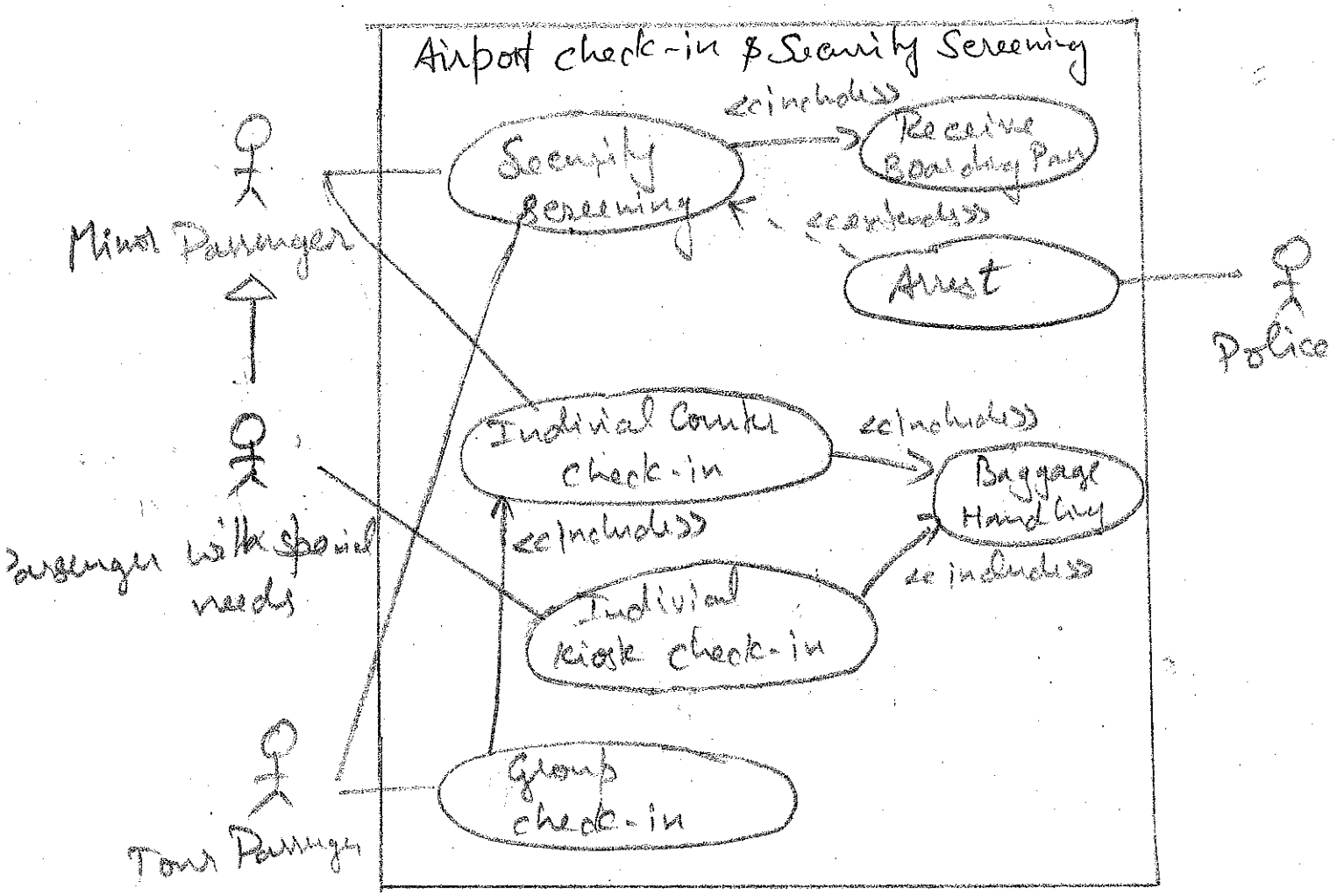
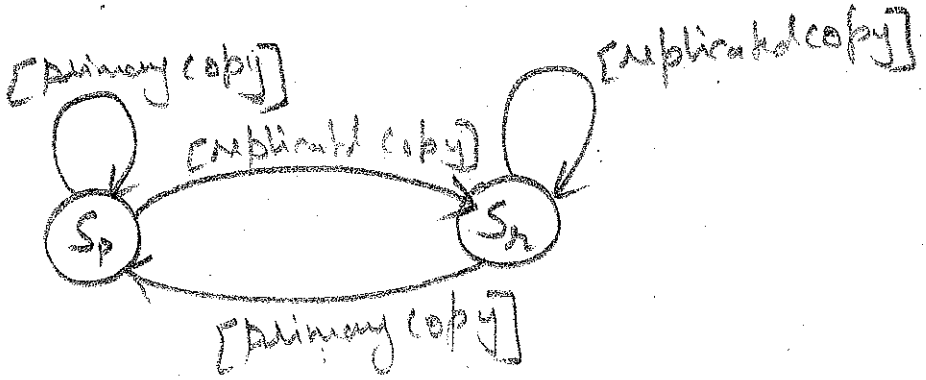
Q1) — 4 marks



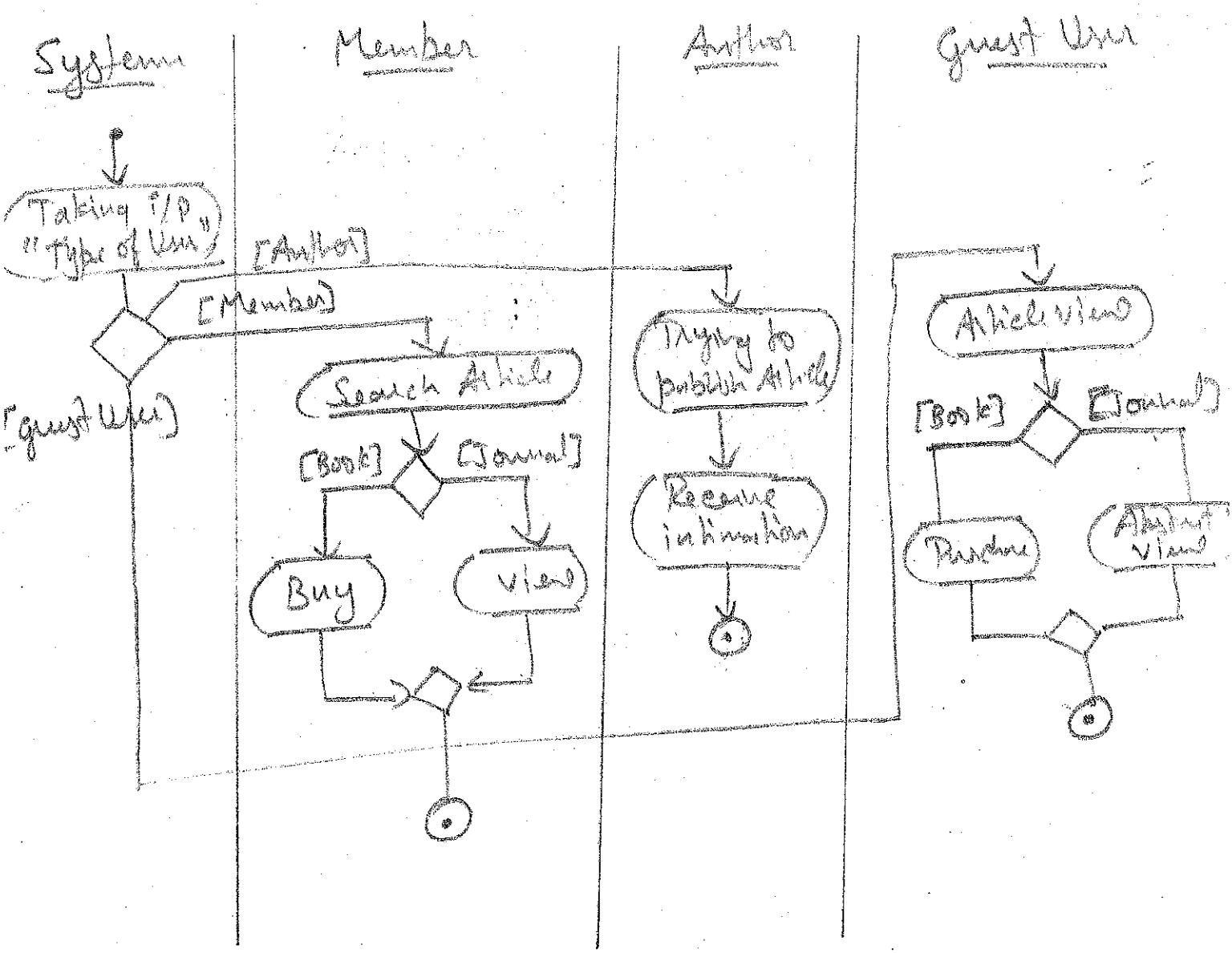
1) Identifying the Actor — 1
2) Identifying the Actor — 1
3) Relationship — 2

Q4)



Q2)

3 marks


System
Member
Author
Guest User
"Taking I/P "Type of User"
[Author]
[Member]
[guest User]
Search Article
[Book]
[Journal]
Buy
View
Trying to publish Article
Receive information
Article view
[Book]
[Journal]
Purchase
Abstract view

MANIPAL INSTITUTE OF TECHNOLOGY

MANIPAL

(A constituent unit of MAHE, Manipal)

V SEMESTER B.TECH. IT/CCE INTERNAL EXAMINATIONS OCTOBER 2020

Mid Term Exam

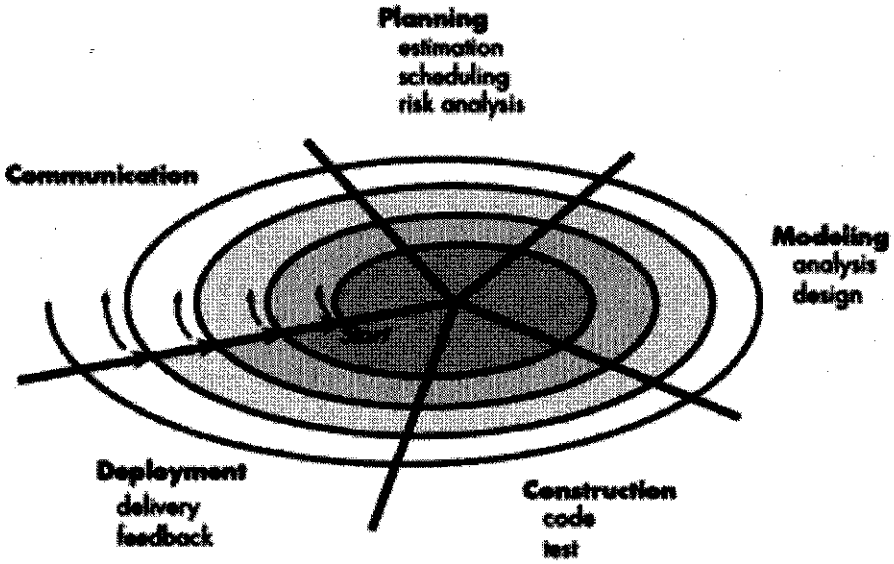
SUBJECT: SOFTWARE ENGINEERING/SOFTWARE DESIGN TECHNOLOGY [ICT 3159] / [ICT 3174]

Date of Exam: **24/10/2020** Time of Exam: **12PM – 1.45PM** Max. Marks: **20**

SOLUTION

3. Which model is a risk-driven process model? Map the activities of that model to the generic software process framework activities and explain. **3**

The spiral development model is a risk-driven process model. The spiral model is an evolutionary software process model that couples the iterative nature of prototyping with the controlled and systematic aspects of the waterfall model. It has two main distinguishing features. One is a cyclic approach for incrementally growing a system's degree of definition and implementation while decreasing its degree of risk. The other is a set of anchor point milestones for ensuring stakeholder commitment to feasible and mutually satisfactory system solutions. **1**



Each of the generic framework activities represent one segment of the spiral path illustrated in figure. The mapping is of spiral model activities to the generic framework activities is also shown in the figure.

As this evolutionary process begins, the software team performs activities that are implied by a circuit around the spiral in a clockwise direction, beginning at the center. Risk is considered as each revolution is made. Anchor point milestones—a combination of work products and conditions that are attained along the path of the spiral—are noted for each evolutionary pass. The first circuit around the spiral might result in the development of a product specification; subsequent passes around the spiral might be used to develop a prototype and then progressively more sophisticated versions of the software. **1**

Each pass through the planning region results in adjustments to the project plan. Cost and schedule are adjusted based on feedback derived from the customer after delivery.

In addition, the project manager adjusts the planned number of iterations required to complete the software. 1

6. What are the criteria defined for good software design with respect to coupling and cohesion? 2

The functional independence of the components while designing a software is assessed using two qualitative criteria: cohesion and coupling

A cohesive module performs a single task, requiring little interaction with other components in other parts of a program. The design should achieve high-cohesion components, each component serving a single function. The low-cohesion components performing many unrelated functions have to be avoided. 1

Coupling is an indication of interconnection among modules in a software structure. Coupling depends on the interface complexity between modules, the point at which entry or reference is made to a module, and what data pass across the interface. The software design should strive for the lowest possible coupling. Simple connectivity among modules results in software that is easier to understand and less prone to a "ripple effect," caused when errors occur at one location and propagate throughout a system. 1

7. What is legacy software? Mention one advantage and disadvantage of the same. 2

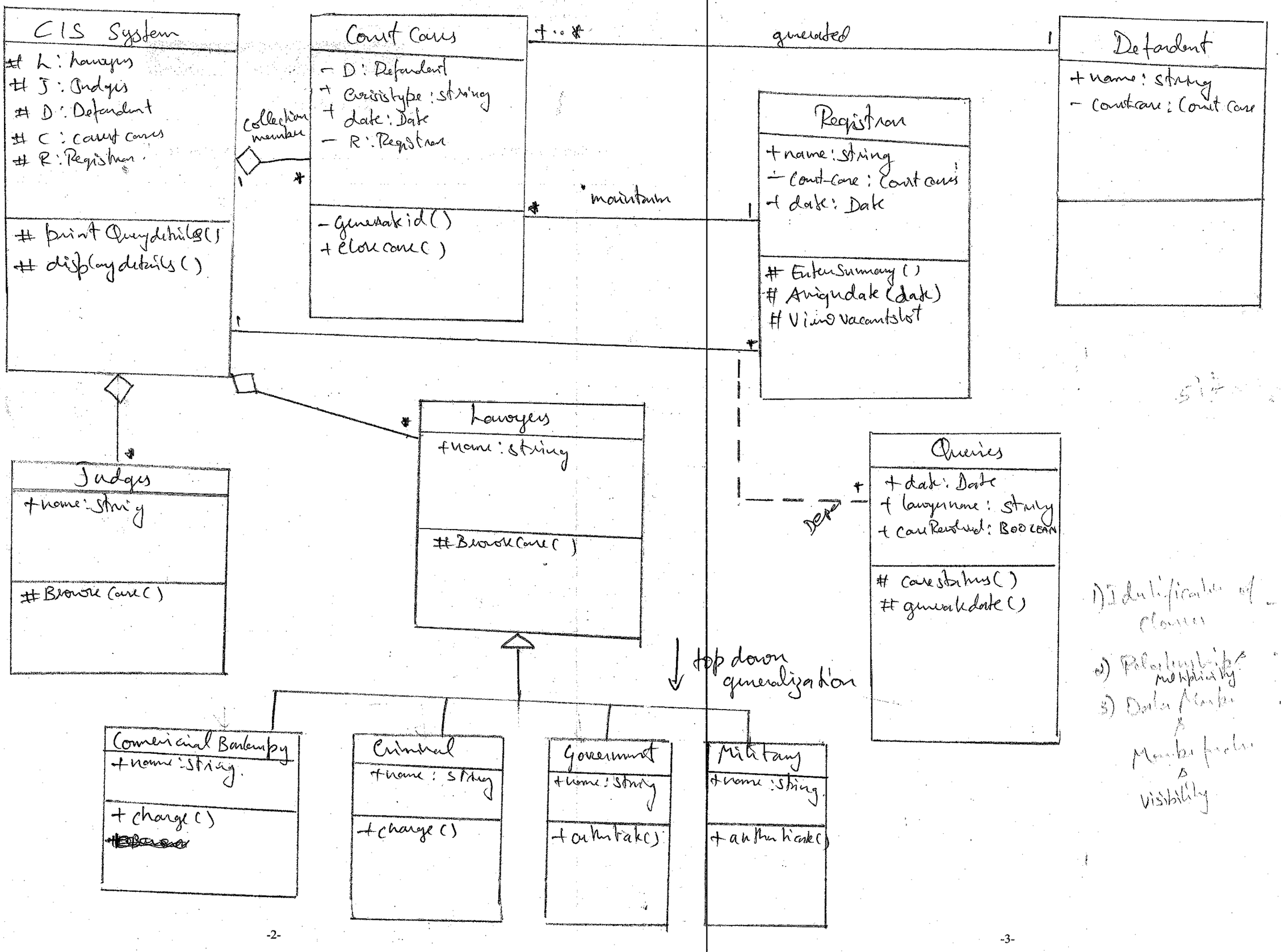
These older programs are referred to as legacy software.

Advantage: Many legacy systems remain supportive to core business functions and are 'indispensable' to the business." Hence, legacy software is characterized by longevity and business criticality. 1

Disadvantage: Legacy systems sometimes are also characterized by their "poor quality" having inextensible designs, convoluted code, poor or nonexistent documentation, test cases and results that were never archived, and a poorly managed change history. 1

Q5) 20

4 marks



1) Identification of classes

2) Relationship & multiplicity

3) Data Members & Member function & visibility