

CMU B.Sc. (HONS) SE/B.Sc. (Hons) SE- ASSIGNMENT FEEDBACK SHEET -ICBT CAMPUS

Student Details (Student should fill the content)				
Name				
Student ID				
Scheduled unit details				
Unit code	CIS6003			
Unit title	Advanced Programming			
Unit enrolment details	Year	3		
	Study period			
Lecturer				
Mode of delivery	Full Time			
Assignment Details				
Nature of the Assessment	Course work 100%			
Topic of the Case Study	Lab Appointment System			
Learning Outcomes covered	1,2,3			
Word count	4000			
Due date / Time	Jan 2024			
Extension granted?	Yes	No	Extension Date	
Is this a resubmission?	Yes	No	Resubmission Date	
Declaration				
I certify that the attached material is my original work. No other person's work or ideas have been used without acknowledgement. Except where I have clearly stated that I have used some of this material elsewhere, I have not presented it for examination/assessment in any other course or unit at this or any other institution				
Name/Signature			Date	
Submission				
Return to:				
Result				
Marks by 1 st Assessor		Signature of the 1 st Assessor		Agreed Mark
Marks by 2 nd Assessor		Signature of the 2 nd Assessor		

Comments on the Agreed Mark.			
For Office use only (hard copy assignments)			
Receipt date		Received by	

STUDENT NAME:		STUDENT NUMBER:	
Module Number & Title:		Semester:	
Assignment Type & Title:			
For student use: <i>Critical feedback on the individual progression towards achieving the assignment outcomes</i>			
For the Assessors' feedback			
Indicate the Task number strengths and Weaknesses and the marks for each task			
Task No/Question No	Strengths		

Task No / Question No	Weaknesses		
Areas for future improvement			
Marks			
Task /Question No	Allocated Marks	Awarded Marks	Remarks
Total Marks			
Name and the Signature of the Assessor			
Date			

Upon successful completion of this module, you will have demonstrated:

- Demonstrate fluency in contemporary programming languages, development tools and environments.
- Evaluate and demonstrate the theory and concepts of contemporary/industry-standard programming and design in the software development life cycle.
- Demonstrate awareness of industry standards of professional and ethical software development, software carpentry and codemanship.

Coursework –Lab Appointment System – 100 Marks

Scenario

ABC Laboratories provides a range of services for patients who require medical tests. Currently, all the appointment details and test records are managed manually. The organization is planning to implement a web-based Lab Appointment System to improve their service and productivity. The proposed Lab Appointment System must help in registering information about patients who are planning to do the medical test and handles patients' queries. A unique ID must be produced for each patient after registration. When the ID is created the patient will receive the appointment time and number to undergo the test accordingly.

The system should be able to manage and store test details that include information about the patient, test and test results, technicians, and the doctor who recommended the test. Patients can pay the bill by credit card and the receipt will be emailed to the patient. The patients must have the facility to download their current and previous lab reports.

The system must be user-friendly and generate necessary reports to support operational and managerial decision-making.

Provide a well-designed, user-friendly system addressing the following features:

- The system should have differential access rights to the system users.
- Interactive user-friendly interfaces
- Clear implementation of the business flow via the system.
- Design & implement suitable sets of reports, which you think will add more value to the entire business
- Use test-driven development and include test classes to test your application

Students are free to make necessary assumptions on system design & granting access permissions other than those mentioned within the scenario, but all suggestions must be well explained with valid reasons.

Students can add any functionality which will enhance the system and make the proposed solution more comprehensive.

Use Harvard reference to properly acknowledge all the external sources you use.

Your tasks

Tasks A:

Provide a requirement specification for the proposed system. (06 marks)

Tasks B:

Provide the UML diagrams for the given problem with clear explanations of the design decisions. Derive detailed Use Case diagram, Class diagram & a sequence diagram. Whenever necessary document the relevant assumptions you made.

(09 marks)

Tasks C:

There are many system design patterns available in system development. Critically evaluate singleton, factory and abstract factory design patterns and apply the most suitable design pattern for your system development. [08]

(15 marks)

Tasks D:

Develop an interactive set of interfaces to get the necessary user inputs. Make sure to implement proper validation mechanisms in order to restrict invalid entries to the system. Come up with a suitable set of reports, which you think add more value to your system

- i. Your program must be a distributed application with web services
- ii. Your program should make use of a proper database to store information

(30 marks)

Tasks E:

Document the test plan and explain how you used test-driven development in this scenario and do a test automation to achieve that. This includes test rationale, test plan, test data and proper application of the test plan (LO II)

(15 marks)

Tasks F:

Create user and technical documentation for the developed solution. (10 marks)

Tasks G:

Create your own Git/ GitHub repository which is public to access and upload /deploy the changes of the software project you have developed in it. Share the report link within the documentation. Update it with several versions where modifications are applied each day, that you have applied the new features into which were initially uploaded. Version control techniques you have used throughout the development should be highlighted and documented properly. Demonstrate workflows deployed with the Git repository.

(15 marks)

Guidelines for the report format

Paper A4 | Margins 1.5” left, 1” right, top and bottom

Page numbers – bottom, right | Line spacing 1.5

Font size

Headings 14pt, Bold | Normal 12pt

Font face- Times New Roman

Referencing and in-text citations should be done strictly using the **Harvard Referencing System**.

Marking Scheme**Task (A) contains 6 marks**

Criteria	Marks
	Out of 6
Functional requirement specification	3
Nonfunctional requirement specification	3

Task (B) contains 9 marks

Diagrams should be evaluated according to the following criteria.

Criteria	Marks
	Out of 9
Proper use of Object-Oriented Design Methodology Use Case Diagram · Identification of correct use cases · Identification of correct Actors and associations Sequence Diagram · Implementing an identified set of use cases (about 3) as sequence diagrams. Class Diagram · Identification of associated methods, with correct signatures and attributes in each class · Correct identification of relationships	1-3
Average Design · Clear identification of private, and public access modifiers & it is visible in the class diagram · Accurate use of <<include> <<extend>> stereo types in use case diagram · Appropriate use of lifelines, messages and objects in proposed sequence diagrams · Correct use of UML notations with minor mistakes Evaluation · Student has given a basic description of the design and given a reasonable justification · Effective judgements have been made	4-6

<p>Excellent Design</p> <ul style="list-style-type: none"> · Highly detailed diagram · Use of OO concepts clearly visible · Backed by relevant assumptions · Multiplicity, navigability aggregation & compositions visible in class diagrams · Excellent use of UML notation <p>Evaluation</p> <ul style="list-style-type: none"> · Good justification of the design · Judge validity of results · Use critical reflection to evaluate the work and justify it with valid explanations <p>Fluency (Of design)</p> <ul style="list-style-type: none"> · Evidence of critical analysis on different perspectives covering how, use case, class & sequence diagrams support designing 	7-9
---	-----

Task (C) contains 15 marks

Criteria	Marks
	Out of 15
Identify the different types of design patterns and their advantages	1-5
Apply the suitable design patterns for system development	6-10
Critically evaluate the impact of design patterns	11-15

Task (D) contains 30 marks

Criteria	Marks
	Out of 30
<p>Pass</p> <ul style="list-style-type: none"> · Basic data management system features. · Use a database (simple design) · Have a simple Web user interface 	0-8
<p>Good</p> <ul style="list-style-type: none"> · Make a good attempt to follow the three-tier architecture. · More sophisticated database design and queries · More sophisticated data representation (e.g. several classes at the business logic level) · Separate UI windows for entering results and viewing overall scores. 	9-20

<p>Excellent</p> <ul style="list-style-type: none"> · More sophisticated UI, · Complex functionality (Email alerts/SMS/Innovative aspects) · 3- tier architecture should exist · Appropriate use of more sophisticated database features (e.g. use of stored procedures/functions/triggers to implement business rules) · Reports being proposed to facilitate decision-making. · Effective use of sessions/cookies 	21-30
---	-------

Task (E) contains

15 marks)

Test Rationale

(5 marks)

- ☐ Provide a concise rationale for the approach adopted. Discuss how you are going to use test-driven development.

Devise your test data

(5 marks)

- ☐ Derive test data for the system.

Produce and apply a test plan

(5 marks)

- ☐ Create test classes for your system
- ☐ You are to carry out relevant tests and provide documentation detailing the tests used to verify your system.
- ☐ Demonstrate that the code passes all the tests (use screen-grabbing software and insert images into your submission).
- ☐ Use of test automation
- ☐ Evaluation of overall success or failure and lessons learned.
- ☐ Traceability showing how each requirement is met by the design.

Task (F) contains 10 marks

Criteria	Marks
	Out of 10
Errors in the documentation	0-3
Acceptable standard of documentation with poor explanations	3-5
High standard of documentation with screenshots & average explanations	5-7
Professional standard of documentation with screenshots & good explanation	5-10

Task (G) contains 15 marks

Criteria	Marks
	Out of 15
Poor/no Git version control, deployment, or workflows used demonstrated	0-3
Git Repo/GitHub is used for creating a repository only and uploading the initial version only	3-6
Git repo was created, the initial version of the project was uploaded, and several versions were updated and deployed with changes, but no workflow or version control technique was demonstrated.	6-10
Git repo creation, accessibility restrictions, versioning and version control techniques, workflow (CI/CD) demonstrated, and deployment of changes and the latest updated version is done and demonstrated in documentation	10-15

Final Grading criteria for the coursework:

Marks	Final Grade
≥ 70	1
69-60	2:1
59-50	2:2
49-40	3
< 40	Fail
