Fr. Conceicao Rodrigues College Of Engineering

Fr. Agnel Ashram, Bandstand, Bandra (W) Mumbai 400 050.

Software Engineering Lab

(CSL501)

Department of Computer Engineering

University of Mumbai

B.E. (Computer, Sem - V)

(AY 2022-23)

SUPERVISOR: Dr. B. S. Daga

GROUP MEMBERS

1. Hudah Ansari (9179)
2. Shreya Palande (9217)
3. Manasvi Patil (9219)

LAB OBJECTIVES

The software engineering program provides project-rich learning experiences to learners for success in a rapidly evolving computing field. The major goal of this course is to teach students how to use software engineering principles to address real-world problems. Students will learn how to analyse, test, and create software development techniques. Software engineering helps demonstrating and assessing software projects used in the real world.

Successful completion of all our experiments allowed us to identify and assess the process models. We are able to plan, schedule and track the progress of projects. This course helped us in designing and testing software projects as well as identify risks and maintain quality in these projects.

INDEX

Course Scheduling System

1. Software Requirement Specification (SRS)

This experiment explains the software requirement for the course scheduling system.

Appendix

1. Introduction
   1.1 Purpose
   1.2 Scope
   1.3 Definitions and abbreviations
   1.4 References
   1.5 Overview

2. General Description
   2.1 Product Perspective
   2.2 Product Functions
   2.3 User Characteristics
   2.4 Constraints
   2.5 Assumptions and Dependencies

3. Specific Requirements

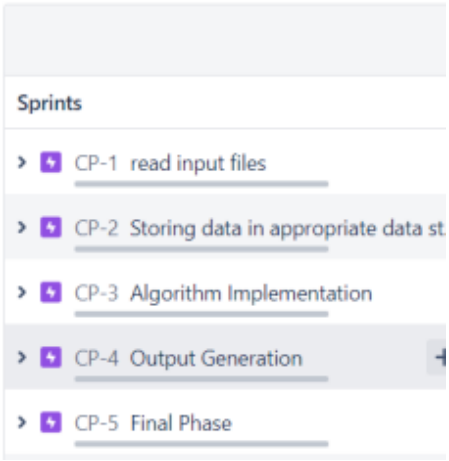
1. Introduction
   - Purpose of this document
     The project aims to ease the registration process of students for various courses and generate course schedules based on availability of classroom and time slots.
   - Scope of this document
     This document aims to provide details about the project so as to clear any doubts or misconceptions that the reader may have regarding the project.
   - Definitions and Acronyms
     None
   - References
     Internet, Instructor
   - Overview
     This document contains the description of the entire project. It also includes the various external inputs that are being utilised for the project.

2. General Description
   - Product Perspective
     The system will be operated within the university environment. It will be utilised to generate a course schedule by taking into account the courses selected by the students
   - Product Functions
     The system will be able to generate course schedules
     The system will ensure the multiple classes are not using the same room at the same time slot
   - User characteristics
     The user is expected to be aware of their desired course and the code associated with it. The user is also expected to know how to provide correct input to the system.
   - General constraints
     The system must run on Windows or Linux Operating System
   - Assumptions and dependencies
     The system is dependent on the input files containing the available time slot and available classrooms

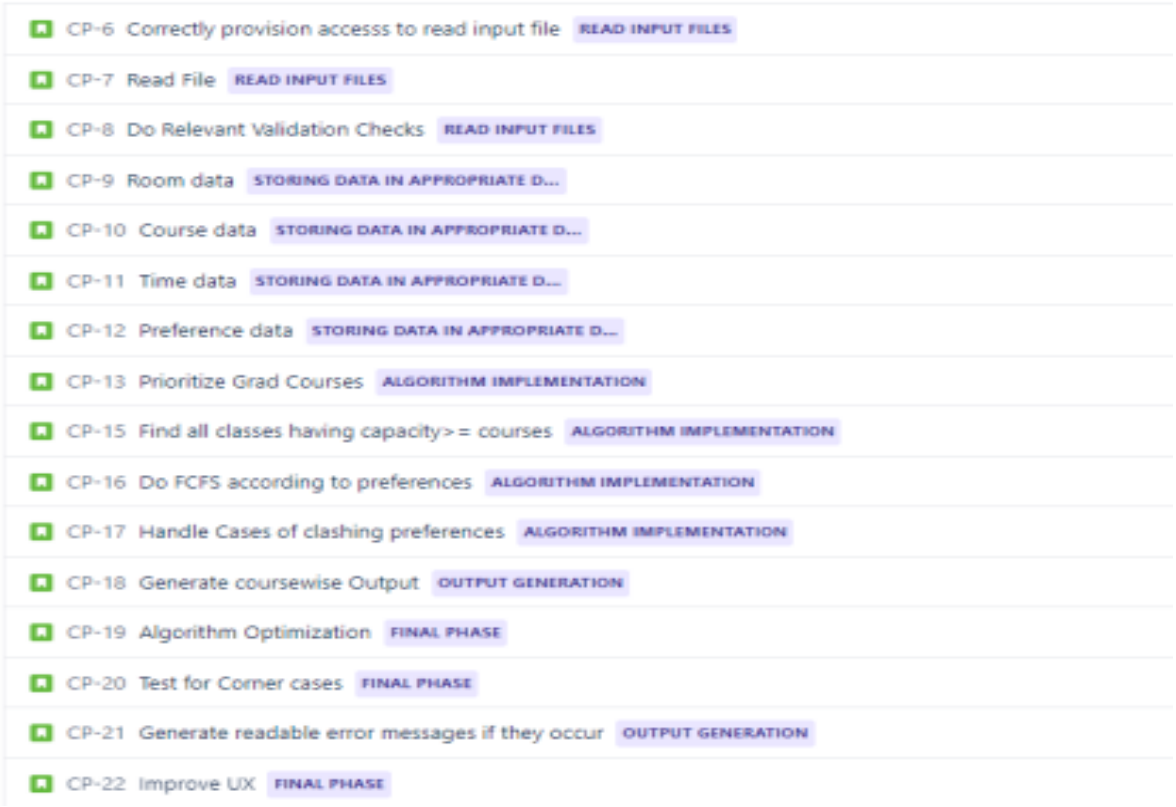
3. Specific Requirements
   - The system requires hardware like PC, mouse, etc to accept input and display output
   - It should be easy to use
   - System should be periodically updated to include all new courses and available classrooms
   - System should allow course instructor to make any desired changes

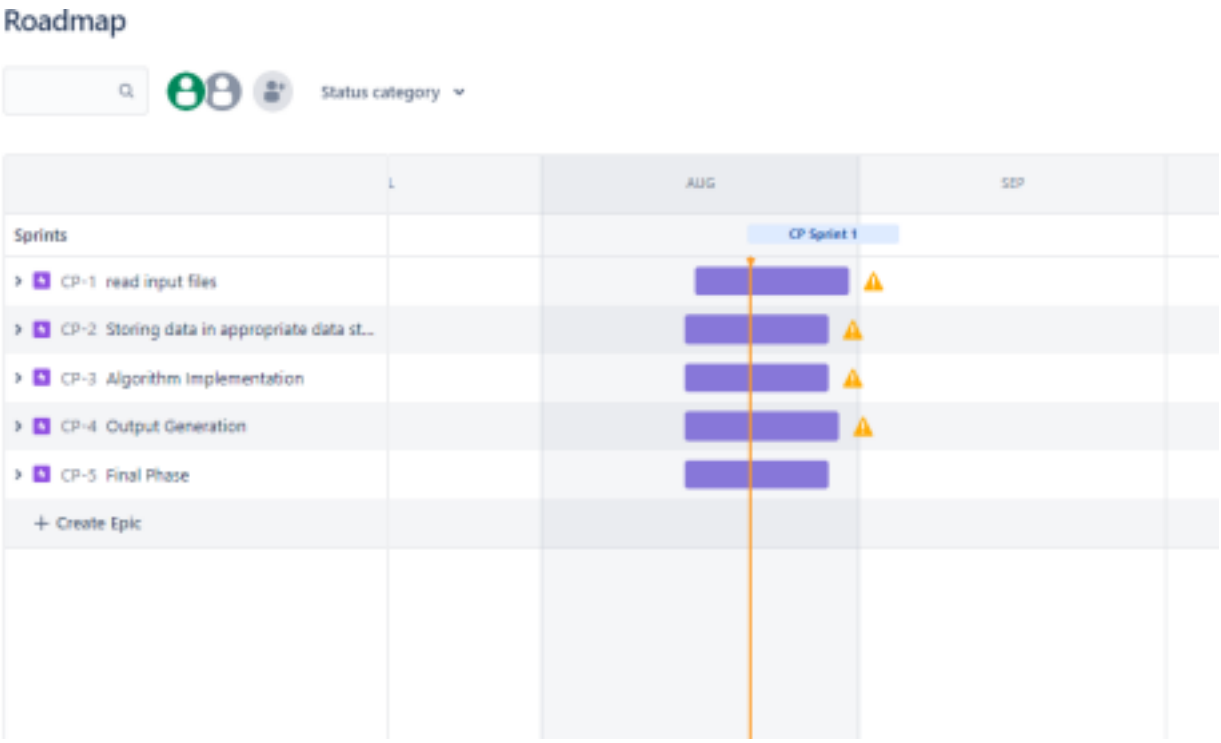
2. Implement Course Scheduling System using SCRUM method on JIRA Tool.

This experiment explains the implementation of the SCRUM model using the JIRA tool for the course scheduling system.

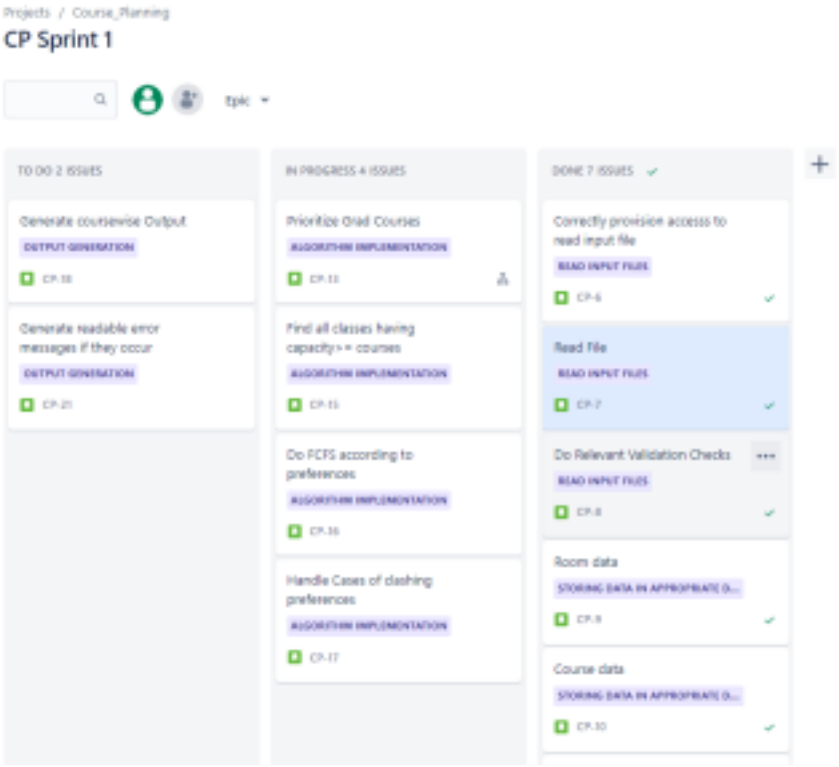


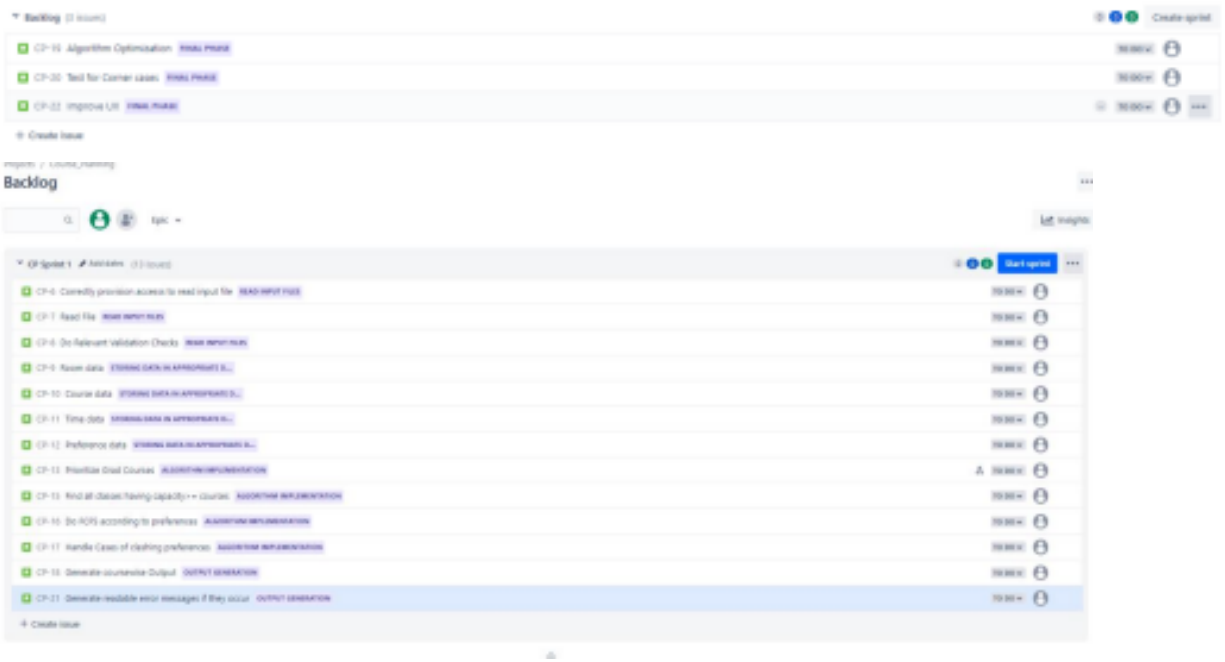
The above image displays the sprints implemented for Course Scheduling System





The below images display the details of each sprint





An SRS is basically an organisation's understanding of a customer or potential client's system requirements and dependencies.

In SRS following aspects are addressed:
a) **Functionality**. What is the software supposed to do?
b) **External interfaces.** How does the software interact with people, the system's hardware, other hardware, and other software?

c) **Performance.** What is the speed, availability, response time, recovery time of various software functions, etc.?

d) **Attributes.** What are the portability, correctness, maintainability, security, etc. considerations?

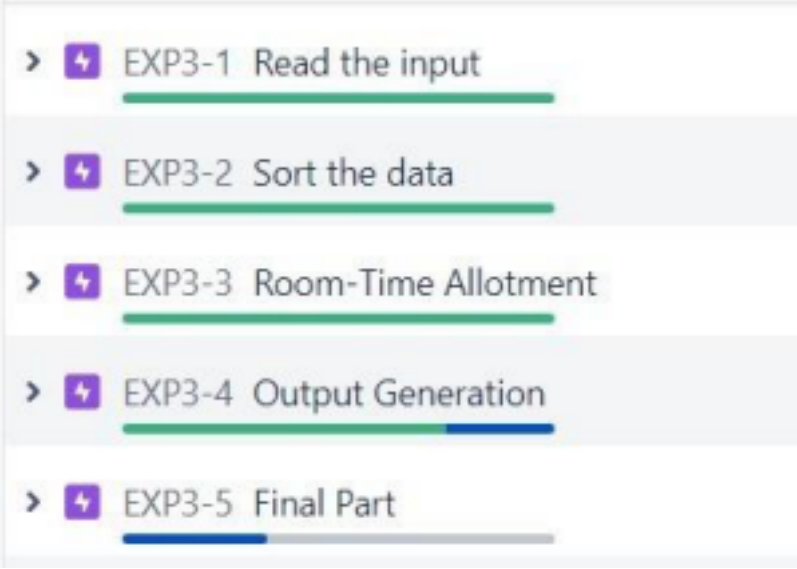
e) **Design constraints** imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.

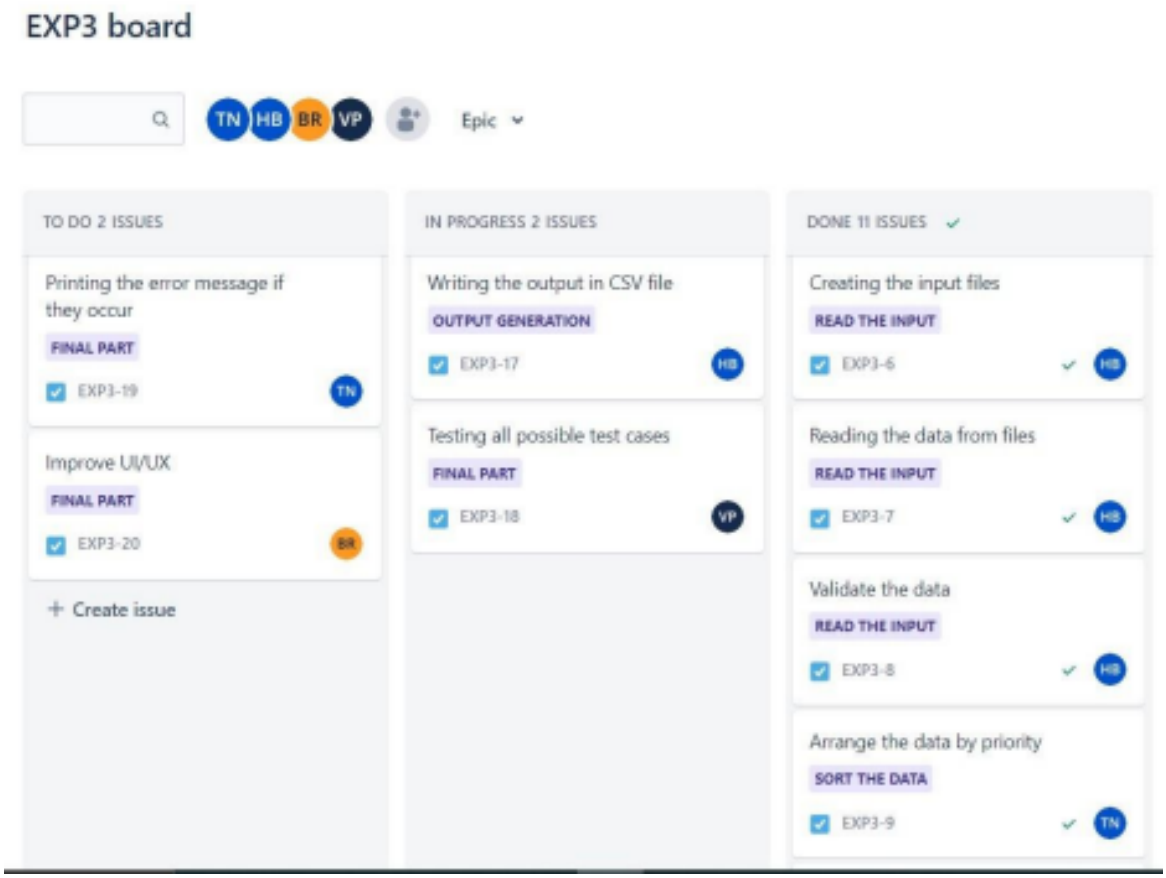
Conclusion:

Following learning was attained during this experiment:
1. Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. The Agile team delivers work in small, but consumable, increments.
2. With Agile methodology ,software products can be developed very fast. 3. This method is more focused on productivity rather than on bureaucracy, which makes agile methods standard choice in industry and natural choice for developers.

3. Implement Course Scheduling System using KANBAN method on JIRA Tool.

This experiment explains the implementation of the KANBAN model using the JIRA tool for the course scheduling system. In this experiment, we have added the images displaying the Kanban Boards that we designed for the Course Scheduling System.





An SRS is basically an organisation's understanding of a customer or potential client's system requirements and dependencies.
In SRS following aspects are addressed:
a) **Functionality**. What is the software supposed to do?
b) **External interfaces.** How does the software interact with people, the system's hardware, other hardware, and other software?
c) **Performance.** What is the speed, availability, response time, recovery time of various software functions, etc.?
d) **Attributes.** What are the portability, correctness, maintainability, security, etc. considerations?
e) **Design constraints** imposed on an implementation. Are there any required standards in effect, implementation language, policies for database integrity, resource limits, operating environment(s) etc.

Conclusion:
Following learning was attained during this experiment:
1. Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. The Agile team delivers work in small, but consumable, increments.
2. With Agile methodology ,software products can be developed very fast. 3. This method is more focused on productivity rather than on bureaucracy, which makes agile methods standard choice in industry and natural choice for developers.

4. To calculate function point for Course Scheduling System.

In this experiment, Function Point Analysis calculation for CSS is explained.

External Input= 2 x 4 = 8
External Output= 1 x 5 = 5
External Inquires= 3 x 4 = 12
Internal Logic files=6 x 10 = 60
External Inference files= 8 x 7 = 56
Unadjusted FP total = 141
Value adjustment Factor determination
1. 3
2. 2
3. 1
4. 3
5. 2
6. 3
7. 2
8. 1
9. 4
10. 2
11. 4
12. 2
13. 3
14. 3
Value adjustment factor = 35

Total = 141 x (0.65 + 0.01 * 35) = 141 x (1)
= 141

Conclusion
Thus calculated function point = 141

SOURCE CODE:

```
import java.util.*;
import java.io.*;
public class fpa {
 public static void main(String[] args) {  int frates[][] = {
 { 0, 50, 0 },
 { 0, 40, 0 },
 { 0, 35, 0 },
 { 0, 6, 0 },
 { 0, 4, 0 }
```

```
};
int fac_rate = 3;
calfp(frates, fac_rate);
}
static void calfp(int frates[][], int fac_rate) {
// Function Units
String funUnits[] = {
"External Inputs",
"External Outputs",
"External Inquiries",
"Internal Logical Files",
"External Interface Files"
};
// Weight Rates
String wtRates[] = { "Low", "Average", "High" };
// Weight Factors
int wtFactors[][] = {
{ 3, 4, 6 },
{ 4, 5, 7 },
{ 3, 4, 6 },
{ 7, 10, 15 },
{ 5, 7, 10 },
};
int UFP = 0;
// Calculating UFP (Unadjusted Function Point)
for (int i = 0; i < 5; i++) {
for (int j = 0; j < 3; j++) {
int freq = frates[i][j];
UFP += freq * wtFactors[i][j];
}
}
// 14 factors
String aspects[] = {
"How many data communication facilities are there?", "How are distributed data and processing functions handled?", "Was response time and throughput required by the user?", "How heavily used is the current hardware platform?", "How frequently are transactions executed?",
"What percentage of information is entered online?", "Was the application designed for end-user efficiency?", "How many internal logical files are updated by online transaction?", "Does the application have extensive logical or maths processing?", "Was the application developed to meet one or many user needs?", "How difficult is conversion and installation?",
"How effective are startup, backup and recovery?", "Was the application designed for multiple organisations?", "Was the application designed to facilitate change?" };
/*
* Rate Scale of Factors
* Rate the following aspects on a scale of 0-5 :-
* 0 - No influence
```

```
* 1 - Incidental
* 2 - Moderate
* 3 - Average
* 4 - Significant
* 5 - Essential
*/
int sumF = 0;
// Taking Input of factors rate
for (int i = 0; i < 14; i++) {
int rate = fac_rate;
sumF += rate;
}
// Calculate CFP
double CAF = 0.65 + 0.01 * sumF;
// Calculate Function Point (FP)
double FP = UFP * CAF;
System.out.println("Function point analysis - ");
System.out.println("Unadjusted Function Points (UFP) : " + UFP);  System.out.println("Complexity
Adjustment Factor (CAF) : " + CAF);  System.out.println("Function Points (FP) : " + FP);
}
}
```

5. To estimate project cost using COCOMO Model for Course Scheduling

This experiment we will observe various modes in COCOMO model.

OBSERVATIONS:
FPA=75
LOC=30*75=2250
KLOC=2.25

1. **Basic COCOMO**

Model	**Organic**	**Semi-detached**	**Embedded**
Effort(E)	5.62 person-months	7.44 person-months	9.53 person-months

2. **Intermediate COCOMO**
   EAF=1.36

Model	**Organic**	**Semi-detached**	**Embedded**
Effort (E)	10.20 person-months	10.12 person-months	10.08 person-months
Development Time (T)	5.38 months	5.05 months	4.75 months

3. **Advanced COCOMO**

Cost Drivers	**RPD**	**DD**	**CUT**	**IT**
PRODUCT				
RELY	0.75	0.85	0.9	0.85
DATA	0.85	1	1	0.85
CPLX	0.85	0.85	0.85	1
PLATFORM				
TIME	1	1	1	1
STOR	1	1	1	1
VIRT	1	1	1	1

TURN	0.85	0.85	0.85	1
PERSONNEL				
ACAP	1	0.9	0.9	1
AEXP	1	1	1	1.2
PCAD	0.75	1	1	1
VEXP	1	1	1	1
LEXP	1	1	1	1
PROJECT				
MODP	0.75	0.85	0.85	1.2
TOOL	0.85	0.85	0.85	1.2
SCED	1	1	1	1

RPD=0.22
DD=0.4
CUT=0.42
IT=1.25

EAF=2.29

Model	**Organic**	**Semi-detached**	**Embedded**
Effort (E)	17.17 person-months	17.04 person-months	16.97 person-months
Development Time (T)	5.38 months	5.05 months	4.74 months

Conclusion

Thus, we observe that in the Basic COCOMO model there is an increase in effort from organic model to semi-detached model and even more increase from semi-detached model to embedded model. Development Time cannot be calculated for Basic COCOMO.

In Intermediate COCOMO, there is a decrease in both effort and development time from organic model to semi-detached model and even more decrease from semi-detached model to embedded model. The same is observed in Advanced COCOMO models.

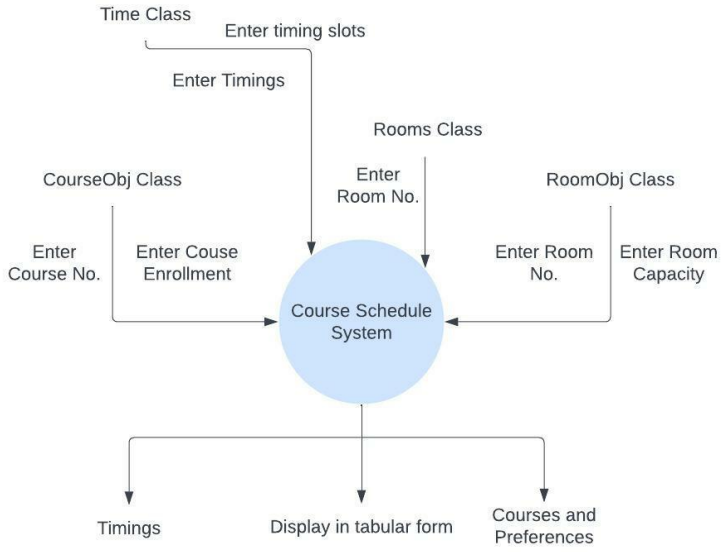
The effort calculated is lesser in Basic model, a little more in Intermediate Model and the most in Advanced Model.

The Development Time is nearly the same for Intermediate and Advanced Model.

6. Develop diagrams for data flow analysis on Course Scheduling System

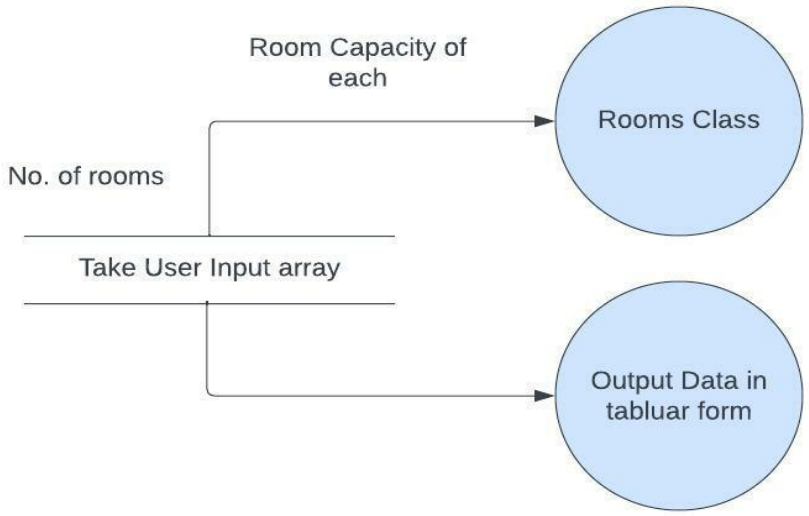
In this experiment, Data Flow Diagrams (DFD) for the course scheduling system are created.

0-Level DFD for Course Schedule System:



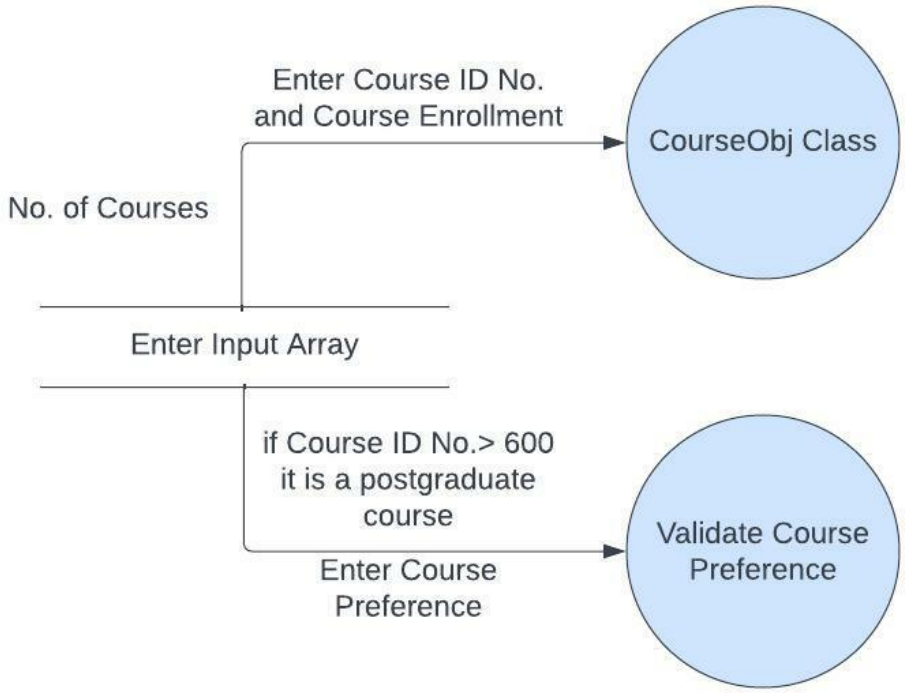
2 Level DFD:

1. Take User Input For Room Class
   The level 2 DFD for this Process is given below



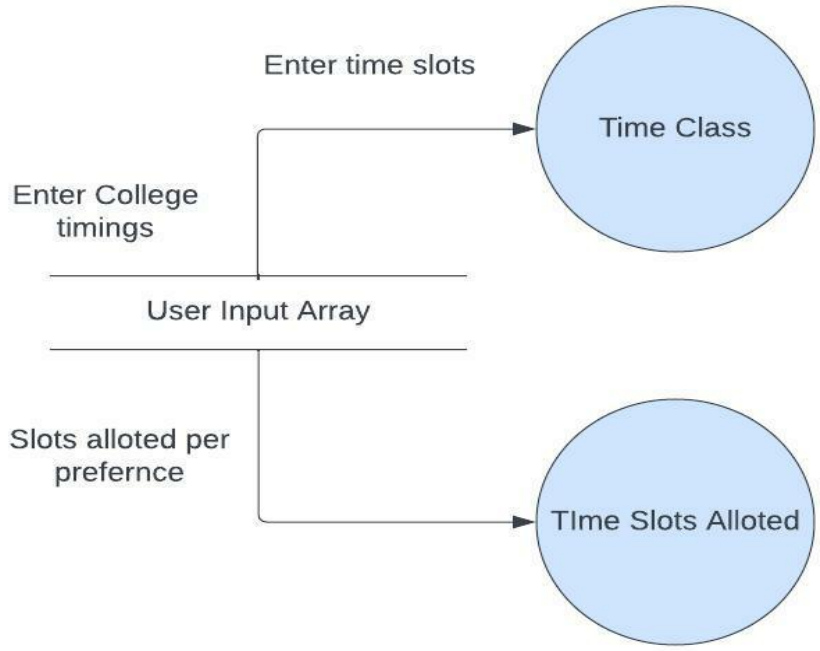
2. Take User Input From CourseObj Class

The level 2 DFD for this process is given below



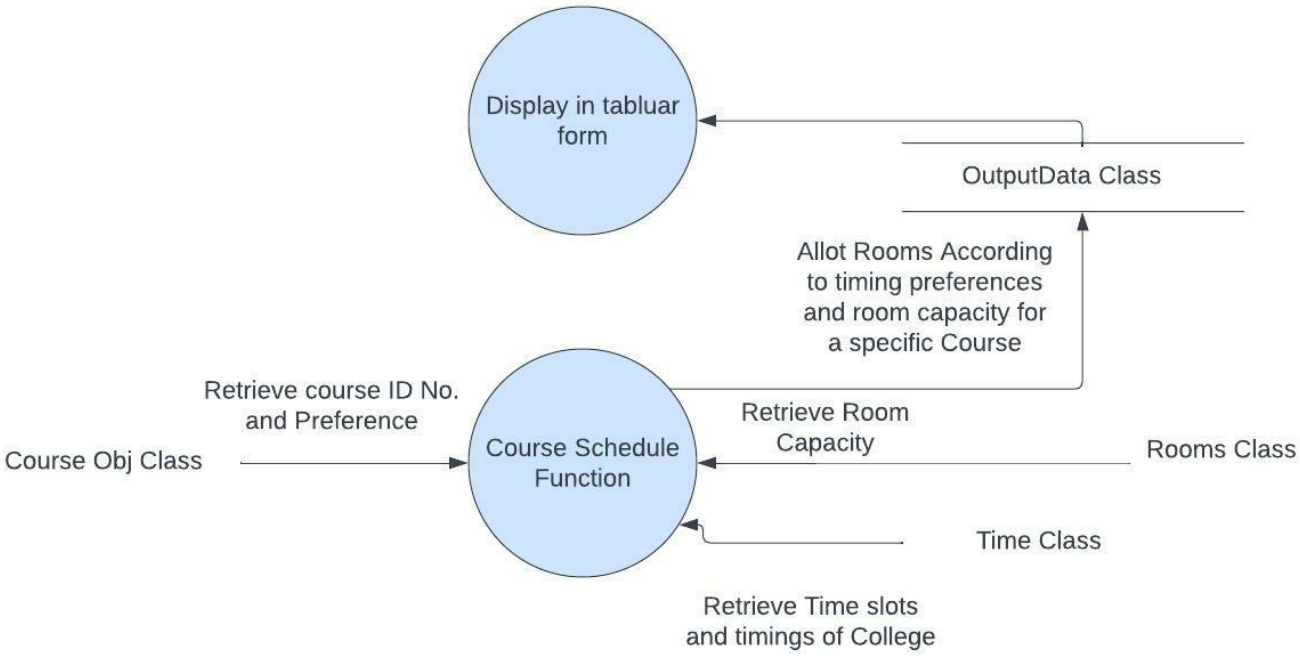
3. Take Input For Time Class

The Level 2 DFD for this process is given below



4. Course Scheduler Function

The level 2 DFD for this Process is given below

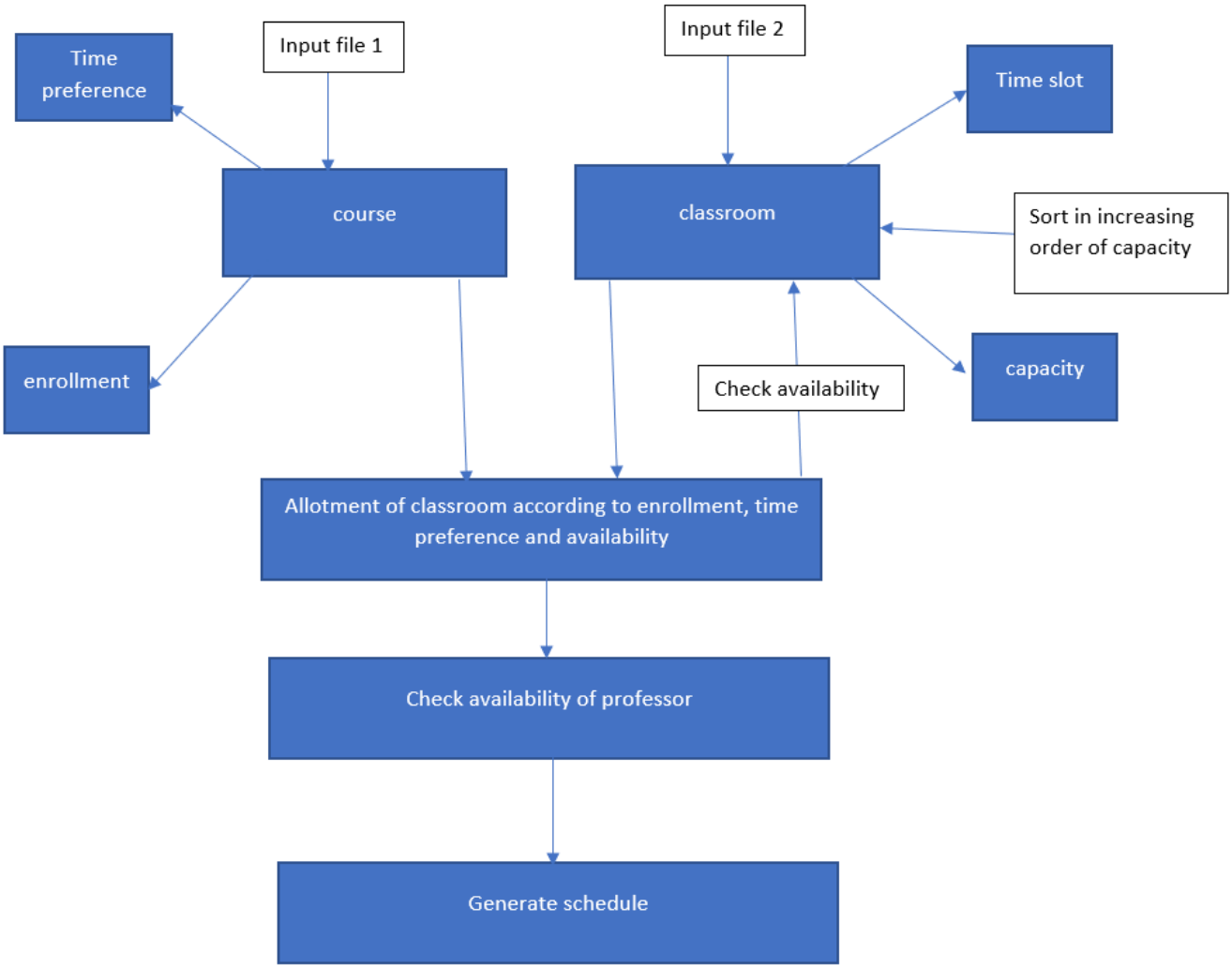


Conclusion :

1. It helps developers understand the process flow of the project easier than reading the SRS Document. Same for non-technical Staff.
2. Helps Divide the various parts of the project into easier visual representations.
3. From 0 level DFD we get the basic idea of the project.
4. From 2 level DFD we understand the various parts of the project.

7. Implementation of data flow design pattern

In this experiment, data flow patterns according to course scheduling system is implemented.



Conclusion:

The software needs the architectural design to represent the design of software. Each style will describe a system category that consists of a set of components(eg: a database, computational modules) that will perform a function required by the system.

The use of architectural styles is to establish a structure for all the components of the system.

8. Do design using OO approach and hence highlight Cohesion and Coupling in the design.

This experiment helps in understanding Cohesion and Coupling using an Object Oriented approach for course scheduling system.

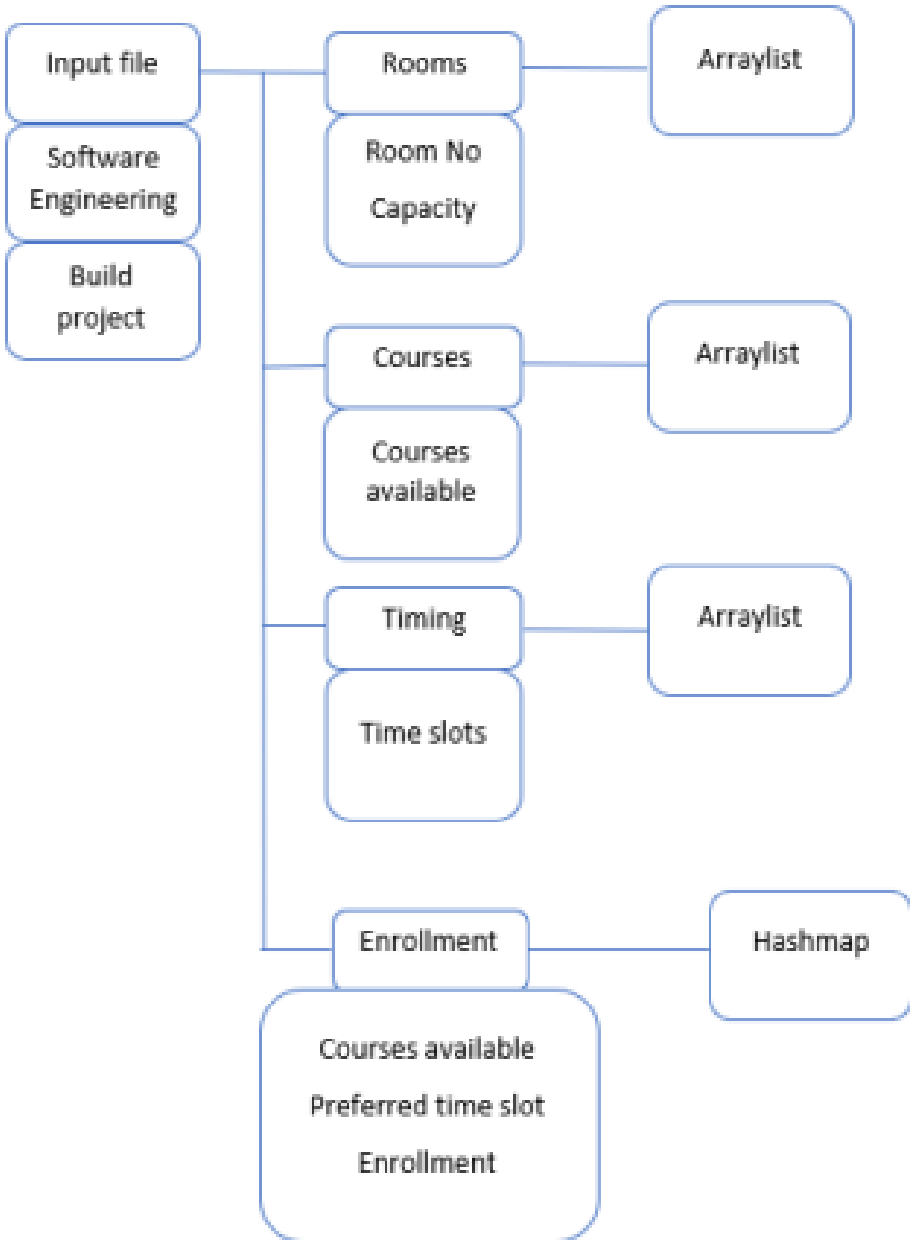
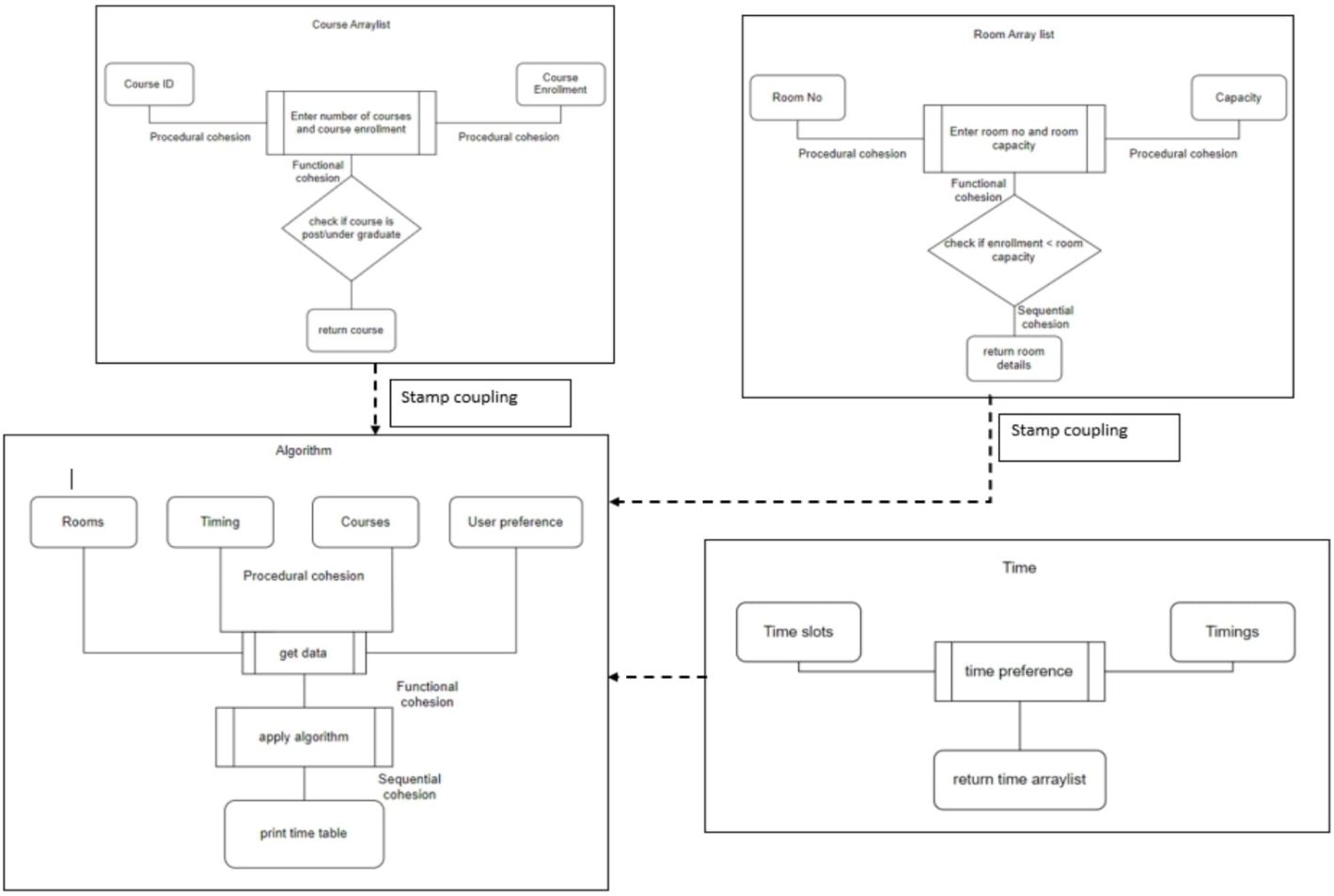


Diagram representing cohesion and coupling



Conclusion

- Helps Convert Course Scheduling System to a simple diagrammatic format which is easy to understand OOD.
- Divides classes into separate modules showing the interrelation between them (Cohesion / Coupling).

● Helped understand the different types of Cohesion and Coupling. ● Helped implement Cohesion and Coupling links in the OOD for Course Schedule System.

9-A. To design test cases for performing black box testing for Course Scheduling

Black box testing is done in this experiment.

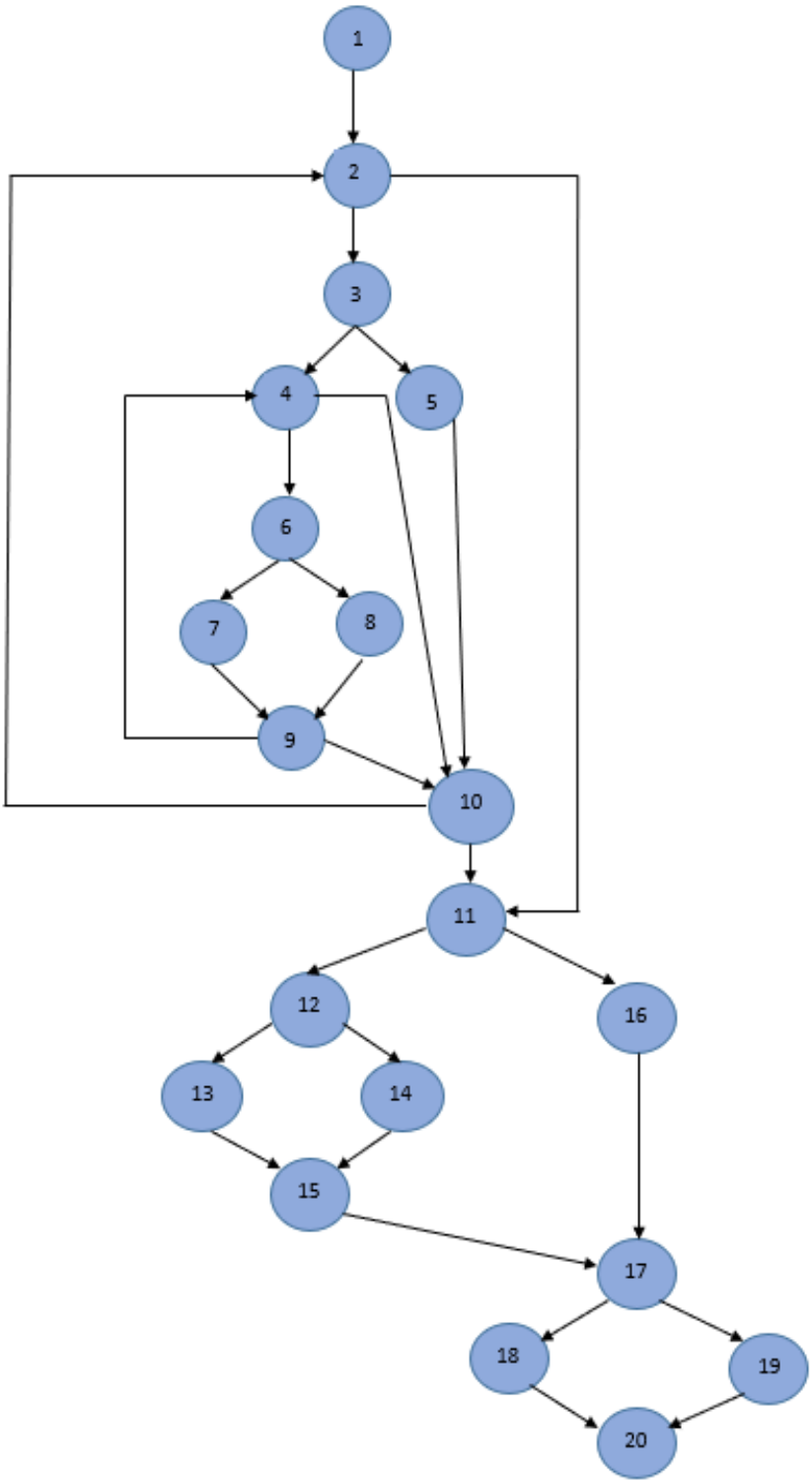
Test Description	Expected Result	Actual Result
Obtaining list of available classrooms	Display available classrooms in increasing order of capacity	Displayed available classrooms in increasing order of capacity
Obtaining list of available time slots	Display list of available time slots	Displayed list of available time slots
Checking availability of professors	Display list of availability of professors with time slot	Did not display list of professors
Generating feasible course schedule	Display course schedule	Displayed course schedule
Allowing HOD to make changes	HOD should have permission to access software	HOD has access to all software used

Conclusion

The black box is a powerful technique to check the application under test from the user's perspective. Black box testing is used to test the system against external factors responsible for software failures.

9-B. To design test cases for performing white box testing for Course Scheduling.

White box testing is done in this experiment.



Set of independent paths:

- path 1: 1-2-3-4-6-8-9-10-11-12-13-15-17-18-20
- path 2: 1-2-3-4-6-8-9-10-11-16-17-18-20
- path 3: 1-2-3-5-10-11-12-13-15-17-18-20
- path 4: 1-2-11-12-13-15-17-18-20
- path 5: 1-2-3-4-10-11-12-13-15-17-18-20
- path 6: 1-2-3-4-10-11-16-17-18-20
- path 7: 1-2-3-4-10-11-12-14-15-17-18-20

Cyclomatic Complexity=E - N + 2
=27 - 20 + 2
=9

Test Case	**Expected Result**	**Actual Result**
For loop to sort classes in ascending order of capacity	data sorted in ascending order	data sorted in ascending order
while loop to allot classes	classes get allotted on basis of availability	classes got allotted on basis of availability
if condition to check availability of classes	assign class if available	available class was assigned

Conclusion

White Box Testing is a testing technique in which software's internal structure, design, and coding are tested to verify input-output flow and improve design, usability, and security.

White box testing can be quite complex. The complexity involved has a lot to do with the application being tested. A small application that performs a single simple operation could be white box tested in a few minutes, while larger programming applications take days, weeks, and even longer to fully test.

White box testing in software testing should be done on a software application as it is being developed after it is written and again after each modification.

10. Version controlling & Risk Analysis of the project.

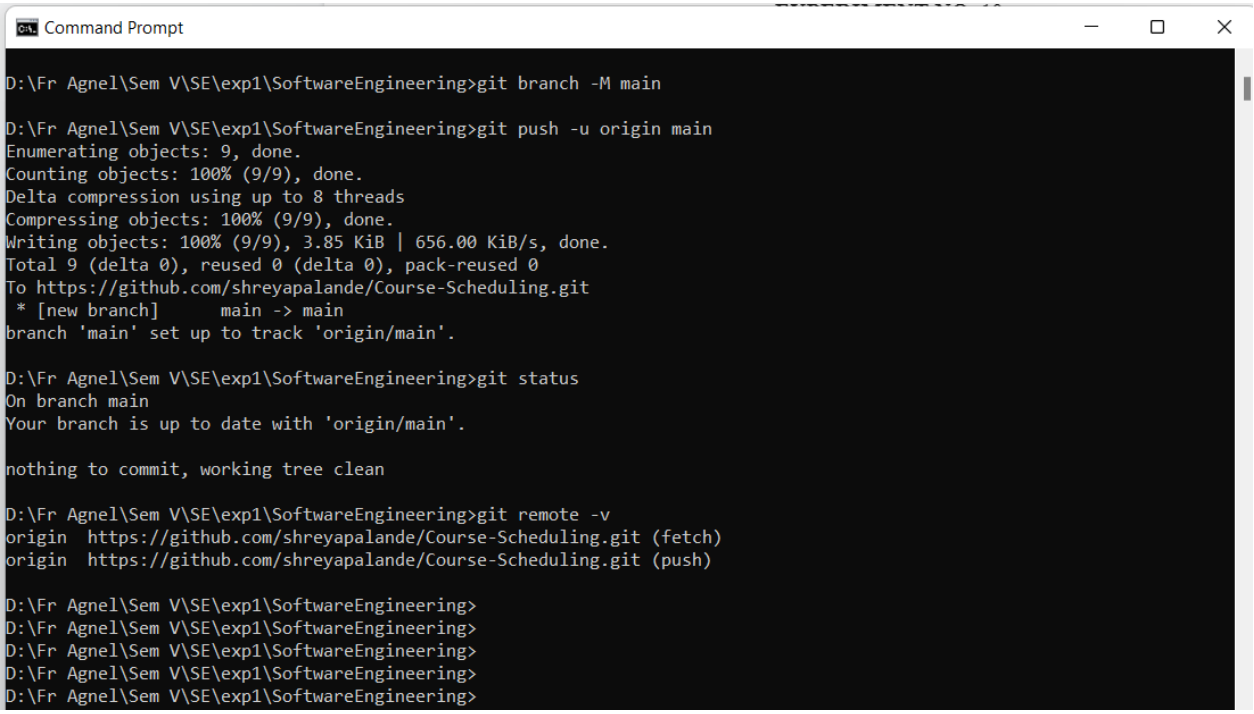
In this experiment, Version controlling and Risk analysis related to our course scheduling system project is done.

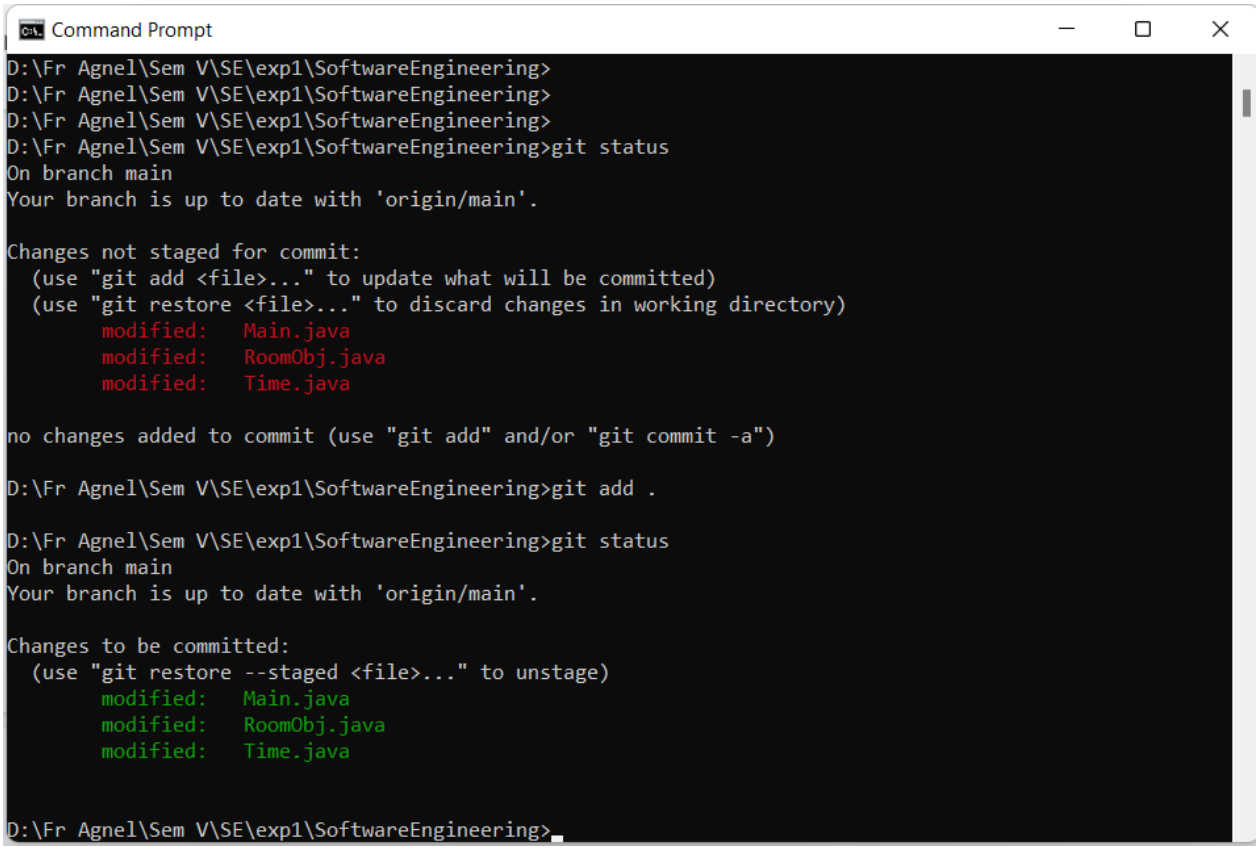
Aim

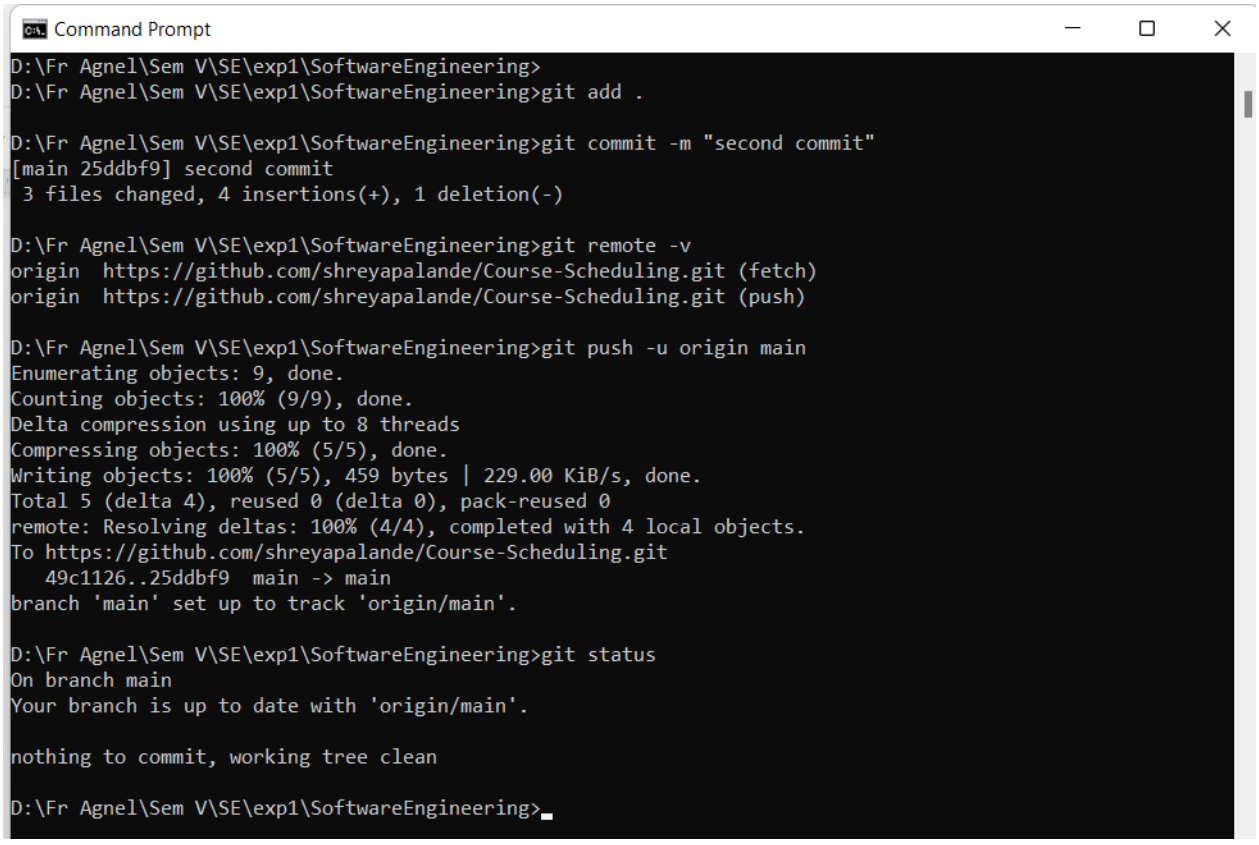
1. Version controlling of the project using Git-Hub
2. Develop a risk table for Course Schedule System

1. Version Controlling

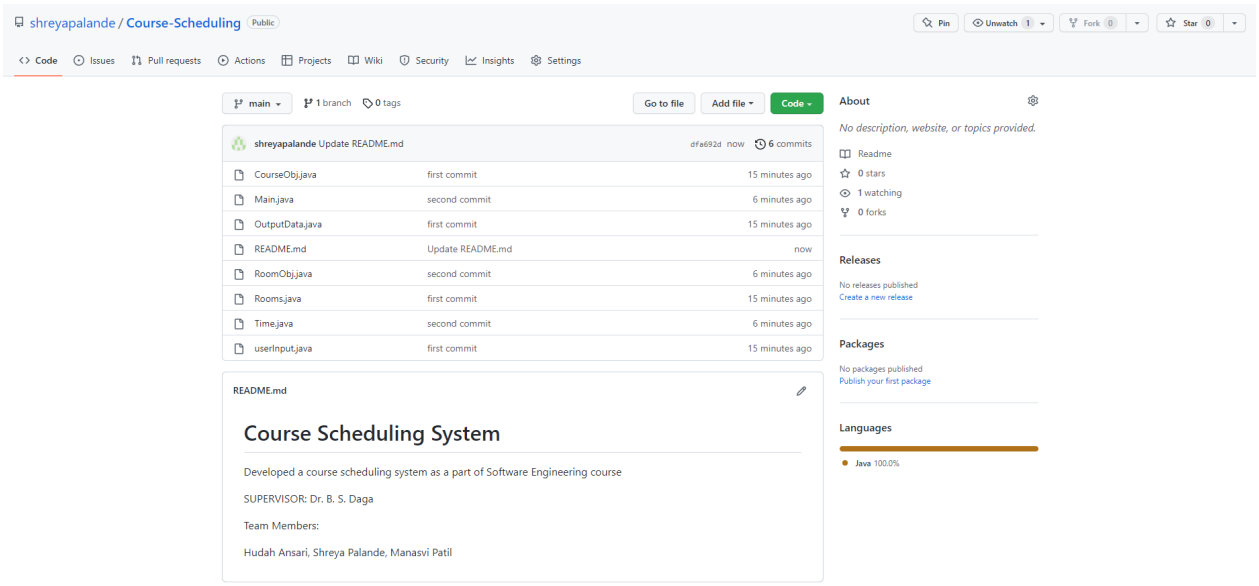
- Git



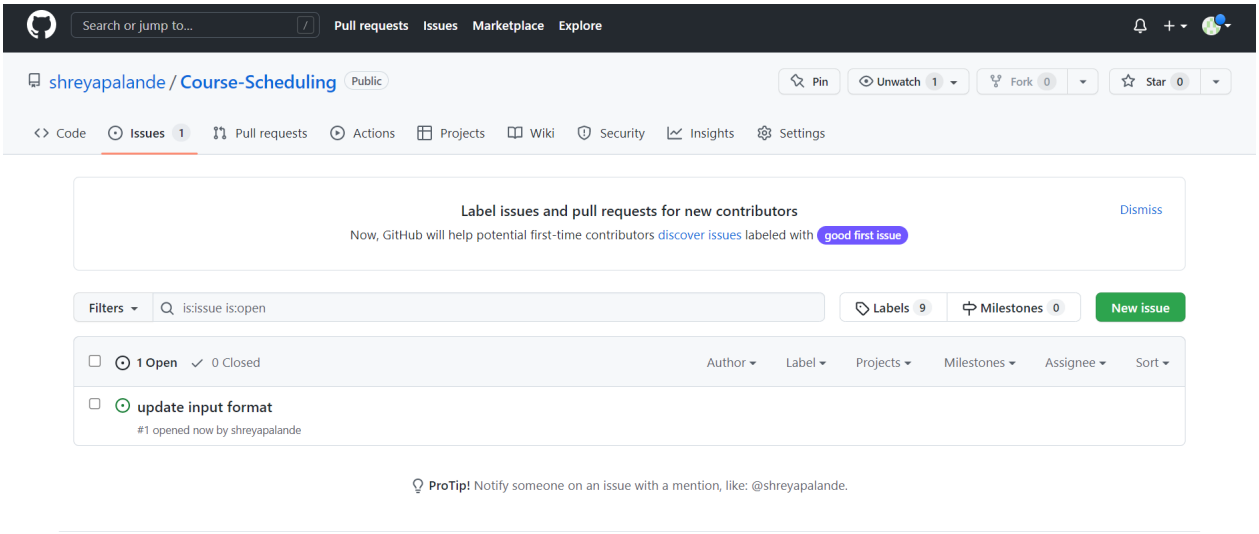

Command Prompt
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git status
On branch main
Your branch is up to date with 'origin/main'.
Changes not staged for commit:
(use "git add <file>..." to update what will be committed)
(use "git restore <file>..." to discard changes in working directory)
modified: Main.java
modified: RoomObj.java
modified: Time.java
no changes added to commit (use "git add" and/or "git commit -a")
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git add .
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git status
On branch main
Your branch is up to date with 'origin/main'.
Changes to be committed:
(use "git restore --staged <file>..." to unstage)
modified: Main.java
modified: RoomObj.java
modified: Time.java
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>

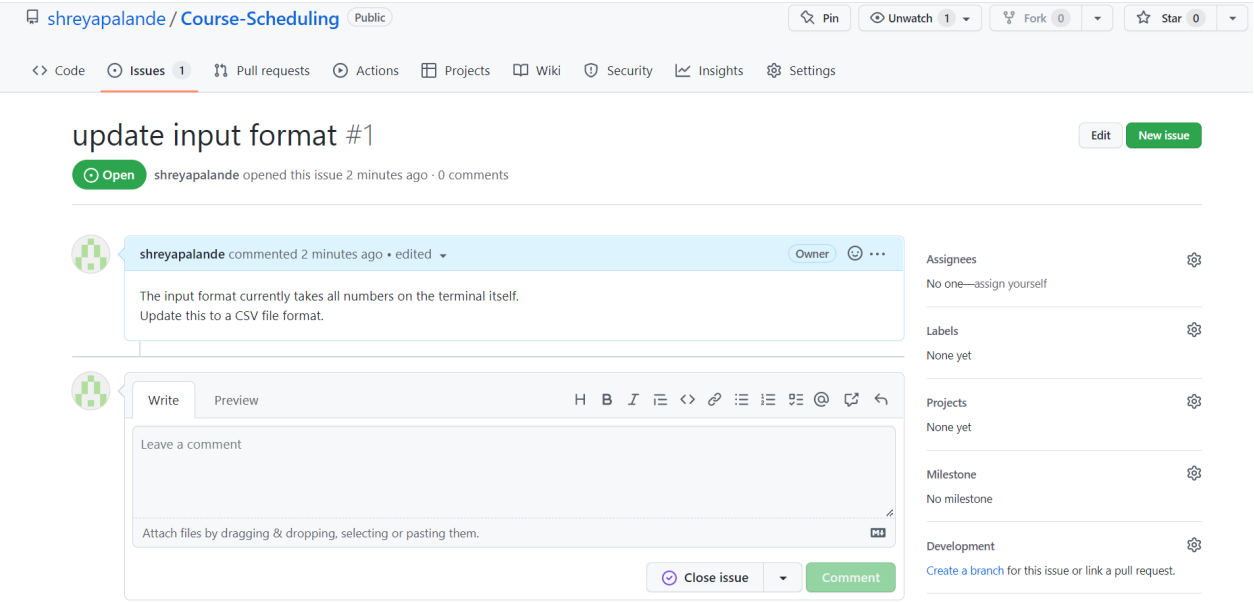

Command Prompt
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git add .
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git commit -m "second commit"
[main 25ddbf9] second commit
3 files changed, 4 insertions(+), 1 deletion(-)
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git remote -v
origin https://github.com/shreyapalande/Course-Scheduling.git (fetch)
origin https://github.com/shreyapalande/Course-Scheduling.git (push)
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git push -u origin main
Enumerating objects: 9, done.
Counting objects: 100% (9/9), done.
Delta compression using up to 8 threads
Compressing objects: 100% (5/5), done.
Writing objects: 100% (5/5), 459 bytes | 229.00 KiB/s, done.
Total 5 (delta 4), reused 0 (delta 0), pack-reused 0
remote: Resolving deltas: 100% (4/4), completed with 4 local objects.
To https://github.com/shreyapalande/Course-Scheduling.git
49c1126..25ddbf9 main -> main
branch 'main' set up to track 'origin/main'.
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>git status
On branch main
Your branch is up to date with 'origin/main'.
nothing to commit, working tree clean
D:\Fr Agnel\Sem V\SE\exp1\SoftwareEngineering>

- Github



Creating an Issue - Issues let you track your work on GitHub, where development happens. When you mention an issue in another issue or pull request, the issue's timeline reflects the cross-reference so that you can keep track of related work. To indicate that work is in progress, you can link an issue to a pull request.





2. Risk Analysis

Risks	Probability	Impact	Mitigation Plan
Project design and deliverable definition is incomplete.	50%	1	Define the scope in detail via design workshops with input from subject matter experts.
Project schedule is not clearly defined	60%	2	Hold scheduling workshops with the project team so they understand the plan and likelihood of missed tasks is reduced.
Unresolved project conflicts not escalated in a timely manner	30%	2	Hold regular project team meetings and look out for conflicts.
Incorrect data for availability of classrooms and capacity	40%	3	Check the data frequently and make any required changes before generating course schedules
There is conflict in availability of professors	60%	3	Check availability of professors well in advance and keep contingency plan ready
Changes to be made in schedule after generating all schedules	50%	2	Make sure all students are aware that once course schedule is generated, changes will only be made in case of emergency

Conclusion :

Risk Analysis in project management is a sequence of processes to identify the factors that may affect a project's success. These processes include risk identification, analysis of risks, risk management and control, etc. Proper risk analysis helps to control possible future events that may harm the overall project. It is more of a pro-active than a reactive process.

These steps can be used to manage risk in an organisation

- Risk identification
- Risk Qualification
- Risk Response
- Risk Monitoring and Controlling