Prisma : For Professional Team Work

Software Engineering Project

Group-16
Pragya Agrawal(2019BCS-040)
Richa Gupta(2019BCS-047)
Sriya Chettebhaktula(2019BCS-063)
Krishna Gandhi(2019BCS-079)

INDEX

INTRODUCTION

Purpose

The purpose of this document is to present a detailed description of the Team Communication System. It will explain the purpose and features of the system, the interfaces of the system, what the system will do, the constraints under which it must operate and how the system will react to external stimuli. This document is intended for both the stakeholders and the developers of the system and will be proposed to the Regional Historical Society for its approval.

Scope of Project

This software is a project management tool for teams to pull off the work smoothly. Prisma encourages communication between team members, clients through public or private channels or DMs rather than sticking to old methods like mails/calls. It strengthens client relationships to provide top-tier enterprise support and drastically decrease response time. Hence, a platform to integrate with tools like Jenkins, GitHub and more for code reviews and deployment.

SIZE ESTIMATION

Function Point Metric

Step 1: UFP (Unadjusted Function Point) Computation

The unadjusted function points (UFP) is computed as the weighted sum of five characteristics of a product.

Five Characteristics are:

- Inputs
- Outputs
- Inquires
- Files
- Interfaces

The weights associated with the five characteristics are determined empirically by Albrecht through data gathered from many projects.

Inputs

During Sign Up/ Registration

- ❖ User Name
- ❖ Password
- ❖ Mobile Number
- ❖ Email
- ❖ Location

After creating a channel

- ❖ Channel Topic
- ❖ Channel Description
- ❖ Channel Participants

Total Number of Inputs: 7

Outputs

- ❖ Confirmation Message: User created this Channel.
- ❖ Confirmation Message: User A added User B to the Channel.
- ❖ Confirmation Message: User A removed User B from the Channel.
- ❖ Confirmation Message: User left the Channel.

- Confirmation Message: User changed Channel topic/ description.
- Confirmation Message: User deleted the Channel.
- Confirmation Message: Channel/ Contact Archived.
- File downloaded in the device.
- Missed voice call at time.
- Error message: Message not Sent.
- Error message: Could not place call. Users not connected to the internet.
- Error message:File not found.
- Error message: Could not join the channel due to limit exceeding.
- #Download symbol for stickers/ Files.
- #deleted message.

Total Number of Outputs: 15

Inquiries

- Request specific file.
- Request specific User from a Channel.
- Request specific Channel.
- Request to show pinned messages.
- Request user information.
- Request Channel Information

Total Number of Inquiries: 6

Files

- List of Registered Users.
- List of Created Channels.
- List of Message Data Entry.
- List of File Data Entry.
- List of Call Data Entry.

Total Number of Files: 5

Interfaces

- Module for Users.
- Module for Developers.

Total Number of Interfaces: 2

UFP Calculation:

UFP = (Number of inputs)*4 + (Number of outputs)*5 + (Number of inquiries)*4 + (Number of files)*10 + (Number of interfaces)*10

= (8 * 4) + (16 * 5) + (6 * 4) + (5 * 10) + (2 * 10)

=**206**

Step 2: Refined Parameters

Refine UFP to reflect the actual complexities of the different parameters used in UFP computation.

We will garde complexity of each parameter into three broad categories—simple, average, or complex and then compute the UFP using the weighted table given below.

The weights associated with these five characteristics were determined empirically by

Albrecht (Scientist who proposed this method) through data gathered from many projects.

TYPE	Simple	Average	Complex
No. of Inputs	3	4	6
No. of Outputs	4	5	7
No. of Inquiries	3	4	6
No. of Files	7	10	15
No. of Interfaces	5	7	10

Table 1: Weight of Various Entities for Different Complexities (used to calculate refined UFP)

Categorizing the parameters:

INPUTS	CATEGORIES

User Name	Simple
Password	Average
Mobile Number	Simple
Email	Complex
Location	Complex
Channel Topic	Simple
Channel Description	Average
Channel Participants	Simple

OUTPUTS	**CATEGORIES**
Confirmation Messages:	
User created this Channel.	Simple
User A added User B to the Channel.	Simple
User A removed User B from the Channel.	Simple
User left the Channel.	Simple
User deleted the Channel.	Simple
User changed Channel topic/ description.	Simpe
Channel/ Contact Archived.	Simple
File downloaded in the device.	Complex
Missed voice call at time.	Average
Error Messages:	
Message not Sent.	Simple
Could not place a call. Users not connected to the internet.	Average
Could not join the channel due to limit exceeding.	Average

#Download symbol for stickers/ Files.	Simple
#deleted message.	Simple

INQUIRIES	CATEGORIES
Request specific file.	Average
Request specific User from a Channel.	Average
Request specific Channel.	Simple
Request to show pinned messages.	Simple
Request user information.	Simple
Request Channel Information	Average

Files	CATEGORIES
List of Registered Users.	Complex
List of Created Channels.	Average
List of Message Data Entry.	Average
List of File Data Entry.	Average
List of Call Data Entry.	Complex

INTERFACES	CATEGORIES
Module for User	Complex
Module for Developer	Complex

Summary

- **Input:** 4 Simple + 3 Average + 1 Complex
- **Output:** 12 Simple + 3 Average + 1 Complex
- **Inquiries:** 2 Simple + 3 Average
- **Files:** 3 Average +2 Complex
- **Interfaces:** 2 Complex

Refined UFP Calculation:

= (4*3 + 3*4 + 1*6) + (12*4 + 3*5 + 1*7) + (2*3 + 3*4 + 0*6) + (0*7 + 3*10 + 2*15) + (0*5 + 0*7 + 2*10) { From Table 1 }

= 12 + 12 + 6 + 48 + 15 + 7+ 6 + 12 + 30 + 30 + 20

= 198

Step 3: Refine UFP based on complexity of the overall project

14 parameters that can impact the overall project size and help to refine the UFP have been identified as given in the table below.
Each of these 14 parameters is assigned a value from 0 (not present or no influence) to 6 (strong influence).

Function Point Relative Complexity Adjustment Factors	Score
Extent of conversion and installation included in the design	3
Extent of online updating of master files	4
Extent of complex data processing	5
Requirement for reliable backup and recovery	4
Requirement for data communication	3
Extent of distributed processing	3
Performance requirements	5
Expected operational environment	4
Extent of multi-screen or multi-operation online data input	3

Extent of multiple installations in an organisation and variety of customer organisations	2
Extent of complex inputs, outputs, online queries and files	5
Extent that currently developed code can be designed for reuse	6
Extent of change and focus on ease of use	5
Extent of online data entries	6
Degree of Influence	**58**

The resulting numbers are summed, yielding the total degree of influence (DI). A technical complexity factor (TCF) for the project is computed as:

$$\boxed{\textbf{TCF = (0.65 + (0.01* DI))}}$$

= (0.65 + 0.58)

= **1.23**

Finally, FP is given as the product of UFP and TCF:

$$\boxed{\textbf{FP = UFP * TCF}}$$

= 198 * 1.23

= **243.54**

EFFORT AND TIME ESTIMATION

COCOMO

[Constructive Cost Model]

Different models of COCOMO have been proposed to predict the cost estimation at different levels, based on the amount of accuracy and correctness required. Barry Boehm postulated that any software development project can be classified into one of the following three categories based on the development complexity—

- Organic,
- Semi Detached, and
- Embedded.

All of these models can be applied to a variety of projects, whose characteristics determine the value of constant to be used in subsequent calculations.

This project deals with developing a well-understood application program and the size of the development team is reasonably small. Hence, this project belongs to the organic category.

General form of the COCOMO expressions The basic COCOMO model is a single variable heuristic model that gives an approximate estimate of the project parameters. The basic COCOMO estimation model is given by expressions of the following forms:

Effort = a1 × (KLOC)^a2 PM

Tdev = b1 × (Effort)^b2 months

where,

- KLOC is the estimated size of the software product expressed in Kilo Lines Of Code.
- a1, a2, b1, b2 are constants for each category of software product.
- Tdev is the estimated time to develop the software, expressed in months.
- Effort is the total effort required to develop the software product, expressed in person-months (PMs)

Estimated Lines of Code ~ 4800 - 5000

The size of this organic software product has been estimated to be 4,900 lines of source code.

Step 1: Estimation of development effort:

Effort = 2.4 * (KLOC)^1.05

= 2.4 * (4.9)^1.05

= **12.7326 person-months**

Step 2: Estimation of development time:

Tdev = 2.5 * (Effort)^0.38

= 2.5 * (12.7326)^0.38

= **6.57 months**

PROJECT SCHEDULE BREAKDOWN

Activity Network

An activity network shows different activities making up a project ,their estimated durations, and their interdependencies. It is a graphical method for showing dependencies between tasks (activities) in a project.

- The network consists of nodes connected by arcs.
- Nodes denote events and represent the culmination of one or more activities. Arcs represent activities and are labeled with the name of the activity and have an estimated time to complete the activity.

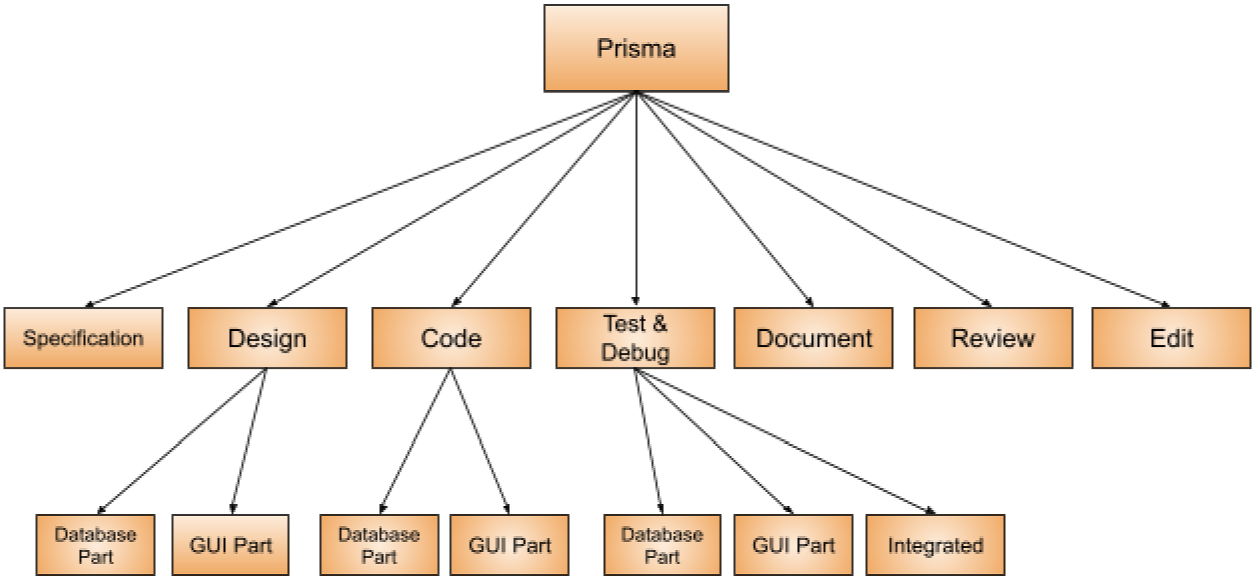
Two equivalent representation for activity networks are possible are in use:

- ❖ Activity On Node
- ❖ Activity On Edge

Step 1: Work Breakdown Structure

Work breakdown structure (WBS) is used to recursively decompose a given set of activities into smaller activities. WBS provides the necessary framework for detailed cost estimating and control along with providing guidance for schedule development and control.

- ❖ The decomposition of the activities is carried out until any of the following is satisfied.
- ❖ A leaf-level sub-activity (a task) requires approximately two weeks to develop.
- ❖ Hidden complexities are exposed, so that the job to be done is understood and can be assigned as a unit of work to one of the developers.
- ❖ Opportunities for reuse of existing software components is identified.



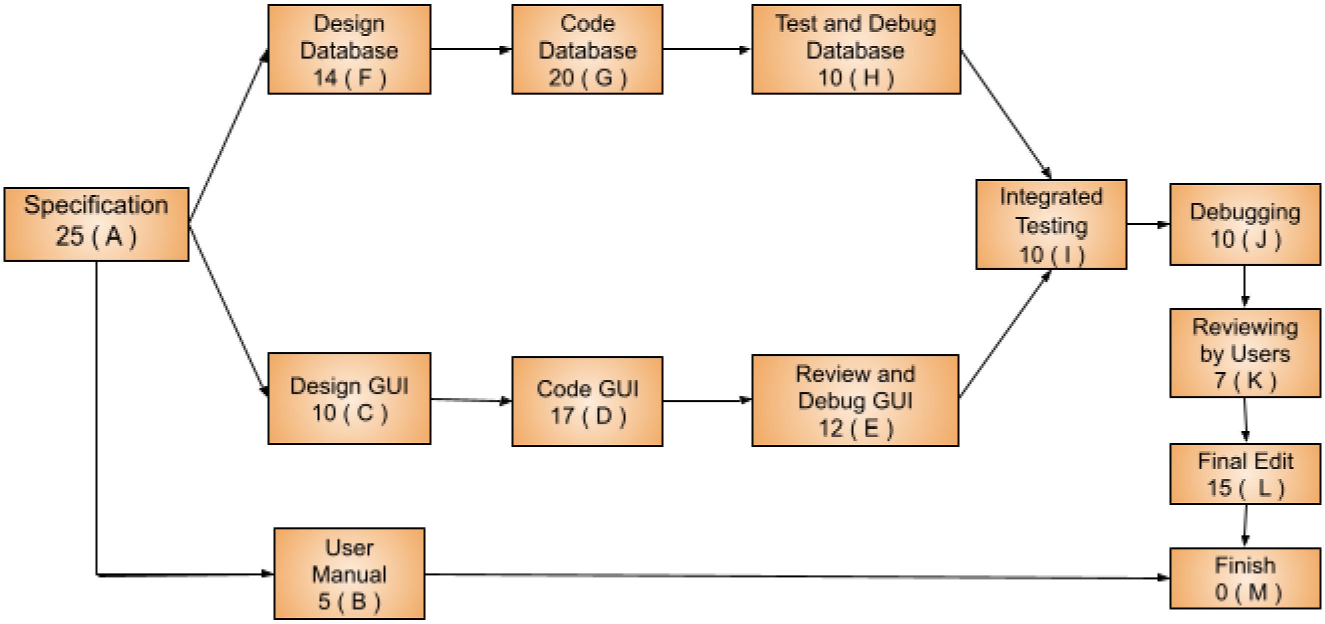
Work Breakdown Structure

Step 2: Activity Network Representation:

Task	Description	Predecessors	Estimated Duration (days)	Slack Time
A	Analyzing Requirement Specifications	-	25	0
B	Documenting Requirements /User Manual	A	5	81
C	Designing GUI (Frontend)	A	10	5
D	Coding GUI (Frontend)	C	17	5
E	Reviewing (By users) and Editing GUI	D	7+5 = 12	5

F	Designing database(Backend)	A	14	0
G	Coding database(Backend)	F	20	0
H	Testing and Debugging Database	G	10	0
I	Integrate And Testing	E,H	10	0
J	Debugging	I	10	0
K	Reviewing product (By Users)	J	7	0
L	Final Editing	K	15	0
M	Finish	B,L	0	0
Total Estimated Duration			**155 days**	

Step 3: Activity (On Node) Network Diagram



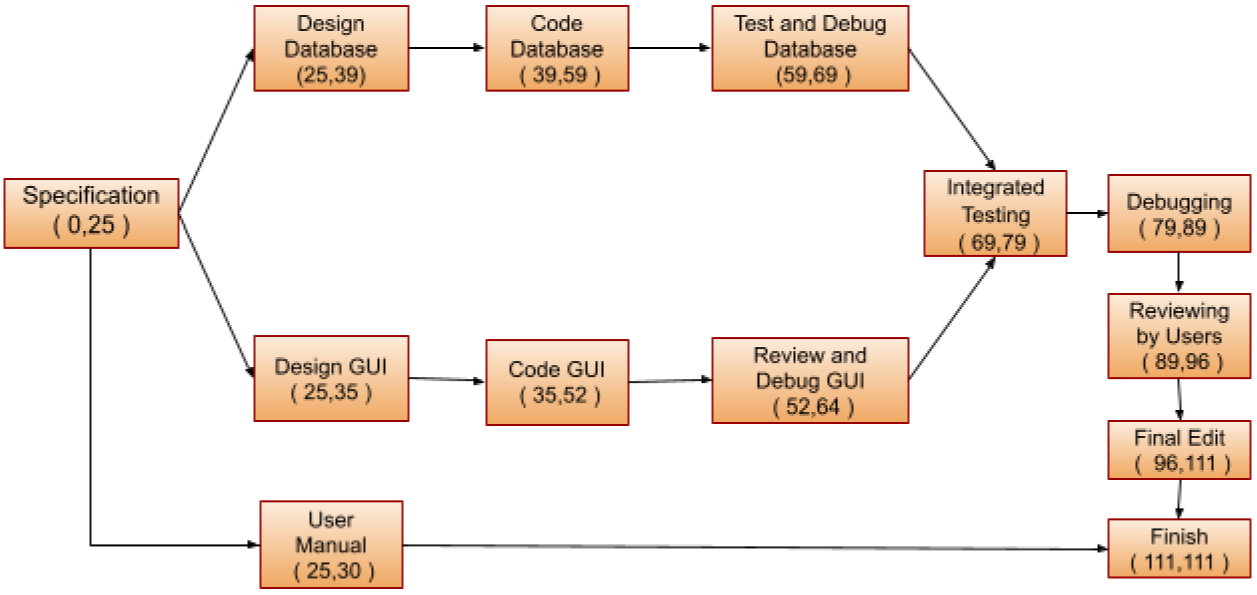
Activity Networks Representation

Critical Path Method

CPM is an algorithmic approach to determine the critical paths and slack times for tasks not on the critical paths. It involves calculating the following quantities:

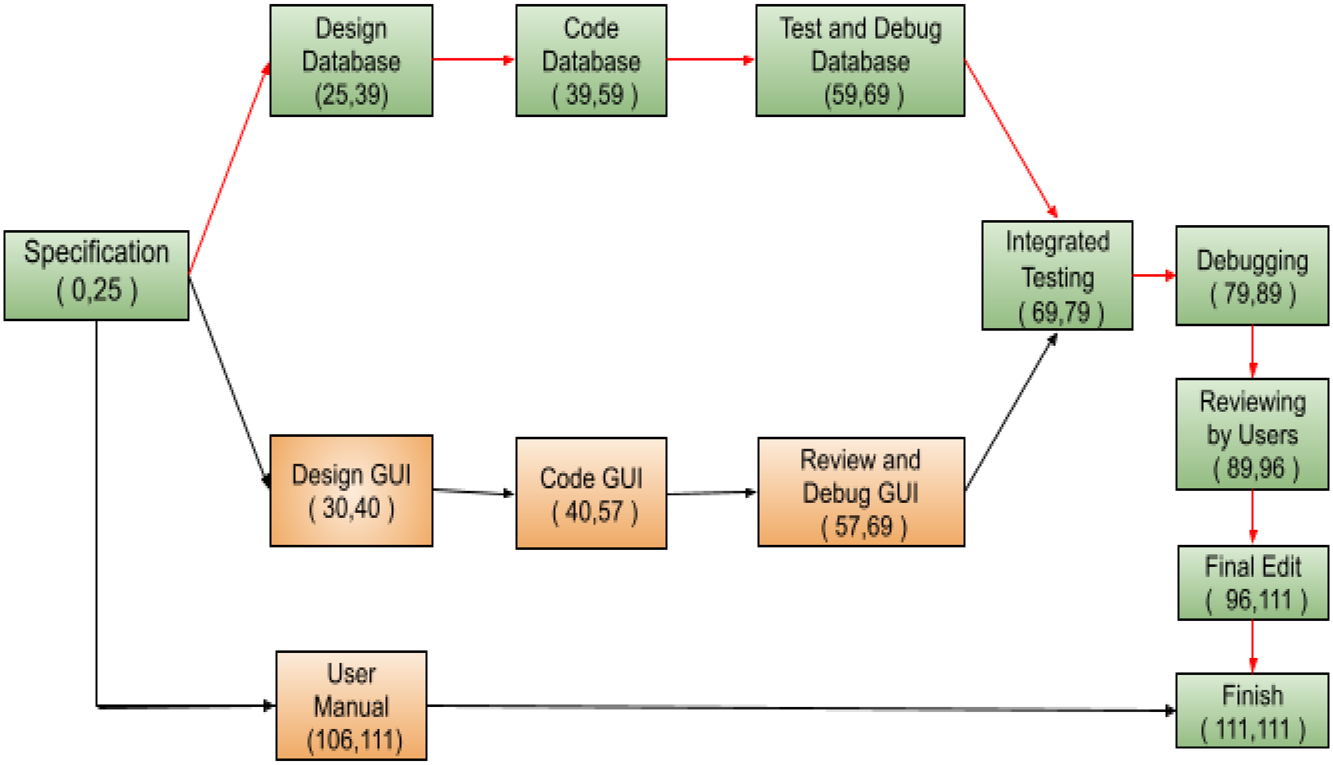
- Minimum time (MT)
- Earliest start (ES)
- Latest start time (LST)
- Earliest finish time (EF)
- Latest finish (LF)
- Slack time (ST)

Critical Path Method Diagram 1 (ES,EF)



Critical Path Method Diagram 1 (ES,EF)

Critical Path Method Diagram 2 (LS,LF)



Critical Path Method Diagram 2 (LS,LF)

Critical Path :

Specification -> Design Database -> Code Database -> Test and Debug Database -> Integrated Testing -> Debugging -> Reviewing -> Final Edit -> Finish

PERT Chart

[Project Evaluation and Review Technique]

PERT is a project management technique, whereby planning, scheduling, organising, coordinating and controlling uncertain activities are done.

PERT charts, like activity networks consist of a network of boxes and arrows. The boxes represent activities and the arrows represent task dependencies.

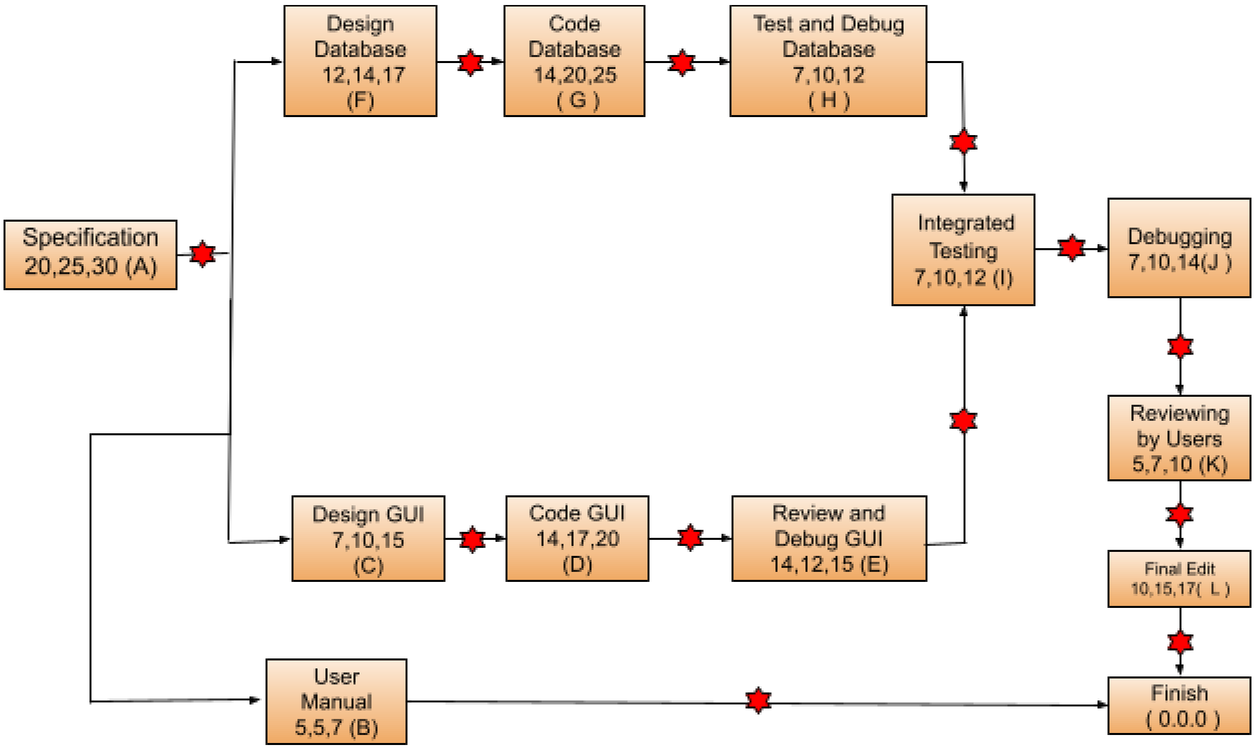
Each task is annotated with three estimates:

- Optimistic (O): The best possible case task completion time.
- Most likely estimate (M): Most likely task completion time.
- Worst case (W): The worst possible case task completion time.

Table: PERT Activity List

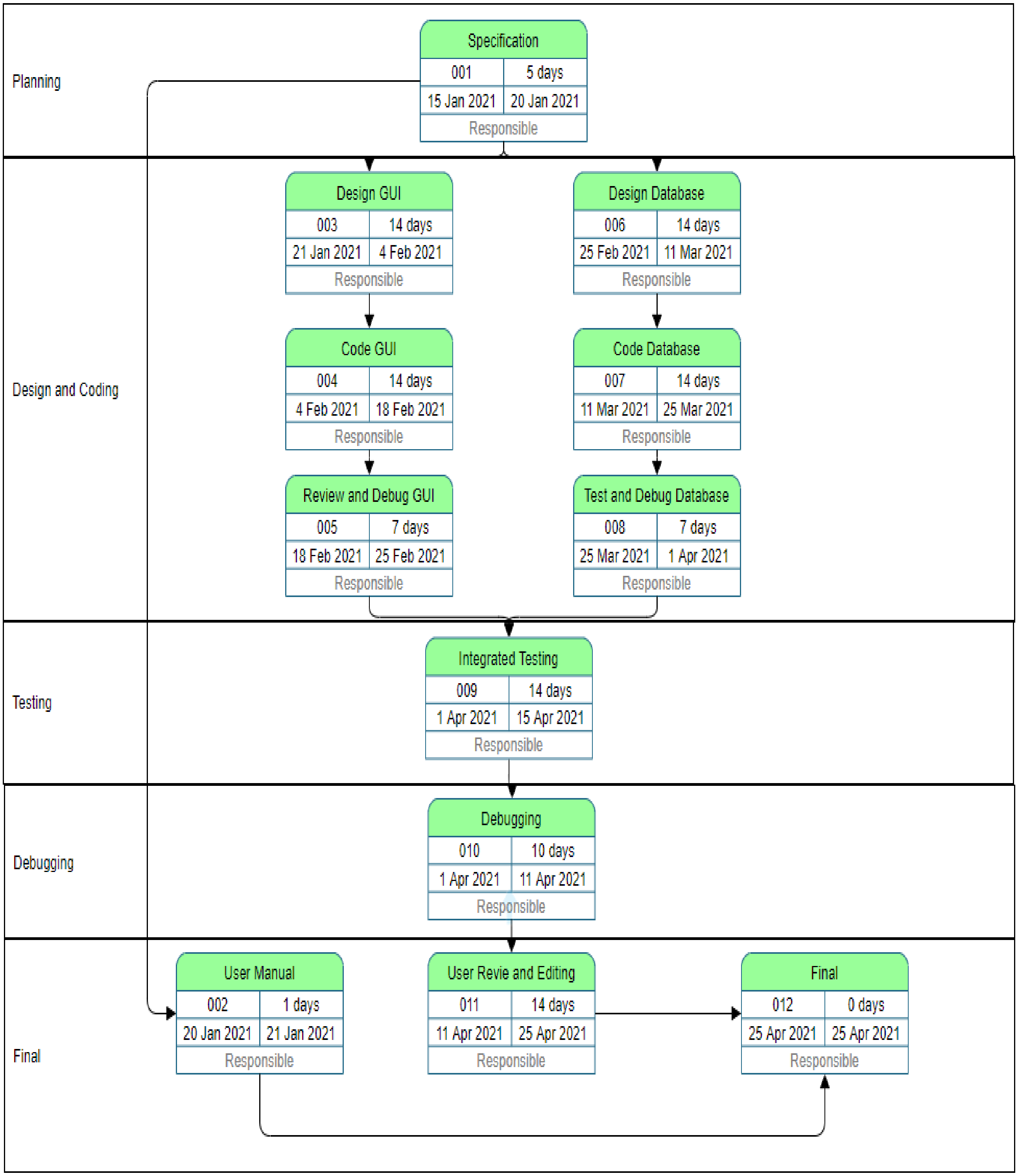
Task	Optimistic(O)	Most Likely Estimate (M)	Worst Case (W)
A	20	25	30
B	5	5	7
C	7	10	15
D	14	17	20
E	14	12	15
F	12	14	17
G	14	20	25
H	7	10	12
I	7	10	12

J	7	10	14
K	5	7	10
L	10	15	17
M	0	0	0
Total	122 days	155 days	194 days


Design Database 12,14,17 (F)
Code Database 14,20,25 (G)
Test and Debug Database 7,10,12 (H)
Specification 20,25,30 (A)
Integrated Testing 7,10,12 (I)
Debugging 7,10,14(J)
Reviewing by Users 5,7,10 (K)
Design GUI 7,10,15 (C)
Code GUI 14,17,20 (D)
Review and Debug GUI 14,12,15 (E)
Final Edit 10,15,17(L)
User Manual 5,5,7 (B)
Finish (0.0.0)

PERT Diagram

PERT CHART


Planning
Specification
001
5 days
15 Jan 2021
20 Jan 2021
Responsible
Design and Coding
Design GUI
003
14 days
21 Jan 2021
4 Feb 2021
Responsible
Design Database
006
14 days
25 Feb 2021
11 Mar 2021
Responsible
Code GUI
004
14 days
4 Feb 2021
18 Feb 2021
Responsible
Code Database
007
14 days
11 Mar 2021
25 Mar 2021
Responsible
Review and Debug GUI
005
7 days
18 Feb 2021
25 Feb 2021
Responsible
Test and Debug Database
008
7 days
25 Mar 2021
1 Apr 2021
Responsible
Testing
Integrated Testing
009
14 days
1 Apr 2021
15 Apr 2021
Responsible
Debugging
Debugging
010
10 days
1 Apr 2021
11 Apr 2021
Responsible
Final
User Manual
002
1 days
20 Jan 2021
21 Jan 2021
Responsible
User Revie and Editing
011
14 days
11 Apr 2021
25 Apr 2021
Responsible
Final
012
0 days
25 Apr 2021
25 Apr 2021
Responsible