ESTIMATION REPORT

MOVIE RECOMMENDATION SYSTEM

Submitted To:

Santosh Singh Rathore Sir

Submitted By:

Donthagani Sai Rohan	2021BCS-025
Mekala Bhavana	2021BCS-041
Nelluri Pavithra Sai Lakshmi	2021BCS-049

Table of Contents

1. Size Estimation

1.1 Function Point Metric

According to function point metrics, the size of a software project is directly related to the range of capabilities it can support. Size will increase as more features are offered. The variety of high-level functions or features that a piece of software offers directly affects how big it is. This assumption is plausible given that it would be more work to implement each feature. The size of the project may be determined using this approach straight from the problem definition, which is very beneficial to project managers during project planning.

Step 1: UFP (Unadjusted Function Point) Computation

INPUTS:

1. The User's Personal Information (During Sign Up)

2. Cellular number and password (During user Login)

Username and password 3. (Admin)

4. Updating fresh movie information.

5. Search terms for movies (Titles,People,Genres) There are a total of 5 inputs.

OUTPUTS:

1. User successfully registered, according to the confirmation message

2. User successfully logged in confirmation message

3. Details about the film, such as actors and genre tags

4. Suggestions based on movies or keyword searches

5. Unable to Successfully Sign Up Error Message

Unsuccessful Login Error Message, number six (for users & Admin)

7. Search Unsuccessful Error Message

8. User Rating Submission Unsuccessful Error Message 8 total outputs were produced.

INQUIRIES:

1. Ask for movie data.

2. History of Searches.

3. Request suggestions for movies Three inquiries in all.

FILES:

1. Movie Database.

2. User Database.

TotalNumberof Files: 2

INTERFACES:

1. User Interface
2. Admin Interface

Total Number of Interfaces: 2

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

UFP = (5*4) + (8*5) + (3*4) + (2*10) +
(2*10)

UFP = 112

Step 2: Refine Parameters

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 Interface	5	7	10

Table: Cost of Various Entities for Different Complexities

INPUTS: 3 Simple + 2 Average

OUTPUTS: 6 Simple + 2 Complex

INQUIRIES: 3 Average

FILES: 2 Average

INTERFACES: 1 Average + 1 Complex

Refined UFP = ((3*3) + (2*4)) + ((6*4) + (2*7)) + (3*4) + (2*10) + ((1*7) (1*10))

= (9+8) + (24+14) + 12 + 20 + (7+10)

=104

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

Function Point Relative Complexity Adjustment Factors	Score
Requirement for reliable backup and recovery	5
Requirement for data communication	3
Extent of distributed processing	1
Performance requirements	4
Expected operational environment	3
Extent of online data entries	4
Extent of multi-screen or multi-operation online data input	4
Extent of online updating of master files	3
Extent of complex inputs, outputs, online queries and files	5

Extent of complex data processing	4
Extent that currently developed code can be designed for reuse	3
Extent of conversion and installation included in the design	2
Extent of multiple installations in an organization and variety of customer organizations	4
Extent of change and focus on ease of use	4
Degree of Influence	49

Technical Complexity Factor (TCF) = 0.65 + 0.01 * DI

= 0.65 + 0.49

= 1.14

Function Point = UFP * TCF

= 104 * 1.14

= 118.56

2. Effort and Time Estimation

2.1 Cost Constructive Model(COCOMO):

Barry W. Boehm created the Constructive Cost Model, a methodology for procedural software cost assessment. A regression model called COCOMO is based on LOC, or the quantity of lines of code. According to Boehm's theory, each software development project may be divided into one of the following three groups based on the complexity of the development process. They are embedded, semi-detached, and organic. This project falls within the COCOMO expressions' general category for organic projects. A single variable heuristic model, the fundamental COCOMO model provides an approximation of the project parameters. Expressions of the following kind provide the fundamental COCOMO estimate model:

Where, Effort = a1 (KLOC) a2, and PM Tdev = b1 (Effort) b2 months

KLOC stands for Kilo Lines of Code, which is the projected size of the software package.

2.2 Estimation of development effort:

= 2.4 * (KLOC)^1.05

= 2.4 * (1.1)^1.05

= 2.653 PM

2.3 Estimation of development time:

= 2.5 * (EFFORT)^0.38

= 2.5 * (2.653)^0.38

= 3.622 months

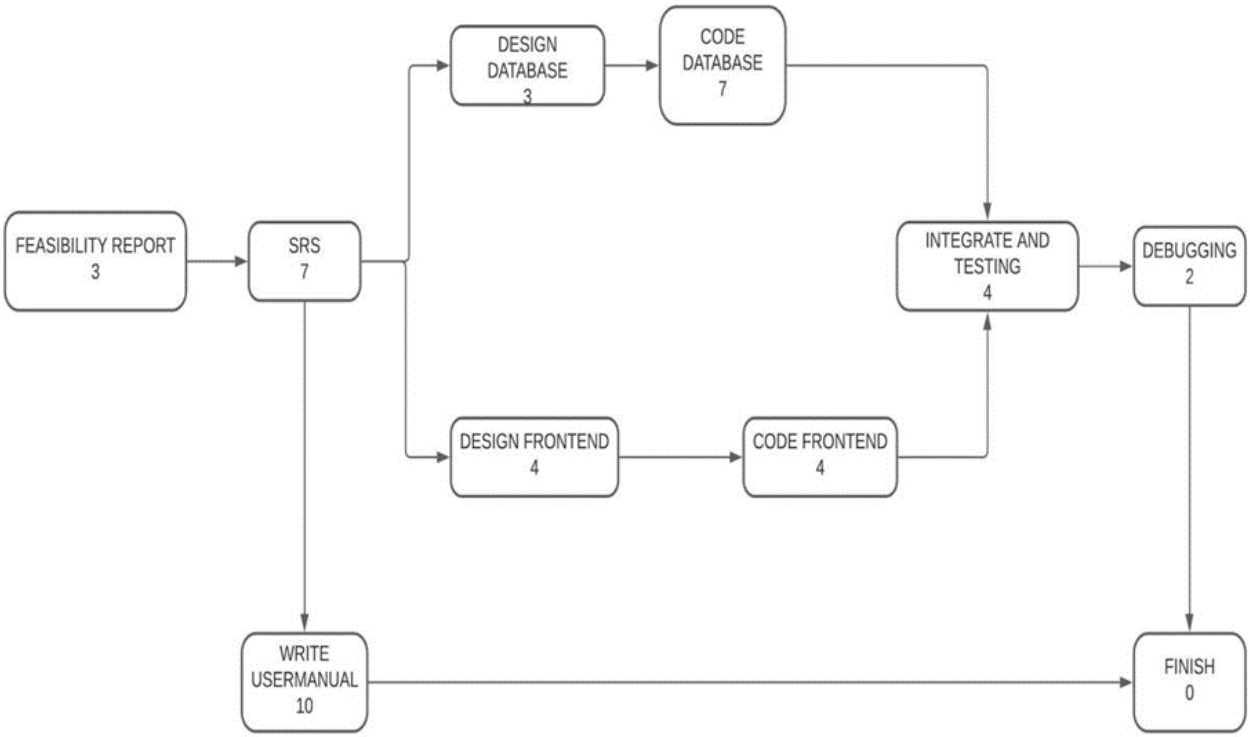
3. PROJECT SCHEDULE BREAKDOWN

3.1 Activity Network

A project's many activities, their expected completion times, and their interdependencies are displayed in an activity network.

We employ an activity-on-node model, in which each activity is represented by a rectangular (some use circular) node, and each task's duration is displayed next to the activity's duration.

Using directed edges, the inter-task relationships are displayed.



Activity On Node

An activity network can only be used to estimate the length of an activity.

Hence, utilizing an activity diagram to estimate the worst case (pessimistic) and best case (optimistic) estimations is not practicable.

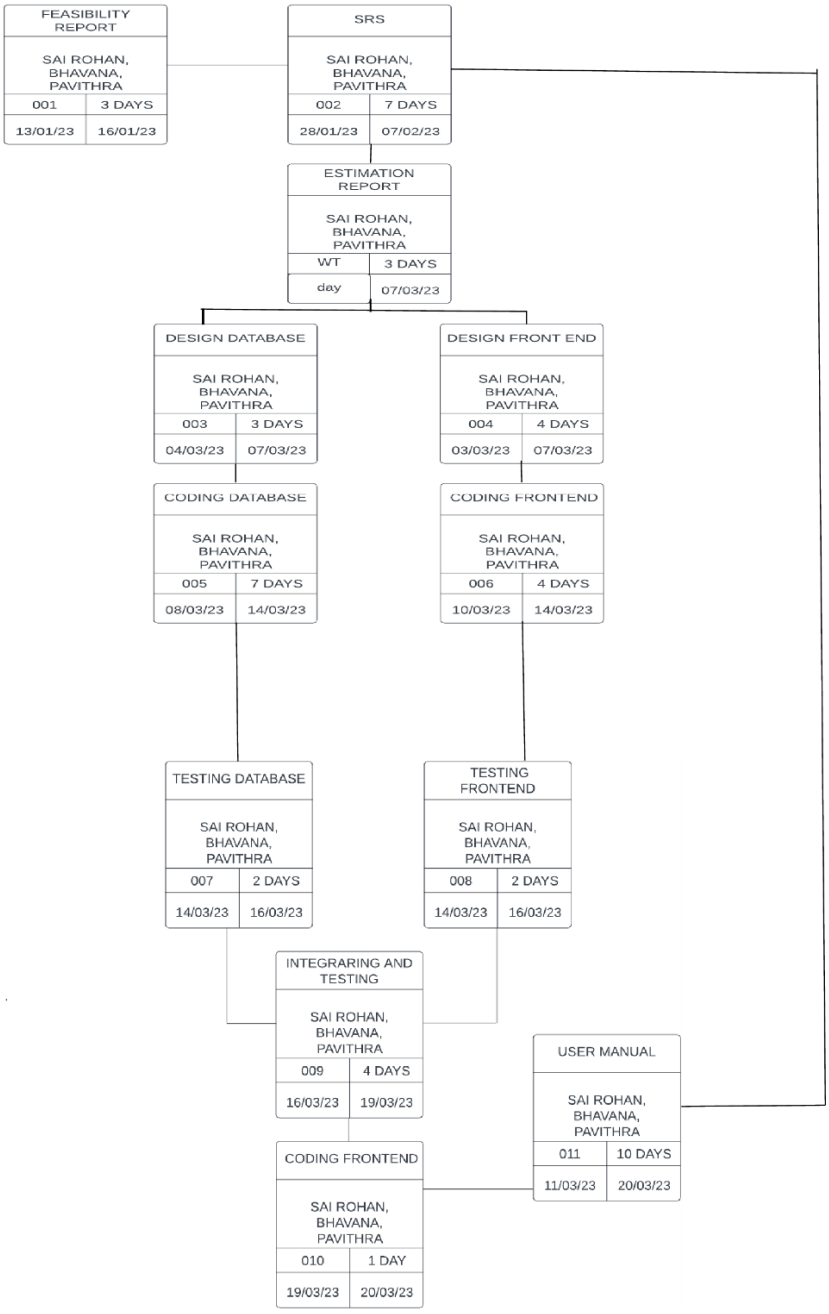
The usefulness of the activity network diagrams is restricted since the real durations could differ from the estimated durations.

3.2 Pert Diagram

A technique called the Program Evaluation and Review Technique (PERT) is used to evaluate the activities in a schedule and choose a different Critical Path Method (CPM). To calculate the bare minimum time needed to finish a project, it examines the length of time needed to do each job and the dependencies that go along with it. Activity charts with a higher level of sophistication are called project evaluation and review technique (PERT) charts.

The statistical fluctuations in the project estimates are shown in a PERT chart under the assumption that they follow a normal distribution. Because PERT allows for some variability in job completion durations, it offers the ability to

calculate the likelihood of hitting project milestones based on the likelihood of finishing each task in the route thereto.
Three estimates are listed next to each task:
Positive (O): The quickest scenario for job completion. Estimate of the task's most likely completion time (M). The task's worst-case completion time is indicated by the letter "W".


FEASIBILITY REPORT
SAI ROHAN, BHAVANA, PAVITHRA
001
3 DAYS
13/01/23
16/01/23
SRS
SAI ROHAN, BHAVANA, PAVITHRA
002
7 DAYS
28/01/23
07/02/23
ESTIMATION REPORT
SAI ROHAN, BHAVANA, PAVITHRA
WT
3 DAYS
day
07/03/23
DESIGN DATABASE
SAI ROHAN, BHAVANA, PAVITHRA
003
3 DAYS
04/03/23
07/03/23
DESIGN FRONT END
SAI ROHAN, BHAVANA, PAVITHRA
004
4 DAYS
03/03/23
07/03/23
CODING DATABASE
SAI ROHAN, BHAVANA, PAVITHRA
005
7 DAYS
08/03/23
14/03/23
CODING FRONTEND
SAI ROHAN, BHAVANA, PAVITHRA
006
4 DAYS
10/03/23
14/03/23
TESTING DATABASE
SAI ROHAN, BHAVANA, PAVITHRA
007
2 DAYS
14/03/23
16/03/23
TESTING FRONTEND
SAI ROHAN, BHAVANA, PAVITHRA
008
2 DAYS
14/03/23
16/03/23
INTEGRARING AND TESTING
SAI ROHAN, BHAVANA, PAVITHRA
009
4 DAYS
16/03/23
19/03/23
USER MANUAL
SAI ROHAN, BHAVANA, PAVITHRA
011
10 DAYS
11/03/23
20/03/23
CODING FRONTEND
SAI ROHAN, BHAVANA, PAVITHRA
010
1 DAY
19/03/23
20/03/23