Assignment No.:1

Topic: System Requirements Specification Document

Objective: To get familiar with preparing requirements document, which is used to capture and document all the requirements at the start of the project. In this assignment we mainly focus on functional requirements and SRS to be prepared

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4 of Program Outcomes.

Problem Statement

CASE STUDY

RAILWAY RESERVATION SYSTEM

The purpose of Railway Reservation System is to provide the train timing details, reservation, billing and cancellation.

Using these systems Ticket Counter person can perform operations like finding out the train timings and to know information about PNR status, seats availability and costs of each ticket, etc.

Design/Methodology/Algorithm/Data Structure

Outputs/Results

INTRODUCTION

1.1 Purpose

The purpose of Railway Reservation System is a software application which provides the train timing details, reservation, billing and cancellation.

Using these systems Ticket Counter person can perform operations like finding out the train timings and to know information about PNR status, seats availability and costs of each ticket, etc.

This system is developed in three categories:

- Ticket Reservation
- Ticket Cancellation
- Status checking

1.2 Document Convention

Font: Times New Roman Size: 10

1.3 Intended Audience

The customers accessing the online reservation system for booking railway tickets.

1.4 Product Scope

·Freight Revenue enhancement.

·Passenger Revenue enhancement.

·Improved & optimized service

1.5 References and Acknowledgments

Articles:-

1.IEEE SRS Format

Websites:-

1. Yatra.com

2.Irctc.co.in

3.Indianrail.gov.in

4.www.google.com

OVERALL DESCRIPTION

2.1 Product Perspective

Before the automation, the system suffered from the following **DRAWBACKS:**

- The existing system is highly manual involving a lot of paperwork and calculation and therefore may be erroneous. This has lead to inconsistency and inaccuracy in the maintenance of data.
- The data, which is stored on the paper only, may be lost, stolen or destroyed due to natural calamity like fire and water.
- The existing system is sluggish and consumes a lot of time causing inconvenience to customers and the airlines staff.
- Due to manual nature, it is difficult to update, delete, add or view the data.
- Since the number of passengers have drastically increased therefore maintaining and retrieving detailed record of passenger is extremely difficult.
- An railways has many offices around the world, an absence of a link between these offices lead to lack of coordination and communication.

Hence the railways reservation system is proposed with the following

- The computerization of the reservation system will reduce a lot of paperwork and hence the load on the airline administrative staff.
- The machine performs all calculations. Hence chances of error are nil.
- The passenger, reservation, cancellation list can easily be retrieved and any required addition, deletion or updation can be performed.
- The system provides for user-ID validation, hence unauthorized access is prevented.

2.2 Product Functionality

Booking agents with varying levels of familiarity with computers will mostly use this system. With this in mind, an important feature of this software is that it be relatively simple to use. The scope of this project encompasses:

Search: This function allows the booking agent to search for train that are available between the two travel cities, namely the "Departure city" and "Arrival city" as desired by the traveller. The system initially prompts the agent for the departure and arrival city, the date of departure, preferred time slot and the number of passengers. It then displays a list of train available with different airlines between the designated cities on the specified date and time.

Payment: The user can also handle payment of the ticket through this.

Cancellation : The system also allows the passenger to cancel an existing reservation. This function registers the information regarding a passenger who has requested for a cancellation of his/her ticket.

Answer Queries: The system also answers certain queries made by the user.

2.3 Users and Characteristics

- **EDUCATIONAL LEVEL**:-At least user of the system should be comfortable with English language.
- **TECHNICAL EXPERTISE: -** User should be comfortable using general purpose applications on the computer system.

2.4 Operating Environment

The system cannot run without

Minimum Hardware Requirements

Processor Pentium III

Hard disk drive 40 GB

RAM 128 MB

Cache 512 kb

Preferred Hardware Requirements

Processor Pentium IV

Hard disk drive 80 GB

RAM 256 MB

Cache 512 kb

2.5 Design and Implementation Constraints

- This system is only made for official purposes and cannot be handled by the general public.

2.6 User Documentation

- Application Specification
- SRS
- User Manual

2.7 Assumptions and Dependencies

- Railway Employees will be having a valid username and password to access the software
- The software needs Railway Employees to have complete knowledge of railways reservation system.
- Software is dependent on access to internet.

EXTERNAL INTERFACE REQUIREMENTS

3.1 User Interfaces

The user interfaces defines the human-computer interaction of the railway reservation system.

3.2 Hardware Interfaces

The software shall interface with the electromechanical that controls the online connection systems. The hardware interface is supported by the main control panels

3.3 Software Interfaces

Software interfaces is supported by the main control panels and operating systems which hosts the algorithm for calculating distributed travel and wait time information.

3.4 Communication interfaces

All system interfaces communicate in order to activate ordered requests. The communication mediums are the external interface that communicates with the control panel of the railway reservation system.

SYSTEM FEATURE

4.1 System Feature 1

Search:-

Input: Departure and Arrival cities.

Output: Available trains.

4.2 System Feature 2

Payment:-

Input: Online banking details

Output: Booking Id.

4.3 System Feature 3

Cancellation:-

Input: Booking Id, Reason for Cancellation

Output: Ticket cancelled.

4.4 System Feature 4

Answer Queries:-

Input: Username and Password to send the completed form

Output: The form is sent to the administrator

OTHER NON-FUNCTIONAL REQUIREMENTS

5.1Performance Requirements

- **User Satisfaction:** - The system is such that it stands up to the user expectations.
- **Response Time: -**The response of all the operation is good. This has been made possible by careful programming.
- **Error Handling:** - Response to user errors and undesired situations has been taken care of to ensure that the system operates without halting.
- **Safety and Robustness:** - The system is able to avoid or tackle disastrous action. In other words, it should be foul proof. The system safeguards against undesired events, without human intervention.
- **Portable:** - The software should not be architecture specific. It should be easily transferable to other platforms if needed.
- **User friendliness:** - The system is easy to learn and understand. A native user can also use the system effectively, without any difficulties.

5.2 Safety and Security Requirements

The system use SSL (secured socket layer) in all transactions that include any confidential customer information. The system must automatically log out all customers after a period of inactivity. The system should not leave any cookies on the customer's computer containing the user's password. The system's back-end servers shall only be accessible to authenticated management.

5.3 Software quality attributes

Software should satisfy following requirements as well:-

- SECURITY
- PORTABILITY
- CORRECTNESS
- EFFICIENCY
- FLEXIBILTY
- TESTABILTY
- REUSABILTY

5.4 Business Rules

Not Applicable

Discussion/Conclusion

The purpose of the requirements analysis phase is to transform the needs and high-level requirements specified in the earlier phases into unambiguous (measurable & testable), traceable, complete, consistent & stakeholder-approved requirements.

There are two distinct steps for this phase:-

→ Requirements gathering & analysis: To collect all relevant info regarding the product to be developed and also need out the incompleteness, & inconsistencies in these requirements.

→ Requirement specification: All identified customer requirements are organised into the software Requirement Specification (SRS) document.

Questionnaire

1) What are the different types of requirements?

→ The different types of requirements are:-

- Functional requirements:- They describe the functions to be supported by the system. Each function can be characterised by input data & the output data to be produced.
- Non-functional requirements:- Identify the performance requirements, the required standards to be followed etc.

2) What is the importance of requirement analysis phase?

→ ~~But it~~ The main aim of this phase is to understand the exact requirements of the customer & to document them properly. If this phase is not completed properly or in a sluggish manner then there might be inconsistencies in the final product. It basically lays the foundation of the project.

Grade awarded: 10

Teacher's signature with date .....

Assignment No.:2

Topic: Software Design Document (Data Design)

Objective: In this assignment we will focus on data design using ERD. The ERD should be translated to relational tables following the rules of normalization

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4, PO5, PO12 of Program Outcomes.

Problem Statement

CASE STUDY

RAILWAY RESERVATION SYSTEM

The purpose of Railway Reservation System is to provide the train timing details, reservation, billing and cancellation.

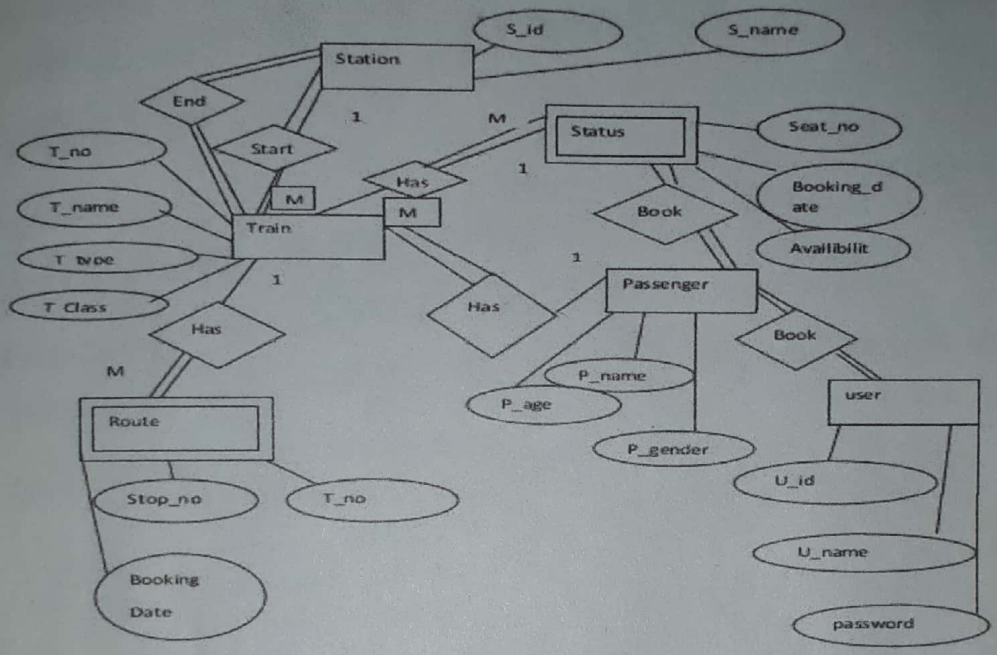
Using these systems Ticket Counter person can perform operations like finding out the train timings and to know information about PNR status, seats availability and costs of each ticket, etc.

Design/Methodology/Algorithm/Data Structure

1. ***Data Design***
   - *1.1 Entities*
   - *1.2 Attributes*
   - *1.3 Relationships*
   - *1.4 ERD*
   - *1.5 Data Dictionary*
   - *1.6 Table Structures*
     - *a. Table 1 - description*
       - *i. Attribute Name*
       - *ii. Data Type*
       - *iii. Constraints*
     - *b. Table 2 - description*
       - *i. Attribute Name*
       - *ii. Data Type*
       - *iii. Constraints*

Elaborate for case study

Output:



Title?

Discussion/Conclusion

An Entity Relationship ~~dig~~ diagram is a data modeling technique that graphically illustrates an information systems entities and the relationship between these entities. An ERD is a conceptual and representational model of data used to represent the entity framework infrastructure.

Steps:

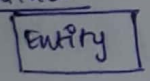
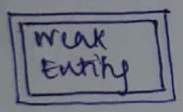
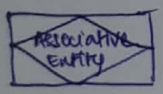
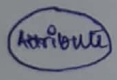
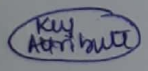
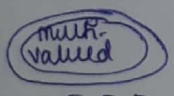
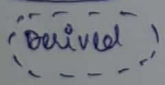
- Identify and define entities.
- determine all interactions between entities.
- Analyse the nature of interactions / determine cardinality.
- Create ERD.

Questionnaire

1. Explain the elements of E-R diagram.

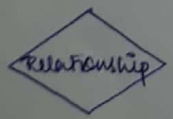
→ There are three basic entities in E-R diagrams:-

- Entities:- These are the things for which we want to store information. An entity is a person, place, thing or event.
- Attributes:- Data we want to collect for entity.
- Relationships:- Describes the between entities.

Symbol	Shape Name	Symbol Description
Entities		
	Entity	An Entity is represented by a rectangle which contains the entity's name.
	Weak Entity	An entity that cannot be uniquely identified by its attributes alone.
	Associative Entity.	An entity used in many-to-many relationship.
Attributes:		
	Attribute	In the chen notation, each attribute is represented by an oval containing attribute name.
	Key Attribute	An attribute the can uniquely identify a particular entity.
	Multivalued Attribute	An attribute the can have many values.
	Derived Attribute	An attribute whose value is calculated from other attributes.

Grade awarded: B

Teacher's signature with date .....

	Strong Relationship	A relationship where entity is existence-independent of other entities and PK of child doesn't contain PK component of Parent Entity.

Assignment No.:3

Topic: Software Design Document (Functional Design)

Objective: In this assignment we will focus on functional design using DFD. The DFD should be decomposed into levels as far as feasible till the lowest level process can be implemented using a single program. The process specification of at least one process should be made.

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4, PO5, PO12 of Program Outcomes.

Problem Statement

CASE STUDY

RAILWAY RESERVATION SYSTEM

The purpose of Railway Reservation System is to provide the train timing details, reservation, billing and cancellation.

Using these systems Ticket Counter person can perform operations like finding out the train timings and to know information about PNR status, seats availability and costs of each ticket, etc.

Design/Methodology/Algorithm/Data Structure

1. *Functional Design*

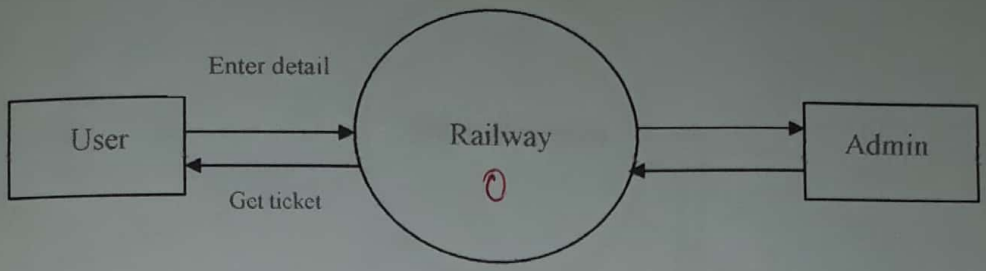
1.1 Overall System description
1.2 DFD
- *a. Context Diagram (Level 0)*
- *b. DFD Level 1*
- *c. DFD Level 2*
- *d. ... ...*

1.3 Functional specification
- *a. Process 1 - description*
  - *i. Inputs*
  - *ii. Outputs*
  - *iii. Files/Tables used*
  - *iv. Process Pseudo Code*
    - *i. Using structured english*
    - *ii. Decision Tree*
    - *iii. Decision Table*
- *b. Process 2 - description*
  - *i. Inputs*
  - *ii. Outputs*
  - *iii. Files/Tables used*
  - *iv. Process Pseudo Code*
    - *i. Using structured english*
    - *ii. Decision Tree*
    - *iii. Decision Table*

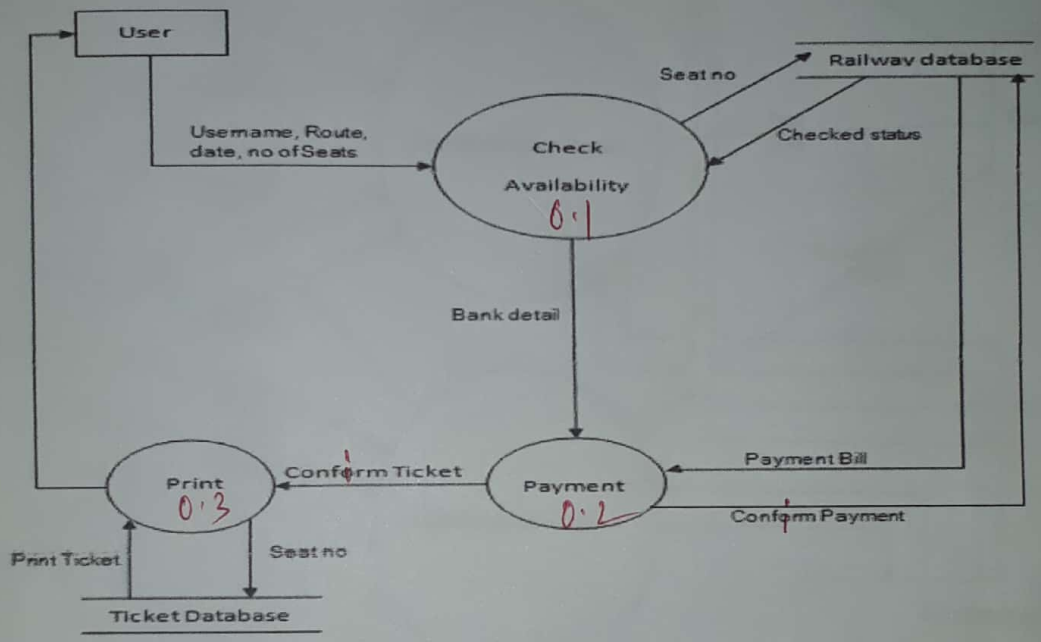
1.4 Assumptions and Dependencies

Outputs/Results

LEVEL 0 DATA FLOW DIAGRAM:

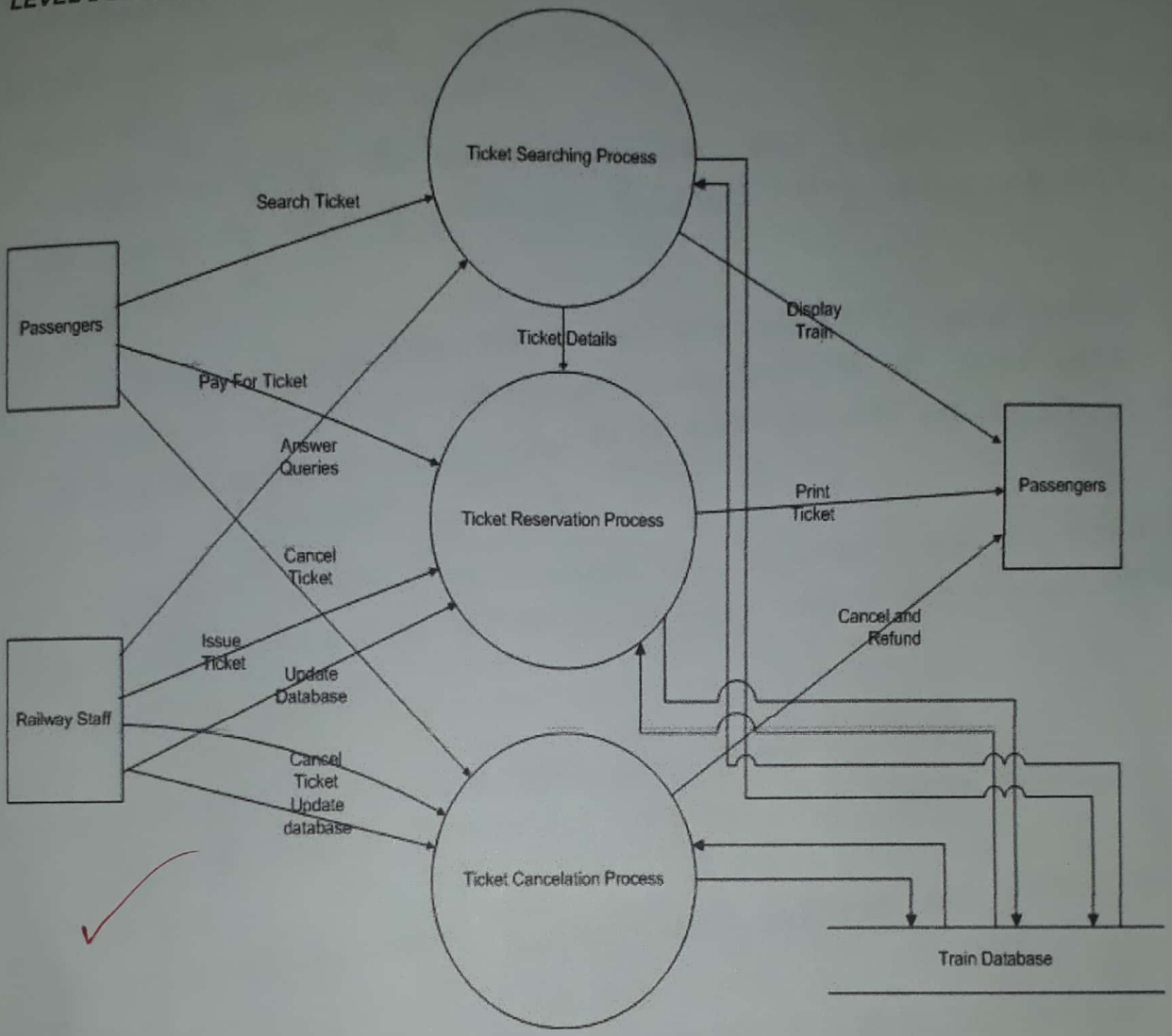


LEVEL 1 DATA FLOW DIAGRAM:



Discussion/Conclusion

LEVEL 2 DATA FLOW DIAGRAM:



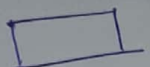
Process Numbering missing.
Which process is expanded?

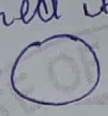
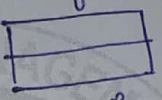
Questionnaire

1) Explain the elements of Data flow Diagram.

Following are the elements.

- External Entity: An outside system that sends or receives data communicating with the system being diagrammed.

  Symbol: 

- Process: Any process that changes the data, producing an output. It might perform computations or sort data based on logic or direct the data flow based on business rules. Symbol:  or 

- Data store: Files or repositories that hold information for later use, such as a database table or a relationship form. Symbol: ——

2) What are the rules for drawing DFD?

→
- Each process should have atleast one input and an output.
- Each data store should have atleast one data flow out.
- Data stored in a system must go through a process.
- All process in a DFD go to another process or data store.

Grade awarded: ~~B~~ E

Teacher's signature with date 31/8/18

Assignment No.:4

Topic: Design Document using OOAD

Objective: To get familiar with preparing design document based on OOAD using the OMG UML diagrams

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4, PO5, PO12 of Program Outcomes.

Problem Statement

CASE STUDY

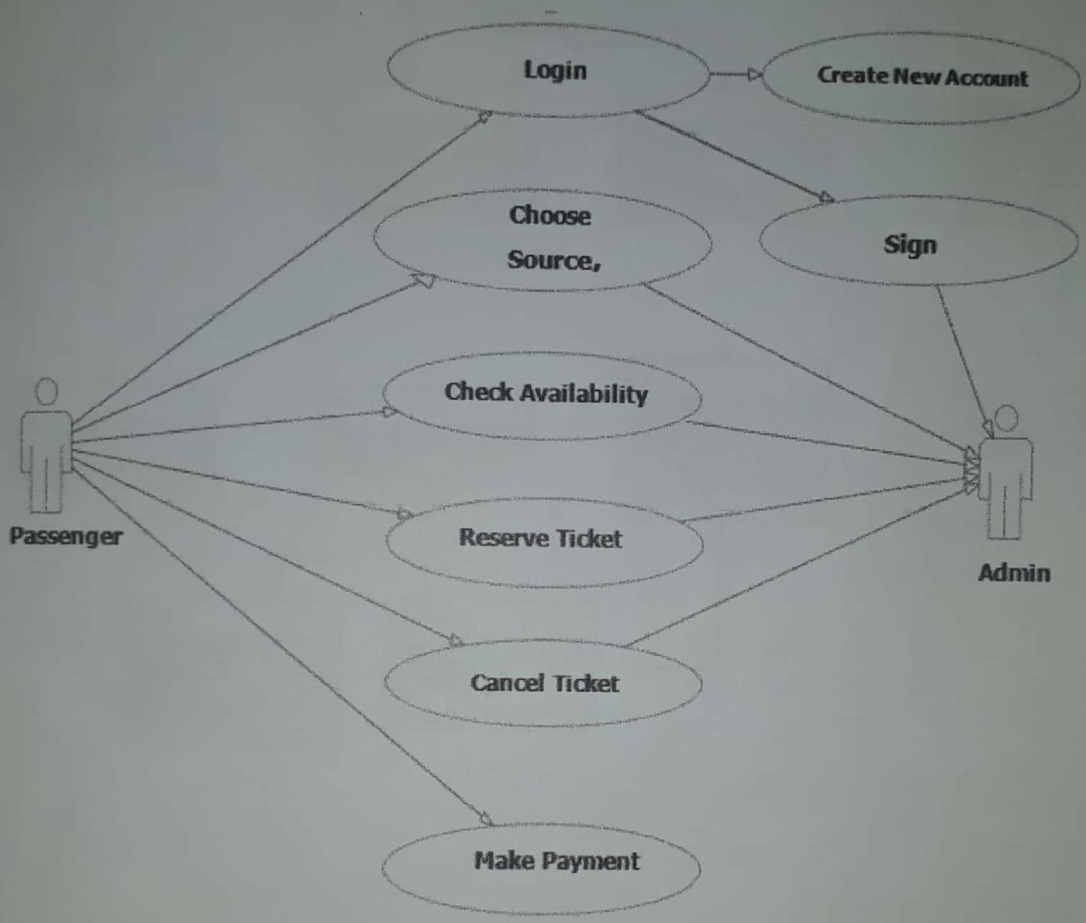
RAILWAY RESERVATION SYSTEM

The purpose of Railway Reservation System is to provide the train timing details, reservation, billing and cancellation.

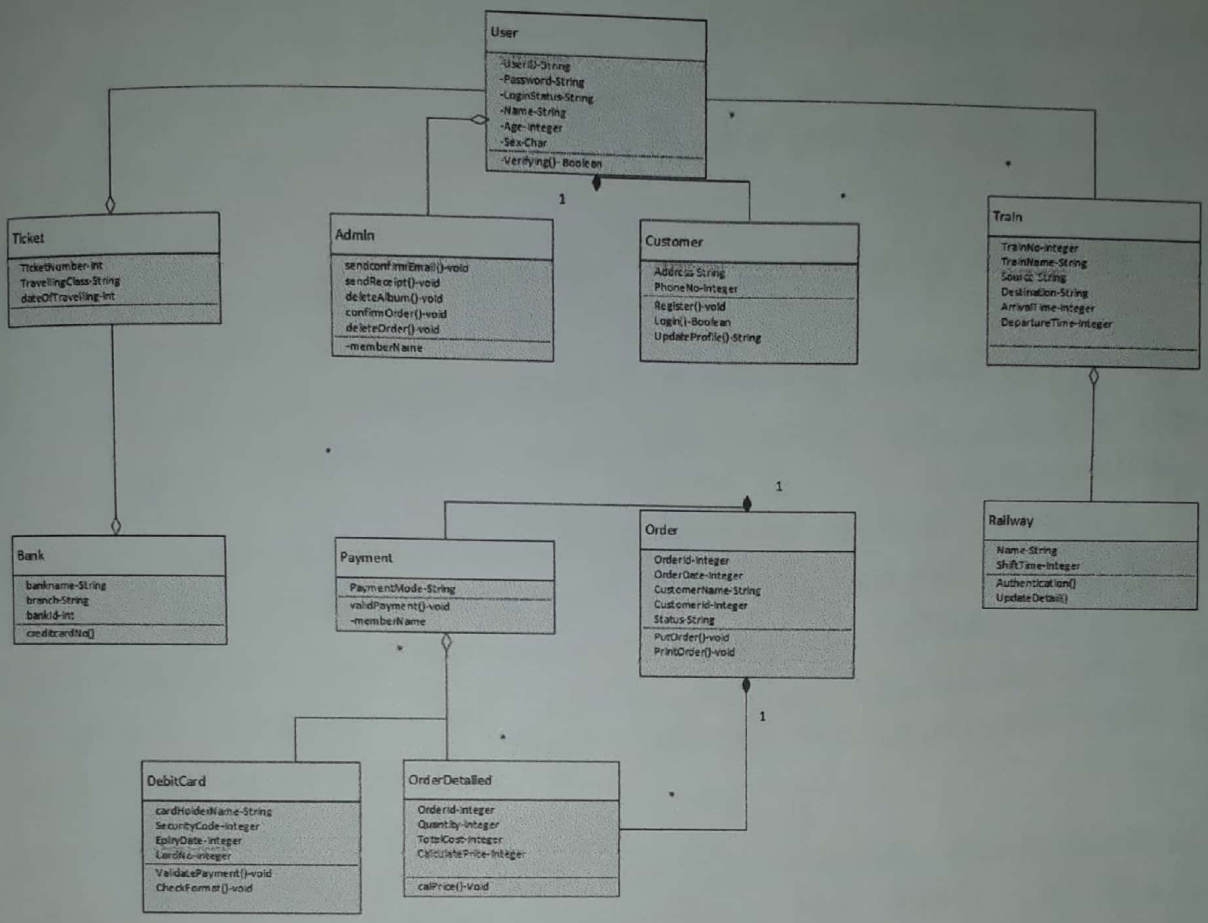
Using these systems Ticket Counter person can perform operations like finding out the train timings and to know information about PNR status, seats availability and costs of each ticket, etc.

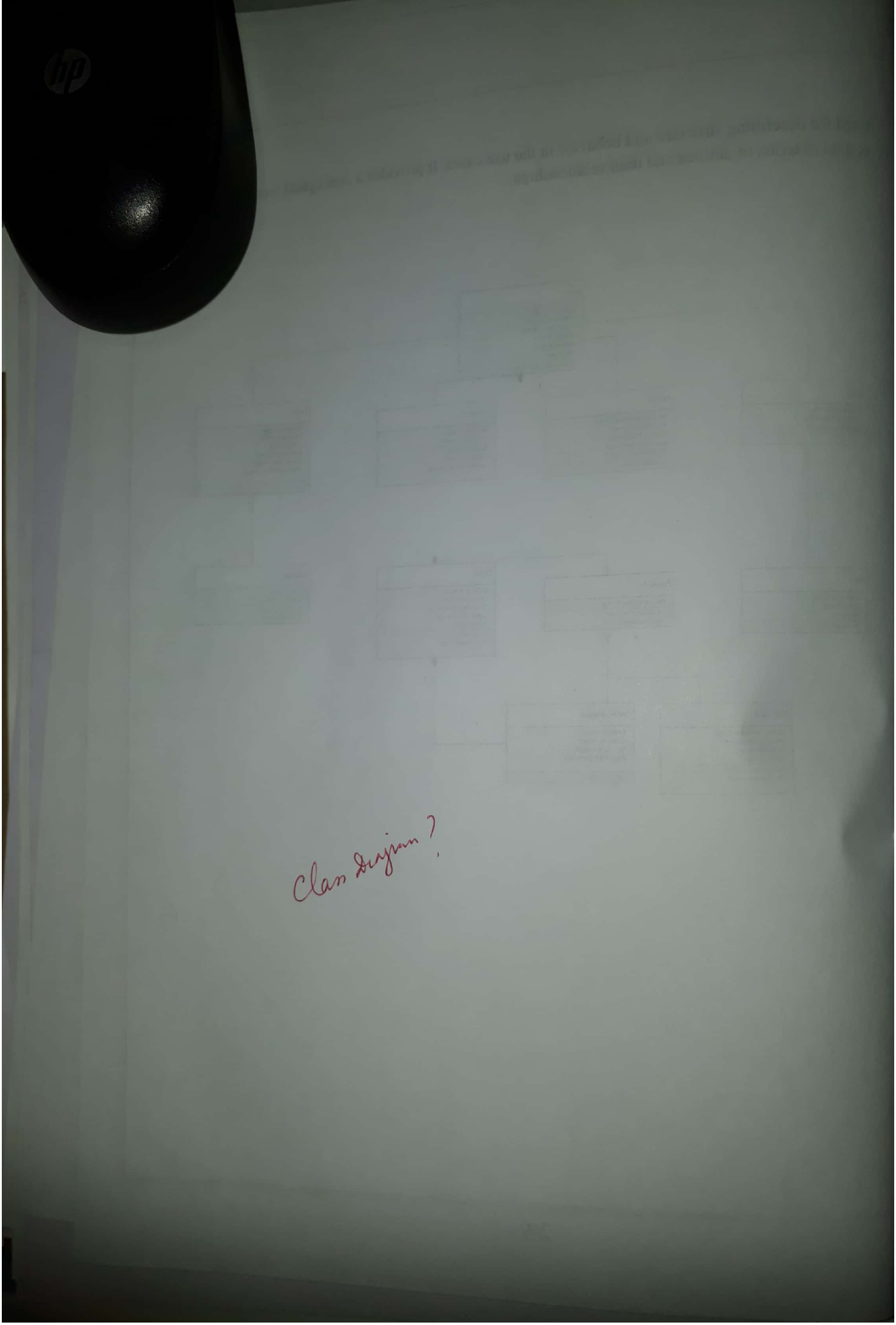
Design/Methodology/Algorithm/Data Structure

A **use case diagram** in the Unified Modeling Language (UML) is a type of behavioral diagram defined by and created from a Use-case analysis. Its purpose is to present a graphical overview of the functionality provided by a system in terms of actors, their goals (represented as use cases), and any dependencies between those use cases. The main purpose of a use case diagram is to show what system functions are performed for which actor. Roles of the actors in the system can be depicted.



Used for describing structure and behavior in the use cases. It provides a conceptual model of the system in terms of entities and their relationships.





Discussion/Conclusion

UML (unified modelling language) is a standard notation for the modelling of real-world objects as a first step in developing an object oriented design methodology. Its notation is derived from the and unifies the notations of three object-oriented design and analysis methodologies:-

- Grady Booch's methodology for describing a set of objects and their relationships.
- James Rumbaugh's object-modeling Technique (OMT).
- Iva. Jacobson's approach which includes a use-case methodology.

UML is not a programming language but tools can be used to generate code in various languages using UML diagrams. UML has a direct relation width object oriented analysis and design. After some standardisation UML has become an OMG (Object Management Group) standard.

An object contains both data & methods that control the data. The data represents the state of the objects. A class describes an object and they also form hierarchy to model real world system. The hierarchy is represented as inheritance and the classes can also be associated in different manners as for the requirements.

The objects are the real world entities that exist around us and the basic concepts like abstraction encapsulation, inheritance, polymorphism, all can be represented using UML. So UML is powerful enough to represent all the concepts exists in object oriented analysis and design.

Questionnaire

1) What are the interactions diagram in UML?

→ Interaction diagrams are models that describe how groups of objects interact among themselves through message passing to realize some behaviours.

2) What does state chart diagram represent?

→ A state chart diagram is normally used to model how the state of an object changes in its life time. State chart diagrams are good at describing how the behaviour of an object changes across several use case executions.

Grade awarded: E

Assignment No.:5

Topic: Project planning using PERT charts & identifying critical path and activities

Objective: In this assignment a PERT chart is given and critical path identification needs to be done.

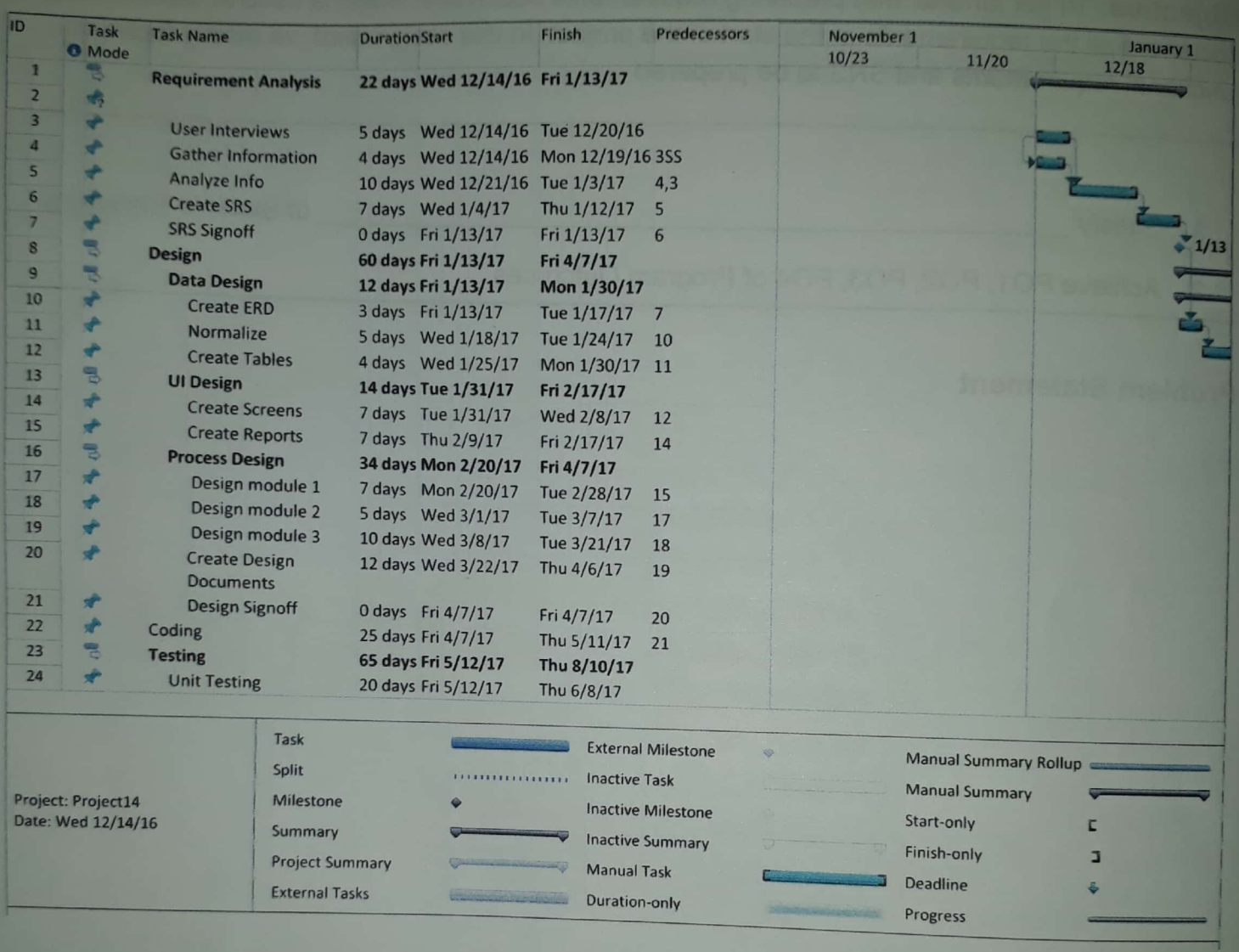
1. Satisfy _ interpret, construct, actualize. _____ of Bloom's Taxonomy
2. Achieve PO5, PO11,PO12 of Program Outcomes.

Problem Statement

Task No	Task Name	Duration (days)	Predecessor
1.	**Requirement Analysis**	**26**	
2.	User Interviews	5	
3.	Gather Information	4	2, Start-to-Start
4.	Analyze Info	10	2,3
5.	Create SRS	7	4
6.	SRS signoff	0	5
7.	**Design**	**60**	
8.	**Data Design**	**12**	
9.	Create ERD	3	6
10.	Normalize	5	9
11.	Create Tables	4	10
12.	**UI design**	**14**	
13.	Create Screens	7	11
14.	Create Reports	7	13
15.	**Process Design**	**34**	
16.	Design Module 1	7	14
17.	Design Module 2	5	16
18	Design Module 3	10	17
19.	Create design Document	12	18
20.	Design signoff	0	19
21.	**Coding**	**25**	**20**
22.	**Testing**	**65**	
23.	Unit testing	20	21
24.	Integration testing	15	23
25.	System Testing	15	24
26	Acceptance testing	15	25
27.	**Implementation & Signoff**	**5**	**26**

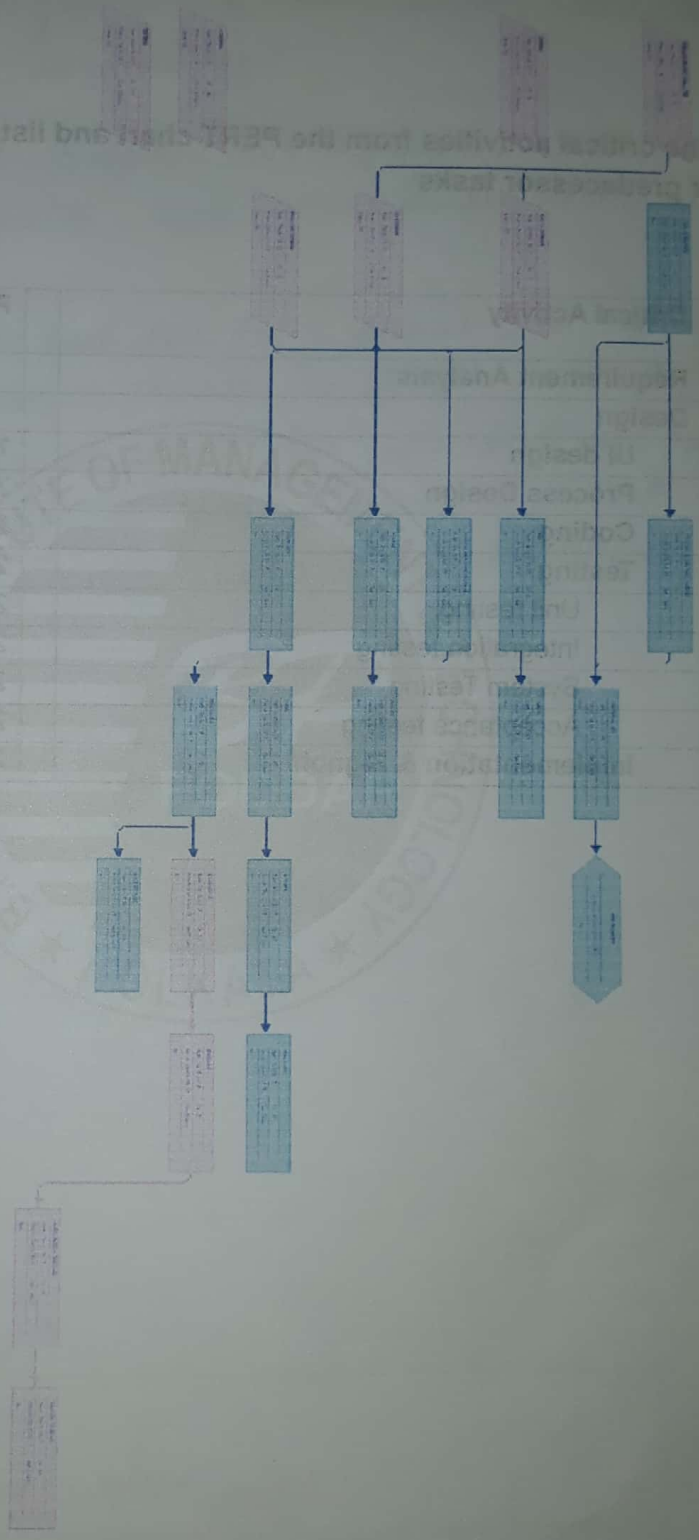
Methodology/Algorithm/Data Structure

ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Requirement Analysis**	**22 days**	**Wed 12/14/16**	**Fri 1/13/17**	
2						
3		User Interviews	5 days	Wed 12/14/16	Tue 12/20/16	
4		Gather Information	4 days	Wed 12/14/16	Mon 12/19/16	3SS
5		Analyze Info	10 days	Wed 12/21/16	Tue 1/3/17	4,3
6		Create SRS	7 days	Wed 1/4/17	Thu 1/12/17	5
7		SRS Signoff	0 days	Fri 1/13/17	Fri 1/13/17	6
8		**Design**	**60 days**	**Fri 1/13/17**	**Fri 4/7/17**	
9		**Data Design**	**12 days**	**Fri 1/13/17**	**Mon 1/30/17**	
10		Create ERD	3 days	Fri 1/13/17	Tue 1/17/17	7
11		Normalize	5 days	Wed 1/18/17	Tue 1/24/17	10
12		Create Tables	4 days	Wed 1/25/17	Mon 1/30/17	11
13		**UI Design**	**14 days**	**Tue 1/31/17**	**Fri 2/17/17**	
14		Create Screens	7 days	Tue 1/31/17	Wed 2/8/17	12
15		Create Reports	7 days	Thu 2/9/17	Fri 2/17/17	14
16		**Process Design**	**34 days**	**Mon 2/20/17**	**Fri 4/7/17**	
17		Design module 1	7 days	Mon 2/20/17	Tue 2/28/17	15
18		Design module 2	5 days	Wed 3/1/17	Tue 3/7/17	17
19		Design module 3	10 days	Wed 3/8/17	Tue 3/21/17	18
20		Create Design Documents	12 days	Wed 3/22/17	Thu 4/6/17	19
21		Design Signoff	0 days	Fri 4/7/17	Fri 4/7/17	20
22		Coding	25 days	Fri 4/7/17	Thu 5/11/17	21
23		**Testing**	**65 days**	**Fri 5/12/17**	**Thu 8/10/17**	
24		Unit Testing	20 days	Fri 5/12/17	Thu 6/8/17	



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
25		Integration Testing	15 days	Fri 6/9/17	Thu 6/29/17	24
26		System Testing	15 days	Fri 6/30/17	Thu 7/20/17	25
27		Acceptance Testing	15 days	Fri 7/21/17	Thu 8/10/17	26
28		Implementation & Signoff	5 days	Fri 8/11/17	Thu 8/17/17	27

Outputs/Results



Discussion/Conclusion

Identify the critical activities from the PERT chart and list them in this page along with their predecessor tasks

Task No	Critical Activity		Predecessor
1.	Requirement Analysis		
7.	Design		
12.	UI design		7
15.	Process Design		7
21.	Coding		20
22.	Testing		21
23.	Unit testing		21
24.	Integration testing		23
25.	System Testing		24
26	Acceptance testing		25
27.	Implementation & Signoff		26

Questionnaire

1) What is the significance of PERT chart? How is it useful for project scheduling?

→ A PERT chart is a project management tool used to schedule, organize, and coordinate tasks within a project.

The Gantt chart and the PERT chart are probably the two most known charts in project management. Each of these can be used for scheduling, but because Gantt charts don't illustrate task dependencies and PERT chart can be confusing, PMs often use both.

Grade awarded: 10

Teacher's signature with date 14/9/18

Assignment No.:6

Topic: Project planning using Gantt chart using Microsoft Project

Objective: In this assignment Microsoft project will be used to formulate the project plan document. Among the different views, Gantt, Tracking Gantt, Network/PERT diagram, Task usage, etc. will be demonstrated

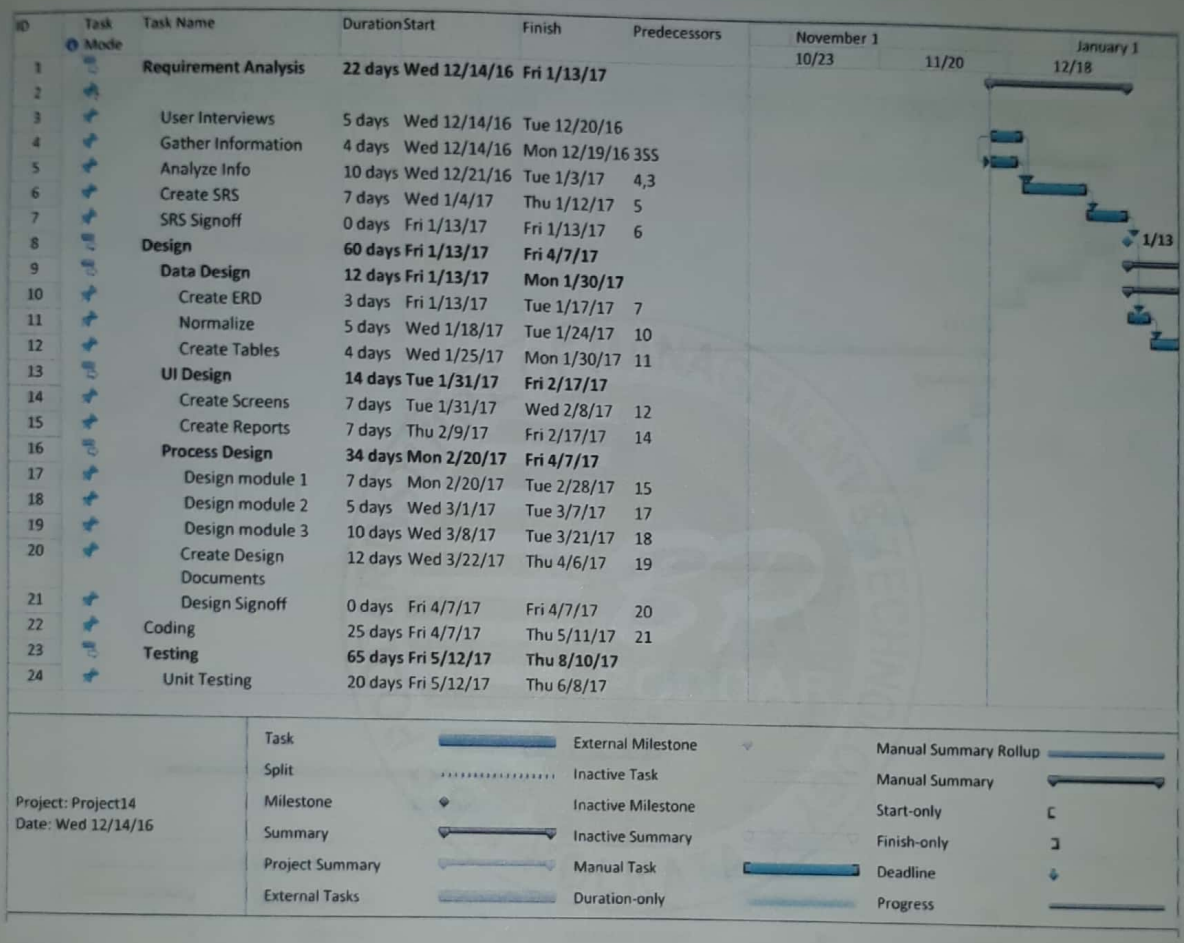
1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO5, PO11 & PO12 of Program Outcomes.

Problem Statement

Task No	Task Name	Duration (days)	Predecessor
1.	**Requirement Analysis**	**26**	
2.	User Interviews	5	
3.	Gather Information	4	2, Start-to-Start
4.	Analyze Info	10	2,3
5.	Create SRS	7	4
6.	SRS signoff	0	5
7.	**Design**	**60**	
8.	**Data Design**	**12**	
9.	Create ERD	3	6
10.	Normalize	5	9
11.	Create Tables	4	10
12.	**UI design**	**14**	
13.	Create Screens	7	11
14.	Create Reports	7	13
15.	**Process Design**	**34**	
16.	Design Module 1	7	14
17.	Design Module 2	5	16
18	Design Module 3	10	17
19.	Create design Document	12	18
20.	Design signoff	0	19
21.	**Coding**	**25**	**20**
22.	**Testing**	**65**	
23.	Unit testing	20	21
24.	Integration testing	15	23
25.	System Testing	15	24
26	Acceptance testing	15	25
27.	**Implementation & Signoff**	**5**	**26**

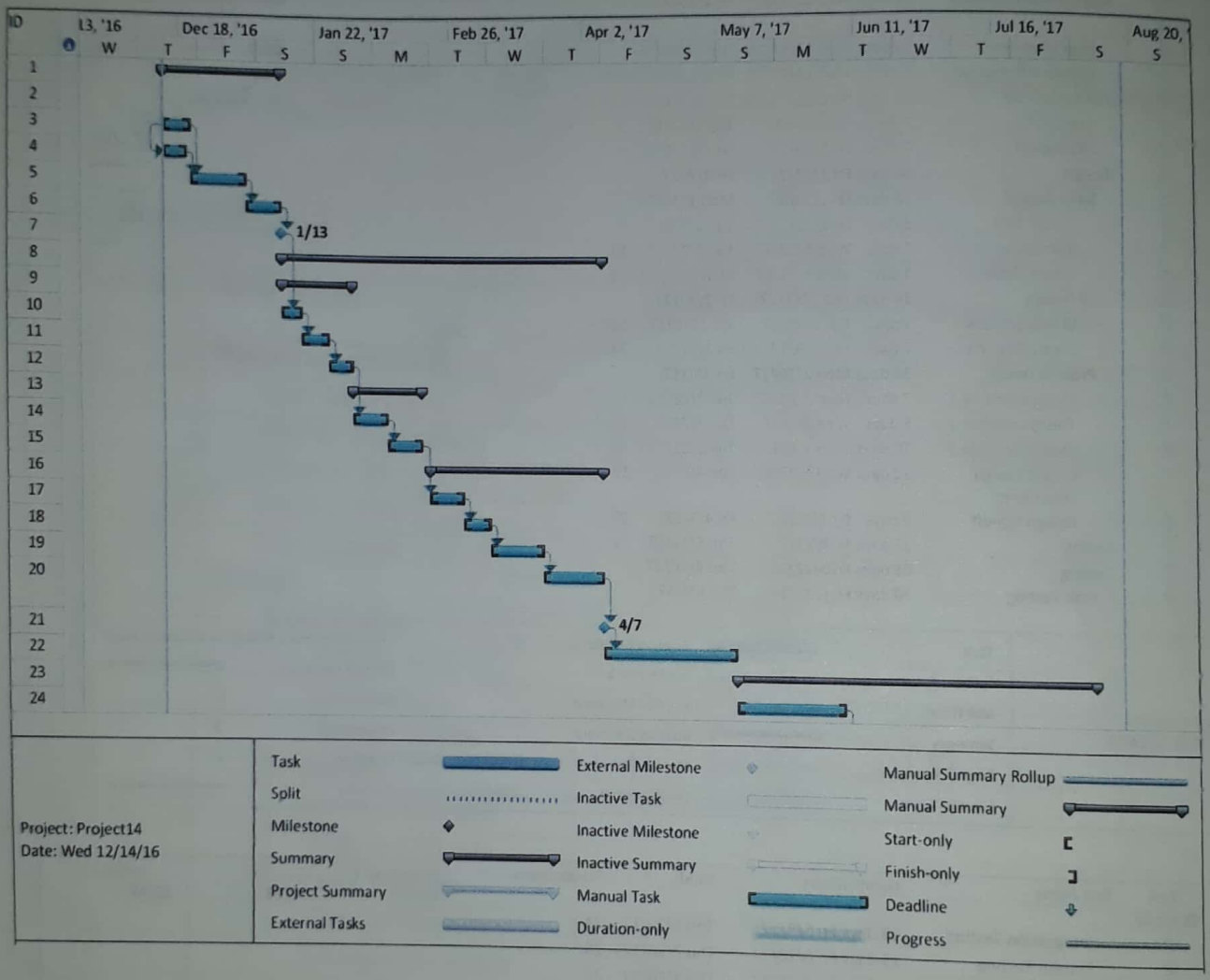
Design/Methodology/Algorithm/Data Structure

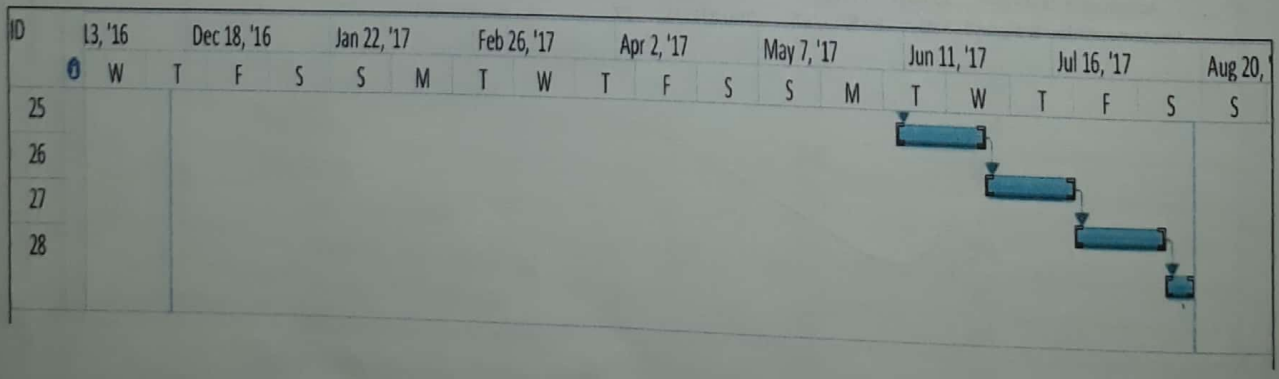
ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
1		**Requirement Analysis**	**22 days**	**Wed 12/14/16**	**Fri 1/13/17**	
2						
3		User Interviews	5 days	Wed 12/14/16	Tue 12/20/16	
4		Gather Information	4 days	Wed 12/14/16	Mon 12/19/16	3SS
5		Analyze Info	10 days	Wed 12/21/16	Tue 1/3/17	4,3
6		Create SRS	7 days	Wed 1/4/17	Thu 1/12/17	5
7		SRS Signoff	0 days	Fri 1/13/17	Fri 1/13/17	6
8		**Design**	**60 days**	**Fri 1/13/17**	**Fri 4/7/17**	
9		**Data Design**	**12 days**	**Fri 1/13/17**	**Mon 1/30/17**	
10		Create ERD	3 days	Fri 1/13/17	Tue 1/17/17	7
11		Normalize	5 days	Wed 1/18/17	Tue 1/24/17	10
12		Create Tables	4 days	Wed 1/25/17	Mon 1/30/17	11
13		**UI Design**	**14 days**	**Tue 1/31/17**	**Fri 2/17/17**	
14		Create Screens	7 days	Tue 1/31/17	Wed 2/8/17	12
15		Create Reports	7 days	Thu 2/9/17	Fri 2/17/17	14
16		**Process Design**	**34 days**	**Mon 2/20/17**	**Fri 4/7/17**	
17		Design module 1	7 days	Mon 2/20/17	Tue 2/28/17	15
18		Design module 2	5 days	Wed 3/1/17	Tue 3/7/17	17
19		Design module 3	10 days	Wed 3/8/17	Tue 3/21/17	18
20		Create Design Documents	12 days	Wed 3/22/17	Thu 4/6/17	19
21		Design Signoff	0 days	Fri 4/7/17	Fri 4/7/17	20
22		Coding	25 days	Fri 4/7/17	Thu 5/11/17	21
23		**Testing**	**65 days**	**Fri 5/12/17**	**Thu 8/10/17**	
24		Unit Testing	20 days	Fri 5/12/17	Thu 6/8/17	



ID	Task Mode	Task Name	Duration	Start	Finish	Predecessors
25		Integration Testing	15 days	Fri 6/9/17	Thu 6/29/17	24
26		System Testing	15 days	Fri 6/30/17	Thu 7/20/17	25
27		Acceptance Testing	15 days	Fri 7/21/17	Thu 8/10/17	26
28		Implementation & Signoff	5 days	Fri 8/11/17	Thu 8/17/17	27

Outputs/Results





Discussion/Conclusion

Difference between Gnatt & PERT chart

Gnatt chart	Pert chart
① Represent with Barchart	① Represent with Flow chart
② Used for small projects	② Used for large and complex projects.
③ Provide Accurate time duration & Percent complete	③ Need to Predict the time.
④ cannot display inter-connecting Tasks that depend on each other.	④ Has numerous inter-connecting networks of Independent Tasks.

Questionnaire

1) What is the significance of Gantt chart in project scheduling?

→ Gnatt charts are useful for planning and scheduling projects. They help us across how-long a project should take, determine the resources needed, and plan the order in which we will complete tasks. They are also helpful for managing the dependencies between tasks.

Grade awarded: 10
Teacher's signature with date 20/9/18

Assignment No.:7

Topic: Software estimation using Function Point metrics

Objective: To develop knowledge about estimation of effort and time in software projects using Function Point approach (FP). Reference website – www.ifpug.org

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4 & PO12 of Program Outcomes.

Problem Statement

A target software system is estimated to have 6 simple, 8 average and 9 complex inputs. In addition, we also expect the system to have 17 average outputs, 51 average enquiries, 10 average master files, which must be updated on-line & 22 average interfaces.

On an overall basis, we may say that the inputs, outputs files or enquiries have average complexity. In addition the following information is also given:—

The system requires significally reliable backup and recovery. Data communications are essential and the system S/W has a significant proportion of distributed processing fn. Hence internal ~~passing~~ processing of the S/W is expected to be significantly complex. The system performance is also significantly critical for smooth and efficient user operation. Finally it is essential that system is designed for multiple installations in different organizations. Therefore, the S/W code must essentially be reusable Assume zero influence for all other factors not mentioned.

Based on the above system description, compute the adjusted function points (FP) for the system.

Design/Methodology/Algorithm/Data Structure

	Simple (S)	Factor (Fs)	Average (a)	Factor (Fa)	Complex (Cc)	Factor (Fc)	Raw total S*Fs+a*Fa+c*Fc
No. of Inputs	6	3	3	4	9	6	84
No. of outputs	0	4	17	5	0	7	85
No. of Enquiries	0	3	51	4	0	6	204
No. of Files	0	7	10	10	0	15	100
No. of Interfaces	0	5	22	7	0	10	154
					UFP = (row wise sum of Row Total)		627

List of adjustment factors (Fi where i = 1 to 14) as stated below. All values should be between 1 to 5 (least significant to most significant)

General system characteristics		Brief Descriptions	value for the stated problem.
1	Data communications (F1)	How many communication facilities are there to aid in the transfer or exchange of information with the application/system?	3
2.	Distinguished data processing (F2)	How are distributed data and processing function handled?	2
3.	Performance (F3)	Was response time or throughput required by the user?	5
4	Heavily used configuration (F4)	How heavily used is the current hardware platform where the application will be executed?	1
5	Transaction rate (F5)	How frequently are transactions executed daily, weekly, monthly etc.	5
6	On-line data entry (F6)	What percentage of the information is entered online?	3
7	End-user efficiency (F7)	Was the application designed for end-user efficiency?	4
8	On-line update (F8)	How many ILF's are updated by on-line transactions	3

Outputs/Results

	General System characteristics	Brief Description	Value for the stated problem
9	Complex Processing (F9)	Does the application have extensive logical or mathematical processing?	1
10	Reusability (F10)	Was the application developed to meet one or many users needs?	4
11	Installation ease (F11)	How difficult is conversion & installation?	5
12	Operational ease (F12)	How effective and for automated are start-up, back-up and recovery procedures?	4
13	Multiple sites (F13)	Was the application specially designed, developed and supported to be installed at multiple sites for multiple organizations?	1
14	Facilitate change (F14)	Was the application specifically designed, developed and supported to facilitate change?	3

Value Adjustment Factor (VAF) = 0.65 + 0.01 * 44

= 1.09

Adjusted Function Point FP = UFP * VAF

= 627 * 1.09 = 679.07

Discussion/Conclusion

Advantages

1. It can be ~~easily~~ early in the software development life cycle.
2. It is independent of the programming language, technology, techniques.
3. It provides reliable relationships to effort.
4. Creation of more functions point can define productivity goal as opposed to LOC
5. Productivity of projects written in different languages can be measured.
6. They can be counted early and after.
7. They can be used for GUI systems.
8. It considers environment factors.

Disadvantages:

1. It needs subjective evaluations with a lot of judgement involved.
2. Many efforts and cost models are based on LOC, so function points as compared to LOC.
3. Less research data is available on function points as compared to LOC.
4. It is performed after creation of design specifications.
5. It has low accuracy of evaluating as a subjective judgement is involved.
6. As the learning curve is quite long its not easy to gain proficiency.
7. It is a time consuming method.

Questionnaire

Q. Is the estimate of effort (in hours) from the size (in function points) is related only to construction activities or the whole software project?

→ The main questions that a software estimating process tries to answer are related to the time required to complete a project & the cost of its development. The answers can only be considered reliable if all the activities involved in the software production are estimated as the factors that affect requirements definitions to product deployment and testing.

Function Point Analysis is a method used to identify one of these factors, the size of the software. In a process of software estimating, the size is the first greatness to be analyzed. Without it, the other estimates (such as effort, term, cost and no. of defects) cannot be determined without having added a high degree of subjectivity to the results.

Based on the amount of function point is possible to obtain indicators such as delivery rate (H/FP), cost (US$/FP) and quality indicators (Defects/FP) of projects already undertaken. This information could be extrapolated for application in estimate processes of new projects.

To simplify if we know "how much" to be produced we can extrapolate the effort, time and cost involved in this work will be.

Grade awarded: ⓒ
Teacher's signature with date 5/10/18

Assignment No.:8

Topic: Software estimation using COCOMO

Objective: To develop knowledge about estimation of effort and time in software projects using Comprehensive Cost Model (COCOMO).

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4 & PO12 of Program Outcomes.

Problem Statement

Estimate the effort put in and development time of the following projects.

a) Project P1 is an organic mode project delivering 50,000 lines of codes.

b) Project P2 is a semi detached mode project delivering 95000 lines of code.

c) Project P3 is an embedded mode project delivering 300,000 lines of code.

cocomo (constructive cost model) has been designed by 1981 by Barry Boehm to given an estimate of the number of man-months it will take to develop a software product and it is referred as cocomo 81.

The basic cocomo equations for estimation are—

$$E = a\,KLOC^b$$

$$T = cE^d$$

where E = Effort, T = Time, a, b, c, d = constants, KLOC = Kilolines of code.

The values of a, b, c, d are given in table.

Software Project	a	b	c	d
Organic	2.4	1.05	2.5	0.38
Semi-detached	3.0	1.12	2.5	0.35
Embedded	3.6	1.20	2.5	0.32

- Organic is a type of software Project (Here P1) in which size of development team is small and team members are experienced.
- Project P2 is semi-detached in which the development team consists of a mixture of experienced & inexperienced staff.
- Project P3 is embedded in which the software being developed is strongly coupled to complex hardware.

Outputs/Results

Project P1 (Organic):-

$E = a\,KLOC^b = 2.4(50)^{1.05} = 145.9 \simeq 146$ PM.

$T = c.E^d = 2.5(146)^{0.38} = 16.61$ months.

Project P2 (semi-detached):-

$E = a\,KLOC^b = 3(95)^{1.12} = 492.23 \simeq 492$ PM.

$T = cE^d = 2.5(492)^{0.35} = 21.88$ months.

Project P3 (Embedded):-

$E = a\,KLOC^b = 3.6 \times (300)^{1.20} = 3379.4 \simeq 3379$ PM

$T = cE^d = 2.5(3379)^{0.39} = 33.66$ months.

Discussion/Conclusion

Advantages

① cocomo is factorial and easy to interpret
② Accounts for various factors that affect cost of the project.
③ works on historical data and hence is more predictable
④ The drivers are very helpful to understand the impact on the different factors that affect the project costs.

Disadvantages

① cocomo model ignores requirements and all documentation.
② It ignores customer skills, co-operation, knowledge and other parameters.
③ It oversimplifies the impact of ~~study~~ safety/security aspects.
④ It ignores hardware issues.
⑤ It ignores personal turnover levels.
⑥ It is dependent on the amount of the time spent in each phase.

Questionnaire

1) How did cocomo handle the disadvantages in their later versions?

→ A new model called cocomo 2 was designed in 1998 and the need for this model came up as software development technology moved from mainframe and overnight batch processing to desktop development code reusability and the use of off-the-self software components.

cocomo consists of a hierarchy of 3 increasingly detailed and accurate forms:

1) Basic cocomo: is a static, single valued model that computes software development effort and cost as a function of program size expressed in estimated lines of code.

2) Intermediate cocomo: computes software development effort as function of program size and a set of "cost drivers" that include subjective assessment of product, hardware, personnel and project attributes.

3) Detailed cocomo: Incorporates all characteristics of the intermediate version with an assessment of the cost drivers impact on each step in SDLC.

Grade awarded: 10

Teacher's signature with date 5/10/18

45

Assignment No.:9

Topic: Unit Testing using White box approach

Objective: To get familiar with identifying test cases and test paths based on McCabe's Control Flow Graph (CFG) and significance of cyclomatic complexity

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4 & PO12 of Program Outcomes.

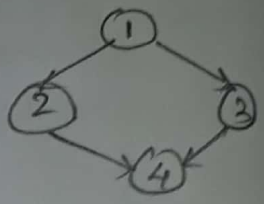
Problem Statement

write a c code for sorting a 500 integers. Use any sorting algorithm or any language. Calculate the number of test cases and identify test paths using mccabe's cyclomatic complexity.

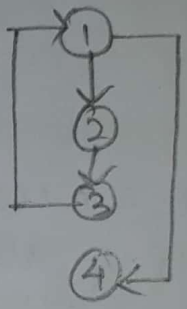
A control flow graph describes the sequence in which the different instructions of a program get executed.

eg:- Sequence :- 1. a=5;
2. b=a*2-1

Selection : 1. if (a>b)
2. c=3;
3. else c=5;
4. c=c*c;



Iteration: 1. while (a>b) {
2. b=b-1;
3. b=b*a; }
4. c=a+b;



Mc Cabe's cyclomatic complexity metric :- This defines an upper bound on the number of independent paths in a program.

Three different ways to compute the cyclomatic complexity are:-

method 1:

Given a CFG (G) of a program, the cyclomatic complexity

$V(G) = E - N + 2$.

where, N = No. of nodes of the CFG.
E = No. of edges of the CFG.

method 2:

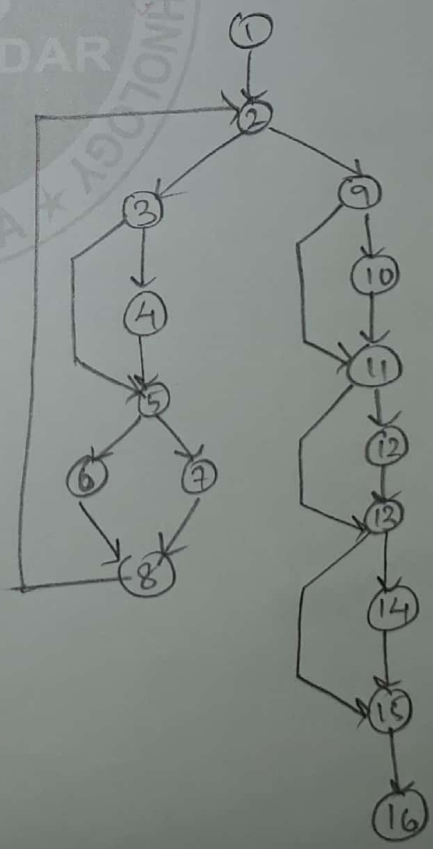
V(G) = Total no. of non-overlapping bounded areas + 1.

method 3:

If N = no. of decision and loop statements of a program.

then Mc Cabe's metric = N+1

```
retcode getList(char *lin, int *i, retCode *status)
/*1*/ {
/*1*/ int num, done;
/*1*/ int2=0;
/*1*/ int1=0;
/*1*/ done=(getone(lin,i,&num,status)!=OK);
/*2*/ while(!done)
/*3*/ {
/*3*/ int1=int2;
/*3*/ int2=num;
/*3*/ ints++
/*3*/ if(lin(+i)==SEMICOL)
/*4*/ curin=num;
/*5*/ if((lin[+i]==COMMA)||(lin[+i]==SEMICOL))
/*6*/ {
/*6*/ +i=+i+1;
/*6*/ done=(getdone(lin,i,&num,status)!=OK);
/*6*/ }
/*7*/ else
/*7*/ done=1
/*8*/ }
/*9*/ int=min(ints,2);
/*9*/ if(ints==0)
/*10*/ int2=curin;
/*11*/ if(ints<=1)
/*12*/ int1=int2;
/*13*/ if(*status!=ERR)
/*14*/ *status=OK;
/*15*/ return(*status);
/*16*/ }
```



McCabe's cyclomatic complexity V(G) is given by any of the following equations:—

V(G) = E - N + 2 (where E = no. of edges, N = No. of nodes in the CFG — (1)

= R + 1 (R = no. of bounded / closed regions in CFG — (2)

= P + 1 (P = no. of predicate nodes i.e, node having out degree > 1) — (3)

For the above example, V(G) = 21 - 16 + 2 = 7 (by equation 1)

= 6 + 1 = 7 (by equation 2 & 3).

List of Test Paths / Independent paths

Path#1: 1-2-9-10-11-12-13-14-15-16.

Path#2: 1-2-9-11-12-13-14-15-16.

Path#3: 1-2-9-10-11-13-14-15-16.

Path#4: 1-2-9-11-12-13-15-16

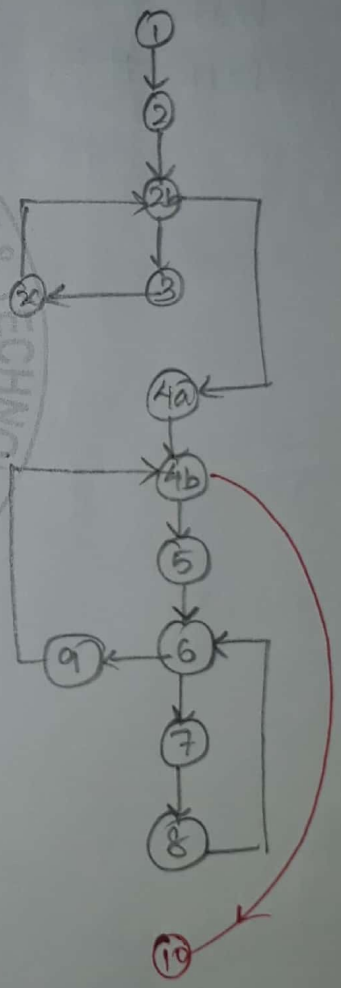
Path#5: 1-2-3-5-6-8-9-11-13-15-16.

Path#6: 1-2-3-5-7-8-9-11-13-15-16.

Path#7: 1-2-3-4-5-6-8-2-9-11-13-15-16.

Outputs/Results

```
insertion(int a[], int p[], int N)
{
(1) int i, j, k;
(2) for ((2a) i=0; (2b) i<=N; (2c) i++)
(3)   p[i] = i;
(4) for ((4a) i=2; (4b) i<=N; (4c) i++)
    {
(5)   k = p[i]; j=1;
(6)   while (a[p[j-1]] > a[k]) {
(7)     p[j] = p[j-1];
(8)     j--
    }
(9)   p[j] = k;
10  }
```



Discussion/Conclusion

$V(G) = E - N + 2 = 14 - 12 + 2 = 4$.

$= R + 1 = 3 + 1 = 4$.

$= P + 1 = 3 + 1 = 4$.

Independent Path

Path #1: 1-2a-2b-4a-4b-5-6-7-8

Path #2: 1-2a-2b-4a-4b-5-6-9-4b-5-6-7-8

Path #3: 1-2a-3b-3-2c-2b-4a-4b-5-6-7-8.

Questionnaire

what is CFG? what is the significance of McCabe's cyclomatic complexity in unit Testing?

→ CFG (control flow graph) is a graph which describes the sequence in which the different instructions of a program get executed.

Significance of McCabe's cyclomatic complexity in unit testing are:-

1. Limiting complexity during development
2. Measuring the "structuredness" of a program
3. Implications for s/w testing.
4. Correlation to number of defects
5. Estimation of structural complexity of code.
6. Estimation of testing effort.
7. Estimation of program reliability.

Grade awarded:	E
Teacher's signature with date	12/10/18

Assignment No.:10

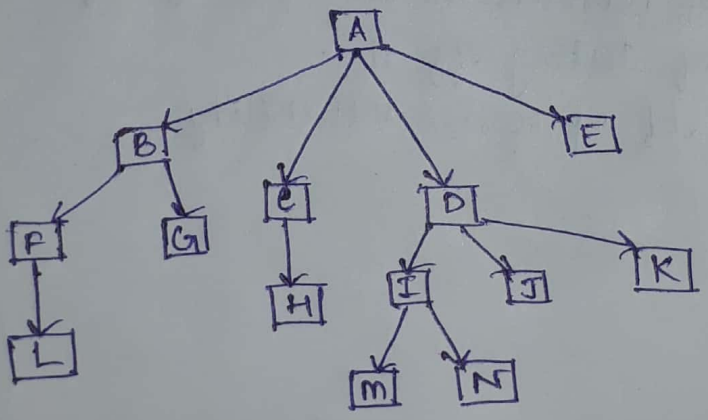
Topic: Integration Testing based on structured chart

Objective: To get familiar different types of integration test techniques and display the integration test paths using a structured chart.

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve PO1, PO2, PO3, PO4 & PO12 of Program Outcomes.

Problem Statement

The structured chart below illustrates the hierarchy of models in a software system. Design order of execution of modules for different types of technique of integration testing.



4953

① Bigbang integration testing: In this approach, all the modules making up a system are integrated in a single step. All the modules of the system are simply linked together and tested

② Bottom-up integration testing: First the modules for each subsystem are integrated. Thus, the subsystems can be integrated separately & independently.

③ Top-down integration testing: This starts with the root module and one or two subordinate modules in the system. This approach requires the use of program stubs to simulate the effect of lower level routines that are called by the routines under test. A pure top-down integration does not require any driver routines.

④ Mixed integration testing: It follows a combination of top-down and bottom-up testing approaches. Testing can start as and when modules become available after unit testing. In this approach both stubs and drivers are required to be designed.

Outputs/Results

Bottom Up → L-F-G-B-H-C-M-N-I-J-K-D-E-A.

TOP-DOWN → A-B-F-L-G-C-H-D-I-M-N-J-K-E.

Big Bang → A-B-C-D-E-F-L-G-H-M-N-I-J-K.

Mixed-Integration → L-F-G-B-H-C-A-D-I-J-K-M-N-E.

Discussion/Conclusion

Integration testing is performed to check the following points:

- To check whether the modules developed by individual developers when combined are according to standards and gives the expected results.
- When modules are combined, sometimes the data travelling between modules has many errors which may not give the expected results. So integration testing is performed to find the defects or bugs in all interfaces.
- To check the integration between any third party tool used.

Questionnaire

What is the difference between integration testing and System testing?

System Testing	Integration Testing
Testing the completed product to check if it meets the specification requirements	Testing the collection and interface modules to check whether they give the expected result.
Both functional & non-functional testing are covered like sanity usability, performance, stress on bad.	Only functional testing is performed to check whether the two modules when combined give correct outcome.
It is a high level testing performed after integration testing	It is a low level testing performed after Unit Testing.
It is a black box testing technique so no knowledge of internal structure or code is required.	It is both black box and white box testing approach so it requires the knowledge of the two modules & the interface.
It is performed by test engineers only.	It is performed by developers as well as test engineers.
Here the testing is performed on the system as a whole including all the external interfaces.	Here the testing is performed on the interface between individual module thus any defect found is only for individual modules.

Grade awarded: 10

Teacher's signature with date 2/11/18

Assignment No.:11

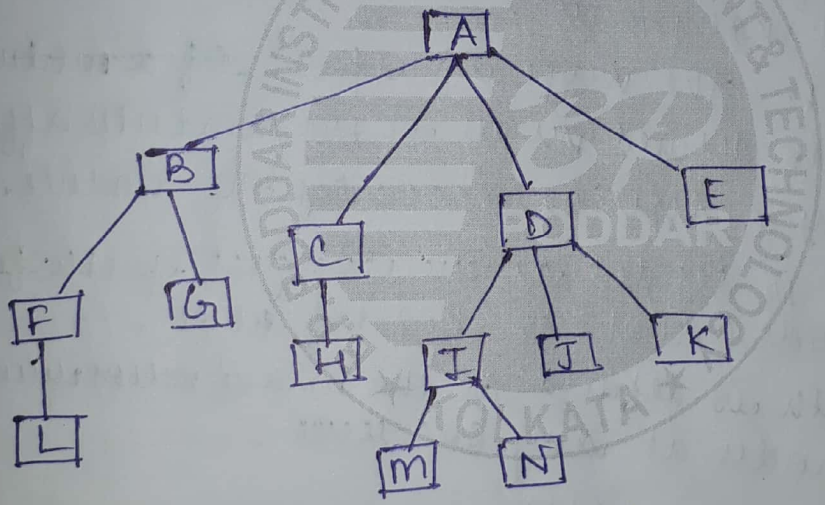
Topic: Beyond Syllabus

Objective: Architectural Metrics based on Structured chart.

1. Satisfy interpret, construct, actualize of Bloom's Taxonomy
2. Achieve _____ of Program Outcomes.

Problem Statement

The structured chart below shows the modules of a software system.



Calculate the following metrics for each node based on the structured chart shown above —

- Fan-in
- Fan-out
- Span of control
- Depth & width of the overall system.

Fan-in :- Fan-in metrics are structural metrics which measure the number of modules that call.

Fan-out :- Fan-out metrics are structured metrics which measure the number of modules that called by a given module.

Scope of control :- Naturally the project management has to manage scope changes to. The world is a collection of changes. But they must be integrated into the existing project scope statement by referring to a defined change process.

Scope of Effect :- The scope of effect of a module is defined as all of the modules affected by a decision made by the module.

Depth :- Depth is defined as the longest path from the top node (root) to the leaf node.

Width :- Width is defined as the maximum number of nodes at any one level.

Features of gooddesign are :-
1. low coupling 2. High cohesion 3. High Fan-in 4. low Fan-out 5. low scope of effect 6. High depth 7. low width.

Outputs/Results

Module No.	Module Name	Fan-in	Fan-out	Span of control
1	A	0	4	13
2	B	1	2	3
3	C	1	1	1
4	D	1	3	5
5	E	1	0	0
6	F	1	1	1
7	G	1	0	0
8	H	1	0	0
9	I	1	2	2
10	J	1	0	0
11	K	1	0	0
12	L	1	0	0
13	M	1	0	0
14	N	1	0	0

Overall Depth of the system = 4.
Overall width of the system = 8.

Discussion/Conclusion

A structure chart is a graphic depiction of the decomposition of a problem. It is a tool to aid in software design. It is particularly helpful for large problems.

A structure chart illustrates the partitioning of a problem into sub-problems and shows the hierarchical relationships among the parts. A classic "organization chart" for a company is an example of structure chart.

The top of the chart is a box representing the entire problem, the bottom of the chart shows a number of boxes representing the less complicated sub-problems.

A structure chart is not a flowchart. It has nothing to do with the logical sequence of tasks. It does not show the order in which tasks are performed. It does not illustrate an algorithm.

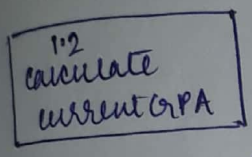
Each block represents some function in the system, and thus should contain a verb phrase, eg: "Print report heading".

Questionnaire

With an example discuss the elements of a structured chart.

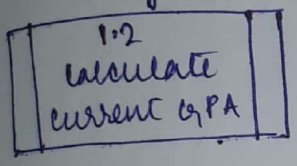
→ Elements used in a structured chart are given below module

module



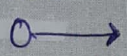
conditional line

Library module



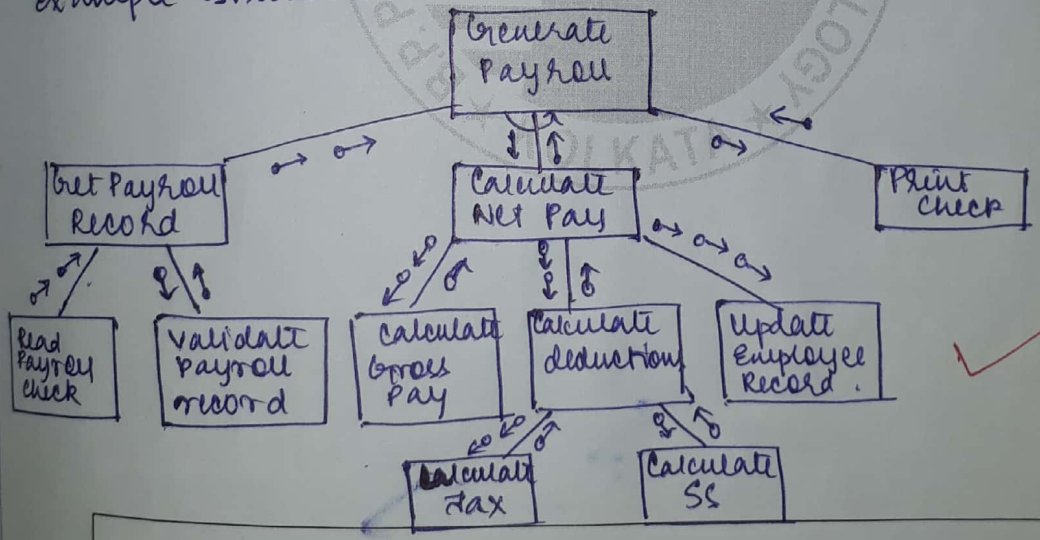
control

Data



Loop

Example structured chart:



Grade awarded: 0

Teacher's signature with date 14/11/18