

**LAB RECORD**

23CSE111 – Object Oriented Programming

***Submitted by***

CH.SC.U4CSE24106 – **C S Deeraj**

**BACHELOR OF TECHNOLOGY**

IN

COMPUTER SCIENCE AND ENGINEERING

AMRITA VISHWA VIDYAPEETHAM

AMRITA SCHOOL OF COMPUTING

CHENNAI

March - 2025

## 

**AMRITA VISHWA VIDYAPEETHAM**

**AMRITA SCHOOL OF COMPUTING, CHENNAI**

**BONAFIDE CERTIFICATE**

This is to certify that the Lab Record work for 23CSE111- Object Oriented Programming Subject submitted by ***CH.SC.U4CSE24106 – C S Deeraj*** in **“Computer Science and Engineering”** is a bonafide record of the work carried out under my guidance and supervision at Amrita School of Computing, Chennai.

This Lab examination held on 13/03/2025

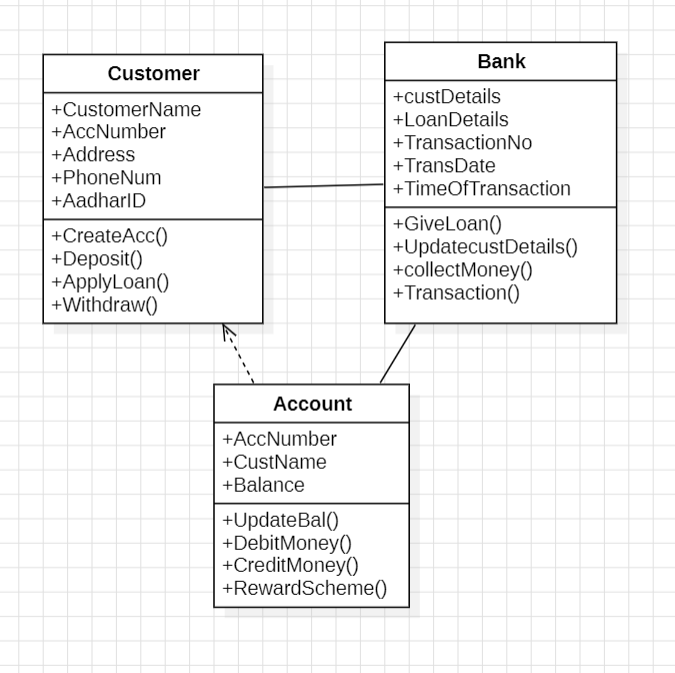
|  |  |
| --- | --- |
| Internal Examiner 1 | Internal Examiner 2 |

# Index

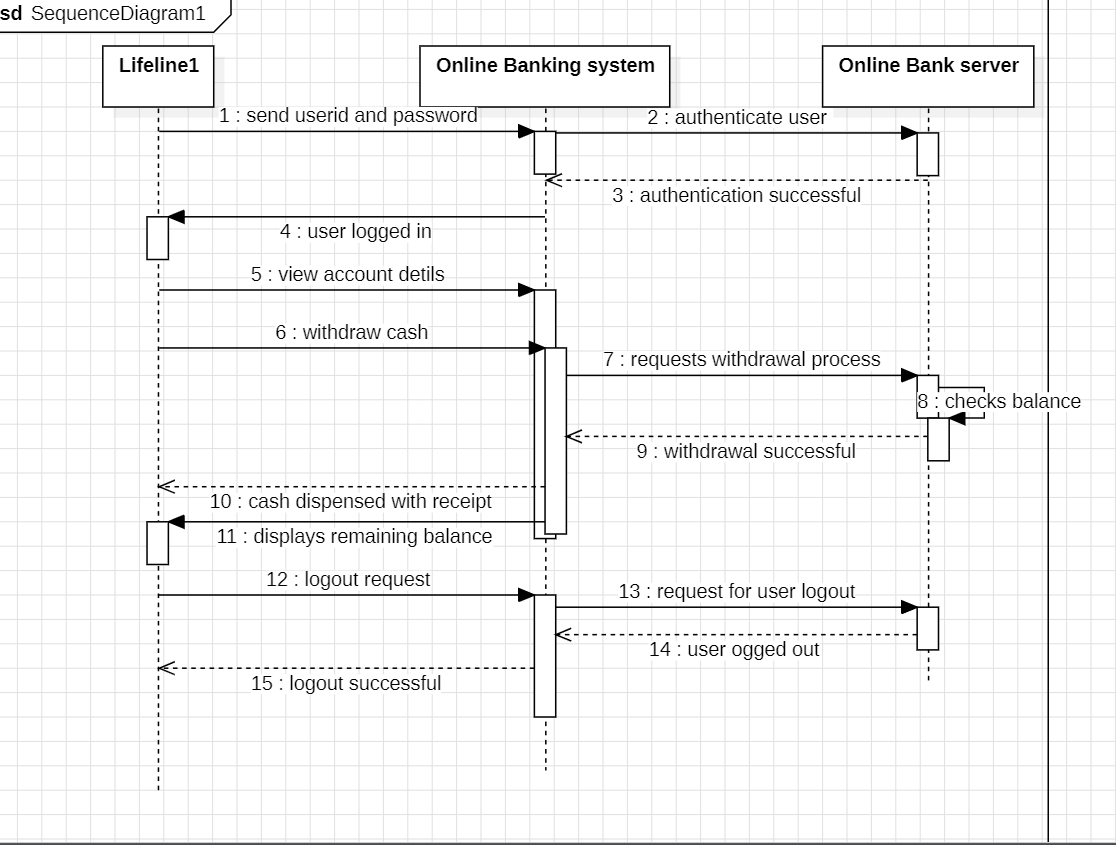
|  |  |  |
| --- | --- | --- |
| **S.NO** | **Experiment Name** | **Page Number** |
| 1. | Bank Management System UML-  *(a)Class*  *(b)Sequence*  *(c)Use Case*  *(d)State*  *(e)Collaboration* | [1](#one) |
| 2. | Restaurant Management System UML-  *(a)Class*  *(b)Sequence*  *(c)Use Case*  *(d)State*  *(e)Collaboration* | [6](#two) |
| 3. | Java Basic Programs: | 57 |
| 4. |  | 60 |
| 5. |  | 61 |
| 6. |  | 64 |
| 7. |  | 76 |

1. Bank Management System UML:

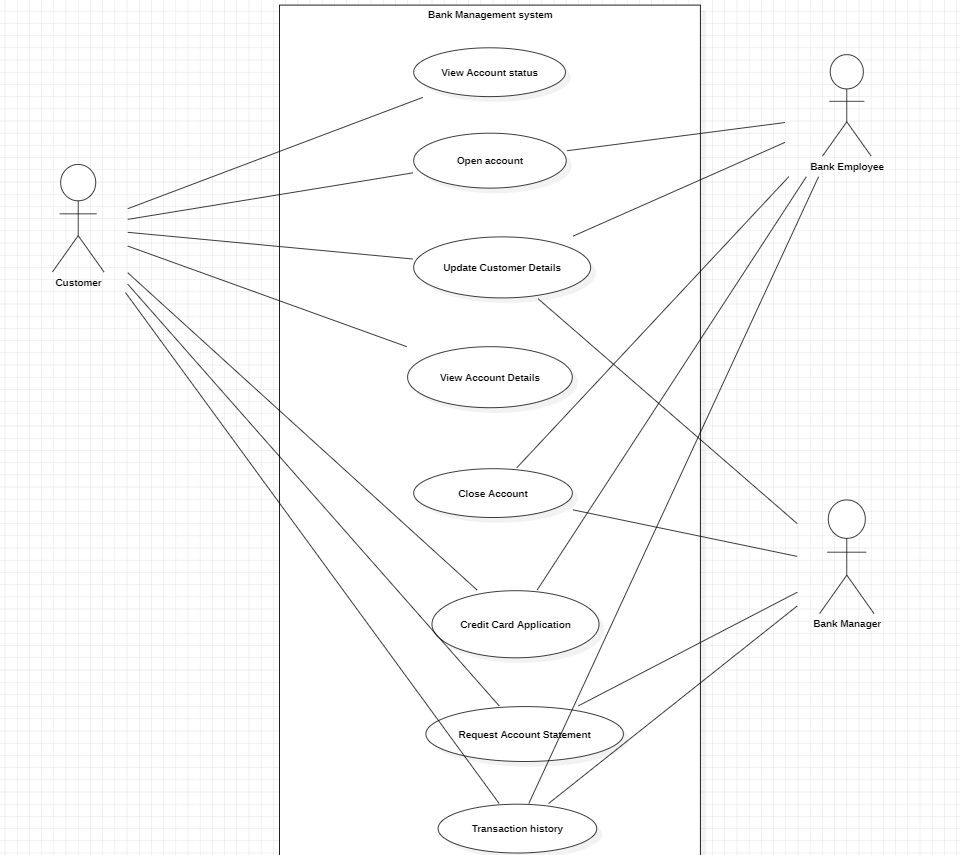
(a) Class Diagram:



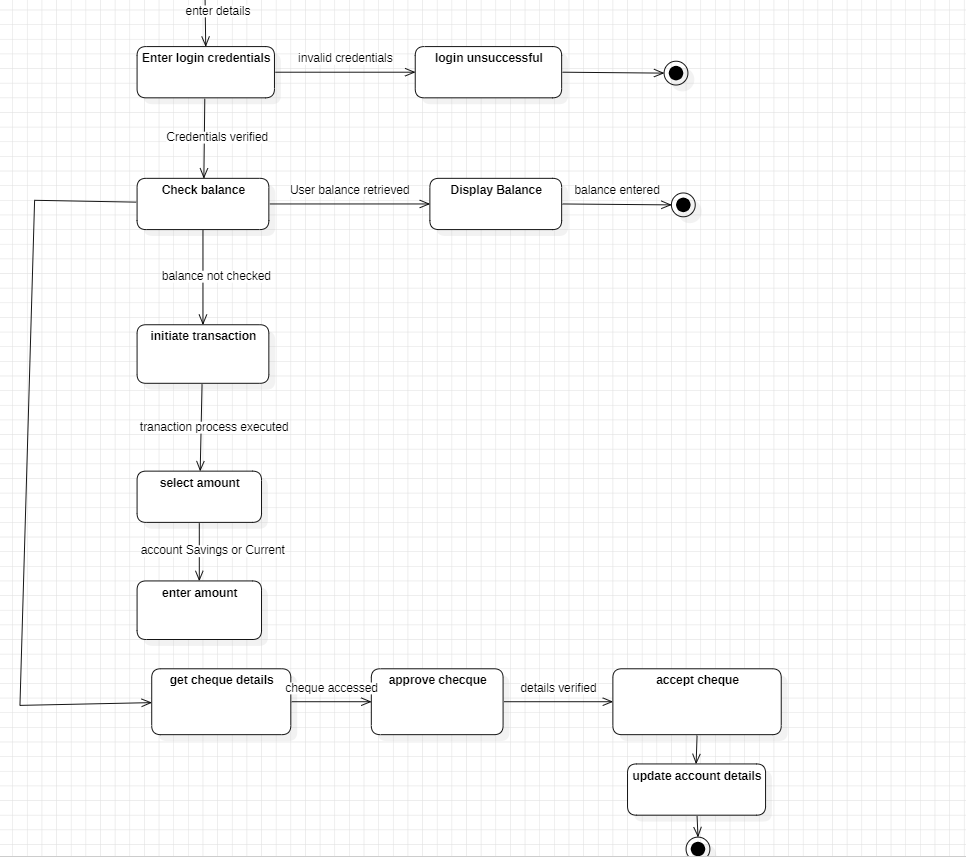
(b) Sequence Diagram:



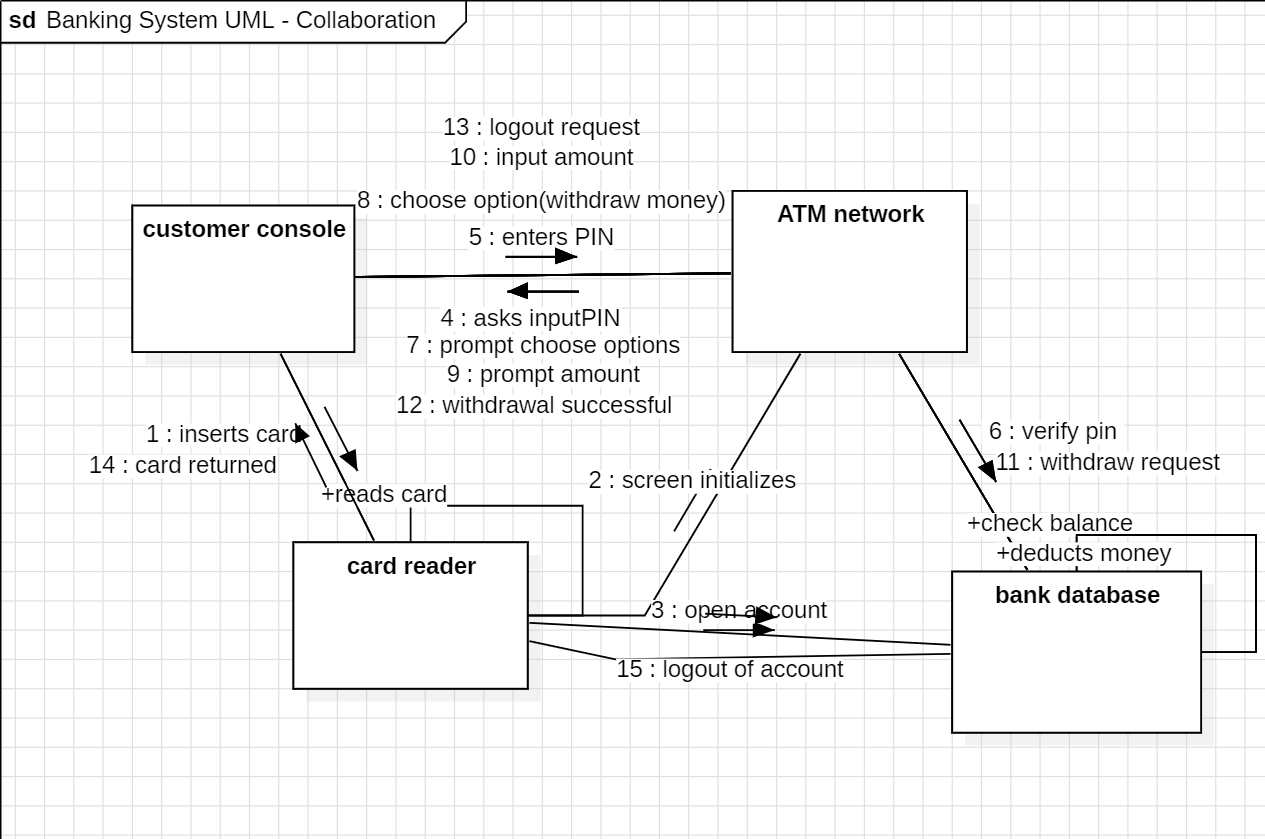
(c) Use Case Diagram:



(d) State Diagram

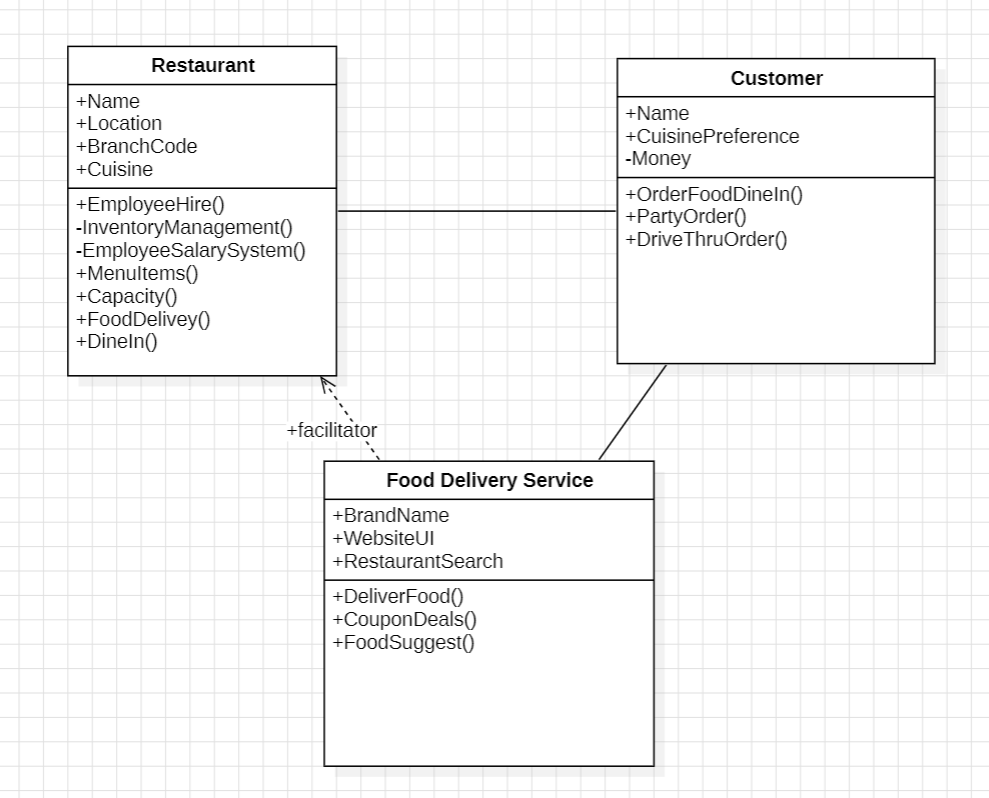


(e)Collaboration Diagram

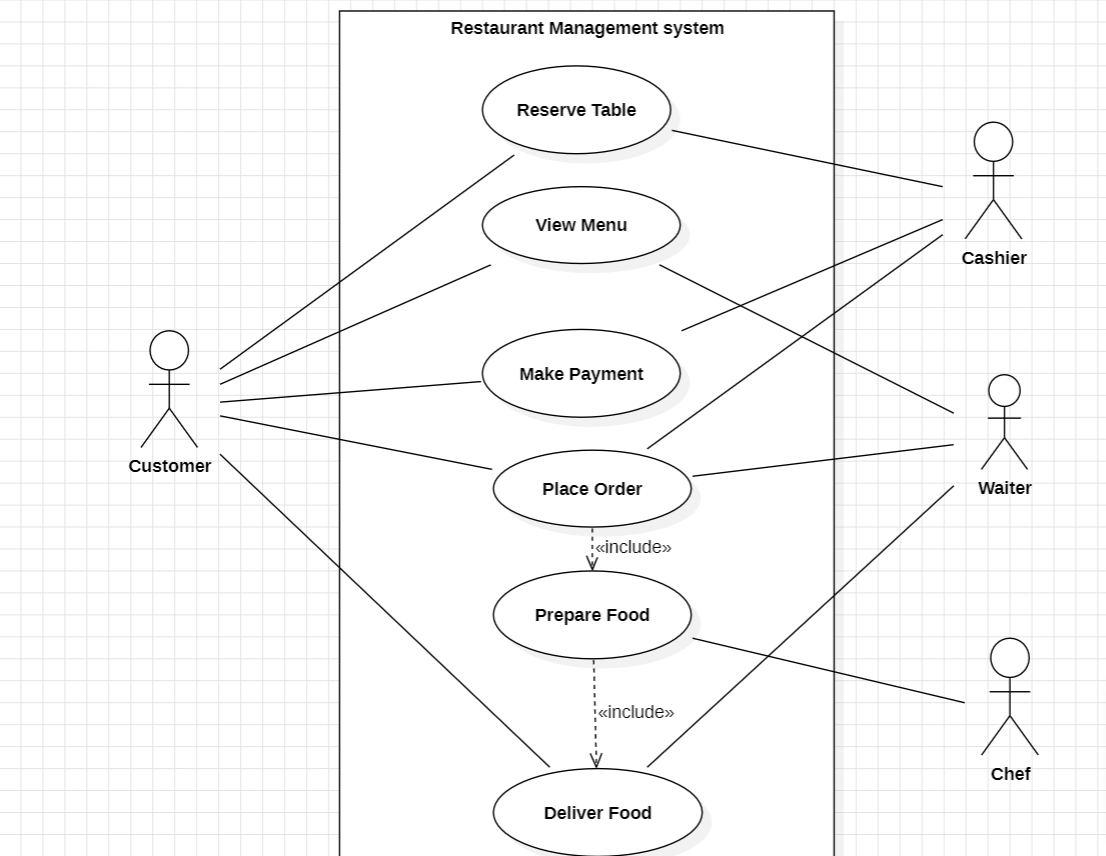


2nd UML Topic: Restaurant Management System:

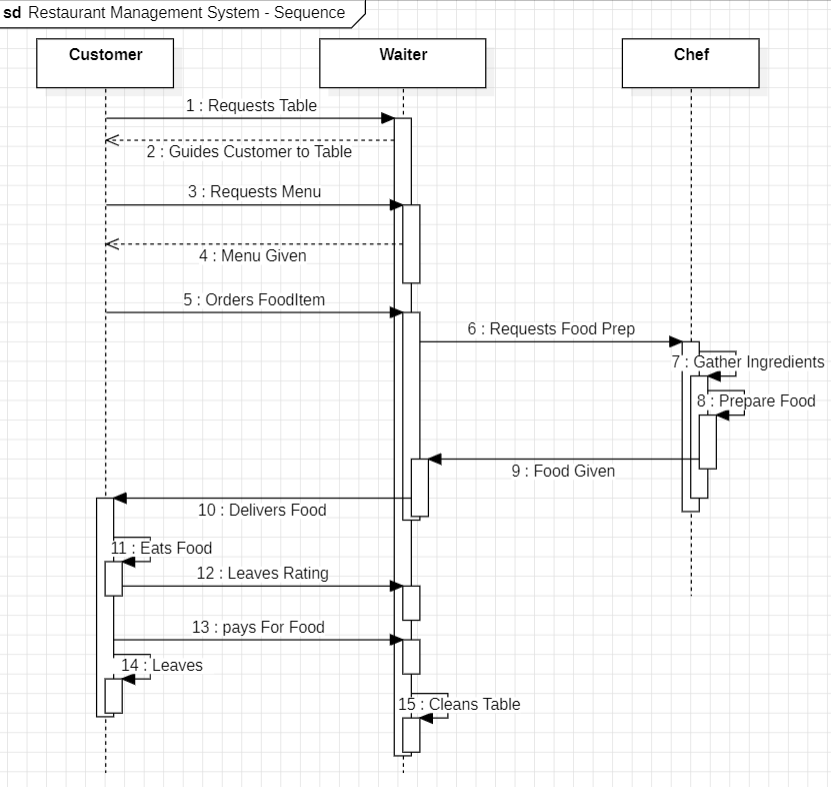
(a)Class Diagram:



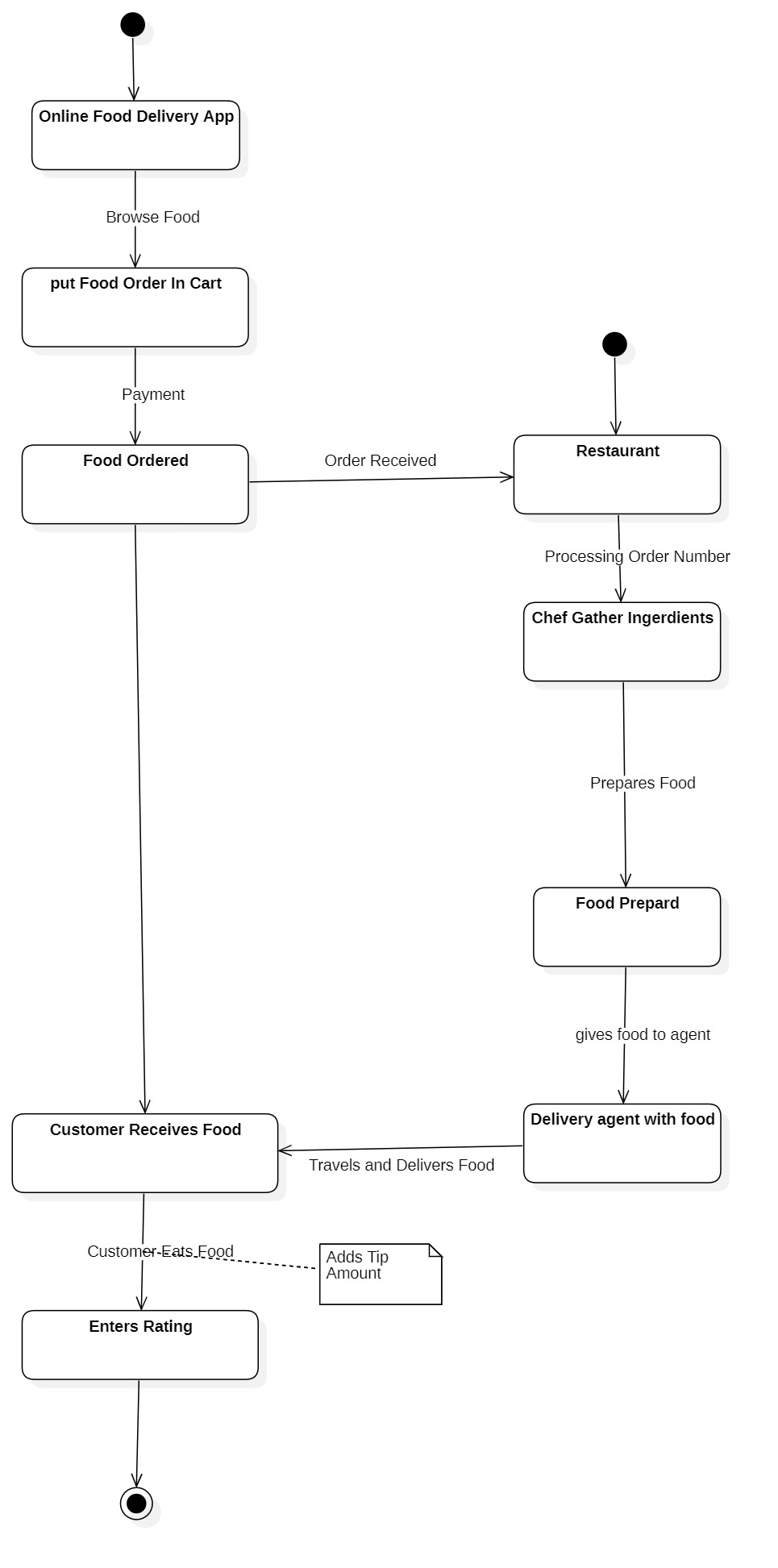
(b) Use Case Diagram:



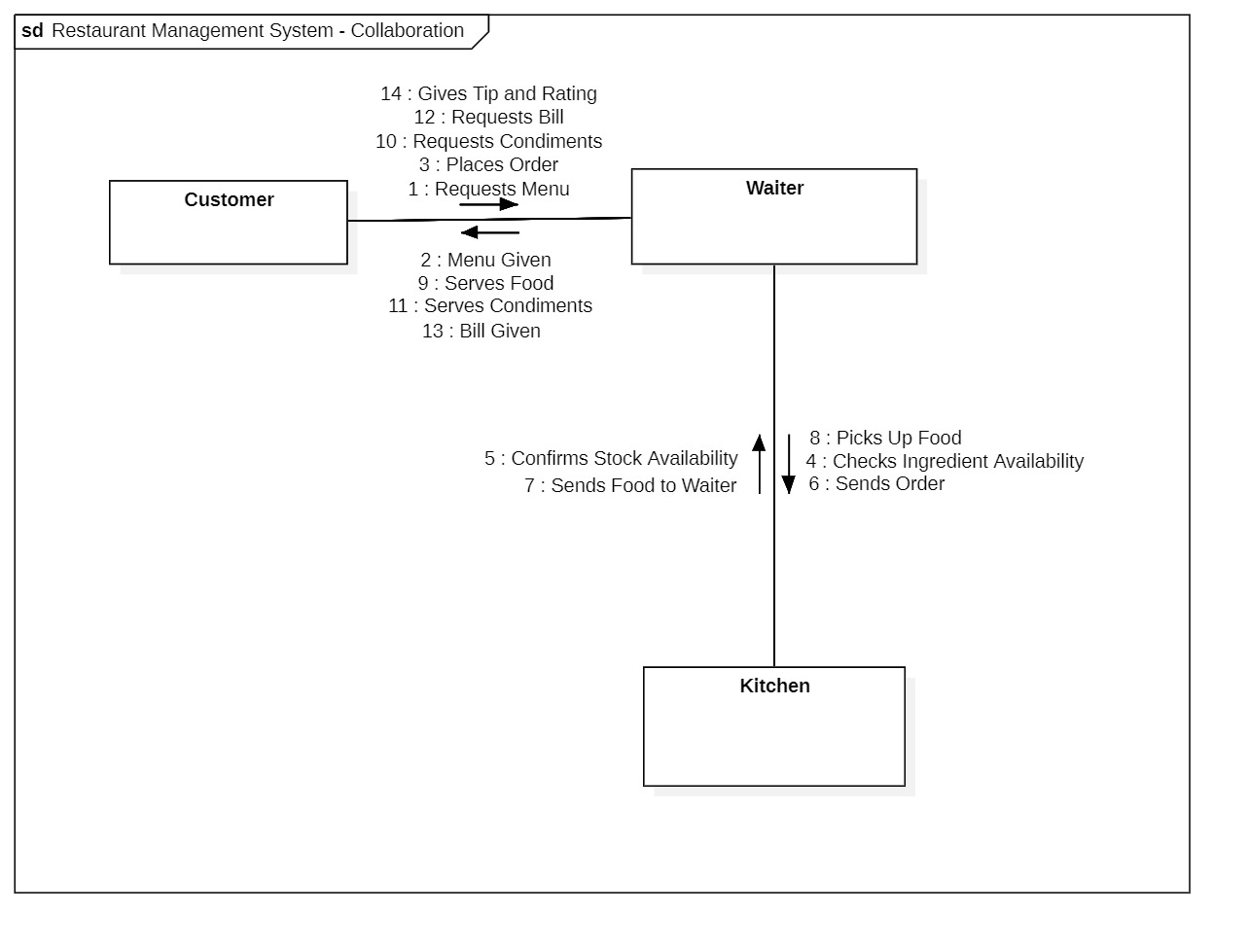
(c)Sequence Diagram:



(d) State Diagram:



(e) Collaboration Diagram:



3. Java Loop Statements, Switch Case, Jump Statements:

1. Print Numbers in Spiral Forms:

**Aim:** To print numbers in a **spiral-like decreasing row format** using loops and conditionally resetting the loop.

**Program Code:**

class spiralnumbers {

public static void main(String[] args) {

int n = 5, num = 1;

for (int i = 0; i < n; i++) {

System.out.print(num++ + " ");

if (i == n - 1) {

i = -1; n--;

System.out.println();

}

if (n == 0) break;

}

}

}

**Output:**

1 2 3 4 5

6 7 8 9

10 11 12

13 14

15

2. Infinite Odd Numbers (While Loop & Break):

**Aim:** To generate an **infinite odd number** and stop the loop using **break**.

**Program Code:**

class oddisinfinite {

public static void main(String[] args) {

int n = 1;

while (true) {

System.out.print(n + " ");

if (n > 100) break;

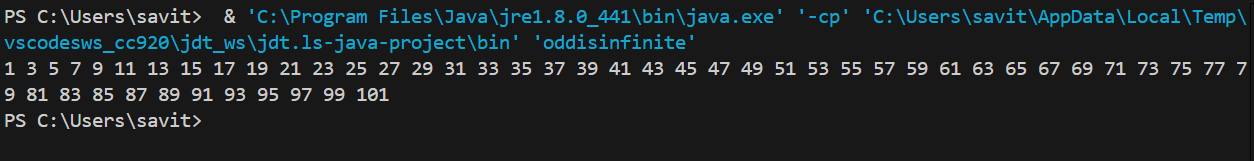
n += 2;

}

}

}

**Output:**



3**. Unique Number Series (Loop & Continue)**

**Aim:** To generate a **unique series of numbers** by **skipping even-indexed multiplications** using the continue statement.

**Program Code:**

class uniquenum {

public static void main(String[] args) {

for (int i = 1, j = 10; i <= 10; i++, j--) {

if (i % 2 == 0) continue;

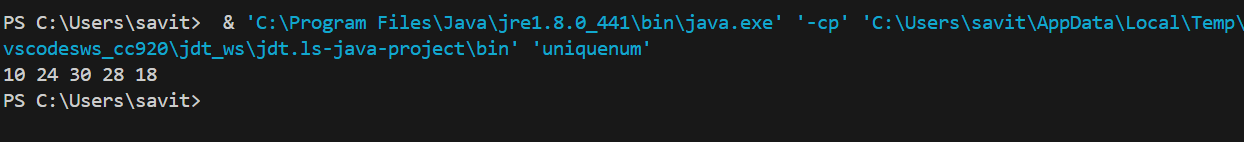
System.out.print((i \* j) + " ");

}

}

}

**Output:**



4. ATM system using switch case:

**Aim:** To **simulate an ATM system** that:**Validates PIN** with limited attempts and Provides options for **withdrawal, deposit, and balance check** using a switch statement

**Program code:**

import java.util.Scanner;

class Atmsystem {

public static void main(String[] args) {

int balance = 5000, pin = 1234;

Scanner sc = new Scanner(System.in);

for (int i = 3; i > 0; i--) {

System.out.print("Enter PIN: ");

if (sc.nextInt() == pin) break;

System.out.println("You Entered the Wrong Pin. Attempts left: " + (i - 1));

}

while (true) {

System.out.print("1.Withdraw 2.Deposit 3.Exit: ");

switch (sc.nextInt()) {

case 1 { System.out.print("Amount: "); balance -= sc.nextInt(); }

case 2 { System.out.print("Amount: "); balance += sc.nextInt(); }

case 3 System.exit(0);

}

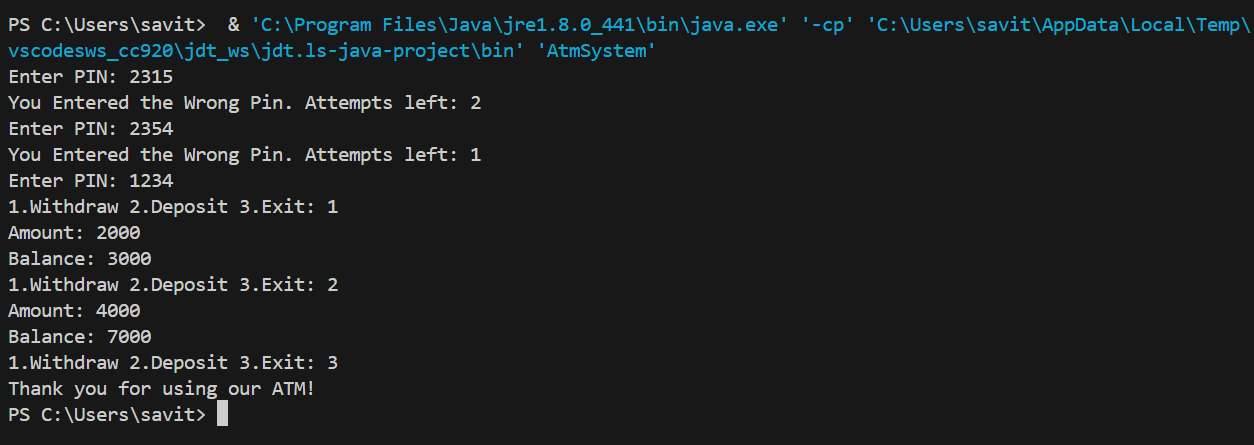
System.out.println("Balance: " + balance);

}

}

}

**Output:**

****

**5. Reverse Number without String (While & Modulus)**

**Aim: To reverse a number mathematically** without converting it into a string.

**Program Code:**

import java.util.Scanner;

class numreverse {

public static void main(String[] args) {

int n = new Scanner(System.in).nextInt(), rev = 0;

while (n > 0) {

rev = rev \* 10 + (n % 10);

n /= 10;

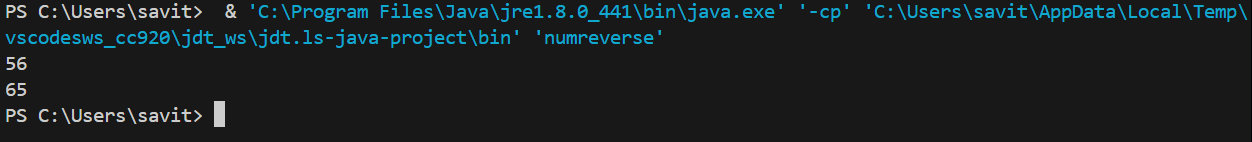
}

System.out.println(rev);

}

}

**Output:**

****

**6. Print Chess Board**

**Aim:** To print an **8x8 chessboard pattern** using **nested loops**.

**Program Code:**

class chessboard {

    public static void main(String[] args) {

        for (int i = 1; i <= 8; i++) {

            for (int j = 1; j <= 8; j++)

                System.out.print((i + j) % 2 == 0 ? "■ " : "O");

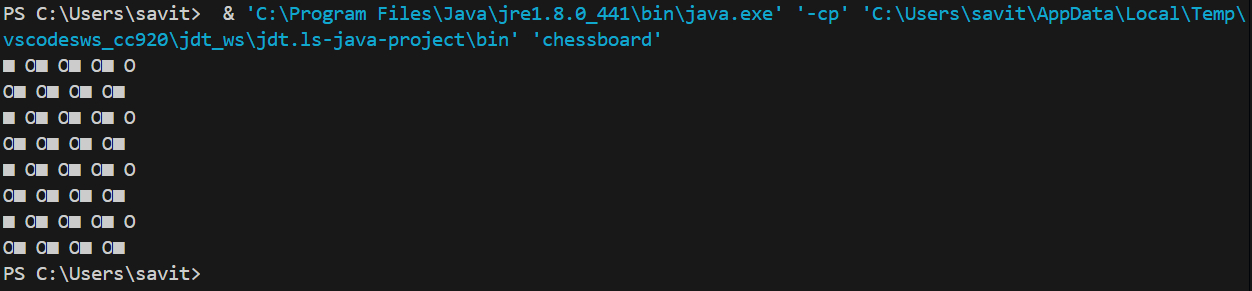
            System.out.println();

        }

    }

}

**Output:**



**7. Number Pyramid (For & Multiplication)**

**Aim:** To generate **a mathematical pyramid pattern** based on **progressive number multiplication.**

**Program Code:**

class numberpyramid {

public static void main(String[] args) {

for (int i = 1, num = 1; i <= 5; i++, num = num \* 10 + 1)

System.out.println(num \* num);

}

}

**Output:**

1

121

12321

1234321

123454321

**8. Skip multiples of 5**

**Aim:** To **iterate through number series** and print the series skipping the number 5 and it multiples

**Program Code:**

class skipmultiples {

public static void main(String[] args) {

for (int i = 1; i <= 20; i++) {

if (i % 5 == 0) continue;

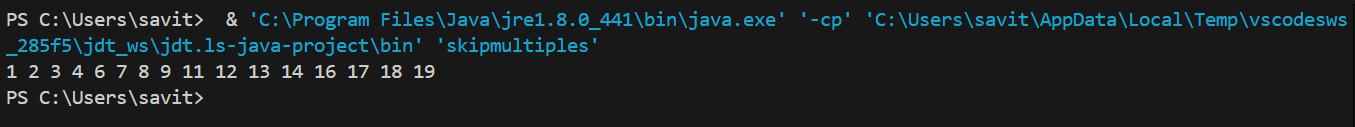
System.out.print(i + " ");

}

}

}

**Output:**

****

1 2 3 4 6 7 8 9 11 12 13 14 16 17 18 19

**9.Fibonacci Series with limit:**

**Aim:** to **generate a Fibonacci series** but stop the series after a certain number

**Program Code:**

class fibonaccibreak {

    public static void main(String[] args) {

        int a = 0, b = 1;

        while (true) {

            System.out.print(a + " ");

            if (a > 100) break;

            int temp = a + b;

            a = b;

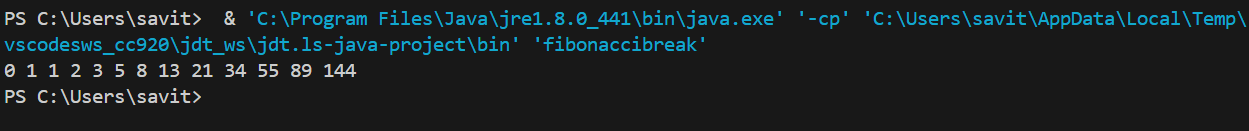
            b = temp;

        }

    }

}

**Output:**



10.**Multiplication Table using for Loop:**

**Aim:**To generate a multiplication table with the user’s number input.

**Program Code:**

import java.util.Scanner;

class multitable {

    public static void main(String[] args) {

        int n = new Scanner(System.in).nextInt();

        for (int i = 1; i <= 10; i++)

            System.out.println(n + " x " + i + " = " + (n \* i));

    }

}

**Output:**

