

SE Experiment 6

Aim: Estimate effort and cost required using FP for project. create WBS and Gantt chart for same.

Theory:

FP estimation:

(EI)

- External Input: user authentication credentials, creating/scheduling meeting inputs, sending messages, Managing participant list, Recorded meeting, change in user settings.
- External output^(EO): Display meeting schedule, Display ~~meet~~ participant lists, playback of recorded meetings, sending notifications.
- External Inquiries (EQ): search for meetings, Retrieve user Data
- Internal Logical Files (ILF): user accounts, meeting records
- External interface files (EIF): calendar API, document editing APIs.

Table:

Information Domain	Count	Simple	Avg	Complex	
EI	6	3	④	6	24
EO	4	4	⑤	7	20
EQ	2	3	④	6	8
ILF	2	7	⑩	15	20
EIF	2	5	⑦	10	14
Total					86

Value adjustment factors:

Answer for all 14 question

1) → 5

8) → 3

2) → 4

9) → 3

3) → 4

10) → 4

4) → 5

11) → 4

5) → 4

12) → 3

6) → 4

13) → 4

7) → 3

14) → 5

$$\Sigma F_i = 55$$

$$FP(\text{estimated}) = \text{total} * [0.65 + 0.01 * \Sigma F_i]$$

$$= 86 * [0.65 + 0.01 * 55]$$

$$= 103.2 \text{ function points}$$

Conclusion:

Hence we estimated effort and cost required using function point for project. Also created WBS and gantt chart for the same.



Academic Year: 2023_24

Team Members :

Riddhi Shah-60004210161

Dhruvin Chawda-60004210159

Implementation :

1. WorkBreakdown Structure :

Task ID	Task Description	Est Person -Hours	Who	Resources	Material and supplies
<u>1</u>	<u>User Authentication and Authorization</u>				
1.1	Design user authentication system	40	Backend Developer	Software development tools	\$250
1.2	Implement email/password authentication	30	Backend Developer	Authentication libraries	
1.3	Implement Single Sign-On (SSO) method	50	Backend Developer	SSO integration tools	\$100
1.4	Develop role-based access control (RBAC) system	60	Backend Developer	RBAC frameworks	
<u>2</u>	<u>Meeting Creation and Scheduling</u>				
2.1	Design meeting creation interface	40	Frontend Developer	Design software	
2.2	Implement meeting scheduling functionality	50	Backend Developer	Scheduling libraries	
2.3	Integrate calendar integration feature	30	Backend Developer	Calendar API	\$25/month
2.4	Develop recurring meeting functionality	40	Backend Developer	Recurring meeting algorithms	
<u>3</u>	<u>Audio and Video Conferencing</u>				
3.1	Design audio and video conferencing system	80	Backend Developer	Conferencing protocols	
3.2	Implement real-time communication features	60	Backend Developer	Real-time communication libraries	
3.3	Develop mute/unmute functionality	20	Backend Developer	Audio processing tools	\$10



Academic Year: 2023_24

3.4	Implement video on/off feature	20	Backend Developer	Video processing tools	\$10
<u>4</u>	<u>Screen Sharing and Content Collaboration</u>				
4.1	Design screen sharing interface	40	Frontend Developer	Design software	
4.2	Develop screen sharing functionality	50	Backend Developer	Screen sharing libraries	
4.3	Implement collaborative document editing	60	Backend Developer	Collaboration tools	
4.4	Develop virtual whiteboard functionality	50	Backend Developer	Whiteboard libraries	
<u>5</u>	<u>Chat and Messaging Features</u>				
5.1	Design in-meeting chat interface	30	Frontend Developer	Design software	
5.2	Implement private messaging feature	40	Backend Developer	Messaging libraries	
5.3	Develop file sharing capabilities	50	Backend Developer	File sharing protocols	
<u>6</u>	<u>Participant Management</u>				
6.1	Design participant management interface	40	Frontend Developer	Design software	
6.2	Implement add/remove participants functionality	50	Backend Developer	Participant management tools	
6.3	Develop participant permission control feature	60	Backend Developer	Permission control frameworks	
6.4	Implement participant status view	30	Backend Developer	Status monitoring tools	
<u>7</u>	<u>Recording and Playback</u>				
7.1	Design meeting recording interface	40	Frontend Developer	Design software	
7.2	Implement secure meeting recording functionality	60	Backend Developer	Recording encryption tools	\$10
7.3	Develop meeting storage system	50	Backend Developer	Storage infrastructure	\$300



Academic Year: 2023_24

7.4	Implement meeting playback feature	40	Backend Developer	Playback algorithms	
<u>8</u>	<u>Virtual Backgrounds and Filters</u>				
8.1	Design virtual backgrounds interface	30	Frontend Developer	Design software	
8.2	Develop background customization options	40	Backend Developer	Background customization tools	\$20
8.3	Implement video filters feature	50	Backend Developer	Video filter libraries	\$15
<u>9</u>	<u>Integration with Calendar and Productivity Tools</u>				
9.1	Design integration interface	40	Backend Developer	Design software	
9.2	Implement calendar synchronization	60	Backend Developer	Calendar API	\$25/month
9.3	Integrate with document editors	50	Backend Developer	Document editing APIs	\$10/month
<u>10</u>	<u>Security Features</u>				
10.1	Design security measures	80	Backend Developer	Security protocols	
10.2	Implement end-to-end encryption	70	Backend Developer	Encryption libraries	
10.3	Develop meeting password protection	50	Backend Developer	Password protection algorithms	
10.4	Implement secure participant verification	60	Backend Developer	Verification protocols	

Estimation :

FP :

- External Inputs (EI): User authentication, Creating/scheduling meetings, Sending messages, Managing participant lists, Recording meetings Changing user settings
- External Outputs (EO): Displaying meeting schedules, Sending notifications, Displaying participant lists, Playback of recorded meetings
- External Inquiries (EQ): Searching for meetings, Retrieving user information
- Internal Logical Files (ILF): User accounts, Meeting records



Academic Year: 2023_24

- External Interface Files (EIF): Calendar API, Document editing APIs

Table :

Information Domain	Count	Simple	Average	Complex	
External inputs	6	3	4	6	24
External Outputs	4	4	5	7	20
External enquiry	2	3	4	6	8
Internal Logical Files	2	7	10	15	20
External interface files	2	5	7	10	14
Count-total					86

1. Does the system require reliable backup and recovery?

5 - It is essential to have reliable backup and recovery for important modules.

2. Are specialized data communications required to transfer information to or from the application?

4 - Application needs to be "realtime" .

3. Are there distributed processing functions?

4 - Yes

4. Is performance critical?

5 - Performance is always critical

5. Will the system run in an existing, heavily utilized operational environment?

4 - No, the project is not based on a pre-existing system.

6. Does the system require online data entry?

4 - Yes

7. Does the online data entry require the input transaction to be built over multiple screens or operations?

3 - If input transactions span multiple screens or operations, it can increase complexity, but it's not the most complex scenario.

8. Are the ILFs updated online?

3

9. Are the inputs, outputs, files, or inquiries complex?

3 - Yes

10. Is the internal processing complex?

4 - Yes



Academic Year: 2023_24

11. Is the code designed to be reusable?

4 - Yes as we are following the best coding practices.

12. Are conversion and installation included in the design?

3

13. Is the system designed for multiple installations in different organizations?

4

14. Is the application designed to facilitate change and ease of use by the user

5

$\Sigma(F_i) = 55$

The estimated number of FP is derived:

FP(estimated) = count-total*[0.65 + 0.01* $\Sigma(F_i)$]

= 86*[0.65 + 0.01*55]

= 103.2 function points.

Gant Chart :

