Question 1 (REST Web services)

Discuss what a REST Web service is, all the concepts and technologies underlying them. Describe how concretely a programmer can develop a REST Web service in Java, by providing simple pseudo-code. Then design the interfaces of a (set of) REST Web service(s) that are able to provide information on public transportation, e.g., lines of buses, stops, time of scheduled arrivals, current (real) time of arrival, etc. In doing this last exercise. please provide motivations on the choices you may do, and develop the solution on the basis of such assumptions

Question 2 (SCRUM)

Describe ALL the basic elements of SCRUM. Then consider a system as INFOSTUD (the system used by Sapienza University of Rome for managing exams and all the students' lifecycle - you know all its features/functionalities) and assume you have a team of 7 persons - including UI designers, database designer, programmers, etc. The length of a sprint of 4 weeks. Propose and discuss a possible product backlog, and show the division of the features over the sprints, by presenting how you would evolve the system over 6 months of project (i.e., you have to present the initial 6 sprints of the project).

Question 3 (Function Points)

Describe the method of Function Points for evaluating software development complexity. Provide all the basic notions and exemplify through examples, whenever possible and appropriate.

REST (REPRESENTATIONAL STATE TRANSFER) IS A SW ARCHITECTURAL STYLE
FOR COMMUNICATING DISTRIBUTED SYSTEMS. IT'S BASED ON HTTP THAT WORKS
BY USING A WELL-DEFINED URL STRUCTURE THAT UNIQUELY IDENTIFIES A
RESOURCE OR SET OF RESOURCES, AND USING SPECIFIC HTTP METHODS FOR
RETRIEVING INFORMATION (GET), MODIFYING (POST, PUT, PATCH, DELETE).

THE TERM "REPRESENTATIONAL STATE TRANSFER" MEANS THAT A SERVER WILL RESPOND
WITH THE REPRESENTATION OF A RESOURCE IN HTML, XML OR JSON DOCUMENT
FORMAT.

REST IS ALSO STATELESS, MEANING THAT EACH REQUEST IS INDEPENDENT
AND CONTAINS ALL THE NECESSARY INFORMATION.

PSEUDO CODE:

```
PUBLIC CLASS BUS{
    PRIVATE STRING ID;
    PRIVATE STRING LINENAME;
    PRIVATE STRING CURRENTSTOP;
    PRIVATE STRING ARRIVALTIME;
}

PUBLIC CLASS BUSCONTROLLER {

    PUBLIC List<Bus> GETALLBUSES(){
        RETURN BUSSERVICE.GETALLBUSES();
    }

    PUBLIC BUS ADDBUS(BUS){
        RETURN BUSSERVICE.ADDBUS(BUS);
    }

    PUBLIC VOID DELETEBUS(ID){
        BUSSERVICE.DELETEBUS(ID);
    }
}

PUBLIC CLASS BUSSERVICE{

    PRIVATE List<Bus> BUSES= NEW ARRAYLIST<>();

    PUBLIC List<Bus> GETALLBUSES(){
        RETURN BUSES;
    }

    PUBLIC BUS ADDBUS(BUS){
        BUSES.ADD(BUS);
        RETURN BUS;
    }
}
```

REST

/BUS
- GET: RETURN ALL BUSES
- POST: ADD A NEW BUS

/BUS/BUSLINES — GET: RETURN ALL BUSLINES

/BUS/{BUSLINE}/STOPS
- GET: RETURN ALL SPECIFIC STOPS
- POST: ADD A NEW SPECIFIC STOP

*/BUS/{BUSLINE}/{STOP}/SCHEDULED ARRIVALS
- GET: RET SCHEDULE ARRIVAL AT A SPECIFIC STOP OF A SPECIFIC BUS LINE
- PUT: MODIFY " "

* WE CAN DO THE SAME WITH /CURRENT ARRIVALS

Q2

PRODUCT BACKLOG:

	PRIORITY
REGISTRATION AND AUTHENTICATION	HIGH
EXAM BOOKING	HIGH
CONSULTATIONS RESULTS	MEDIUM
STUDENT PROFILE MANAGEMENT	MEDIUM
ADMINISTRATIVE MANAGEMENT	MEDIUM
UI SYSTEM DESIGN	HIGH
TESTING AND IMPROVEMENTS	MEDIUM

6 MONTH OF 4 WEEKS SPRINTS → 6 SPRINTS

SPRINT	MAIN OBJECTIVE	DEVELOPED FEATURES
1	CREATION OF THE TECHNICAL BASE	REGISTRATION, AUTHENTICATION
2	KEY FEATURES FOR STUDENTS	EXAM BOOKING, BASIC UI
3	CONSULTATION AND MANAGEMENT OF RESULTS	CONSULTATIONS RESULTS, UI IMPROVEMENTS
4	ADDING ADMINISTRATIVE FEATURES	ADMINISTRATIVE MANAGEMENT, DATA LOADING
5	TESTING AND OPTIMIZATION	FUNCTIONAL TESTING AND BUG FIXES
6	DELIVERY	FINAL IMPROVEMENTS