



EXERCISE SUBMISSION 2

**ISEN1000 Introduction to Software
Engineering Trimester 3, 2023**

Dhrubo Jouti Das Troyee

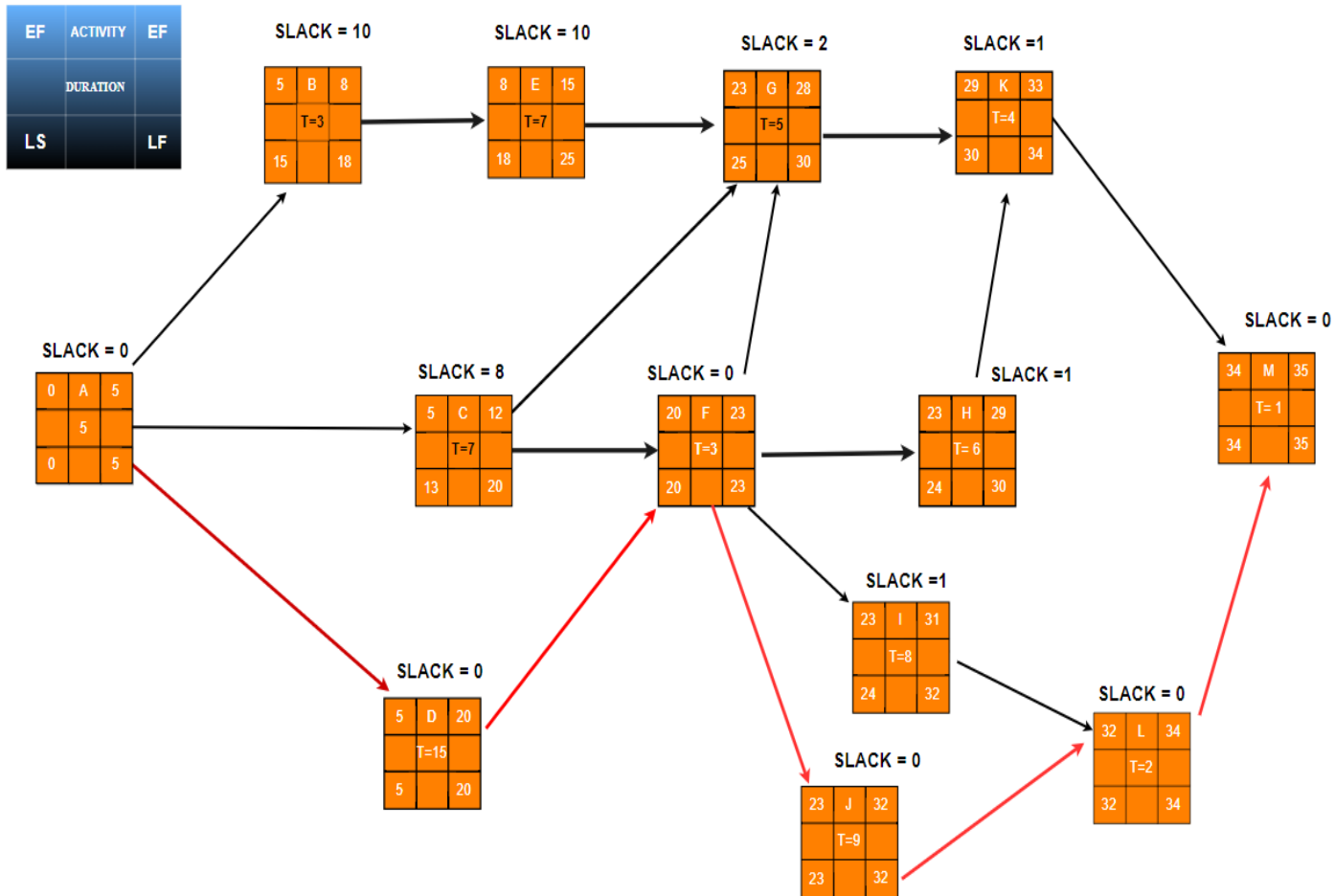
Question 1: Planning, and Agile Software Project Management

❑ Q.A TASK TABLE completed with missing dependencies:

Serial No	Task	Estimated (weeks)	Duration	Dependencies
1.	A	5		----
2.	B	3		A
3.	C	7		A
4.	D	15		A
5.	E	7		B
6.	F	3		C ,D
7.	G	5		E, C,F
8.	H	6		F
9.	I	8		F
10	J	9		F
11.	K	4		G, H
12.	L	2		I, J
13.	M	1		K,L

❑ Q.(B) Table for estimated Early Start (ES), Late Start (LS), Early Finish (EF) and Late Finish (LF) for each task in the AON

Task	TIME(t) WEEK	EARLY START (ES)	EARL FINISH(EF) $EF = ES + t$	LATESTART(LS) $LS = LF - t$	LATE FINISH(LF)
A	5	0	5	0	5
B	3	5	8	15	18
C	7	5	12	13	20
D	15	5	20	5	20
E	7	8	15	18	25
F	3	20	23	20	23
G	5	23	28	25	30
H	6	23	29	24	30
I	8	23	31	24	32
J	9	23	32	23	32
K	4	29	33	30	34
L	2	32	34	32	34
M	1	34	35	34	35



Our Critical Path is

A → D → F → J → L → M

According to the graph above, the project's completion time is 35 WEEKS.

❑ Q.(C) CRITICAL PATH METHODS IN PROJECT MANAGEMENT:

Task	SLACK TIME =LS – ES = LF-EF	CRITICAL PATH
A	0	YES
B	10	NO
C	8	NO
D	0	YES
E	10	NO
F	0	YES
G	2	No
H	1	No
I	1	No
J	0	YES
K	1	NO
L	0	YES
M	0	YES

Here, activities with zero slack make up the critical path and are highlighted in yellow. In project management, the critical path consists of the tasks, having zero slacks, that must be completed to finish the project within the shortest possible time frame. Any delay in any of the tasks on the critical path will cause a delay in the overall project completion. Besides, tasks that are not on the critical path might start and conclude at any time, therefore delays in these tasks may not necessarily affect the project's final finishing schedule.

- ❑ Q.(D)Calculation of expected time: Program Evaluation and Review Technique (PERT):

Task	Estimated Duration (WEEK)			Expected = $\frac{\text{Optimistic} + (4 \times \text{Most Likely}) + \text{Pessimistic}}{6}$ Time
	Optimistic Time Estimate	Most likely Time Estimates	Pessimistic Time Estimate	
A	3	6	9	6
B	5	8	16	8.333333
C	2	5	8	5
D	4	5	9	5.5
E	3	7	11	7
F	3	9	15	9
G	8	12	16	12
H	5	8	10	7.83333333
I	9	10	11	10
J	2	5	10	5.33333333
K	7	14	17	13.333333
L	7	10	15	10.33333333
M	6	9	12	9

Expected project duration

$$=6 + 5.5 + 9 + 5.33333333 + 10.333333 + 9$$

=45.17 Weeks (it's around 10 months)

The expected project duration is **45.17 Weeks (it's around 10 months)**

The project completion time is **35weeks (about 8 months).**

- ❑ **Q.(E)** You project is in at week 9 and you are using Scrum based project management, there is a sprint in 2 weeks, where client wants to add three activities (say N, O, P, in WBS) as backlog, what necessary action would you consider in this case, and describe the responsibilities of each team to deal with the situation

If a client wishes to add new activities (N, O, P) to the backlog during the ninth week of a sprint that is scheduled for two weeks, the following necessary actions can be taken in a Scrum-based project management approach:

1. Development (preparing): To assess and focus on the freshly introduced exercises (M, N, and O), the Scrum group should call an excess modification meeting. During this session, the team works together to rank the tasks based on their importance and value, estimate the amount of work needed for each task, and discuss and clarify the criteria.

2. The Product Owner's Responsibilities:

The Product Owner is very important in this situation. Such as:

- Of course, having a conversation with the client to understand the details and requirements of the newly introduced activities.
- Analyzing the impact of adding these activities into the workflow and overall project goals.
- Making sure the product backlog is updated with the additional tasks.
- Speaking with the customer about how to control their expectations and clarify any possible effects on the project's duration or deliveries

3. Duty of the Scrum Expert:

The Scrum Master oversees the following duties:

- Helping with the adjustment meeting to ensure effective collaboration and mutual understanding among coworkers.
- Identifying and addressing any risks or conflicts that may arise from the addition of new activities.
- Making sure that the team's capacity and velocity are considered when organizing the upcoming sprint.
- Assist the item owner in managing and focusing on the extra stock.
- Assisting the team in calculating the amount of work required for the new tasks.
- Leads the sprint retrospective meeting, establishing a secure environment for the team to reflect and pinpoint opportunities for improvement during the sprint planning.

4. The Function of the Development Team:

The Development Team oversees the following:

- Taking an active part in the backlog modification meeting to discover more about the new tasks and the requirements they call for.
- If new activities are scheduled for the coming sprint, they must modify their sprint planning according to their capabilities.
- Discussing any possible issues or connections to the new projects.
- Estimating the amount of work that each task will require in collaboration with the product owner.

By following these steps and maintaining effective collaboration between the Product Owner, Scrum Master, and Development Team, the Scrum-based project can manage the addition of new activities to the backlog and make accurate choices about the upcoming sprint while taking the project's timeline and goals into consideration.

- ❑ Q.(F) You has completed activity “H” by week 20 and all dependencies of “H”. However, all other activities are not complete. Estimate actual performance on the Burn-up Chart and describe the overall project progress in text. (4 marks)

To determine the actual performance on the Burn-up Chart and explain the overall development of the project, we need to take into consideration the final stage of each activity. We may assess our progress on the project by looking at the entire amount of work accomplished, since activity "H" and its dependencies will be completed by week 20.

- ❑ Q.(G)Show Kanban Board when both (5 marks): 1. “G” and “H” are under review and “I” and “J” are done/completed with all their dependencies. 2. When activity “K” and “L” are under review.

- This Kanban board is divided into four sections: To-do, Coding/Testing, Review, and Done. There are four columns in To-Do, three in Coding, two in Review, and three in Done.
- Therefore, to prevent any issue, one must ensure that each column has three sections. So, we can work on a few tasks at a time as the WIP (Work in progress) limitation is 3.
- If anyone has more than three, they can switch between their columns and, in that case, merge them using the same techniques.
- For example, there are 3 tasks in the “Coding/Testing” column. We cannot just move one task to “Review” as it exceeds the WIP limit of 3 tasks per column. One of the tasks in the “Coding/Testing” column must be completed or moved first to

the “Review” column. Then From the “To-do” column task must be moved to the “Coding/Testing” column to maintain the WIP condition.

- Also, we need to choose committed team members to work exclusively on this assignment, avoiding multitasking.
- Keep an eye on the task's progression and take quick action to remove any obstacles to keep it moving forward.
- Moreover, we must shorten the approval and merging steps to increase the speed of workflow stages.

To Do	Testing	Review	Done
A D	B E	I J K F	C G H

Question 2: Functional and Non-functional requirements

- ❖ Mobile application for weather updates.
- ❖ Online application for rental accommodation booking.

Q2.A at least three stakeholders and three actors for each application

1. Mobile application for weather updates.

Stakeholders:

- Users (general Public)
- Advertisers/ Sponsors
- Weather-Related Agencies:

Actors:

- Weather data provider (Human Actor)
- App User ---- primary actor (Human Actor)
- App developer/ Administrator (Human Actor)
- Database ----- (Non-Human Actor)
- Staff member
- Notification system
- Customer support representation
- Admin / Management Staff

2.Online application for rental accommodation booking.

Stakeholders:

- Application Developers/ Administrators
- Renters/ Tenants
- Property Owners/Managers:
- Payment service Provider

Actors:

- Payment Gateway ---- Non-Human Actor
- User((Renter/Tenant) ---- primary actor (Human Actor)
- Notification System ----- (non-human Actor)
- Database ----- (Nonhuman Actor)

Q2.B one user story for each identified actor for specific application

❑ **User Stories:**

❖ **Mobile application for weather updates.**

1. **As a User**, I want to receive all notifications about weather conditions or alerts, so that I can take necessary precautions and stay safe.

2. **As a user**, I want to customize my weather preferences, such as temperature units or preferred locations, so that I can use the application to meet my specific needs.

3. **As a staff member**, I want to update information so that passengers can easily find their way to the airport.

4. **As an App Developer**, I want to receive user feedback for using application, so that I can identify the areas for improvement and update the changes.

5. **As an Administrator**, I want to monitor and manage the application's server performance so that I can ensure it works properly during the extreme weather events

6. **As a Notification system**, I want to send timely and relevant weather updates to users, so that they stay informed about the changes in weather conditions

❖ **Online application for rental accommodation booking.**

1. **As a Renter**, I want to have the ability to save and revisit my favorite listings, so that I can able to compare and make decisions about potential accommodations over time.

2. **As a User**, I want to be able easily search for rental accommodations based on my preferences, such as location, distance, price range, other facility, so that I can quickly find suitable options.

3. **As the Database**, I want to support quick and accurate search functionality for users, so that allow them to find and view property listings efficiently.

4. **As a Renter**, I want to easily communicate with property owners or managers through the application, seeking additional information, so that I can address any specific requirements.

5. **As the Notification System**, I want to send reminders to users about upcoming check-in dates, so that it helps users prepare for their stay and enhancing the overall user experience

6. **As a database**, I want to store user account information, so that I can ensure a seamless user experience during login and registration process.

7. **As the Database**, I want to facilitate data analytics and reporting functionalities for administrators, enabling them to gain insights into user behavior, booking trends, and overall system performance.

8. **As a Renter**, I want to view detailed information about each rental property, including high-quality photos, descriptions, availability dates, and user reviews, so that I can make informed decisions.

9. **As a User**, I want to receive real time updates on availability of rental properties, so that I can quickly secure a booking for my preferable locations

□ Q2.B Use case diagrams:

Use Case 1:	Receive weather Notifications
Goal:	Receive timely and relevant weather notifications about changing weather condition, ensuring users take precautions and stay safe
Primary:	User (all age – type)
Secondary Actor(s):	Notification system (Non- Human actor)
Pre-conditions:	<ul style="list-style-type: none"> • User has downloaded and install the weather updates mobile applications • User has granted necessary permission for the applications to end push notifications. • Mobile data or Wi-Fi
Trigger:	The user needs to be informed of any changes in the weather and receive pertinent warnings or information.
Flow of Events:	
	1. First , the user opens the weather updates mobile applications on their devices.
	2. The applications requests users to provide preferable location for weather information
	3. Using the user's location as a basis, the program receives current weather information.
	4. The necessity for a notification arises from changes in the weather from the application
	5. Temperature changes, alerts for severe weather, and other important information are displayed in the message regarding the weather change.
	6. Getting information from the application system users takes necessary precautions based on provided weather information such as carrying an umbrella, dressing warmly, or adjusting travel plans and so on.
Extensions	
	3. IF the User want to adjust notification preferences
	3.1 they can navigate application settings to adjust the types and other functionality for weather notification they receive.
	3.2 will resume at step 2
	4. If the users missed any of notifications
	4.1. The applications may continue to provide updates as needed.

	4.2 The users also go back to the notification history within the app
	4.3 Continue from step3
	6. If the application fails to be sending notifications
	6.1 application system provides support options or assistance, or provide alternative methods to the user to access information
	6.2 Continue from step3

Use Case 2:	View detailed Rental property Information
Goal:	Aim is to provide all information that needed for making wise choice in selection the accommodation
Primary:	Renter
Secondary Actor(s):	Database (Non – Human Actor) Notification system (Non- human actor)
Pre-conditions:	<ul style="list-style-type: none"> • Renter has logged into the online application using their information • The application's database includes the listing for the rental property
Trigger:	The tenant searches for a particular property or navigates to the list of available rental units.
Flow of Events:	
	1. After accessing the online application for rental details, tenants search for specific areas to find the rental accommodations
	2. After specifying criteria for searching the application collects a list of available rental properties from the database.
	3. The renter selects a specific rental property from the list
	4. The application gets detailed information, such as property descriptions, availability dates, and user evaluations, about the chosen property from the database.
	5 The Image Hosting Service provides high-quality images of the rental property that are accessed by the application.
	6. Then the application comes with details about the selected property from the database including property description, availability dates, high quality image and so on.

	7. The renter reads the information that displayed on the application carefully, for the further proceed
	8. Then when they decided to go through the further process, they try to contact the owner via (email, phone call, SMS or face-to-face) and refund details more briefly
Extensions	
	4. If there is incomplete information about the selected property
	4.1 The tenant may be requested to contact the property owner once it is displayed available on the applications
	4.2 Continue from step 3
	6. IF the renters could not find any property according to the preferable locations
	6.1 The application notifies the renter and may suggest adjusting the search parameters.
	6.2 will resume at step 2
	7. if the selected property has limited availability dates,
	7.1 The system will notify the users and provide them with alternate options for confirmation
	7.2 will continue from step 6

- ❑ **Q2.(D) Identify which one would be the best reliability metric (MTTF, MTBF, ROCOF, POFOD or Availability) for each identified user story in part B. Justify your answer by providing reason of selection**

❖ **User story 1: Mobile application for weather updates.**

Reliability metric: Availability

A major factor in the whole user experience is the weather update system's reliability. Users will be more satisfied if the system is continuously available since they will be able to rely on it without any disruptions. As a metric, availability ensures reliability of the system and gives clients continuous access to the most recent weather information. Availability reflects the reliability of the weather update system in providing timely information. Customers expect frequent access to up-to-date climatic information, especially when facing situations where weather can change rapidly.

❖ User story 2.(Online application for rental accommodation booking.)

Reliability metric : MTBF

"Mean Time Between Failures (MTBF)" is a reliability metric that measures the average time a system operates before experiencing a failure. In the context of a Rental Accommodation Booking Application, MTBF can be relevant for certain user stories that focus on the reliability and stability of the system.

Customers expect reliable service and fast booking confirmations. The average time between successful booking confirmations, or MTBF, can be utilized to determine how reliable the system is at processing and confirming bookings without any errors.

❑ Q2 E ----Non-Functional Requirement Analysis – Usability requirement

User story 1: Mobile application for weather updates

- The weather provider system must be organized, easy to understand, enjoyable experience.
- The system must allow the user to change preferable location to with almost 2 mouse clicks.
- The system also shows all the weather-related options such as temperature unit, and others on one display screen.
- The application must provide quick and efficient retrieval of real-time weather updates at least 99% of the time.
- Weather notifications must be clear, concise, and timely, providing users with relevant information without causing confusion within 30 seconds, on average.
- The system also has accessibility providing features such as text- to-speech, voice commands, and font size for displaying the weather information
- Users should be able to look at complex weather patterns with the application's interactive weather maps, which include zoom and pan features.
- The application should also provide essentials weather information even the device is not connected with mobile data or Wi-Fi

User story 2.(Online application for rental accommodation booking.)

- **User-Friendly Interface:** The system must have a user-friendly interface to ensure easy use.

So that they can search for preferable location, distance from bus stop, shopping market, location must be trackable by the GPS location, availability and complete their booking without any troubles.

The system must allow the user to see previous rental house search history with at most 2 mouse clicks.

- **Finding efficient Data:** The system must allow users to access the application system to find relevant information (preferable location, distance, price range, facility, Wi-Fi details, roommate details, rules of the house, available time,) within 30 seconds, on average.
- Thus, tenant and property Owner of the house will be able to collect relevant information easily
- **Accessibility:** The software must facilitate the users getting correct weather information at least 99% of the time.
- The system must provide quick response times to user actions to avoid frustration.
- The system must show all preferable location information based on different category and options on one screen.

❑ Question 3: Unit Testing _____

Q3.A Testcase

Equivalence Partitioning testing

SR No.	Test Case	Test Data	Expected Output
1.	num1 <= num2, num3 >= num4	(3,5,7,2)	-1
2.	num1 > num2, num3 >= num4	(5,4,7,2)	4.5
3.	num1 < num2, num3 >= num4	(2,6,7,3)	-1
4.	num1 > num2, num3 < num4	(5,3,1,6)	-1
5.	num1 == num2, num3 >= num4	(6,6,9,2)	-1
6.	num1 < num2, num3 < num4	(2,4,1,3)	-1
7.	num1 > num2, num3 == num4	(9,5,3,3)	5
8.	num1 == num2, num3 == num4	(6,6,8,8)	-1

Q3.B the Boundary Values Analysis (BVA) test Case:

SR No.	Test Case	Test Data	Expected Output
1.	Age< 6	Age = 3	“Not Eligible”
2.	Age <13	Age = 10	“Eligible for 1 dose”
3.	Age<17	Age = 15	“Eligible for 2 dose”
4.	Age <26	Age = 20	“Eligible for 3 dose”
5.	Age<46	Age = 35	“Eligible for optional dose”
6.	Age<61	Age = 50	“Must take optional dose”
7.	Age<86	Age = 70	“See your GP for eligibility”.
8.	Age>85	Age = 100	“Invalid age”
9.	Age<0	Age = -1	“Invalid age”

Q3.C Find median of 3 numbers. White box testing (test Case):

SR No.	Test Case	Test Data	Expected Output
1.	Value1 is median	Value1 = 5 Value 2 = 3 Value 3= 7	“5”
“2.	Value2 is median	Value1 = 8 Value 2 = 12 Value 3= 4	“8”
3.	Value3 is median	Value1 = 10 Value 2 = 15 Value 3= 6	“10”
4.	All value is equal	Value1 = 7 Value 2 = 7 Value 3= 7	“7”
5.	All value are zero	Value1 = 0 Value 2 = 0 Value 3= 0	“0”
6.	Non – Integer value	Value1 = “abc” Value 2 = 5 Value 3= 8	“-1”

Q3.D Implement your test design in parts A, B and C using python (unittest) modules, run show the screenshots in the report and submit the implemented python codes.

```
1 #Unittesting.py
2
3 import unittest
4 from unittest.mock import patch
5 from io import StringIO
6
7 from Q3PartA import funmean
8 from Q3PartB import El_COVID
9 from Q3PartC import median
10
11 class AllTest(unittest.TestCase):
12
13     #testcase 1 to testcase 6 is the testcode for median of 3 numbers Q3.PartC.py
14
15     def testcase1(self): #v1 =M
16         with patch('builtins.input', side_effect=['5', '3', '7']), \
17             patch('sys.stdout', new_callable=StringIO) as mock_stdout:
18             result = median()
19             actual_output = mock_stdout.getvalue().strip()
20             self.assertEqual(actual_output, '5')
21             self.assertEqual(result, 5)
```

```

def testcase2(self): # npn - interger input
    with patch('builtins.input', side_effect=['abc', '5', '8']), \
        patch('sys.stdout', new_callable=StringIO) as mock_stdout:
        result = median()
        actual_output = mock_stdout.getvalue().strip()
        self.assertEqual(actual_output, 'Invalid input')
        self.assertIsNone(result)

def testcase3(self): #all 0
    with patch('builtins.input', side_effect=['0', '0', '0']), \
        patch('sys.stdout', new_callable=StringIO) as mock_stdout:
        result = median()
        actual_output = mock_stdout.getvalue().strip()
        self.assertEqual(actual_output, '0')
        self.assertEqual(result, 0)

def testcase4(self): #v1 = M
    with patch('builtins.input', side_effect=['10', '15', '6']), \
        patch('sys.stdout', new_callable=StringIO) as mock_stdout:
        result = median()
        actual_output = mock_stdout.getvalue().strip()
        self.assertEqual(actual_output, '10')
        self.assertEqual(result, 10)

```

```

def testcase5(self): #all value same
    with patch('builtins.input', side_effect=['6', '6', '6']), \
        patch('sys.stdout', new_callable=StringIO) as mock_stdout:
        result = median()
        actual_output = mock_stdout.getvalue().strip()
        self.assertEqual(actual_output, '6')
        self.assertEqual(result, 6)

def testcase6(self):
    with patch('builtins.input', side_effect=['0', '0', '0']), \
        patch('sys.stdout', new_callable=StringIO) as mock_stdout:
        result = median()
        actual_output = mock_stdout.getvalue().strip()
        self.assertEqual(actual_output, '0')
        self.assertEqual(result, 0)

```

```

# testfucntions for testing Covid doage  accordign to age Q3PartB.py
def test_El_COVID(self):
    self.assertEqual(EI_COVID(-1), "Invalid age")
    self.assertEqual(EI_COVID(3), "Not eligible")
    self.assertEqual(EI_COVID(10), "Eligible for 1 dose")
    self.assertEqual(EI_COVID(15), "Eligible for 2 doses")
    self.assertEqual(EI_COVID(20), "Eligible for 3 doses")
    self.assertEqual(EI_COVID(46), "Eligible for optional dose")
    self.assertEqual(EI_COVID(60), "Must take optional dose")
    self.assertEqual(EI_COVID(85), "See your GP for eligibility")
    self.assertEqual(EI_COVID(100), "Invalid age")

def test_funmean(self): # test code for mean of 4 numbers Q3.PartA.py
    self.assertEqual(funmean(3,5,7,2),-1 )
    self.assertEqual(funmean (5,4,7,2 ), 4.5)
    self.assertEqual(funmean (2,6,7,3) , -1)
    self.assertEqual(funmean (5,3,1,6) , -1 )
    self.assertEqual(funmean(6,6,9,2), -1 )
    self.assertEqual(funmean (2,4,1,3) , -1)
    self.assertEqual(funmean(9,5,3,3), 5 )
    self.assertEqual(funmean(6,6,8,8) , -1 )

if __name__ == "__main__":
    unittest.main()

```

```

=====
FAIL: test_El_COVID (__main__.AllTest.test_El_COVID)
-----
Traceback (most recent call last):
  File "c:\Users\dhrub\Codes\Unittesting.py", line 74, in test_El_COVID
    self.assertEqual(EI_COVID(46), "Eligible for optional dose")
AssertionError: 'Must take optional dose' != 'Eligible for optional dose'
- Must take optional dose
+ Eligible for optional dose
-----
Ran 8 tests in 0.009s

```

Here one of my test code (Q3PartB.py) is showing error.. Others are successfully unittest is ok.

❑ **Question 4: Modularity**

❑ **A. Identify the modularity issues in all function**

Sr no	Function Name	Identified Issue	Category of Issue	Description/ comment	Solution
1.	printNameID	Global variable "value"	Coupling Issue	The "value" parameter is used as a global variable in printNameID, which increase coupling between function	In printNameID, pass 'value' as a parameter to the global variable to remove reliance on the global
2.	sizeofValue	Global Variable	Coupling Issue	"name_size" and "ID_size" are global variables increasing coupling between functions	pass "student_name " and "student_ID" as parameter to the global variable to remove reliance on the global.
3.	printNameChar	Global variable	Coupling Issue	"student_name" is global variables increasing coupling between functions	Pass "student_name as parameter to the global variable to remove reliance on the global.
4.	printIDChar	Global variable	Coupling Issue	"ID_size" are global variables increasing coupling between functions	Pass "student_ID" as parameter to the global variable to remove reliance on the global.
5.	InputRecord	Global variable	Coupling Issue	"student_name" and "ID_name" are global variables increasing coupling between functions	pass "student_name " and "student_ID" as parameter to the global variable to remove reliance on the global.

```
PS C:\Users\dhruv> & "C:/Program Files/Python312/python.exe" c:/Users/dhruv/Codes/newQ4.py
Enter student name : Dhrubo jouti das troyee
Enter student ID :90028995
Size of student Name: 25
Size of student ID: 8
Student Name:  Dhrubo jouti das troyee
Student ID:  90028995
Name : 25
ID : 8
Name characters:
  D h r u b o   j o u t i   d a s   t r o y e e
ID characters:
9 0 0 2 8 9 9 5
```

After modifying modularity, the issues this is how it is working. The filename also changed from “Q4.py “ to “ newQ4.py”

❑ Question 5: Ethics and Professionalism _____

Q5. A An online software for tracking flights (google flightradar24)

Identifying the harm:

Financial harm:

- 1.The airline , air travelers may suffer large financial losses because of possible legal fees, lawsuits for damages, higher insurance costs and all.
2. incorrect information about flights causing accidents or crashes in the sky.

2.Physical Harm:

Bad flight tracking has given rise to collisions and accidents that resulted in stress among both passengers and controllers.

Scenario:

Think about an extremely busy day at an Air Traffic Control where multiple flights have been planned using Google Flightradar24. However, a combination of rushed development and neglected testing has left the software malfunctioning.

Number of problems that can occur for this. Such as:

1. The location of an aircraft is shown by the flight tracking program incorrectly.

As a result, the Air traffic Controller will consider that the Pilot is taking the plane in another direction than it is.

- 2.Air traffic controllers unintentionally allow another aircraft to enter the same airspace by relying on inaccurate data.

As a result, it will create a near – collision scenario.

Software Engineering Error process:

During the testing phase, important defects might go under the radar if the testing team neglects to thoroughly test the software or fails to follow a range of real-life situations. Incomplete testing, for example, may fail to identify edge cases involving sudden changes in flight path, resulting in pilots and air traffic controllers receiving incorrect data.

Ethics/ Professionalism issue:

1. Testing: Neither white box testing nor any testing was carried done.

Scenario: Testing white box/ Code Development issue:

1.A lack of understanding of the complexities of the flight tracking algorithms may result in poor test coverage if the testing team and developers are not properly communicating.

2.The dangers are further increased if testing takes place quickly and there is pressure to start the program soon, which could lead to the neglecting of important issues.

3.Thorough test planning, sufficient test coverage, requiring performance of tests, effective communication between the development and testing teams are all necessary to ensure ethical testing methods. There's a much smaller chance that important issues will go undiscovered when full investigative and automated testing approaches are used.

Scenario: Functional Requirements (NFR):

There should be a backup system in the plane that automatically activates in case the main one malfunctions or stops working.

- **Multiple Aircraft Support:**

The software must have the ability to monitor and present data for many airplanes at the same time as Air traffic controllers have a lot of flights to handle at once.

- **Real –time- Aircraft Tracking:**

The system must provide real – time tracking of aircraft, displaying accurate and updated information on their location, speed and other functionality.

- **Flight Path Direction:**

Using past data and present conditions, the system ought to predict and show each aircraft's expected flight route that helps in identifying and organizing future aircraft movements for controllers.

- **Emergency Alert:**

When an emergency occurs, such as a sudden change in height or a departure from the planned flight path, the software must recognize it and notify air traffic controllers as soon as possible, which is essential for swift handling of possible safety risks.

- **Authentication and Authorization:**

It is recommended that authorized staff members only have access to the flight tracking software, and that user classifications be set up to control system functions so that safety measures protect against system abuse and illegal access.

- **Alert priority:**

Prioritizing different alert types (such low fuel and emergency) should be done according to their impact and immediately which Supports controllers in identifying essential concerns during high-pressure scenarios.

Q5.B. Video calling software.

Identifying the harm

1.financial Harm:

Insufficient project management in the development of video calling software may increase the Illegal activities throughout the video calling network or software issues impacting payment systems may cause users or the service provider to face financial losses.

Scenario:

Unexpected expenses are caused by insufficient resource planning and budget management, which affects the video calling software project's capacity to make money.

2. Physical Harm:

When project management mistakes cause delays in the use of video calling software for vital applications like emergency services or healthcare, the results could be seen immediately, affecting patient care or emergency response times.

Scenario:

A telemedicine-focused video calling software project experiences challenges that affect the capacity to offer fast medical consultations.

3.Social Harm:

Project management mistakes may cause delays in the introduction of video calling software, which could negatively affect user expectations and possibly lead to dissatisfaction

Scenario: Due to project management delays, users who were expecting the quick release of an update for the video calling software are disappointed and frustrated.

Software Engineering Error process for Video calling Software:

In the situation when a software development project is underway, and the project manager fails to do the tasks related with it, leads a shortage of skill developers for a critical phrase of this project. Moreover, unclear communication with team member results the priority of the project in a low and the project fails behind schedule.

More things are added here:

1. Look for signs of poor project management that are impacting the development of the video calling software, such as delays, poor use of resources, or insufficient.
2. According to their seriousness and potential impact on the video calling software project's financial, social, or physical well-being, assign project management problems a priority.
3. Review the entire project management process, identify areas for improvement, and implement changes for future projects.
4. To ensure, more efficient project execution in next software development projects, conduct an ongoing evaluation of the project management processes to find opportunities for improvement
5. Record the specifics of detected project management mistakes, including effects on the economic, social, and environmental sides.
6. Charge the project management group or other pertinent parties with looking into and fixing the issues related to project management that have been found.
7. To find out why project management mistakes happened and affected the development of the video calling software, do a root cause.
8. This procedure for handling software errors highlights the need for effective project management to avoid the financial, physically and socially harm sufferings.

Q5.C Online stock market application.

Identifying the harm:

Financial harm:

1 Hardware or software issues affecting essential features of an online stock market applications.

2.Physical Harm:

This mal functionality of Online stock market mobile application also increase stress among both investors and customers

Scenario:

Consider a scenario in which investors mostly depend on an internet stock market app to complete transactions. An unexpected software problem in the application stops customers from performing buy or sell orders at a time of extreme fluctuations in the markets. Investors that are unable to respond quickly in the face of market moves risk financial loss through missed opportunities or slow responses.

Software Engineering Error process:

Functional Requirements:

1. Reliability and Stability:

The program must have stability and dependability, which decreases the possibility of malfunctions or breakdowns so that it can ensure smooth and continuous trading experience.

2. Real –time Order Executions:

The application should facilitate real-time execution of buy and sell orders on the stock market, so that Investors must react quickly to progress in the market.

Non - Functional Requirements:

Safety Methods:

Have strong safety precautions in place to safeguard sensitive financial data and user accounts and thus security measures against unwanted entry and possible money fraud.

The capacity:

During busy trading hours, the application should to be able adjust to the increased user activity without any problems so that it can prevent system overload in times with higher market activity.

Software Engineering Error process:

1. Identify signs of software issues impacting the stock market application's ability to process orders.
2. Develop and implement fixes or solutions to address the identified errors, focusing on ensuring real-time order execution
3. Assign the development and maintenance team the task to examine into and fixing the issues that have been found.
4. Record detailed information regarding detected problems, including how they affect the execution of orders and payments.
5. Examine and improve software development and testing processes to avoid same kinds of errors that impact financial transactions.
6. Perform an in-depth investigation to identify the actual causes of the software issues impacting the processing of orders.
7. Analyze problems according to their seriousness and potential for causing users' money to be lost.

❑ VERSION CONTROL:

Branching Strategy: There will be 3 branches:

1. Main Branch: The primary branch that contains the stable version of the working, where other branches will be merged and quality assurance checks. It should only be updated with thoroughly tested and reviewed code.

```
mkdir ToDoProject  
cd ToDoProject  
git init
```

2. Development Branch: The branch that contains the latest version of the work. Developers will create this branch to work on user stories, add features or fix issues. To maintain the branch with the most recent stable code, we need to update this branch weekly depending on the final work states,

```
git checkout -b development  
git checkout -b bugfix
```

3. Feature Branches: These branches are used to develop new features, major changes or update existing ones in the WBS. For user stories given, each feature branch will be created from the development branch. They will work on the feature branch, change and execute them. All new features and updates are merged into it. Then it should be tested here and merged back into the development branch

```
git checkout -b features
```

When should they be created and when should they be merged and pushed.

At the beginning of the project, the main branch and the development branch will be created. (Day1)

After this stage, the Development Branch will be tested and the change should be merged into the main branch, which will represent a new version of the work that has been done. (Ongoing development weekly)

Then, a feature branch will be created for user stories and will be used. Then it should be merged into the Development branch and pushed (Weekly)

When the developer or programmer finishes a use story and tests it on the feature branch it will be merged into the development branch and pushed. (Every 2-3week)

Overall, after all the user stories have been completed and tested the development branch will be merged with the main branch and pushed.

Pushing TO a remote repo:

- Upload “mynewbranch” to “origin” (and make “origin” the current remote):

```
git push -u origin mynewbranch
```

- Upload the current branch to the current remote:

```
git push
```