

A
Laboratory File
On
Software Engineering Lab (CS1051)

Masters of Technology in Computer Science And
Engineering

submitted by
Arghya Bandyopadhyay
RollNo. 20CS4103

submitted to
Dr. Anirban Sarkar
Associate Professor
Dept. of CSE



National Institute of Technology, Durgapur

Set 1: Control Flow Diagram (CFG)

A. Write a program to store and display the CFG for given program segment.

B. Write a program to identify the number of bounded region in a generated CFG.

C. Write a Program to identify the maximal set of independent paths in a generated CFG. D. Write an application with suitable UI to generate the CFG from a given program file. On editing of the program file the CFG should also modify accordingly. The UI should also show the basic characteristics (like, no. of nodes, no. of edges, predicate nodes, no. of bounded region, no. of paths. Etc.) of the CFG (OPTIONAL)

Assume the if-then-else program be

i) Without loop

```
i = 4
if i % 2 == 0 :
    print("Even Number")
else :
    print("Odd Number")
```

CFG Program:

```
import pydot
import os
from IPython.display import Image, display
from collections import defaultdict
class Graph:

    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)

    def addEdge(self, u, v):
        self.graph[u].append(v)

    def printAllPathsUtil(self, u, d, visited, path):

        visited[u] = True
        path.append(u)

        if u == d:
```

```

        print(path)
    else:
        for i in self.graph[u]:
            if visited[i]== False:
                self.printAllPathsUtil(i, d, visited, path)
    path.pop()
    visited[u]= False

def printAllPaths(self, s, d):

    visited =[False]*(self.V)
    path = []
    self.printAllPathsUtil(s, d, visited, path)

G = pydot.Dot(graph_type="digraph")
nodes = [0,1,2,3,4,5]
edges = [(0,1),(1,2),(1,3),(2,4),(3,4),(4,5)]
g = Graph(len(nodes))

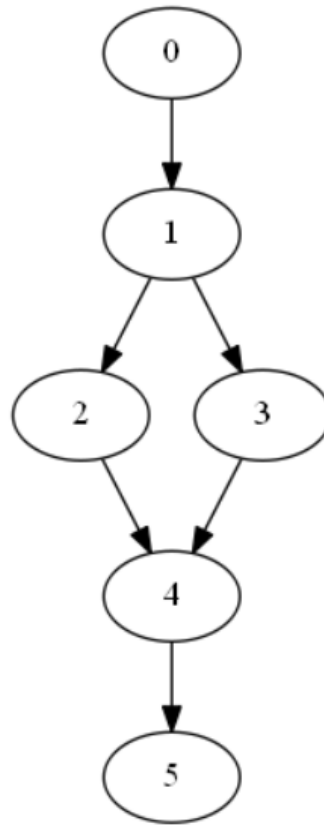
for i in nodes:
    x = pydot.Node(i)
    G.add_node(x)
for i in edges:
    x = pydot.Edge(i[0], i[1])
    G.add_edge(x)
    g.addEdge(i[0], i[1])

im = Image(G.create_png())
display(im)

print("No. of Nodes: ", len(nodes))
print("No. of Edges: ", len(edges))
print("No of Bounded Regions: ", (len(edges) - len(nodes) + 2))
print("Independent Paths: ")
g.printAllPaths(nodes[0], nodes[-1])

```

Output:



No. of Nodes: 6
No. of Edges: 6
No of Bounded Regions: 2
Independent Paths:
[0, 1, 2, 4, 5]
[0, 1, 3, 4, 5]

ii) With loop

```
for i in range(1,5):
    if i % 2 == 0:
        print("Even Number")
    else:
        print("Odd Number")
```

CFG Program:

```
import pydot
import os
from IPython.display import Image, display
from collections import defaultdict

class Graph:

    def __init__(self, vertices):
        self.V = vertices
        self.graph = defaultdict(list)

    def addEdge(self, u, v):
        self.graph[u].append(v)

    def printAllPathsUtil(self, u, d, visited, path):

        visited[u] = True
        path.append(u)

        if u == d:
            print(path)
        else:
            for i in self.graph[u]:
                if visited[i] == False:
                    self.printAllPathsUtil(i, d, visited, path)
            path.pop()
            visited[u] = False

    def printAllPaths(self, s, d):

        visited = [False] * (self.V)
```

```

        path = []
        self.printAllPathsUtil(s, d, visited, path)

G1 = pydot.Dot(graph_type="digraph")
nodes = [0,1,2,3,4,5,6]
edges = [(0,1),(1,2),(2,3),(2,4),(3,5),(4,5),(5,1),(5,6)]
g1 = Graph(len(nodes))

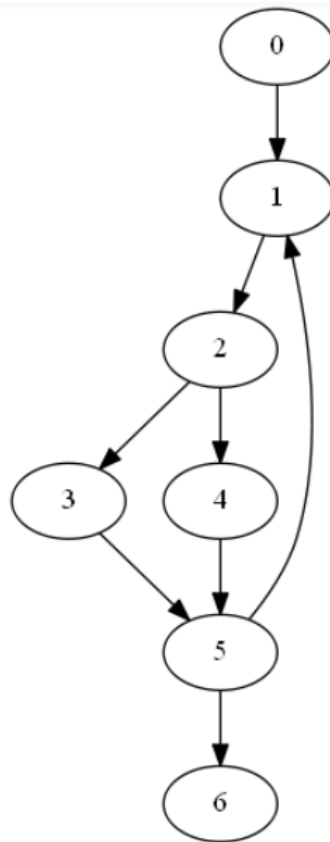
for i in nodes:
    x = pydot.Node(i)
    G1.add_node(x)
for i in edges:
    x = pydot.Edge(i[0], i[1])
    G1.add_edge(x)
    g1.addEdge(i[0], i[1])

im = Image(G1.create_png())
display(im)

print("No. of Nodes: ", len(nodes))
print("No. of Edges: ", len(edges))
print("No of Bounded Regions: ", (len(edges) - len(nodes) + 2))
print("Independent Paths: ")
g.printAllPaths(nodes[0], nodes[-1])

```

Output:



No. of Nodes: 7
No. of Edges: 8
No of Bounded Regions: 3
Independent Paths:
[0, 1, 2, 3, 5, 6]
[0, 1, 2, 4, 5, 6]
[0, 1, 2, 4, 5, 1, 2, 3, 5, 6]

Set 2: Entity-Relationship / Extended Entity Relationship Diagram (ER/EER)

A. NIT Durgapur wants to automate the Student Registration System and seek for intranet based solution. The Candidates need to register for some specific semester may go online with the system and provide the necessary data (roll no., name, department, course, semester and last semester grade point etc.) as input data. The system must be able to check the input data entered by student from the pre-existing student database and then automatically generate the registration slip for the specific semester as student copy. The system also will maintain a log of the students already registered and it will restrict any duplicate registration. System Administrator can view and generate report of the list of registered students and list of unregistered students. Draw Suitable ER/EER Diagram.

Solution:

Student Registration System entities and their attributes:

1. **Student Entity:** roll_no, full_name, email, course_id, dept_id, semester
2. **Course Entity:** course_id, course_name, dept_id
3. **Department Entity:** dept_id, dept_name, hod
4. **Academics Entity:** roll_no, sem_marks, current_sem_reg_flag
5. **System administrator Entity:** user_id, pwd, admin_name
6. **Reports Entity:** sr_no, roll_no, reg_flag
7. **Registration Entity:** reg_id, roll_no, trans_id
8. **Fees Entity :** trans_id, trans_date

Assumptions:

- It is assumed that this Registration System is prepared for NIT Durgapur students and only those students who had taken admission and already submitted the fees of current semester, can go online to generate the registration slip of that semester as Student's Copy.
- System Administrator can view or modify the student or registration database if needed.
- No fee is charged by the System from the students to access their accounts for registration.
- The student and registration database are confidential and is NOT available to be used by anyone except the system administrator.
- Initially, the student will access the Registration System, provide his/her details like name, roll number, department name, birthdate, course, semester etc., to register for the current semester and then this data will be received by the Registration System to compare it with Student and Registration Database. If the student had not registered previously, then he/she will get registered and the Student Copy of registration will be generated.

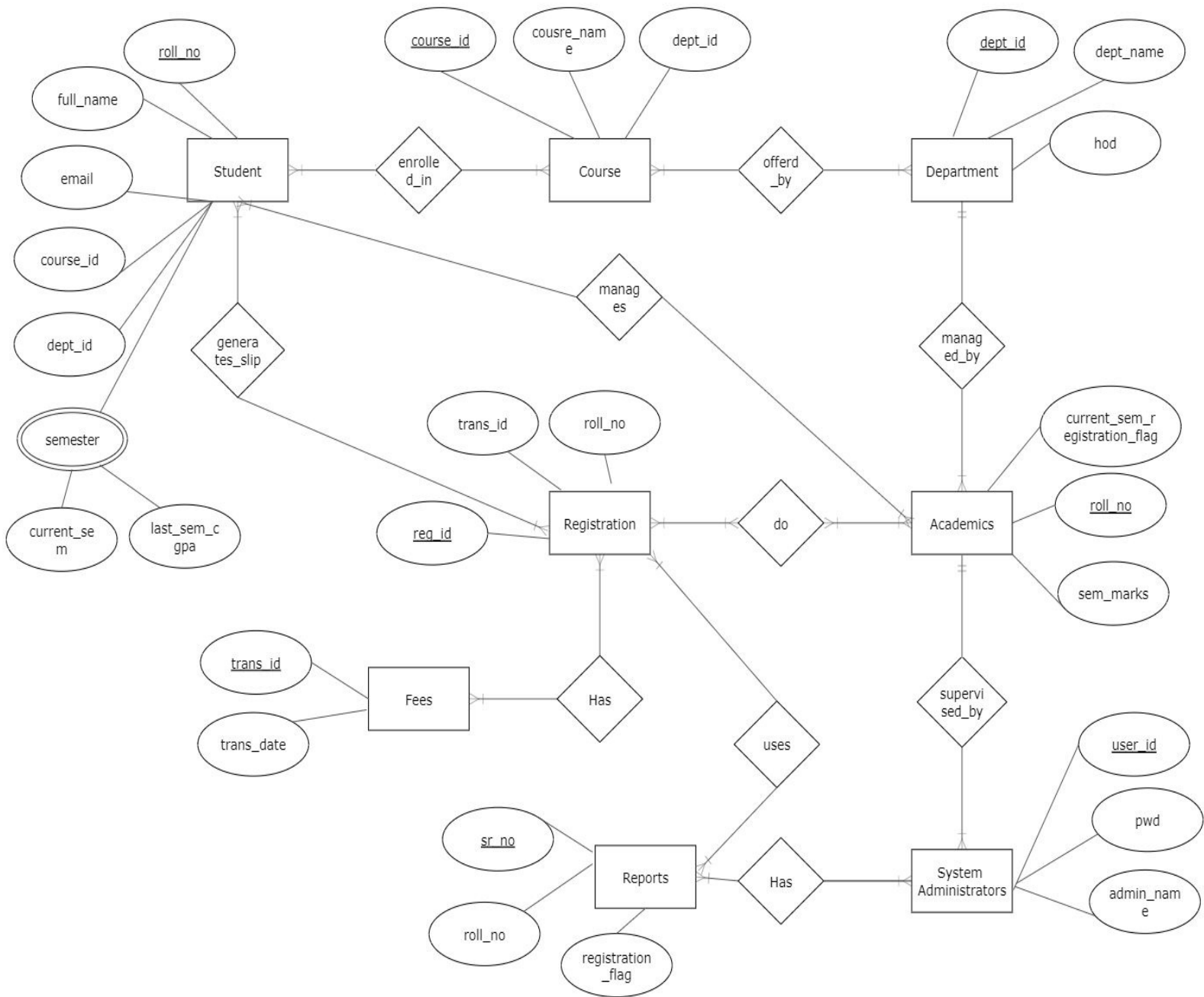


Figure 1 Student Registration System ER Diagram

B. XYZ Bank is a facilitator for public banking system with many branches in different region. They would like to automate the banking Processing System. The bank facilitates deposit, withdrawal and fixed deposit system from its saving accounts. Customer may have joint account as well as many account in any branch of the bank. Withdrawal or deposit can be done from any branch of the bank. The bank also has facility of ATM. On any transaction, the system will be able to maintain the transaction record in some log. Using which, system administrator can generate report for account wise transaction per day. Draw Suitable ER/EER Diagram.

Solution:

Bank Processing System entities and their attributes:

- 1.) **Branch Entity:***branch_id, branch_name, branch_address, branch_manager*
- 2.) **Customer Entity:***cust_id, cust_name, aadhar_no, mobile, dob, cust_add*
- 3.) **Accounts Entity:***acc_no, cust_id, branch_id, acc_bal, acc_type*
- 4.) **ATM Entity:***atm_id, atm_add, trans_id*
- 5.) **Transaction Entity:***trans_id, trans_date, trans_amt, trans_type, from_acc*
- 6.) **System Administrators Entity:***user_id, pwd, admin_name* **Assumptions:**

- It is assumed that the customers who have Bank Account and Bank Passbook as proof or have ATM card with valid Pin Code will only use this Bank Processing System.
- It is also assumed that the customer database such as name, addresses and other details, already exists in the Processing System.
- No fee will be charged by the Processing System from the customers to process deposit or withdrawal from his/her account.
- The resources and database of the Processing System are highly confidential and are NOT available to be used by anyone except system administrator.
- The documentation shows the relationship among the Branch, Customer and their account in that branch, including cash transaction slips, ATMs, customer databases and transaction logs.
- Initially the customer will go the bank branch or ATM to deposit or withdraw the cash and then get his/her transaction slip. Also, the system administrator can be able to generate daily transaction logs.

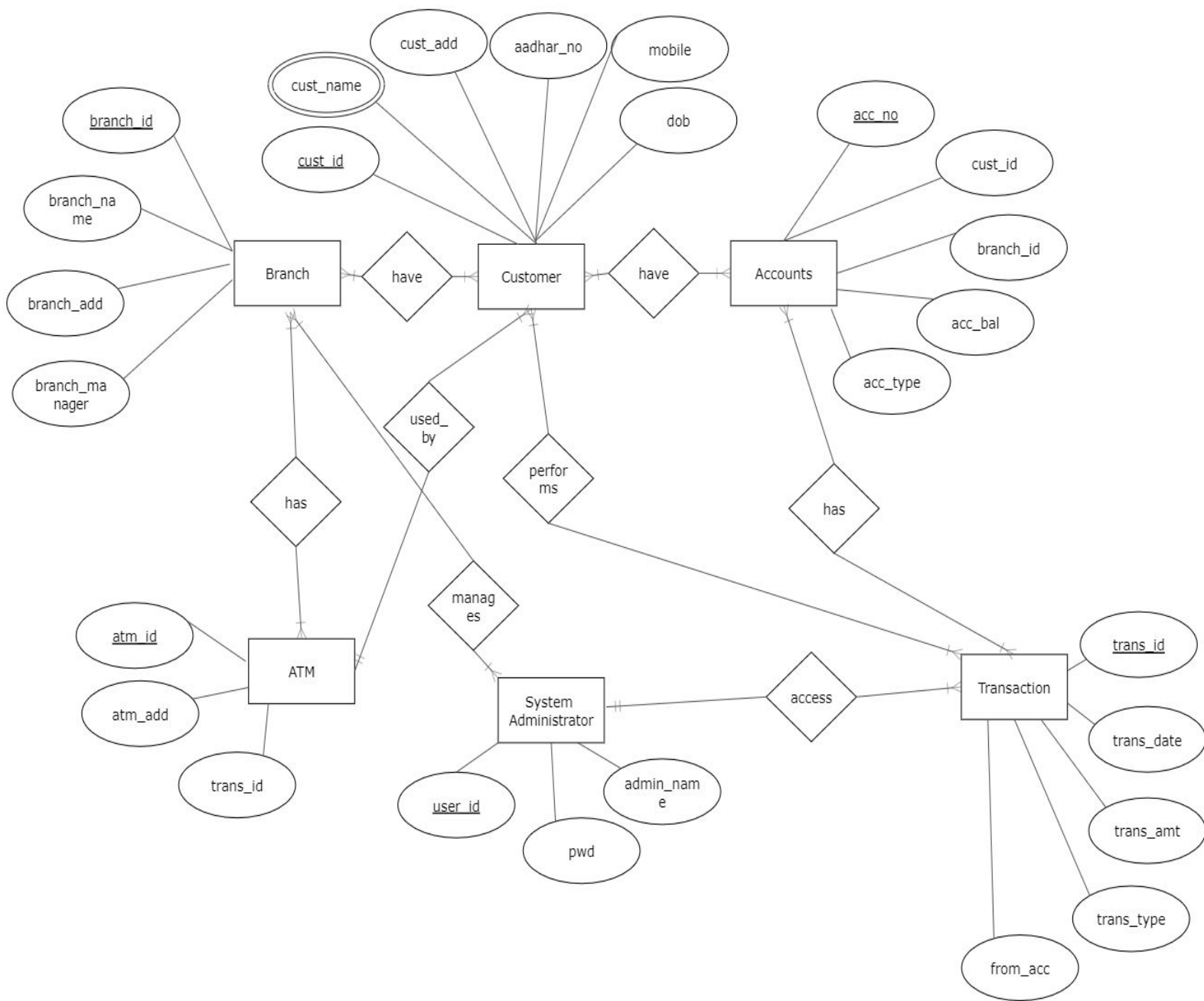


Figure 2 Bank Processing System ER Diagram

Set 3: Unified Modelling Language (UML)

A. Consider Automated Teller Machine (ATM) based withdrawal of cash. For any account it will check if Minimum balance is available for withdrawal and can disburse cash maximum of INR 10,000/-. ATM can accept debit cards of pre-specified banks (more than one). On every transaction, the machine will generate a report. Draw Use Case diagram, Class diagram, Object Diagram, Sequence Diagram for the above activities.

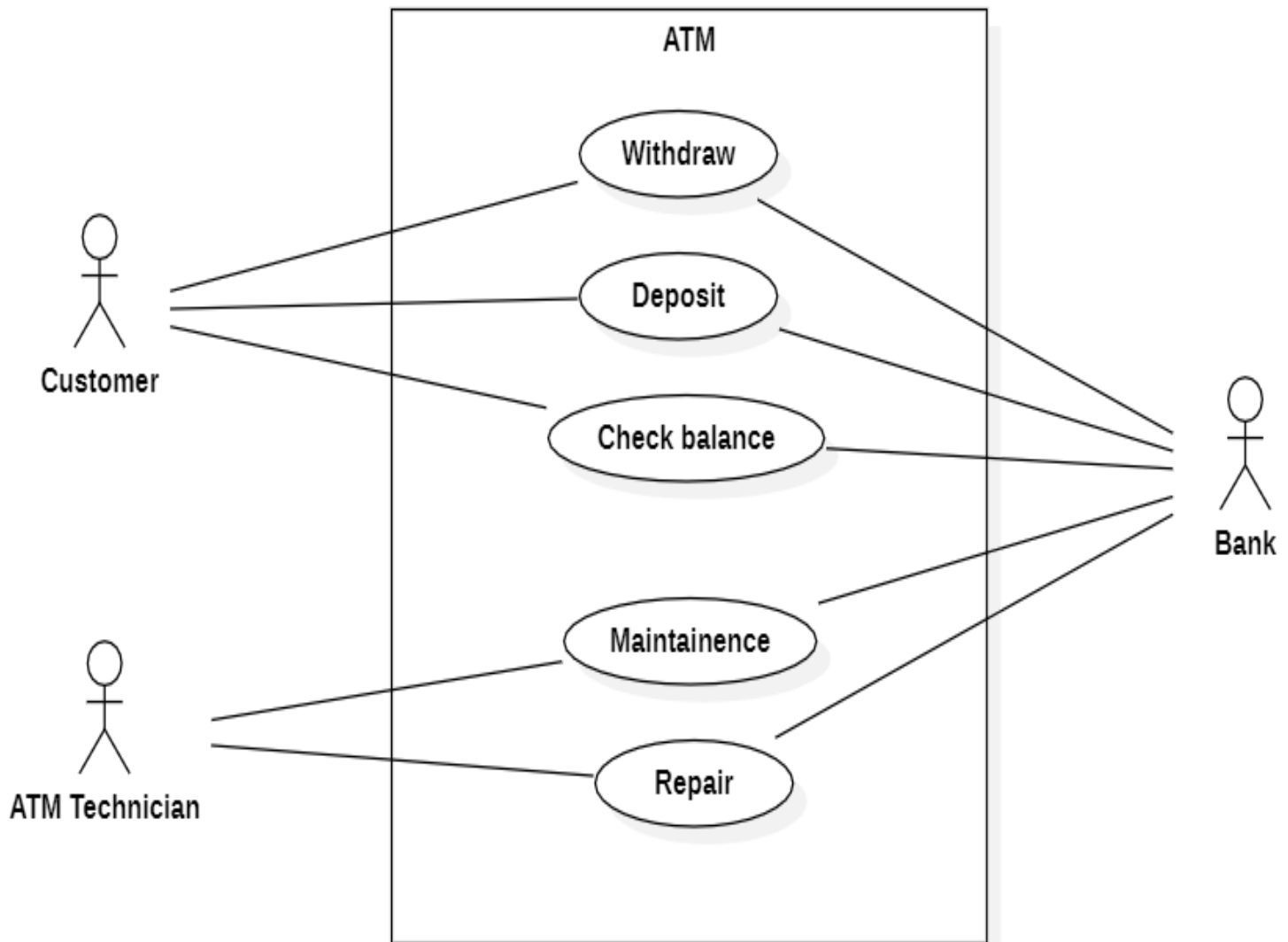


Figure 3 Use Case Diagram for ATM

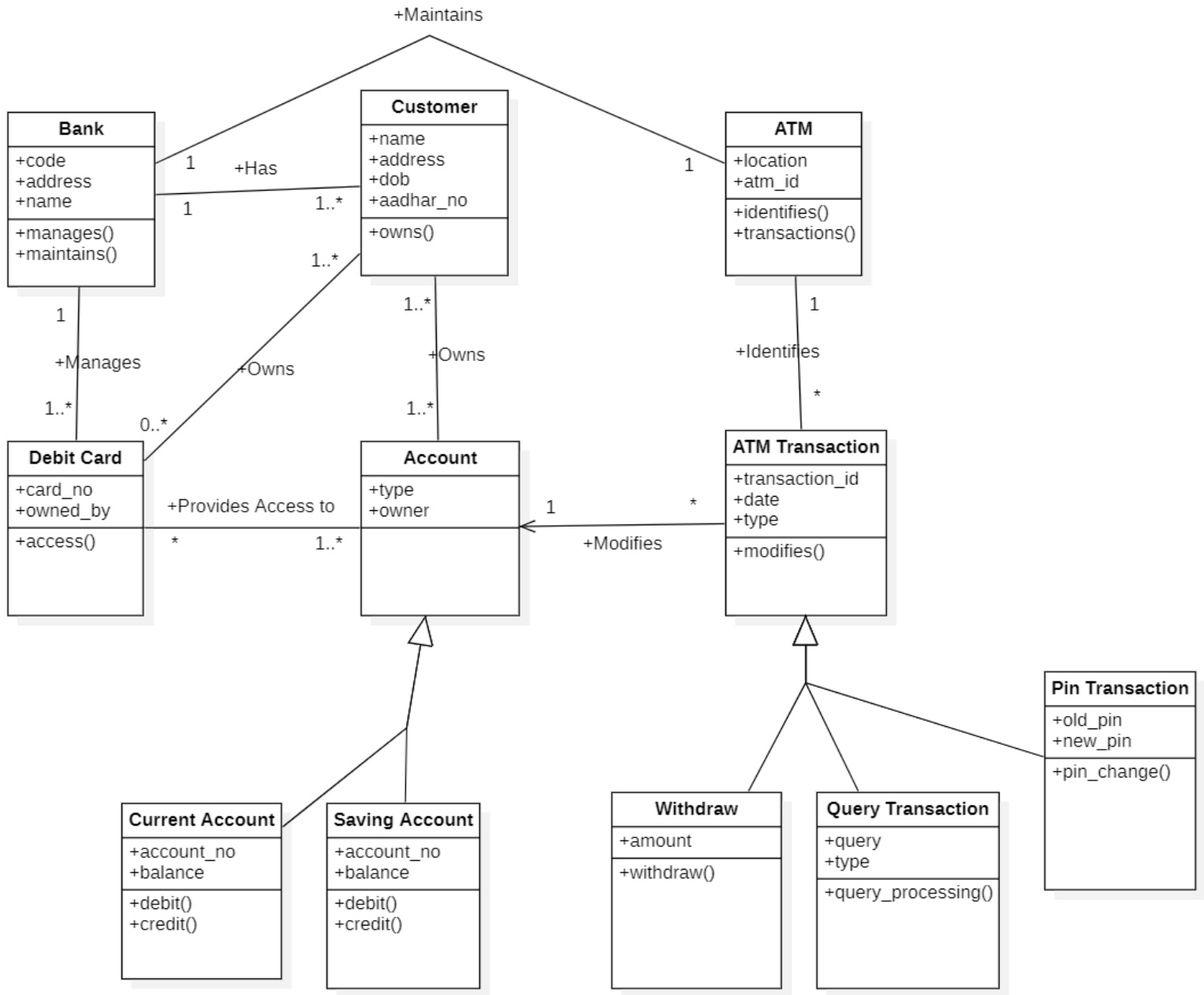


Figure 4 Class diagram for ATM

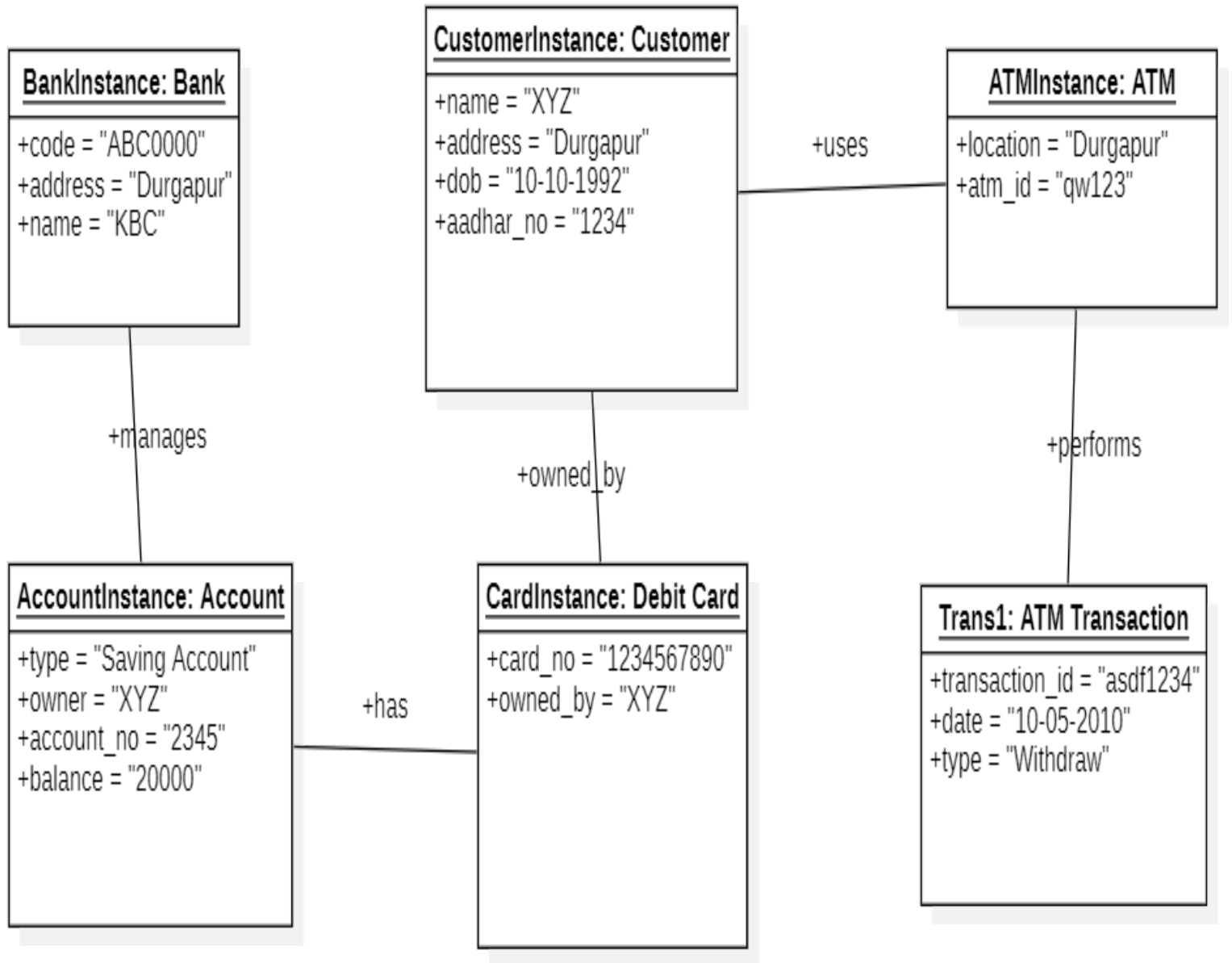


Figure 5 Object Diagram for ATM

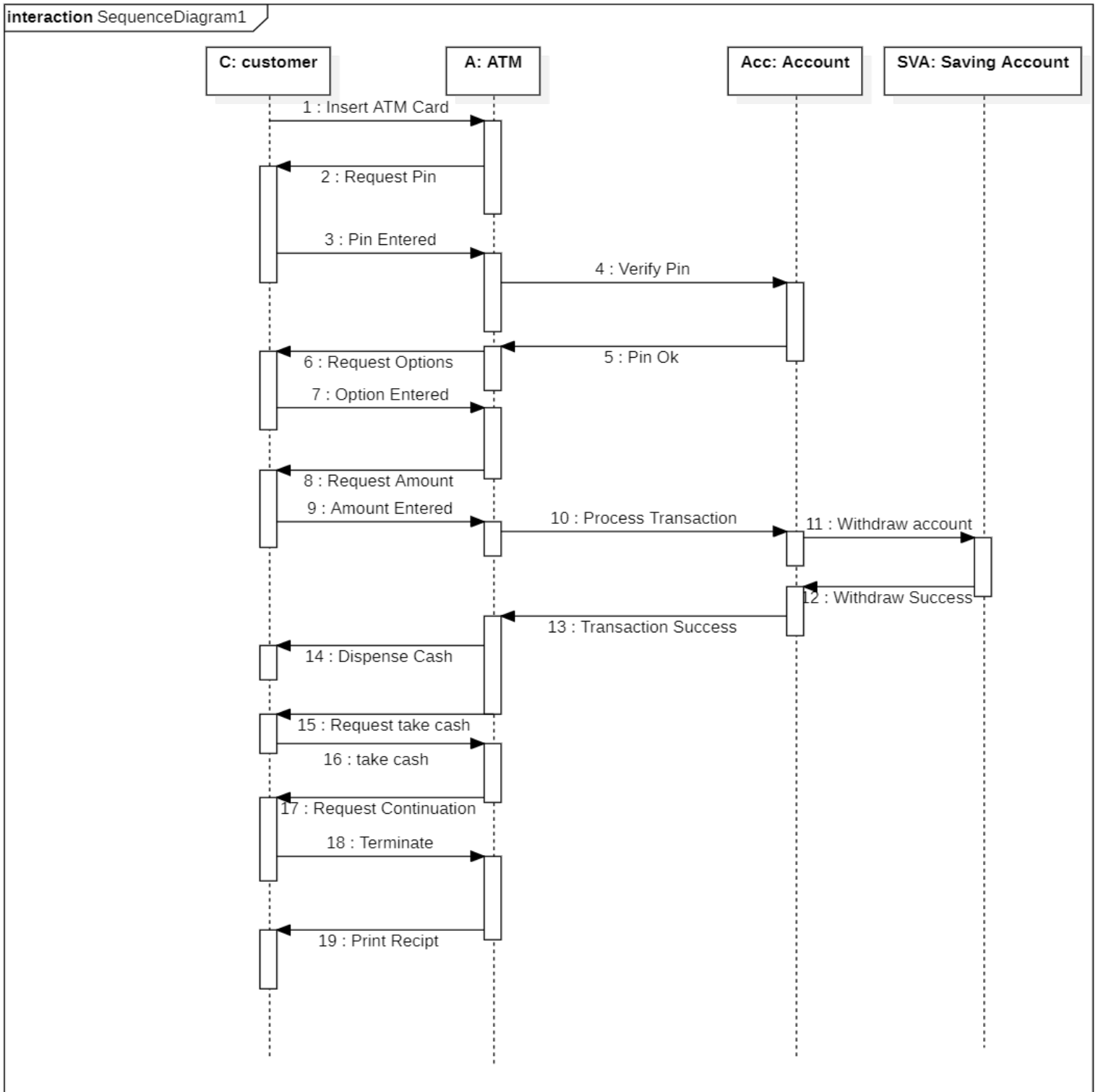


Figure 6 Sequence Diagram for ATM

B. A Clinic Centre would like to implement a system to maintain the Patient's Visit Records. The system will keep and update the information related to Registration, Out Patient Department visits, related Doctor Information and Medical history (Patient Wise). All records will be maintained with respect to the Registration Number of Patients. On Registration, each Patient is normally attached with a medical advisor (Role of Doctor) for necessary guidance. One advisor may guide several Patients at any point of time. Draw Use Case diagram, Class diagram, Object Diagram, Sequence Diagram for the above activities.

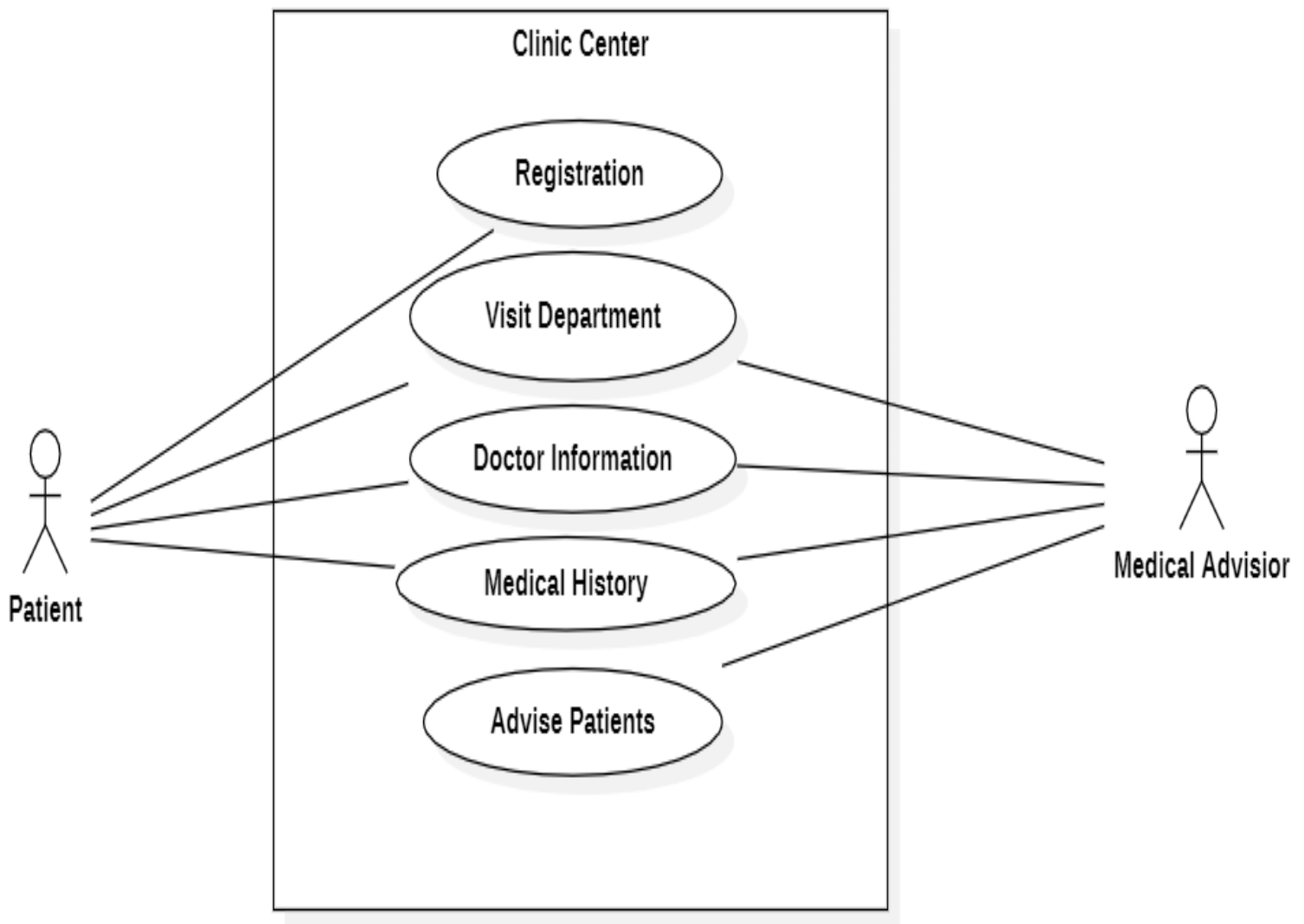


Figure 7 Use case Diagram for Clinic Center

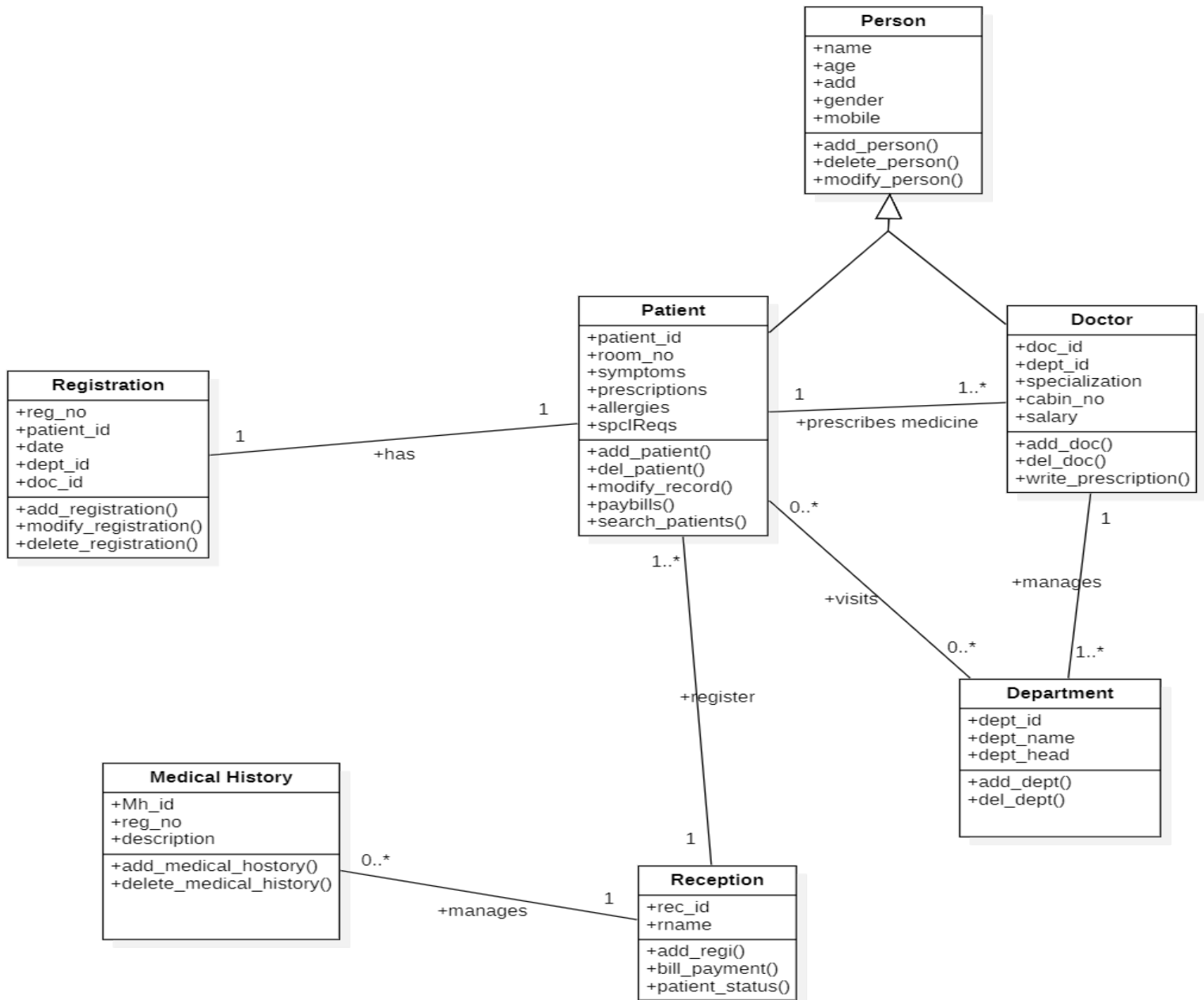


Figure 8 Class Diagram for Clinic Center

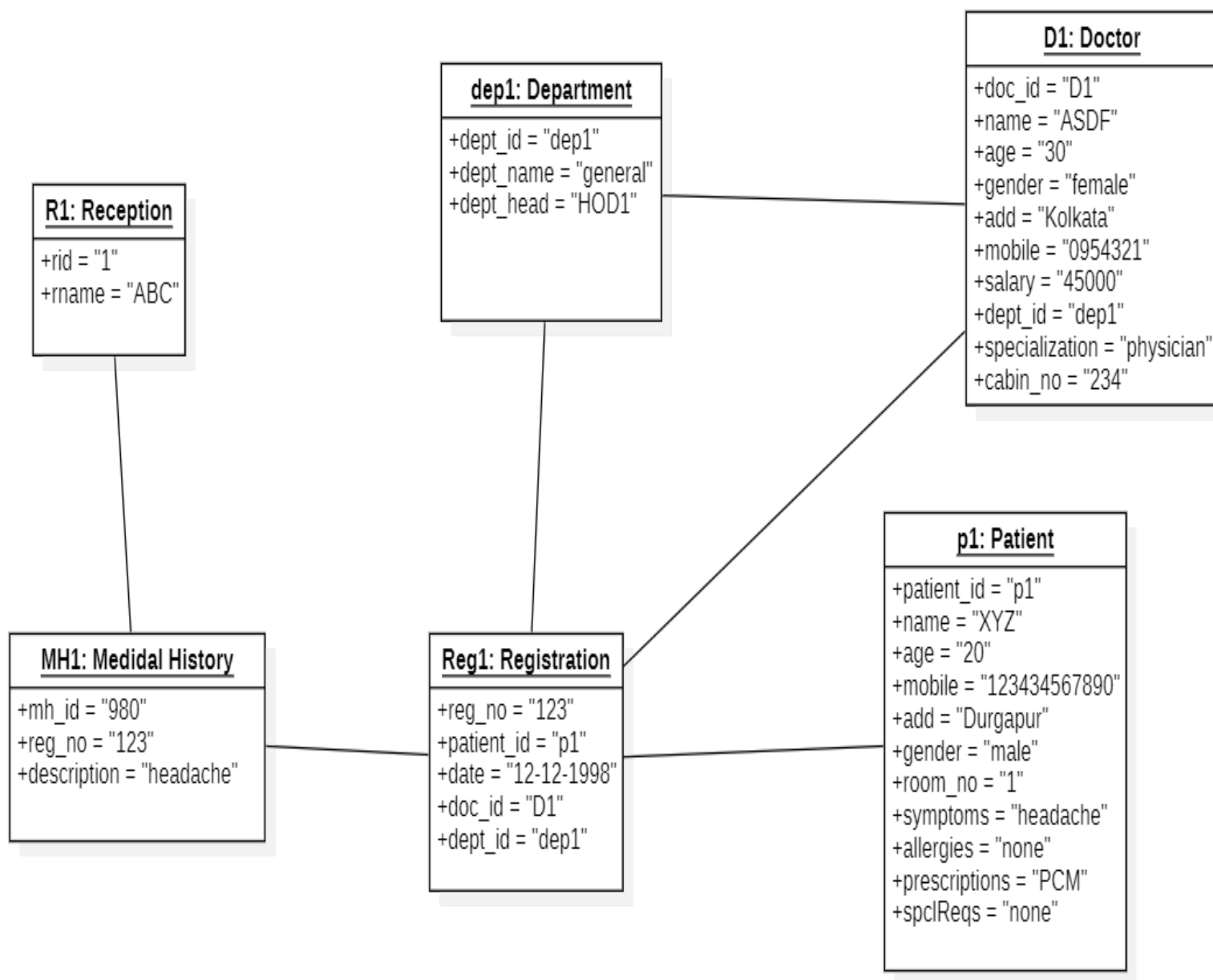


Figure 9 Object diagram for Clinic Center

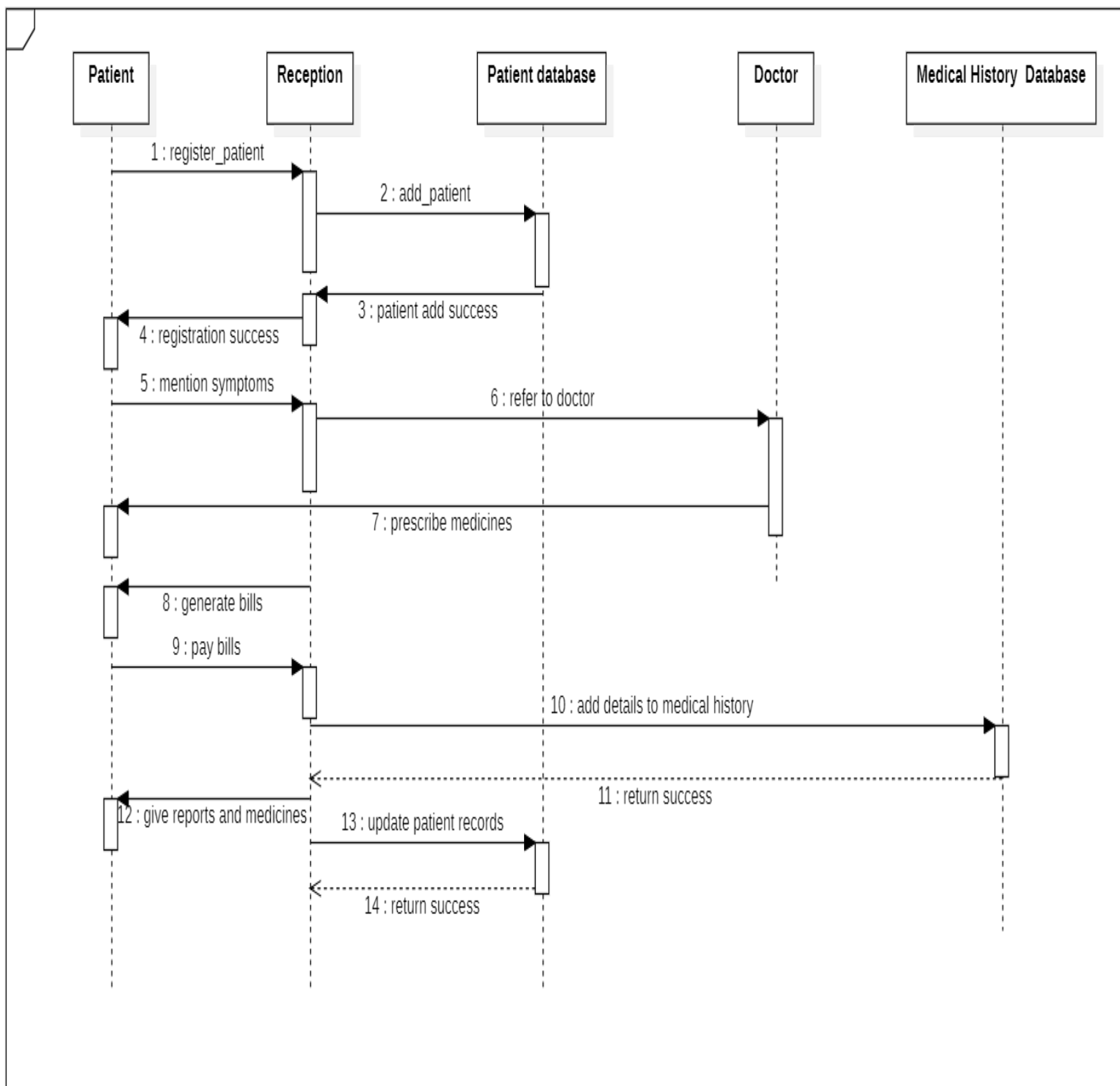


Figure 10 Sequence Diagram for Clinic Center