

Project – Software Development-1 – CSE-2340 – MAS

Scenario and Attributes																													
Scenario:																													
<p>The International Islamic University Chittagong (IIUC) currently manages its courses across nine departments manually through Excel sheets and departmental coordination. This approach often causes inconsistency, and difficulty in tracking course offerings each semester.</p> <p>To solve this, the university's IT Cell has proposed building a centralized RESTful API using Spring Boot that will manage course-related data digitally and uniformly across all departments.</p> <p>The API will allow CRUD operations for courses, departments, and faculty assignments, serving as the backend for future web or mobile applications.</p>																													
Attributes:																													
<table style="width: 100%; border-collapse: collapse;"> <thead> <tr> <th style="text-align: left; width: 25%;">Field</th><th style="text-align: left; width: 25%;">Type</th><th style="text-align: left; width: 50%;">Example</th></tr> </thead> <tbody> <tr> <td>id</td><td>Long</td><td>100</td></tr> <tr> <td>code</td><td>String</td><td>CSE-2340</td></tr> <tr> <td>title</td><td>String</td><td>Software Development-1</td></tr> <tr> <td>credit</td><td>Double</td><td>2.0</td></tr> <tr> <td>type</td><td>String</td><td>Lab</td></tr> <tr> <td>semester</td><td>String</td><td>Autumn 2025</td></tr> <tr> <td>departmentId</td><td>Long</td><td>2</td></tr> <tr> <td>teacher</td><td>String</td><td>MAS</td></tr> </tbody> </table>			Field	Type	Example	id	Long	100	code	String	CSE-2340	title	String	Software Development-1	credit	Double	2.0	type	String	Lab	semester	String	Autumn 2025	departmentId	Long	2	teacher	String	MAS
Field	Type	Example																											
id	Long	100																											
code	String	CSE-2340																											
title	String	Software Development-1																											
credit	Double	2.0																											
type	String	Lab																											
semester	String	Autumn 2025																											
departmentId	Long	2																											
teacher	String	MAS																											

Requirements	Marks and COs
Sprint Planning (CIE): <ul style="list-style-type: none"> 1. Are all entities identified? 2. Are CRUD endpoints planned for each entity? 3. Is the data flow (Model → Service → Controller) well understood? 4. Is the project package structure clearly designed? 5. Is task distribution among members clear? 	5 CO1

Scrum 1 – Model Layer (CIE):	
<ol style="list-style-type: none"> 1. Create necessary classes for Course. 2. Use proper data types and field names. 3. Implement constructors, getters, setters. 4. Use Long or int as unique IDs (manually handled). 5. Relationships can be represented by IDs or nested objects (optional). 	5 CO2
Scrum 2 – Service Layer (CIE):	
<ol style="list-style-type: none"> 1. Maintain List<Model> for in-memory data (e.g., List<Course> courses = new ArrayList<>()); 2. Implement add, getAll, getById, update, and delete methods. 3. Use loops or removeIf() for search and delete operations. 4. Simulate ID generation using a counter variable (e.g., private Long nextId = 100L;). 5. Ensure services are independent and reusable. 	5 CO2
Scrum 3 – Controller Layer (CIE):	
<ol style="list-style-type: none"> 1. Use @RestController and @RequestMapping. 2. CRUD endpoints with appropriate annotations: 3. @GetMapping, @PostMapping, @PutMapping, @DeleteMapping 4. Use @RequestBody for object inputs, @PathVariable for IDs. 5. Return simple responses with status codes. 	5 CO2
Final Scrum – Integration & Testing (CIE): <i>Verify all layers work together using Postman.</i>	
<ol style="list-style-type: none"> 1. Run the Spring Boot app and test all CRUD endpoints. 2. Validate correct creation, update, and deletion from ArrayList. 3. Test invalid IDs (should return error message or false). 4. Observe data persistence within runtime (not permanent). 	10 CO3
Presentation (SEE):	
<ol style="list-style-type: none"> 1. Each member explains their own layer (model/service/controller). 2. Another member will give live demonstration of CRUD using Postman. 3. Clear explanation of code structure and logic. 4. Smooth teamwork and communication. 	10 CO3

Final Report (SEE):	
<ol style="list-style-type: none">1. Layer diagram (Model → Service → Controller).2. API endpoint list with sample JSON request/response.3. Code snippets of core methods.4. Screenshots of successful Postman tests.5. Well-documented .pdf file.	10 CO3