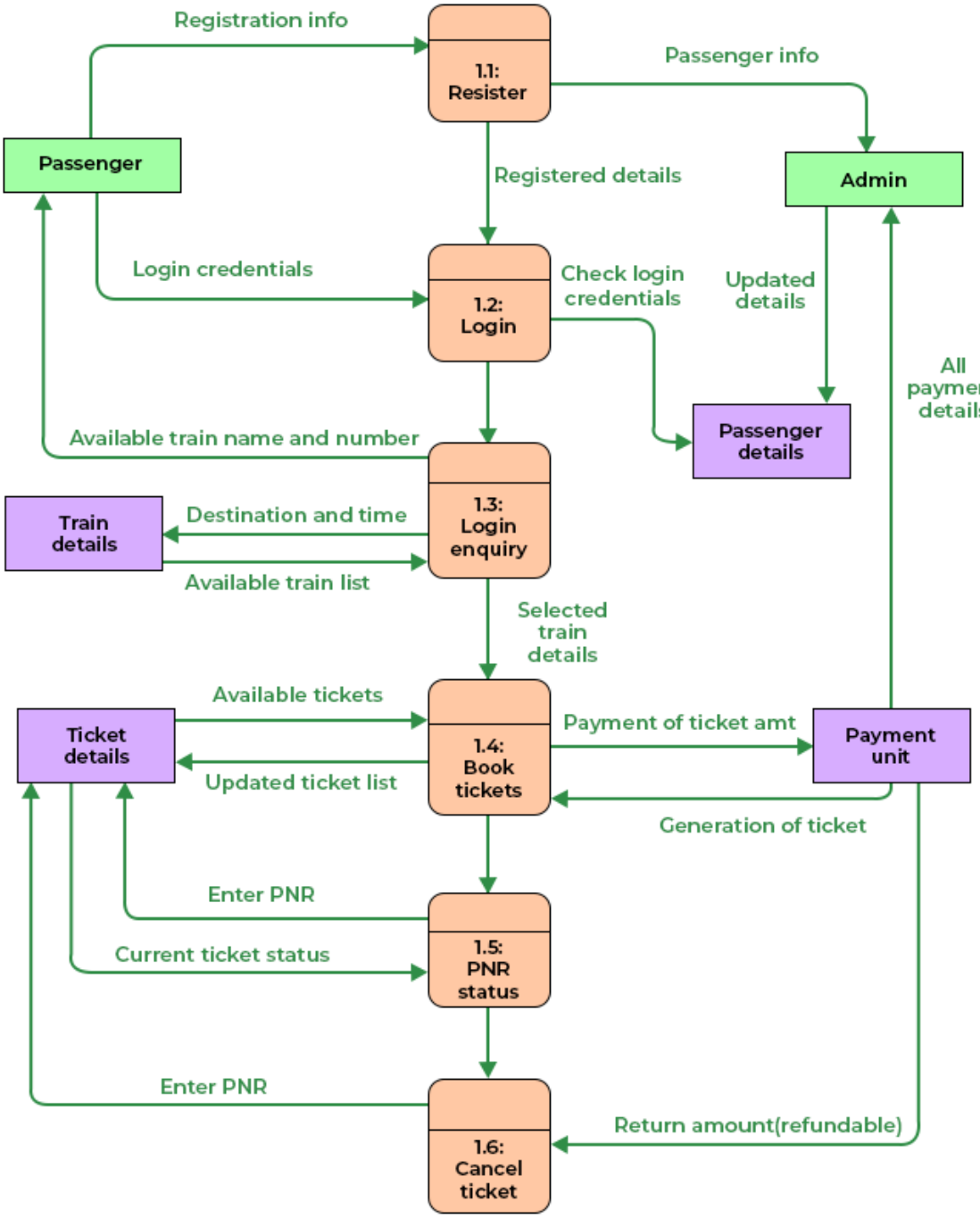
- The CASE STUDIES which are under consideration
    - Hotel/Restaurant management system
    - Railway reservation system
    - Tours and Travels booking system
    - Online banking system
    - Online inventory management system
    - Online Movie Booking System
    - Library Management System
    - Course Scheduling System

1. On the Hotel/restaurant management system, write SRS in IEEE Format

srs-restaurant

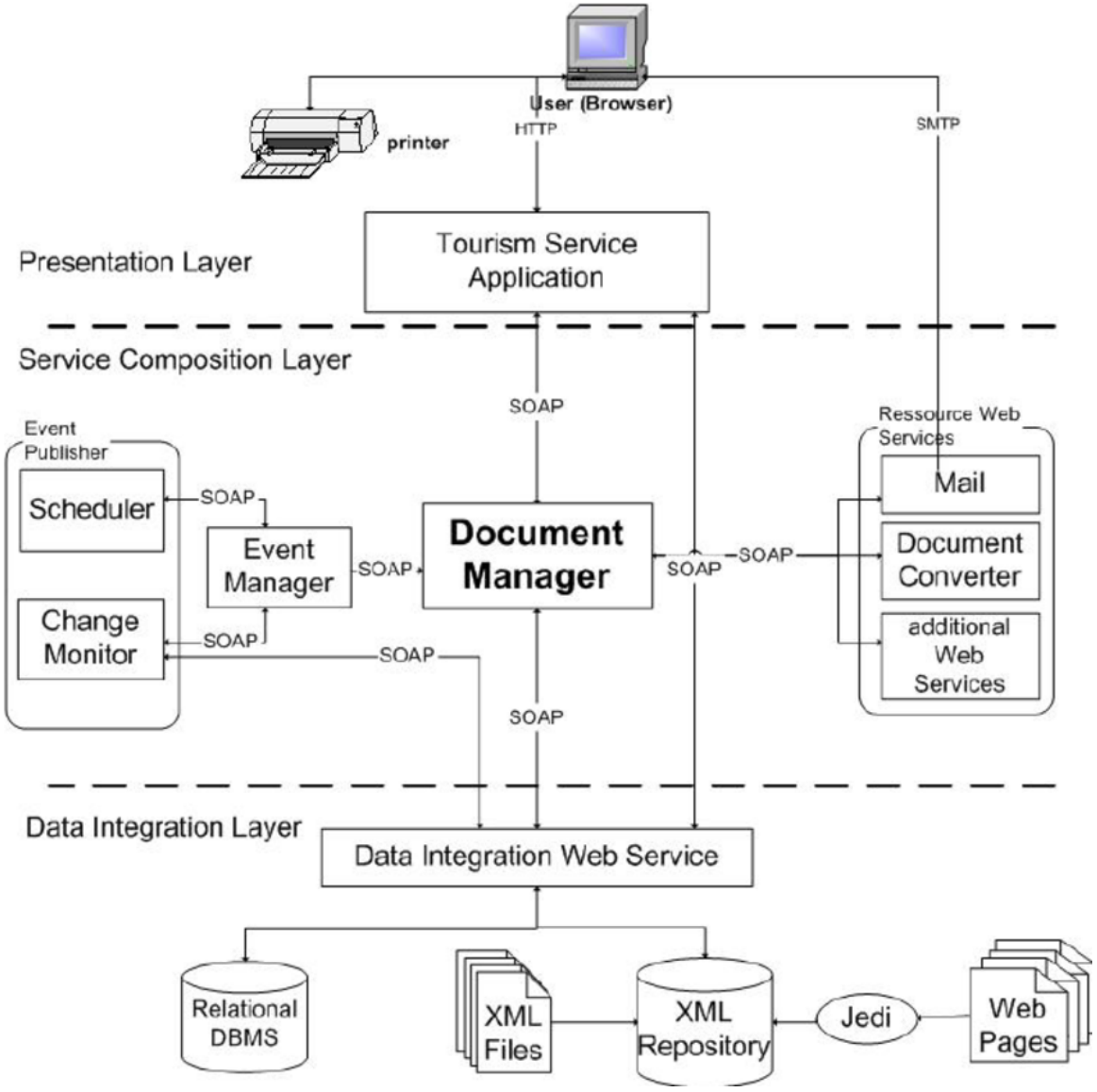
2. Mention design of Railway reservation system by considering Cohesion & Coupling


Registration info
1.1: Resister
Passenger info
Passenger
Registered details
Admin
Login credentials
1.2: Login
Check login credentials
Updated details
All payment details
Passenger details
Available train name and number
Train details
Destination and time
1.3: Login enquiry
Available train list
Selected train details
Available tickets
Ticket details
Payment of ticket amt
Payment unit
1.4: Book tickets
Updated ticket list
Generation of ticket
Enter PNR
Current ticket status
1.5: PNR status
Enter PNR
Return amount(refundable)
1.6: Cancel ticket

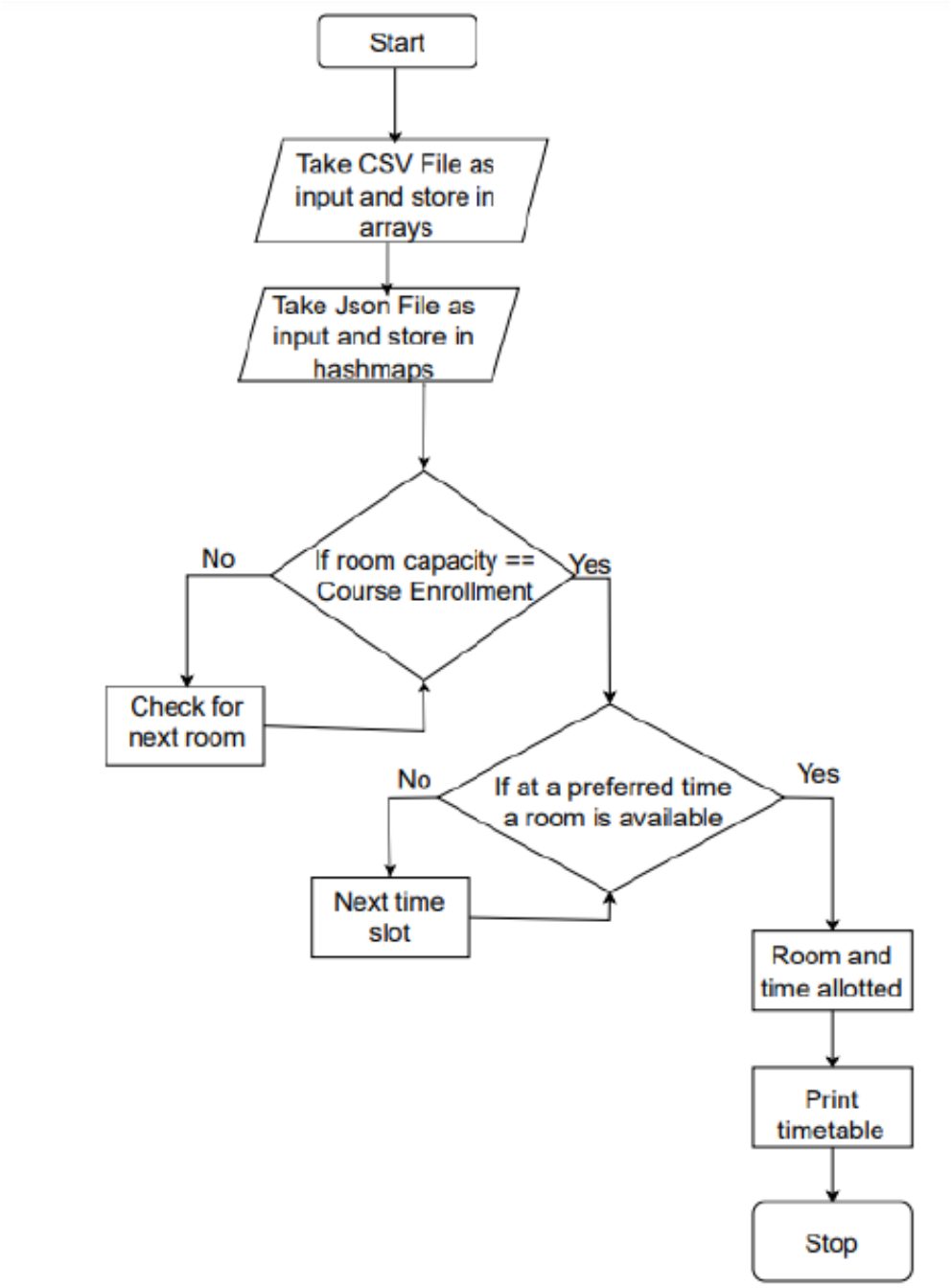
3. On the tours and travels booking system create any two architectural Style

Types of architectural styles are

1. Data Centered architecture
2. Data Flow
3. Call n return
4. Object-oriented architecture


User (Browser)
printer
HTTP
SMTP
Presentation Layer
Tourism Service Application
Service Composition Layer
SOAP
Event Publisher
Scheduler
Change Monitor
SOAP
Event Manager
SOAP
Document Manager
SOAP
SOAP
Ressource Web Services
Mail
Document Converter
additional Web Services
SOAP
SOAP
SOAP
Data Integration Layer
Data Integration Web Service
Relational DBMS
XML Files
XML Repository
Jedi
Web Pages

4. Design of white Box Testing (only on Course Scheduling System)



Number of edges: 13 ; Number of nodes: 10

V(G) = No. of Edges - No. of Nodes + 2 = 13 - 10 + 2 = 5

No. of independent paths :

1-2-3-4-6-8-9-10

1-2-3-4-5-6-8-9-10

1-2-3-4-6-7-8-9-10

1-2-3-4-5-6-7-8-9-10

Testing:

1. First For Loop to check for room available with required capacity

â— Expected Result:

Allot classroom to course if itâ€™s capacity matches the course enrollment or notify the user that a course has no suitable classroom

â— Actual Result:

The capacity and enrollment are compared to find the most suitable classroom and incase of unavailability has mechanisms to inform the user.

2. Second For Loop to check if room is available at the required time

slot.

â— Expected Result:

Once a classroom is allotted for a course, the available time slots are checked and a time slot according to the preferences is allotted for that course.

â— Actual Result:

The time slots are checked and based on preferences an attempt is made to allot the time slot, if not an empty time slot is selected and allotted.

Conclusion

Performed White box testing of our Course Scheduling System to understand and derive test cases based on the internal logic of the system components.

5. Perform Black box testing On the online banking system

Black box testing becomes useful to check the system against external factors responsible for software failures

Steps to perform black box testing: -

Sr.	Test Case Name	Input Criteria	Priority
1.	Check if user can enter amount exceeding his balance amount	1. Take amount input 2. Compare with available balance	High

Sr	Action	Output	Exp. Output	Result (pass/fail)
1.	Inputted an amount which was greater than a user's balance	System prompted 'Invalid Input'	System promp 'Invalid Input'	Pass

Sr. No	Test Case	Test Steps	Expected Result	Actual Result	Pass/Fail
1.	Check if user can login without signing up	Add unregistered user name and password	Informs user that the input login do not exist	Displays a message regarding wrong input	Pass
2	Check if user can login with correct login and password	Add registered user name and password	Login successful	Login successful	Pass
3	Check whether accurate changes are made after a debit/credit	Keep an account balance of 1000 and debit 500 and then login into the system	Total balance shows 500	Total balance shows 500	Pass

6. Perform Risk Analysis On Online inventory management system

ID	Date raised	Risk description	Likelihood of the risk occurring	Impact if the risk occurs	Severity	Owner	Mitigating action	Contingent action	Progress on actions	Status
1	11/08/2022	Project purpose and need is not well-defined. The project objectives haven't been defined to all the membersn working in the team and they are sort of lost as to where to begin the work from	Medium	High	High	Prachiti Patil	Complete a business case if not already provided and ensure purpose is well defined on Project Charter and PID.	Escalate to the Project Board with an assessment of the risk of runaway costs/never-ending project.	Business case re-written with clear deliverables and submitted to the project board for approval.	Open
2	18/08/2022	Project design and deliverable definition is incomplete. The design team hasn't gotten the hold of the design that needs to be implemented for the project	Low	High	High	Prachiti Patil	Define the scope in detail via design workshops with input from subject matter experts.	Document assumptions made and associated risks. Request high risk items that are ill-defined are removed from scope.	Design workshops scheduled.	Open
3	25/08/2022	Project schedule is not clearly defined or understood. The team has many doubts with regards to the prject and the stakeholklders also aren't having a stable time to work out the doubts of the desgin team	Low	Medium	Medium	Rhea Cutinho	Hold scheduling workshops with the project team so they understand the plan and likelihood of missed tasks is reduced.	Share the plan and go through upcoming tasks at each weekly project progress meeting.	Workshops scheduled.	Open
4	15/10/2022	The customers aren't enjoying the services that are being provided to them. They feel that the UI is a bit outadated and isn't as interactive as they expect it to be	Medium	Medium	Medium	Dhruti Vartak	Improving the basic aspects of how the web app looks whilst also adding interactive features so that the interface doesn't look monotonous	Rebuilding the website and taking the users opinion on what part of the interface they think should be made more interactive	15-20% of the website reuilding has been done as of now and more 80-85% is expected to be completed in a couple of weeks time	Waiting
5	25/10/2022	Project conflicts aren't solved in a timely manner, hence leading to delay in pushing out updates	Medium	Low	Medium	Dhruti Vartak	Add another member to the team that helps with solving any creative/program based conflicts	Ask for assistance from anoter team and according implement it in all future projects	The creative issues within the team have been solved and all other issues are still being discussed	Open
6	31/10/2022	The scope of the project hasn't been discussed by all the members of the project team	High	Medium	High	Prachiti Patil	There should be a collaborative approach from both, the project team and development team	Inquire with the project developers to see if there are better alternatives present to the current objects in use	The discussions and talks are held and positive outcomes are expected	Open

7. Create Quality Assurance plan for Online Movie Booking System

SQA Plan is as follows:

1.0 Purpose:

The purpose of this Software Quality Assurance (SQA) Plan is to establish the goals, processes, and responsibilities required to implement effective quality assurance functions for the <Project Name> project.

The <Project Name> Software Quality Assurance Plan provides the framework necessary to ensure a consistent approach to software quality assurance

throughout the project life cycle. It defines the approach that will be used by the SAM and Software Quality (SQ) personnel to monitor and assess software development processes and products to provide objective insight into the maturity and quality of the software. The systematic monitoring of <Project Name> products, processes, and services will be evaluated to ensure they meet requirements and comply with the National/state authorities <Project Name> policies, standards, and procedures, as well as applicable Institute of Electrical and Electronic Engineers (IEEE) standards.

1.1 Scope

2. Reference document
3. Management

   3.1 Mgmt organisation

   3.2 Test

4. Problem Reporting and Corrective Action
5. Tools, Techniques and Methodologies: SQ personnel will require access to the following:
6. Project Tools
7. Record Collection, Maintenance, and Retention

8. Risk Management
9. SQ personnel will assess the project's risk management process against the <Project Name> Risk Management Plan and GPG 7120.4. SQ participates in <weekly/monthly> risk management meetings and reports any software risks to the SAM and the project's Risk Manager.

8. CALCULATE FUNCTION POINT for Course Scheduling

System

1.1 Determine Unadjusted Function Point Count

No.	Measurement Parameter	Count		Low	Average	High		Total
1	External Inputs	2	x	3	**4**	6	=	**8**
2	External Outputs	1	x	4	**5**	7	=	**5**
3	External Inquires	0	x	3	**4**	6	=	**0**
4	Internal Logical Files	2	x	7	**10**	15	=	**20**
5	External Logical Files	0	x	5	7	10	=	**0**
Unadjusted Function Point Total ----------→								**33**

1.2 Determine Value Adjustment Factor

Rate Each Factor: (0 – No Influence, 1 – Incidental, 2 – Moderate, 3 – Average, 4 – Significant, 5 - Essential)

1.How many data communication facilities are there?	5
2. How are distributed data and processing functions handled?	2
3. Was response time or throughput required by the user?	0
4. How heavily used is the current hardware platform?	0
5. How frequently are transactions executed?	0
6. What percentage of the information is entered online?	0
7. Was the application designed for end-user efficiency?	5
8. How many internal logical files are updated by on-line transaction?	0
9. Does the application have extensive logical or math processing?	5
10. Was the application developed to meet one or many user needs?	2
11. How difficult is conversion and installation?	0
12. How effective/automated are stamp, backup, and recovery?	4
13. Was the application designed for multiple sites/organizations?	1
14. Was the application designed to facilitate change?	5
Value Adjustment Factor ----------→	**29**

1.3 Determine Function Points

Unadjusted Function Points x (0.65 + 0.01 x Value Adjustment Factor) ----------→	**31.02**

9. On the Hotel/restaurant management system, create COCOMO

MODEL ESTIMATION

→ Consider a KLOC Value say 400 KLOC, this is the lines of code value for the restaurant mgmt system. Calculate the Effort, Scheduled time for development by considering the developer having high application experience and very low experience in programming.

Very Low Experience in programming and AEXP is high

Take the cost driver table:

AEXP High = 0.82

VLEXP Very Low= 1.14

Calculate the EAF - Effort adjustment factor

EAF = 0.82 x *1.14 = 0.9348*

Type of system: SEMI-DETACHED

Effort = a(KLOC)b x EAF = 3.0(300)^1.12 X 0.9348

= 1,903.13

Scheduled time = c(E) d

= semi detached

= 2. 5x1903x 0.35

RATINGS						
COST DRIVERS						
PRODUCT ATTRIBUTES	Very Low	Low	Nominal	High	Very High	Extra High
RELY	0.75	0.88	1.00	1.15	1.40	..
DATA	..	0.94	1.00	1.08	1.16	..
CPLX	0.70	0.85	1.00	1.15	1.30	1.65
COMPUTER ATTRIBUTES						
TIME	..	..	1.00	1.11	1.30	1.66
STOR	..	..	1.00	1.06	1.21	1.56
VIRT	..	0.87	1.00	1.15	1.30	..
TURN	..	0.87	1.00	1.07	1.15	..

RATINGS						
COST DRIVERS						
Personnel Attributes	Very Low	Low	Nominal	High	Very High	Extra High
ACAP	1.46	1.19	1.00	0.86	0.71	..
AEXP	1.29	1.13	1.00	0.91	0.82	..
PCAP	1.42	1.17	1.00	0.86	0.70	..
VEXP	1.21	1.10	1.00	0.90	..	..
LEXP	1.14	1.07	1.00	0.95	..	..
PROJECT ATTRIBUTES						
MODP	1.24	1.10	1.00	0.91	0.82	..
TOOL	1.24	1.10	1.00	0.91	0.83	..
SCED	1.23	1.08	1.00	0.04	1.10	..

10. On Library Management System, Use JIRA TOOL(Scrum Implementation)

11. On the Tours and travels booking system, Use JIRA TOOL(Kanban Implementation)

12. Do version controlling for Online banking system (GIT HUB & JIRA)

https://medium.com/@fredrick.adegoke/version-control-systems-source-code-banking-efcbb9272aee

13. Set up a procedure for change control in course scheduling system

https://www.guru99.com/change-control-business-analyst.html

14. Set up a procedure for change control in library management system

Change Control

15. Set up a procedure for change control in the ticket booking system

16. Write user stories, epics and at least two Sprints on Course Scheduling System

User stories:

Epics:

Sprint:

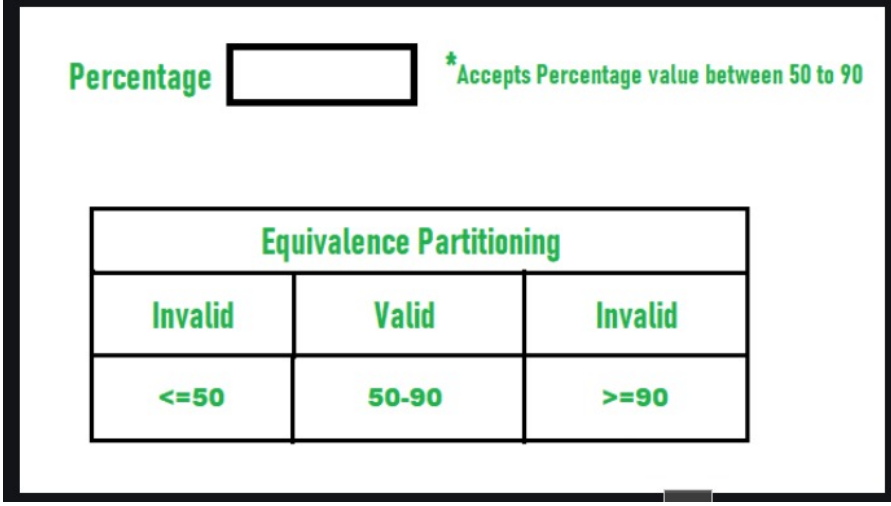
17. Write user stories, epics and at least two Sprints on online banking system

18. Use equivalence partitioning testing strategy for executing the test cases on Course Scheduling System

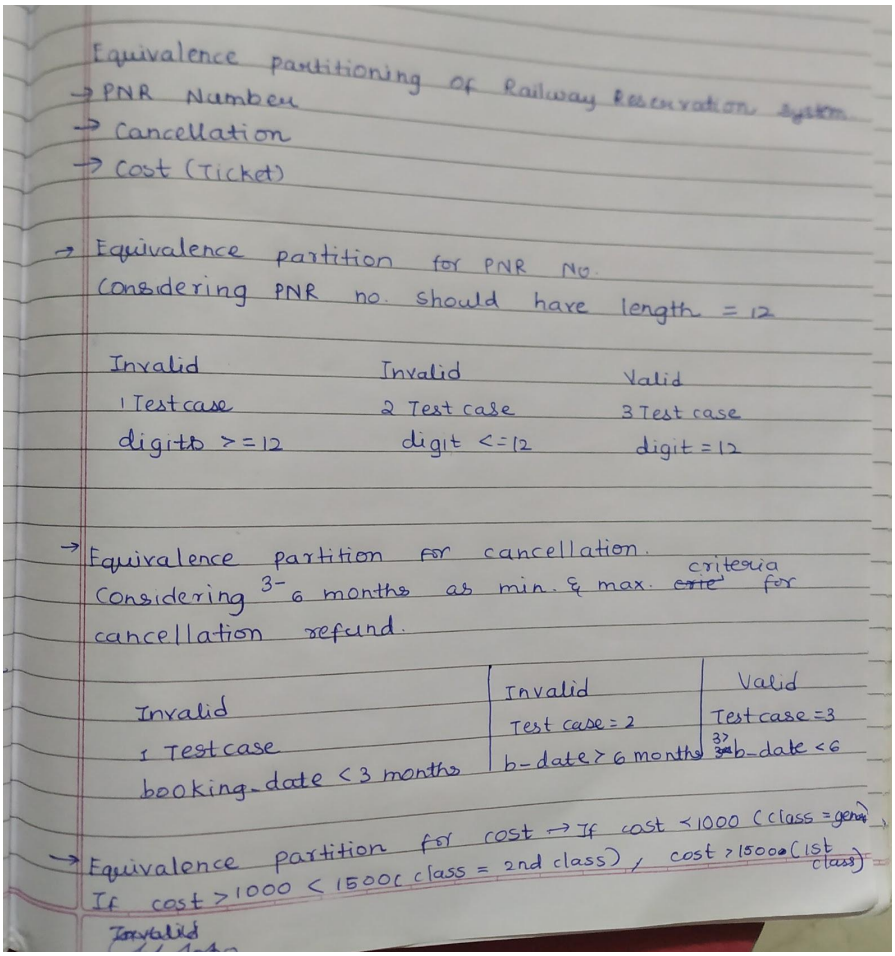
Example-1:

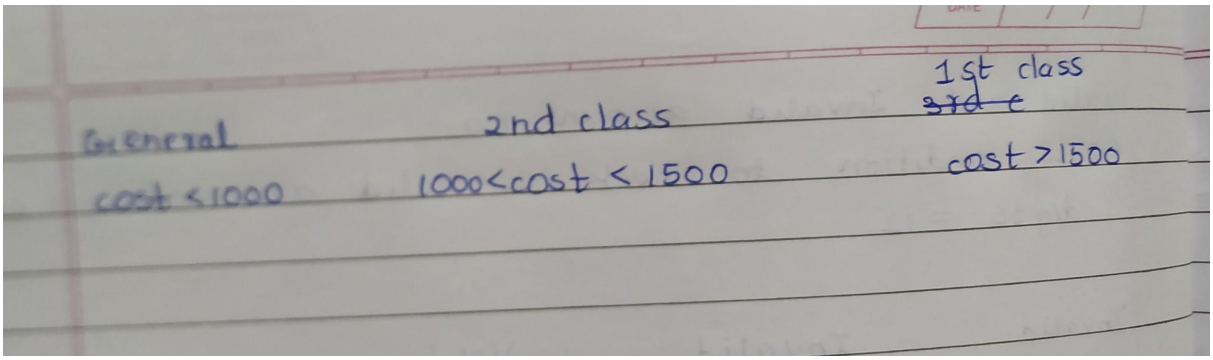
Let us consider an example of any college admission process. There is a college that gives admissions to students based upon their percentage.

Consider percentage field that will accept percentage only between 50 to 90 %, more and even less than not be accepted, and application will redirect user to an error page. If percentage entered by user is less than 50 %or more than 90 %, that equivalence partitioning method will show an invalid percentage. If percentage entered is between 50 to 90 %, then equivalence partitioning method will show valid percentage.



19. Use equivalence partitioning testing strategy for executing the test cases on railway reservation system





Train example for EP

20. Write a program in JAVA for a calculator and hence show its testing[Black Box And White Box]

```
import java.util.Scanner;


class Main {

  public static void main(String[] args) {


    char operator;

    Double number1, number2, result;


    // create an object of Scanner class

    Scanner input = new Scanner(System.in);


    // ask users to enter operator

    System.out.println("Choose an operator: +, -, *, or /");

    operator = input.next().charAt(0);


    // ask users to enter numbers
```

```
System.out.println("Enter first number");
number1 = input.nextDouble();

System.out.println("Enter second number");
number2 = input.nextDouble();

switch (operator) {

  // performs addition between numbers
  case '+':
    result = number1 + number2;
    System.out.println(number1 + " + " + number2 + " = " + result);
    break;

  // performs subtraction between numbers
  case '-':
    result = number1 - number2;
    System.out.println(number1 + " - " + number2 + " = " + result);
```

```
    break;


  // performs multiplication between numbers
  case '*':
    result = number1 * number2;
    System.out.println(number1 + " * " + number2 + " = " + result);
    break;


  // performs division between numbers
  case '/':
    result = number1 / number2;
    System.out.println(number1 + " / " + number2 + " = " + result);
    break;


  default:
    System.out.println("Invalid operator!");
    break;
}
```

```
    input.close();
  }
}
```

21. Develop a design document for any Mini Project undergone [Any sem SE/TE]

https://www.geeksforgeeks.org/design-documentation-in-software-engineering/

22. Development of DFD and ER Diagram for any Mini Project undergone [Any sem SE/TE]

23. Implementation of Course Scheduling System using Data Centered Architecture style

Cohesion

24. Write the program in Java OR C OR C++ which exhibits Functional cohesion

```
import java.util.*;
```

```
import java.util.ArrayList;
public class Main{
   static void sales_tax(String product, ArrayList<String> arr,Dictionary
dict)
   {
      double sales_tax;
      try{
         int price=Integer.parseInt(dict.get("Snacks").toString());
         if(arr.contains(product))
         {
            sales_tax=0;
            //System.out.println("The sales tax of "+product+" is Rs.
"+sales_tax );
         }
         else{
            if(price<1000)
            sales_tax= price*0.2;
            else
            sales_tax= price*0.35;
         }
         System.out.println("The sales tax of "+product+" is Rs. "+sales_tax );
      }
      catch(NullPointerException e){
         System.out.println("Caught excp");

      }
```

```
   }
        public static void main(String[] args) {
            ArrayList<String> arr = new ArrayList<String>(3);
      arr.add("milk");
      arr.add("flowers");
      arr.add("fruits");
      System.out.println("The array you entered is "+arr);

      Dictionary<String,Integer> dict = new Hashtable<String,Integer>();
      dict.put("Smartphones", 20000);
      dict.put("Snacks", 20);
      dict.put("Chocolates", 100);

      sales_tax("Snacks",arr,dict);
      System.out.println(dict.get("Snacks"));
        }
}
```

https://www.linkedin.com/pulse/types-cohesion-ahmed-adel/

25. Write the program in Java OR C OR C++ which exhibits Sequential cohesion

```
public void playWithPartyEffect(Audio audio) {
```

```
    SoundEffect soundEffect = new SoundEffect(400, 800, 0);
    soundEffect = new LiveEffectFilter().apply(soundEffect);
    soundEffect = new EchoFilter().apply(soundEffect);
    soundEffect = new ExtraBassFilter().apply(soundEffect);
    soundEffect = new DelayFilter().apply(soundEffect);
    soundEffect.playSound(audio);
}
public class SoundEffect {

    private int highFrequency;
    private int lowFrequency;
    private long repeatAfterMillis;

    public SoundEffect(int highFrequency, int lowFrequency, long
repeatAfterMillis) {
        this.highFrequency = highFrequency;
        this.lowFrequency = lowFrequency;
        this.repeatAfterMillis = repeatAfterMillis;
    }

    public void updateHighFrequency(int hertz) {
        highFrequency += hertz;
    }

    public void updateLowFrequency(int hertz) {
```

```
        lowFrequency += hertz;
    }

    public void updateRepeatAfterMillis(long millis) {
        repeatAfterMillis += millis;
    }
  public SoundEffect copy() {
        return new Sound(highFrequency, lowFrequency, repeatAfterMillis);
    }

    public void play(Audio audio) {
        // ...
    }

}

// the following functions will be grouped in the same package :

public class DelayFilter implements Function<SoundEffect, SoundEffect> {

    @Override
    public SoundEffect apply(SoundEffect sound) {
        SoundEffect newSound = sound.copy();
        newSound.updateHighFrequency(100);
        newSound.updateRepeatAfterMillis(20);
        return newSound;
```

```
    }
}

public class EchoFilter implements Function<SoundEffect, SoundEffect> {


   @Override
    public SoundEffect apply(SoundEffect sound) {
        SoundEffect newSound = sound.copy();
        newSound.updateLowFrequency(50);
        newSound.updateRepeatAfterMillis(200);
        return newSound;
    }
}

public class ExtraBassFilter implements Function<SoundEffect,
SoundEffect> {


    @Override
    public SoundEffect apply(SoundEffect sound) {
        SoundEffect newSound = sound.copy();
        newSound.updateLowFrequency(8000);
        return newSound;
    }
}
```

```
public class LiveEffectFilter implements Function<SoundEffect,
SoundEffect> {


    @Override
    public SoundEffect apply(SoundEffect sound) {
        SoundEffect newSound = sound.copy();
        newSound.updateHighFrequency(800);
        newSound.updateLowFrequency(400);
        return newSound;
    }
}
```

26. Write the program in Java OR C OR C++ which exhibits

Communicational cohesion

```
public class Ticket {
```

```
    // only controlled by Ticket class, not updated from outside
    private int id = (int) (Math.random() * 10000);
    public int getId() {
        return id;
    }
}
public class TicketsTeller {
    // only controlled by TicketsTeller class, not updated from outside
    private final List<Integer> soldTicketIds = new ArrayList<>();

    public Ticket buyTicket() {
        Ticket ticket = new Ticket();
        soldTicketIds.add(ticket.getId());
        return ticket;
    }
    public int soldTicketsCount() {
        return soldTicketIds.size();
    }
}
```

27. Write the program in Java OR C OR C++ which exhibits Procedural cohesion

```
public void updateFiles(){

    StorageManager manager = new StorageManager();
```

```
    manager.readOldFileFromDisk();
    manager.scanOldFileForNewLines();
    manager.scanOldFileForWhiteSpaces();
    manager.fillInNewFile();

}

// the procedures that handles files are put in the same class or package :

public class StorageManager {

    private static final String OLD_FILE_PATH = "my_old_text_file.txt";
    private static final String NEW_FILE_PATH = "my_new_text_file.txt";
    private File oldFile;
    private File newFile;

    public void readOldFileFromDisk() {
        oldFile = new File(OLD_FILE_PATH);
    }

    public void scanOldFileForWhiteSpaces() {
        if (oldFile != null) {
            // process on "oldFile" variable
        } else {
            throw new UnsupportedOperationException("invoke
readOldFileFromDisk() first");
```

```
        }
    }

    public void scanOldFileForNewLines() {
        if (oldFile != null) {
            // process on "oldFile" variable
        } else {
            throw new UnsupportedOperationException("invoke
readOldFileFromDisk() first");
        }
    }

    public void fillInNewFile() {
        newFile = new File(NEW_FILE_PATH);
        // ...
    }
}
```

28. Write the program in Java OR C OR C++ which exhibits Temporal cohesion

```
public class AndroidApplication extends Application {

    @Override
```

```
    public void onCreate() {
        super.onCreate();
        ApplicationInitializer initializer = new ApplicationInitializer();

        initializer.initializeDatabase();

        Date startDate = initializer.detectApplicationStartDate();
        initializer.writeStartDateToDatabase(startDate);

        initializer.initializeExternalLibraries();
    }

}

// the steps that are invoked at the initialization of the application
// will be grouped in the same package or class :


class ApplicationInitializer {

    public void initializeDatabase(){
        // ...
    }

    public Date detectApplicationStartDate(){
        return new Date();
```

```
    }

    public void writeStartDateToDatabase(Date date){
        // ...
    }


   public void initializeExternalLibraries(){
        // ...
    }

}
```

29. Write the program in Java OR C OR C++ which exhibits Logical cohesion

Logical cohesion

30. Write the program in Java OR C OR C++ which exhibits Coincidental cohesion

31. Write the program in Java OR C OR C++ which exhibits Content coupling

32. Write the program in Java OR C OR C++ which exhibits Common coupling

33. Write the program in Java OR C OR C++ which exhibits External coupling

Code:

```
import java.util.*;
//this is an example of functional cohesion
//this is also an example of external coupling -> as modules access the same global data
//variable i.e. sales_tax
import java.util.ArrayList;
public class Main
{
   static void sales_tax(String product, ArrayList<String> arr,Dictionary dict)
   {

      double sales_tax;
      try{
         int price=Integer.parseInt(dict.get("Snacks").toString());
```

```
            if(arr.contains(product))
        {
            sales_tax=0;
            //System.out.println("The sales tax of "+product+" is Rs. "+sales_tax
);
        }
        else{
            if(price<1000)
            sales_tax= price*0.2;
            else
            sales_tax= price*0.35;
        }
        System.out.println("The sales tax of "+product+" is Rs. "+sales_tax );
        }
        catch(NullPointerException e){
             System.out.println("Caught excp");

        }



  }
          public static void main(String[] args) {
                ArrayList<String> arr = new ArrayList<String>(3);
        arr.add("milk");
        arr.add("flowers");
```

```
        arr.add("fruits");
        System.out.println("The array you entered is "+arr);

        Dictionary<String,Integer> dict = new Hashtable<String,Integer>();
        dict.put("Smartphones", 20000);
        dict.put("Snacks", 20);
        dict.put("Chocolates", 100);

        sales_tax("Snacks",arr,dict);
        System.out.println(dict.get("Snacks"));
          }
}
```

34. Write the program in Java OR C OR C++ which exhibits

Control coupling

```
public class Main
{
    //function for area calculation
    static void area(int length, int breadth){
        int area_cal = length * breadth;
        System.out.println("Area is "+area_cal);
    }
         public static void main(String[] args)
         {
                  //System.out.println("Hello World");
                  int l = 12;
                  int b = 3;
                  area(l,b);
                  //these 2 parameters l and b will have the control to
```

```
the path that the
                //function area will choose thus it is displaying
control coupling
        }
}
```

35. Write the program in Java OR C OR C++ which exhibits Stamp coupling

Stamp coupling occurs when modules share a composite data structure. Composite means that the data has some internal structure to it.

JAVA Code:

36. Write the program in Java OR C OR C++ which exhibits Data coupling

Data Coupling: It is the manner/degree to which one software component influences the execution of another software component.

JAVA Program:

37. Re-evaluation of Course Scheduling System

- *VIVA WilL BE BASED ON THE ENTIRE SYLLABUS AS WELL AS ON THE ASSIGNED PRACTICAL EXAM QUESTION*

- *EXAM IS OF 25 MARKS DURATION 2 HRS FOR EACH STUDENT*