Last Assignment

Instructions

1. This assignment contains six questions
2. Each group (G,H,L, I) is required to form not more than each group with the index number. i.e., Group 1, Goupe2…., Group 6. No Group should have more than 12 members.
3. One group will choose one question. The group index corresponds to the question number. i.e. Group 1 to Question 1, Group 2 to Question 2,., Group 6 to Question 6.
4. Only a group leader is required to submit to represent a group
5. Deadline is indicated in e-learning platform where you will submit it.
6. Any similarities, no matter how small, will be treated as collusion and penalized accordingly. Your work must be your own.
7. You may make (reasonable) use of ideas from third-party sources, provided these are cited as References within your answers and included in the Reference list at the end. Do not copy or closely paraphrase text. If any text is reproduced verbatim (i.e. if you quote a source – even the lecture slides), then this must be placed inside quotation marks "__". Make sure you follow these rules, otherwise your use of third-party sources and ideas may well be considered as plagiarism. Standard University policies apply to this and you can check them online for more guidance.

Questions1

The AUCA plans to digitize its entire Gishushu campus operations by developing a Smart Campus Management System (SCMS**)**. The system must integrate student registration, course management, hostel allocation, smart ID card access, automatic attendance tracking using RFID, digital payment for cafeteria services, and a campus-wide notification service. Students will register online, receive a digital ID, view allocated courses, and check their attendance. Lecturers will manage course materials, submit attendance, and issue assessments. Administrators will oversee student records, fees, staff profiles, and campus resources.

The system must interact with external services such as mobile money APIs, biometric scanners, and the national student verification database. Different user roles—students, lecturers, administrators, and support staff—must be authenticated differently. Data consistency and real-time synchronization are required across departments. The system must support high availability and offline-to-online syncing when network connectivity is poor.

Task:

Design the system using UML. Provide the following UML diagrams:

1. Use-Case Diagram showing all actors and major use cases
2. Class Diagram with attributes, methods, and relationships
3. Sequence Diagram for "Student Course Registration" and "Biometric Attendance"
4. Activity Diagram for the "Hostel Allocation Process"
5. Deployment Diagram showing servers, databases, and external systems

Question 2

A University Teaching Hospital-Kigali (CHUK) wants a digital Telemedicine Platform

allowing patients in rural areas to consult doctors remotely. Patients book appointments using a mobile app, describe symptoms, and upload medical documents. Doctors review patient history, conduct video consultations, prescribe medications, and request lab tests. The system automatically sends electronic prescriptions to partner pharmacies and receives lab test results from accredited laboratories.
The platform should integrate Artificial Intelligent (AI)-based symptom checking, secure payment processing, and an emergency alert button that immediately shares GPS location with the nearest ambulance service. Patient data must be encrypted in transit and at rest. Nurses, doctors, pharmacists, system administrators, and lab technicians each have different access rights. The system must log all activities for auditing and comply with health data privacy laws.

Task:

Develop UML diagrams including:

- Use-Case Diagram
- Class Diagram
- Sequence Diagram for "Video Consultation"
- Activity Diagram for "Lab Test Request and Result Handling"
- State Machine Diagram for "Patient Appointment Lifecycle"
- Deployment Diagram for mobile app, cloud servers, and third-party systems

Question 3

A startup plans to build a large-scale **multi-vendor e-commerce marketplace** where sellers can register, upload products, manage orders, and monitor sales analytics. Buyers browse products, add items to their cart, pay using multiple gateways, track orders in real time, and communicate with sellers via integrated chat. The system must support dynamic discounting, product recommendations, and an Artificial Intelligent (AI) fraud detection module.
Orders must pass through several stages: placed, confirmed, packed, shipped, delivered, or returned. Logistics partners update delivery status through APIs. Administrators handle dispute resolutions, refunds, and platform-wide configurations. Sellers have dashboards for revenue analytics and stock forecasting.

Task:

Develop the required UML diagrams:

1. Use-Case Diagram showing administrators, sellers, buyers, and logistics partners
2. Class Diagram detailing product, order, payment, cart, and seller modules
3. Sequence Diagrams for "Order Placement" and "Refund Process"
4. Activity Diagram for "Seller Product Upload Workflow"
5. Component Diagram for system architecture
6. Deployment Diagram showing cloud infrastructure

Question 4

A transportation authority wants to implement a real-time ticketing and fleet management system for buses across the city. Passengers should pay using smart cards, QR codes, mobile money, or NFC. Bus drivers authenticate at the start of a shift, check route assignments, and monitor passenger count. Each bus is equipped with GPS to provide real-time location tracking, speed monitoring, and breakdown reporting.
Inspectors must verify tickets using handheld devices. The system should detect route violations, over-speeding, and unauthorized halts. A central control center monitors all buses and generates operational reports. Integration with weather forecasts and city-wide traffic systems is required for route optimization.

Task:

Design the UML diagrams:

1. Use-Case Diagram
2. Class Diagram
3. Sequence Diagram for “Passenger Ticket Validation”
4. Activity Diagram for “Bus Breakdown Handling”
5. State Machine Diagram for “Bus Operational States”
6. Deployment Diagram for onboard devices, cloud servers, and control center systems

Question 5

A cooperative of farmers from Nyagatare wants to automate irrigation across large farmlands using IoT sensors, drones, and cloud analytics. Soil moisture sensors, rainfall detectors, and temperature monitors send data to the cloud. The system recommends irrigation cycles and can automatically activate water pumps based on predefined thresholds. Farmers access the system through mobile apps to monitor farm conditions, manage irrigation schedules, and receive alerts for anomalies.
The system must support multi-farm management, role-based access, remote pump control, drought prediction analytics, and maintenance scheduling for sensors and equipment.
Task:
Create the following UML diagrams:

1. Use-Case Diagram
2. Class Diagram representing sensors, actuators, analytics engine, and user modules
3. Sequence Diagram for “Automatic Irrigation Triggering”
4. Activity Diagram for “Sensor Maintenance Workflow”
5. Deployment Diagram showing IoT gateways, cloud servers, and mobile clients

Question 6

Bank of Kigali wants a comprehensive loan management system where customers submit applications online, upload supporting documents, and track loan status. Loan officers verify information, credit analysts evaluate credit scores, and managers approve or reject applications. The system must automatically detect fraudulent submissions using machine learning and cross-check documents with national ID and employment databases.
The system also manages repayment schedules, penalty calculations, loan restructuring, and early settlement. Notification services must alert customers of approvals, rejections, due dates, or missed payments.
Task:
Provide UML diagrams:

1. Use-Case Diagram
2. Class Diagram
3. Sequence Diagram for “Loan Application Evaluation”
4. Activity Diagram for “Fraud Detection Workflow”
5. State Machine Diagram for “Loan Lifecycle”
6. Deployment Diagram for bank data center infrastructure

All the best!