**PRACTICAL - 6**

**AIM:** Study and apply software effort estimation techniques.

**Task:**

1. Choose two estimation techniques from the list (e.g., Bottom-up estimating, Expert judgment, Function Point Analysis, COCOMO).
2. Draw up an outline program structure diagram for a given scenario. For each box on your diagram, estimate the number of lines of code needed to implement the routine in a programming language (eg.Java)

(Use External input types none, External output types the report, that is, 1, Logical internal file types none, External interface file types payroll file, staff file (timetabling), courses file (timetabling), that is, 3, External inquiry types none)

1. Calculation of SLOC from Albrecht function points.
2. Apply COCOMO to estimate the effort.(Use Table C.7 Assessing scale factors)

**Solution:**

**Team Details:**

| **Sr. No.** | **Name** | **Enrollment No.** |
| --- | --- | --- |
| **Team Leader** | **Angat Shah** | **202203103510097** |
| **Team Member 1** | **Yash Patel** | **202203103510228** |
| **Team Member 2** | **Gati Shah** | **202203103510261** |
| **Team Member 3** | **Fenil Shilodre** | **202203103510041** |
| **Team Member 4** | **Sarth Chaudhari** | **202303103510106** |

## 

## Estimation Techniques Used

1. **COCOMO (Constructive Cost Model)**
   * Used for estimating effort, time and cost in software projects.
   * Formula :
   * We determine whether the project falls under Organic, Semi-Detached or Embedded.

## Function Point Analysis (FPA)

* + Used for estimating the size and complexity of the software. (SLOC)
  + Categories include External Inputs (EI), External Outputs (EO), Internal Logical Files (ILF), External Interface Files (EIF) and External Queries (EQ).
  + Assigns weightage to components based on their complexity (Low, Average, High).

## 

## 

## 

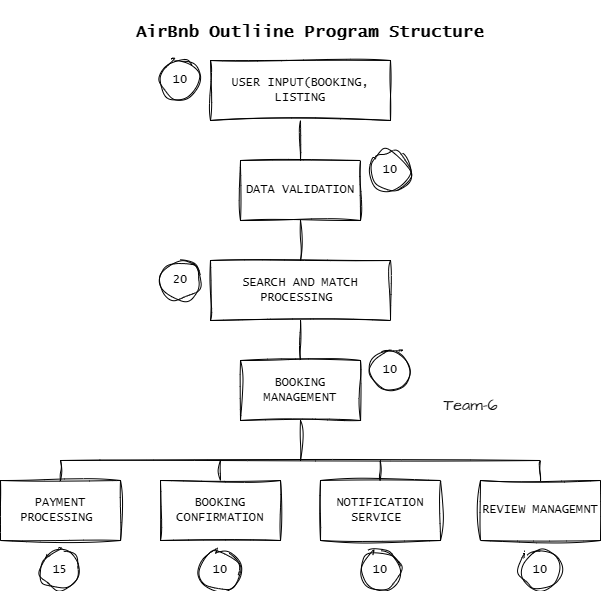
## Albrecht Complexity Multipliers :

| **External User Type** | **Multiplier** | | |
| --- | --- | --- | --- |
| **Low** | **Average** | **High** |
| External Input Type | 3 | 4 | 6 |
| External Output Type | 4 | 5 | 7 |
| External Inquiry Type | 3 | 4 | 6 |
| Logical Internal File Type | 7 | 10 | 15 |
| External Interface File Type | 5 | 7 | 10 |

**Project Title : Airbnb System**

## Program Structure Diagram

Here is the detailed program structure including blocks and the estimated LOC for each component :



* User Input (Booking, Listing) : 10 LOC
* Data Validation : 10 LOC
* Search and Match Processing : 20 LOC
* Booking Management : 10 LOC
* Payment Processing : 15 LOC
* Booking Confirmation : 10 LOC
* Notification Service : 10 LOC
* Review Management : 10 LOC

**Total LOC : 95 LOC**

### Function Point Analysis (FPA) Calculation :

| **Component** | **Count** | **Complexity** | **Weight (Albrecht)** | **Total FP** |
| --- | --- | --- | --- | --- |
| External Input Type (EI) | 3 | Average | 4 | 12 |
| External Output Type (EO) | 2 | Average | 5 | 10 |
| External Inquiry Type (EQ) | 2 | Low | 3 | 6 |
| Internal Logical Files (ILF) | 3 | High | 15 | 45 |
| External Interface File Type (EIF) | 1 | Average | 7 | 7 |
| **TOTAL FUNCTION POINTS** | | | | **80** |

* **SLOC =** FP x 50 LOC (common for Java-based applications)

= 80 x 50

**= 4000 SLOC**

### Effort Estimation Using COCOMO :

* **Effort (Person-Months) =**

=

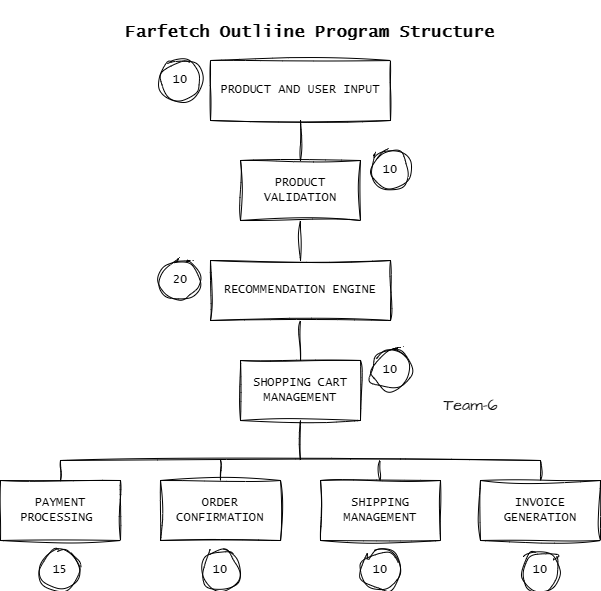
=

**≈ 14 Person-Months**

**Project Title : Farfetch E-commerce System**

## Program Structure Diagram

Here is the detailed program structure including blocks and the estimated LOC for each component :



* Product & User Input : 10 LOC
* Product Validation : 10 LOC
* Recommendation Engine : 20 LOC
* Shopping Cart Management : 10 LOC
* Payment Processing : 15 LOC
* OrderConfirmation : 10 LOC
* Shipping Management : 10 LOC
* Invoice Generation : 10 LOC

**Total LOC : 95 LOC**

### Function Point Analysis (FPA) Calculation :

| **Component** | **Count** | **Complexity** | **Weight (Albrecht)** | **Total FP** |
| --- | --- | --- | --- | --- |
| External Input Type (EI) | 4 | Average | 4 | 16 |
| External Output Type (EO) | 3 | Average | 5 | 15 |
| External Inquiry Type (EQ) | 2 | Low | 3 | 6 |
| Internal Logical Files (ILF) | 3 | High | 15 | 45 |
| External Interface File Type (EIF) | 4 | Average | 7 | 28 |
| **TOTAL FUNCTION POINTS** | | | | **110** |

* **SLOC =** FP x 50 LOC (common for Java-based applications)

= 110 x 50

**= 5500 SLOC**

### Effort Estimation Using COCOMO :

* **Effort (Person-Months) =**

=

=

**≈ 20 Person-Months**

## Conclusion

## Effort estimation plays a crucial role in predicting the time, cost and resource allocation for software projects. Based on the analysis :

* **Airbnb requires fewer resources and effort** due to a lower function point count.
* **Farfetch requires more effort** due to its complexity, larger dataset management and additional security needs.
* The estimation methodologies used provide a realistic measure aligned with industry standards.