incorporate a search functionality that allows users to locate facilities based on their location and available services.
- Enable users to view detailed information about medical facilities and the range of services they provide.

- ★ Appointments
  - Empower patients with the capability to conveniently schedule appointments with healthcare providers.
  - Implement a notification system to remind patients of upcoming appointments, enhancing appointment adherence.

- ★ Health Records and Analytics
  - Develop a dedicated feature enabling healthcare providers to effectively manage and access patient health records.
  - Implement rudimentary analytics to track prevalent health issues and trends within the rural community.
  -

- ★ Multithreaded Notifications
  - Leverage multithreading to automate the process of sending timely reminders to patients about upcoming appointments and prescribed medications.

- ★ Admin Dashboard
  - Provide administrators with a comprehensive dashboard to efficiently manage medical facilities, healthcare providers, and patient data.
  - Enable administrators to generate detailed reports highlighting patient visits, provider availability, and facility utilization.

- ★ Database Interaction and CRUD Operations
  - Implement Create, Read, Update, and Delete (CRUD) operations for diverse components, including patient records, appointments, medical facilities, and services.
  - Harness the capabilities of Hibernate for optimized data management and retrieval.

- ★ Caching and Performance Optimization
  - Integrate caching mechanisms to enhance the performance of frequently accessed patient data, reducing database load and latency.
  - Utilize caching annotations from frameworks like Spring Boot or Hibernate to streamline data retrieval.

- ★ Error Handling and Notifications
  - Implement a proactive error handling system capable of automatically identifying and addressing critical system issues.
  - Incorporate a notification mechanism that promptly alerts administrators or providers in case of system failures.

Deliverables

1. Database Schema design
2. Application source code
3. Video Demonstration
   - A video showcasing the application's features and user interactions.
4. Project Report:

- A short report discussing challenges faced, solutions implemented, and lessons learned.

Timing: 10 hours

Resources

Tools

- Windows Operating Systems
- Linux
- Internet
- Java
- Spring boot
- Hibernate
- DBMS
- Code Editors software.

Equipment	Computer
Materials/ Consumables	Internet, Electricity

Assessable outcomes	Assessment criteria (Based on performance criteria)	Indicator	Observation		Marks allocation
			Yes	No	
Learning outcome 1: Apply Java Web Application Technologies (20%)	1.1. The Java Servlets are appropriately described and developed based on java documentation	Ind.1.1. controllers are used			5
	1.2. Java server pages are properly described and developed.	Ind.2.1. JSP are used			10
	1.3. The java expression language and JSP Standard Tag Library are efficiently applied based on Java EE documentation.	Ind.2.1. jstl are used			5
		Ind.2.2. El is used			5
	1.4 Web application security is properly handled in accordance with the defined security measures.	Ind.2.1. The application is secured			5
		Ind.2.2. SQL injection is handled			5
		Ind.2.3. Faked URL are handled			5
		Ind.2.4. Authentication and authorisation are managed			5
Learning outcome 2: Develop backend using Java	2.1. Spring Boot concepts are properly applied	Ind.3.1. Spring boot is used			10
	3.2. Hibernate is properly integrated with Spring Boot's JPA Support.	Ind.3.1. Hibernate is integrated in Spring boot			10
		3.2 Repository, services are created			5
		Ind.3.3. End points are generated in respective controller			5

(30%)	3.4. Web sockets are properly implemented using Spring Boot	Ind.3.1. Sockets programming are used			5
	3.5 Reporting & analytics tools are swiftly integrated in java application	Ind.3.1 Reports are generated			5
		Ind.3.2 Dashboard is available			5
Learning outcome 3: Build microservices (30%)	4.1. Microservices are efficiently designed using Spring Boot	Ind.4.1. Backend is a microservice			5
	4.2 Microservices are properly dockerized and deployed according to docker documentation	Ind 4.1 Microservices are dokerised			5
Total marks					
Percentage Weightage					100%
Minimum Passing line % (Aggregate): 70%					

References:

1. Herbert Schildt, **Java: The Complete Reference,** Twelfth Edition 12th Edition, McGraw Hill, 2021

2. Ian Darwin, **Java Cookbook: Problems and Solutions for Java Developers 4th Edition,** O'Reilly Media, 2020

3. Joshua Bloch, **Effective Java 3rd Edition,** Addison-Wesley Professional, 2017

4. Cay S. Horstmann, **Core Java for the Impatient 3rd Edition**, Addison-Wesley, 2023

5. Linda Demichiel, Christian Bauer, Gavin King, Gary Gregory, **Java Persistence with Hibernate,** Manning, 2015

6. Craig Walls, **Spring Boot in Action,** Manning Publications Co, 2016

7. Magnis Larsson, **Microservices with Spring Boot and Spring Cloud: Build resilient and scalable microservices using Spring Cloud, Istio, and Kubernetes**, 2nd Edition 2nd ed. Edition, Packt Publishing; 2nd ed. edition,2021

8. Juha Hinkula, **Full Stack Development with Spring Boot and React: Build modern and scalable web applications using the power of Java and React**, 3rd Edition 3rd ed. Edition, Packt Publishing, 2022

Online Resources

1.https://docs.oracle.com/javaee/7/api/javax/servlet/Servlet.html

2. https://docs.oracle.com/javaee/7/tutorial/servlets.htm

3. https://docs.oracle.com/javase/tutorial/jdbc/basics/processingsqlstatements.html