Size Estimation: -

Function Point:

Rate Service:

When a customer finishes his or her meal and checks out and fills all the required data for the payment, a function called rate service will appear to the customer. Rate service a 5-star rating, 1(very bad) to 5 (very good), and comment feedback on how was the service and staff during their night in our restaurant. This function will help our restaurant in improving the service and will give instructions to our staff on how to treat our customer in a better way and in a more civilized way; and also, it will help our staff to be more motivated with the good comments and rating and this will help them feel more open to working.

Rate service function will include staff names attached with a picture and a 5-star rating each. The customer will rate the staff that was in charge of the customer’s evening, write a comment in the comment box provided, and write his/her number to send a thank you message.

Count of components:

* External Input types (EI): 3

1. Star rating
2. Comment
3. Phone number

* External Output types (EO): 1

1. Thank you, message

* External Inquiry types (EQ): 2

1. Check if the phone number exists or not
2. Insert the rating and feedback in the database

* Logical internal file types (LIF): 2

1. Staff table
2. Customer table

* External Interface file types (EIF): 0

Identify Datatypes, Record Types and Complexity:

EI:

* 3 data types: Star rating, Comment, Phone number
* 1 record type: User input
* Low complexity

EO:

* 1 data type: message sent to the customer
* 1 record type: staff table
* Low complexity

EQ:

* 3 data types: Star rating, Comment, Phone number
* 2 record type: customer table, staff table
* Low complexity

LIF:

* 3 data types: Star rating, Comment, Phone number
* 2 record type: customer table, staff table
* Low complexity

EIF:

* 0 datatype
* 0 record type
* Low complexity

Size estimation = 3\*3 +1\*4+ 2\*3 +2\*7+ 0\*5 = 33 FP

Language used: Java

33 FP \* (55) = 1815 LOC = 1.815 KLOC

Albrecht II Function point:

Hide dishes:

The staff has some privileges in our system one of them is hiding the dishes that are no longer available in the kitchen or had a low rating due to customer’s review. Hide dish’s function is important in our restaurant because it minimizes the customer’s confusion regarding whether this specific dish is available or not.

Hide dish’s function includes the following: staff must click on the dish he/she needs to hide from the menu, enters a code to complete the process, and clicks on yes to confirm the operation.

Count of components:

Inputs:

1. Click on hide dish button
2. Click on the unwanted dish
3. Enter a code (known by the staff only)
4. Continue button (to hide the dish)

Outputs:

1. Confirmation message
2. Dish will be unavailable to the customer

Entities (Storage):

1. Staff table (to check if the code is correct or not)
2. Item Table
3. Menu table

Size Estimation: Ni \* 0.58 + Ne\*1.66 + No\*0.26

4\*0.58 + 3\*1.66 + 2\*0.26 = 7.82 FP

Language used: Java

7.82 FP \* (55) = 430.1 LOC = 0.4301 KLOC

Effort Estimation:

COCOMO 81:

Rate Service:

Effort = C (size)^k

C= organic system (2.4)

K= 1.05

Effort = 2.4(33) ^1.05 = 94.33 PM

COCOMO II:

Hide dishes:

Effort (PM) = A (Size)^sf x (Product of exponent multipliers)

A = 2.94

Sf= B +0.01\* estimation (exponent driver ratings)

B= 0.91

Estimation (exponent driver ratings) = 5 factors (PREC +FLEX+RESL+TEAM+PMAT)

1. PREC: very low 6.20
2. FLEX: low 4.05
3. RESL: High 2.83
4. TEAM: very high 1.10
5. PMAT: Extra high 0.00

6.20+ 4.05+2.83+ 1.10+ 0.00 = 14.18

Sf = 0.91 + 0.01\*14.18 = 1.0518

Product of exponent multipliers:

Product:

* RELY: 1 (Nominal)
* DATA: 1.14 (High)
* DOCU: 0.91 (Low)
* CPLX: 0.73 (Very low)
* REUSE: 1.15(Very low)

Computer:

* TIME: 1.00 (Nominal)
* STOR: 1.17 (Very high)
* PVOL: 0.87 (Low)

Personnel:

* ACAP: 0.71 (Very high)
* ALEXP: 0.81 (Very high)
* PCAP: 0.76 (Very high)
* PEXP: 0.91 (High)
* LEXP: 0.91 (High)
* PCON: 1.00 (Nominal)

Project:

* TOOL: 0.78(Very high)
* SITE: 1.09 (Low)
* SCED: 1.00(Nominal)

Product of exponent multipliers = 1\*1.14\*0.91\*0.73\*1.15\*1\*1.17\*0.87\*0.71\*0.81\*0.76\*0.91\*0.91\*1\*0.78\*1.09\*1 = 0.272

Effort = 2.94(7.82) ^ 1.0518 \* 0.272 = 0.3239311757 KLOC