ANIMAL SHELTER WEBSITE

SHIRISH SHETTY -2103164
MAITHALI SHINDE -2103166
ADARSH SHUKLA -2103167

Experiment: 8

Aim:

Conduct Function Point Analysis (FPA) for the project.

Theory:

No. of User Inputs (UI):

User registration or login: 1 UI

Reporting distressed animals: 1 UI

Chatbot interactions: 1 UI

Posting messages and engaging in discussions: 1 UI

Selecting pets for adoption: 1 UI

Volunteering for animal shelters: 1 UI

Making merchandise purchases: 1 UI

Making monetary donations: 1 UI

Total UIs: Approximately **8** UIs

No. of User Outputs (UO):

Displaying the results of animal rescue requests: 1 UO

Chatbot responses: 1 UO

Showing community posts and content: 1 UO

Displaying pet adoption details: 1 UO

Providing information on volunteering opportunities: 1 UO

Showing merchandise for purchase: 1 UO

Confirmations of successful donations: 1 UO

Total UOs: Approximately **7** UOs

No. of User Inquiries (UI):

Inquiries to the Chatbot: 1 UI

Total UIs: Approximately **1** UI

No. of Files:

The number of files can vary based on the data storage and organization within your system. However, you might have separate files for user profiles, animal rescue data, community posts, pet adoption details, merchandise information, and potentially others. Let's estimate around **6** files.

No. of External Interface Files (EIFs):

Integration with Google Maps for location marking: 1 EIF

Integration with payment gateways for merchandise purchases and donations: 1 EIF

Total EIFs: Approximately **2** EIFs

Counting Function Point (FP):

Step 1:

Backup & Recovery:

Value: 4

Explanation: While there is important data related to animal rescue requests and shelter coordination, the fiscal loss due to data loss may not be as critical as in a financial system.

Data Communications:

Value: 3

Explanation: The project involves communication between users (admin and store owners) and real-time data updates, but it's not highly complex or extensive.

Distributed Processing:

Value: 1

Explanation: The project operates within a single environment, and there's no need for distributed processing.

Performance Critical:

Value: 4

Explanation: Accurate tracking of inventory and cost calculations are crucial to prevent fiscal loss, but it may not be as critical as a financial system.

Existing Operating Environment:

Value: 2

Explanation: The project is not highly dependent on a specific operating environment, but it does require basic CRUD operations.

Online Data Entry:

Value: 3

Explanation: Real-time data updates are essential for efficient inventory management, but the complexity is moderate.

Input Transaction Over Multiple Screens:

Value: 4

Explanation: Simultaneous updates from multiple store owners on different screens require accuracy and real-time synchronization.

Master Files Updated Online:

Value: 4

Explanation: Real-time updates of product orders and billing information are crucial for effective inventory management.

Information Domain Values Complex:

Value: 2

Explanation: The data used in the project is relatively straightforward, with average data volume and relationships. Data transformation is moderate in complexity.

Internal Processing Complex:

Value: 2

Explanation: Internal complexity related to algorithms is minimal, but basic data manipulation and performance optimization are required.

Code Designed for Reuse:

Value: 3

Explanation: Some code can be designed for reuse, such as modularization and encapsulation, to enhance maintainability.

Conversion/Installation in Design:

Value: 4

Explanation: Incorporating installation and data migration considerations in the design is crucial for efficient deployment and cost savings.

Multiple Installations:

Value: 3

Explanation: The project should support installation on different devices and OS environments but may not support multiple views simultaneously for security reasons.

Application Designed for Change:

Value: 4

Explanation: Designing the application to be scalable and adaptable to future requirements is important for long-term viability.

Total Value Adjustment Factor: Σ Fi = 42

Step 3: Calculate Unadjusted Function Point (UFP)

Measurement Parameter	**Weighting Factor**

	Simple		**Average**		**Complex**	
Number of user input	8x 3	24	8 x 4	32	8 x 6	48
Number of user output	7x 4	28	7 x 5	35	7x 7	49
Number of user inquiries	1 x 3	3	1x 4	4	1x 6	6
Number of files	6x 7	42	6x 10	60	6x 15	90
Number of external interfaces	2 x 5	10	2x 7	14	2x 10	20
	Simple Total	**107**	**Average Total**	**145**	**Complex Total**	**213**

Step 4: Calculate Function Point.

1. **Simple**

   Unadjusted Function Point = 107

   Function point = (Unadjusted Function Point) x (Complexity Adjustment Factor)
   = 107 * [0.65 + 0.01 * 42]
   = 107 * [0.65 + 0.42]
   = 107 * 1.07
   = 114.49
   =114

2. **Average**

   Unadjusted Function Point = 145

   Function point = (Unadjusted Function Point) x (Complexity Adjustment Factor)

   = 145 * [0.65 + 0.01 * 42]
   = 145 * [0.65 + 0.42]
   = 145 * 1.07
   = 155.15
   = 155

3. **Complex**

Unadjusted Function Point = 213

Function point = (Unadjusted Function Point) x (Complexity Adjustment Factor)
= 213 * [0.65 + 0.01 * 42]
= 213 * [0.65 + 0.42]
= 213 * 1.07
= 227.91
=227

1. Productivity Factor Assumption: Let's assume a productivity factor of 4 hours per function point.
2. Hourly Rate Assumption: Let's assume an hourly rate of Rs.50 per hour.

Now, we can calculate the effort, productivity cost per function point, and total cost for each complexity level:

For Simple (107 Function Points):

- Effort = 107 Function Points × 4 hours/FP = 428 hours
- Productivity Cost per Function Point = Total Effort / Total Function Points = 428 hours / 107 FP = 4 hours/FP
- Total Cost = Effort × Hourly Rate = 428 hours × Rs.50/hour = Rs.21,400

For Average (145 Function Points):

- Effort = 145 Function Points × 4 hours/FP = 580 hours
- Productivity Cost per Function Point = Total Effort / Total Function Points = 580 hours / 145 FP = 4 hours/FP
- Total Cost = Effort × Hourly Rate = 580 hours × Rs.50/hour = Rs.29,000

For Complex (213 Function Points):

- Effort = 213 Function Points × 4 hours/FP = 852 hours
- Productivity Cost per Function Point = Total Effort / Total Function Points = 852 hours / 213 FP = 4 hours/FP
- Total Cost = Effort × Hourly Rate = 852 hours × Rs.50/hour = Rs.42,600