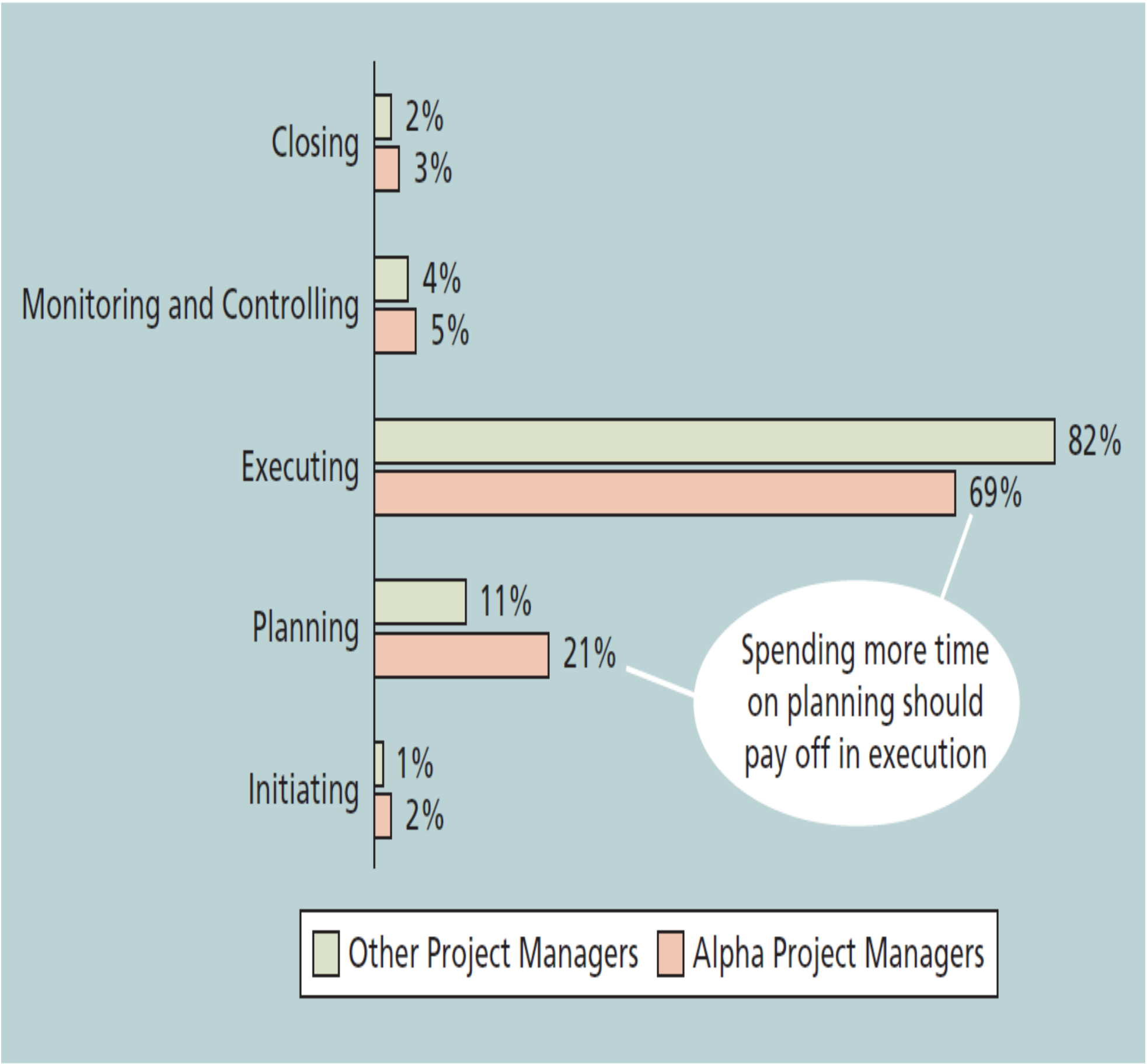
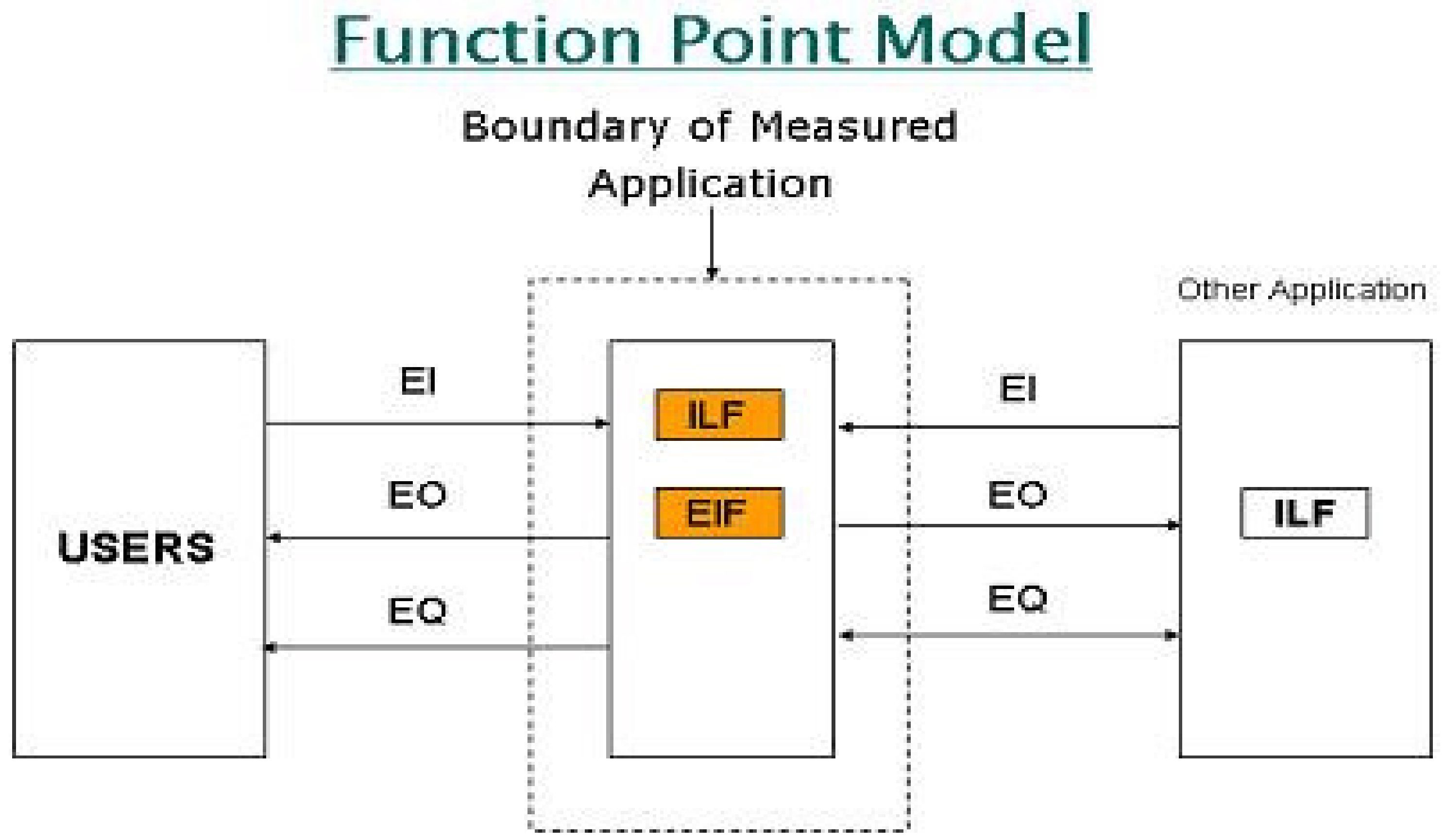
Size Estimation of Machine Vision based Retail Self-Checkout System Software

Author: Tanjimul Ahad Asif

Background



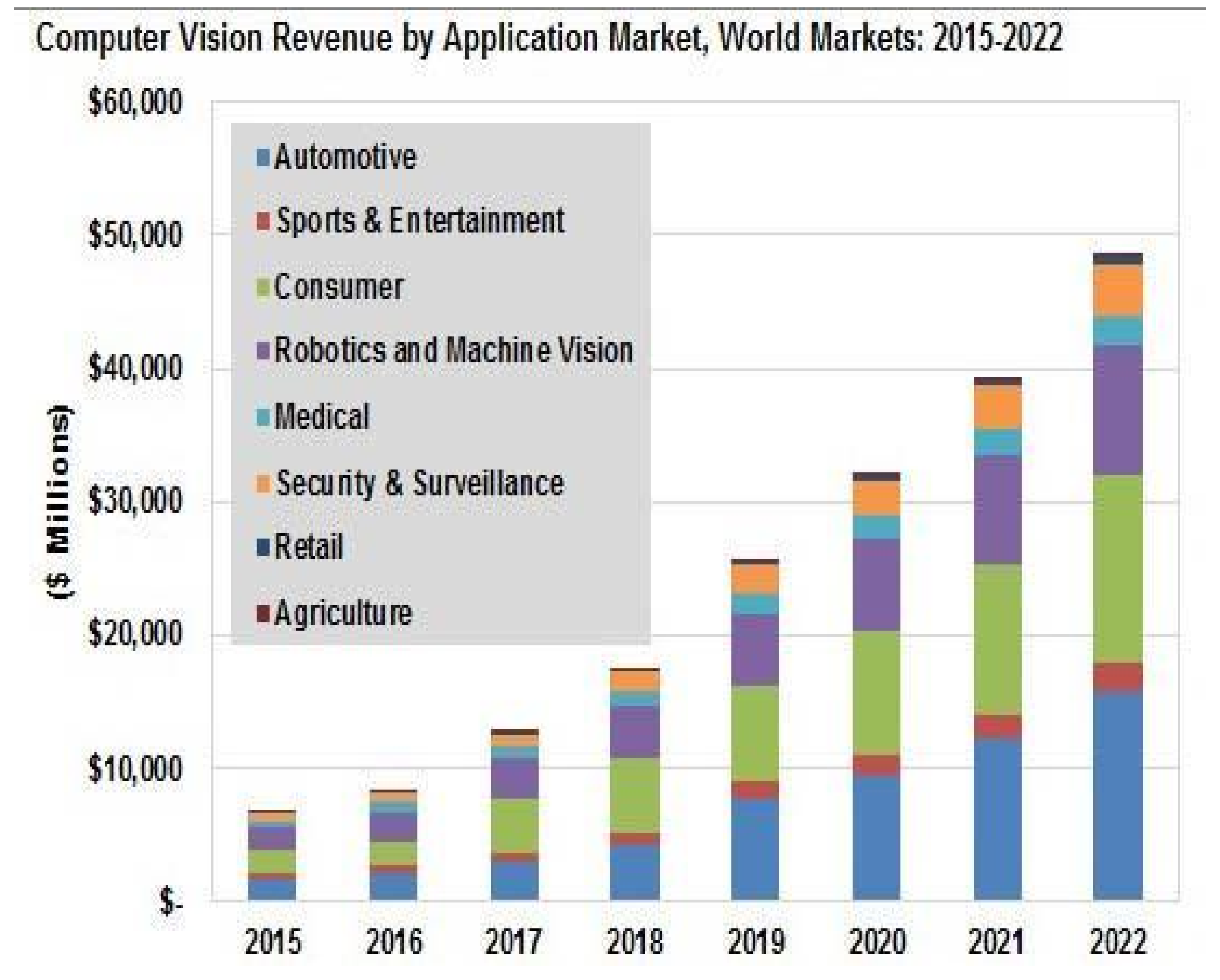


Function Point = Crude Function Point x (0.65 + 0.01 x RCAF)

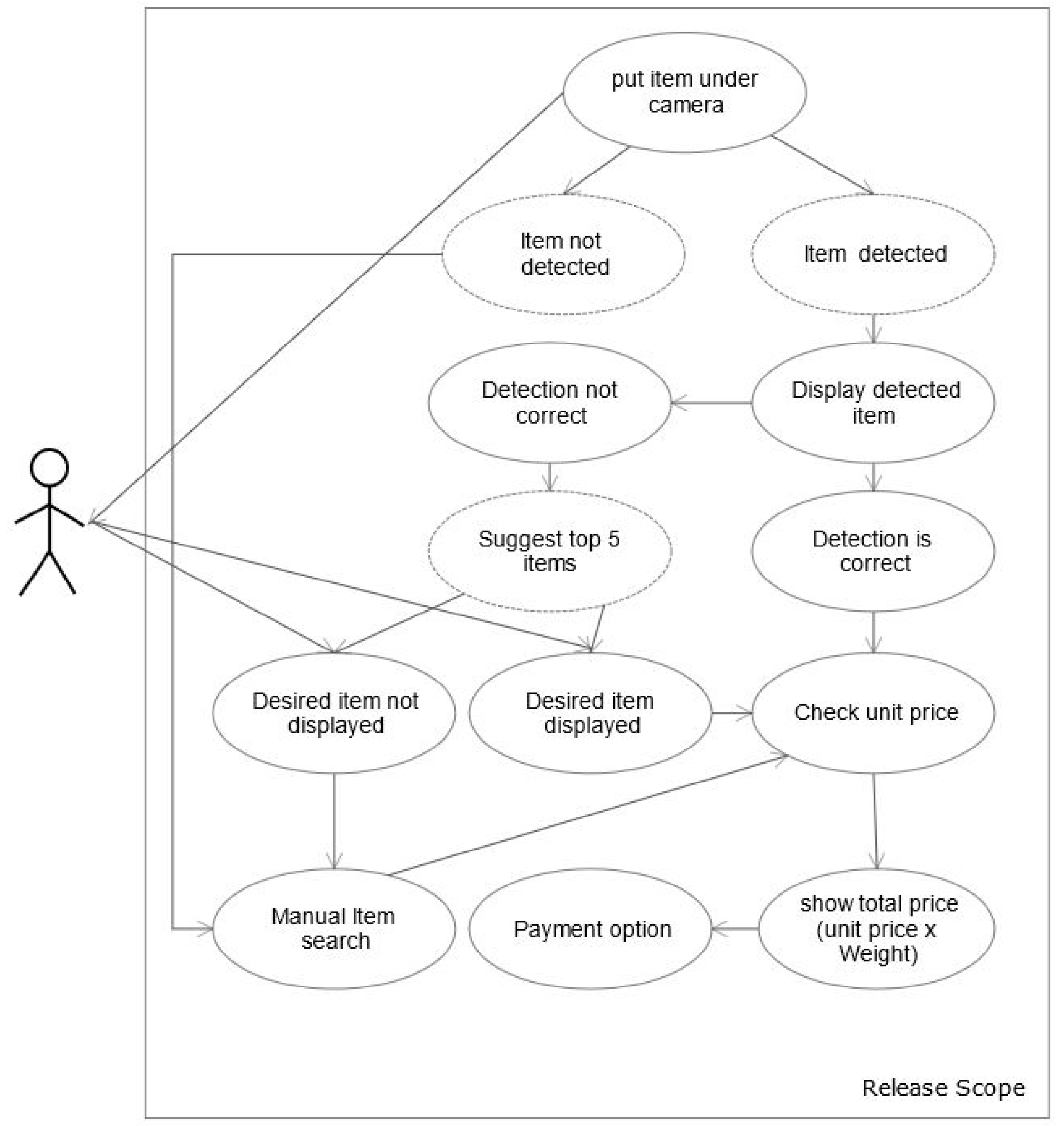
= 99 x (0.65 + 0.01 x 53)

= 116.82

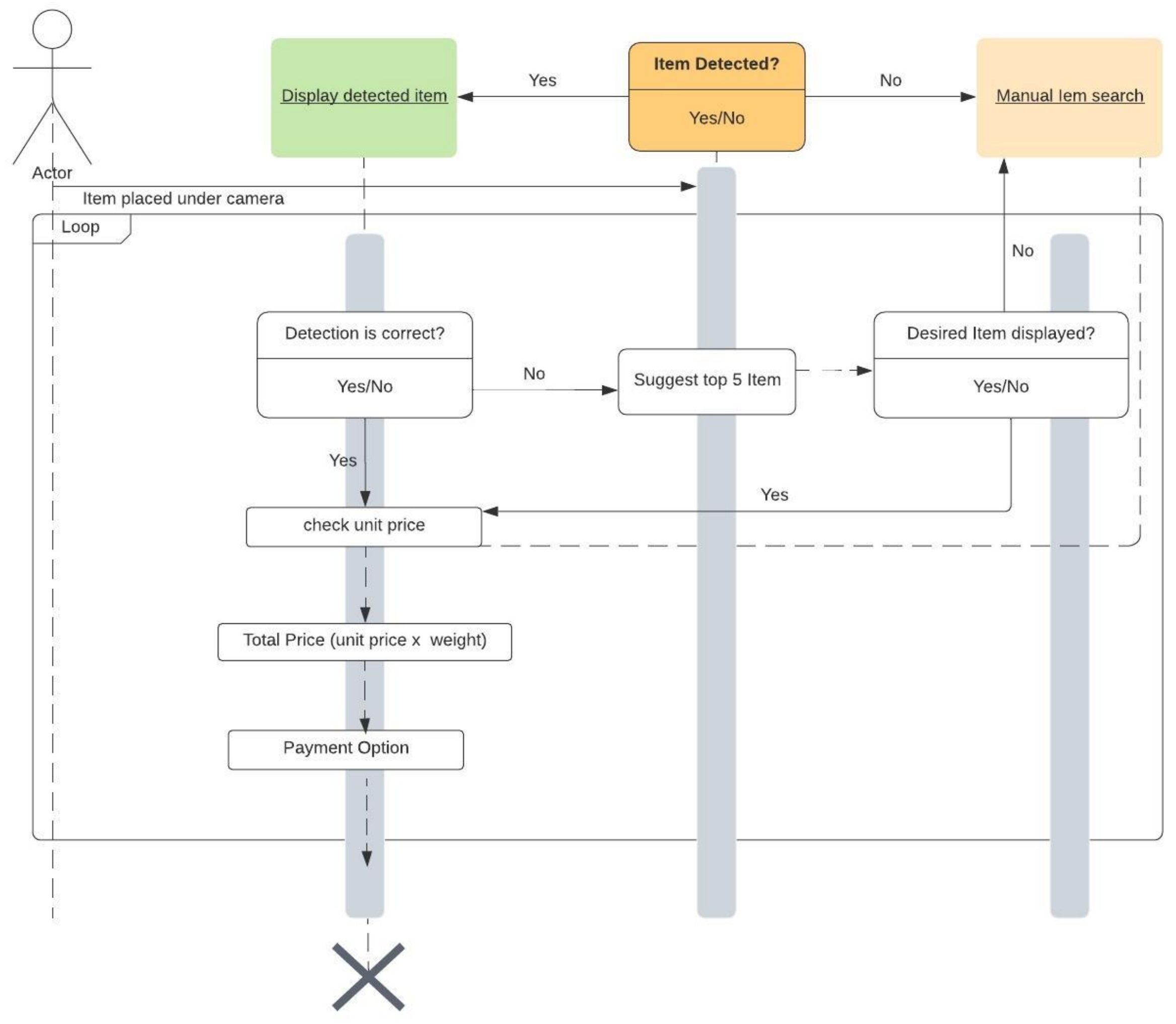
System Overview



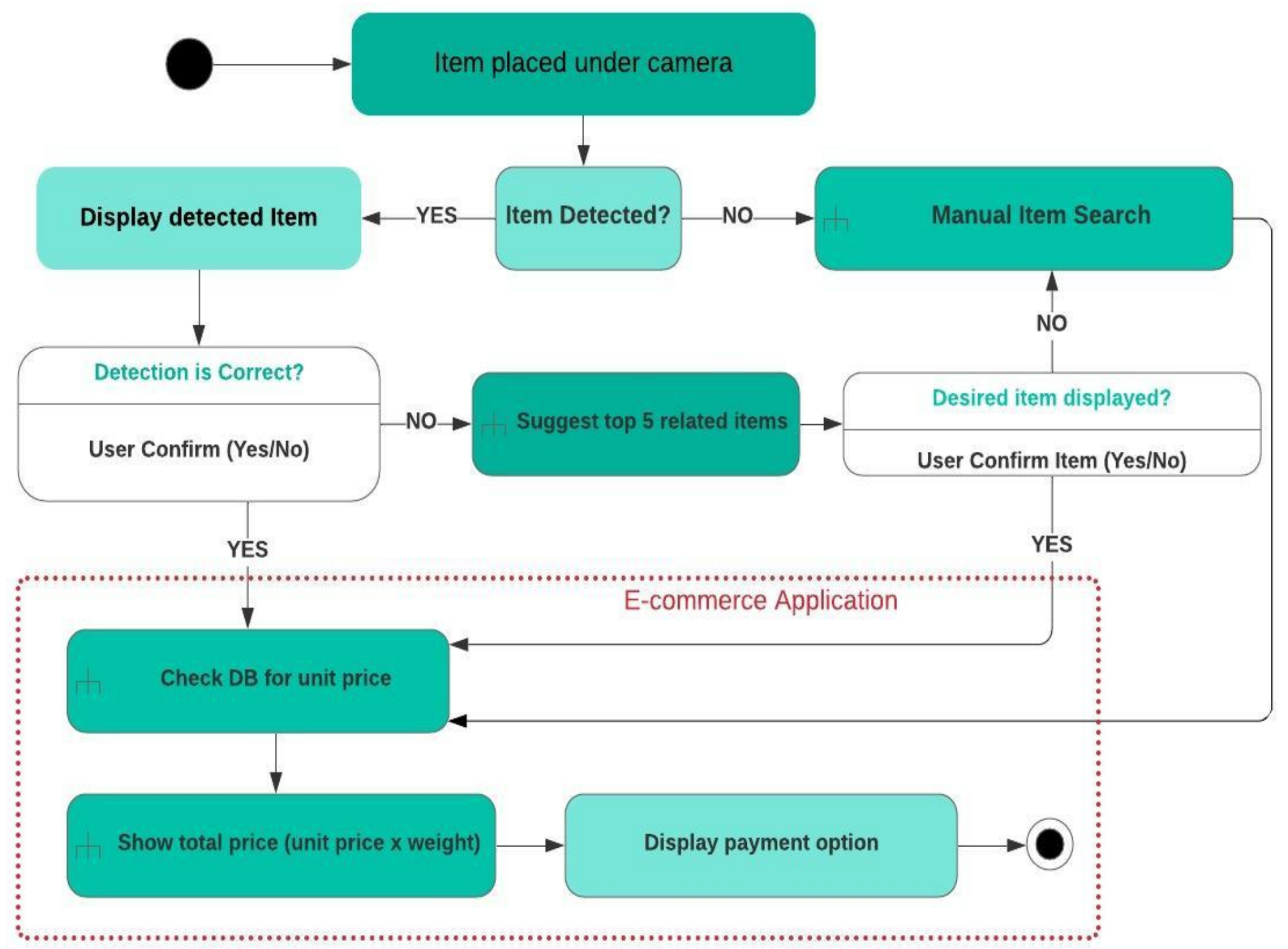
Use Cases



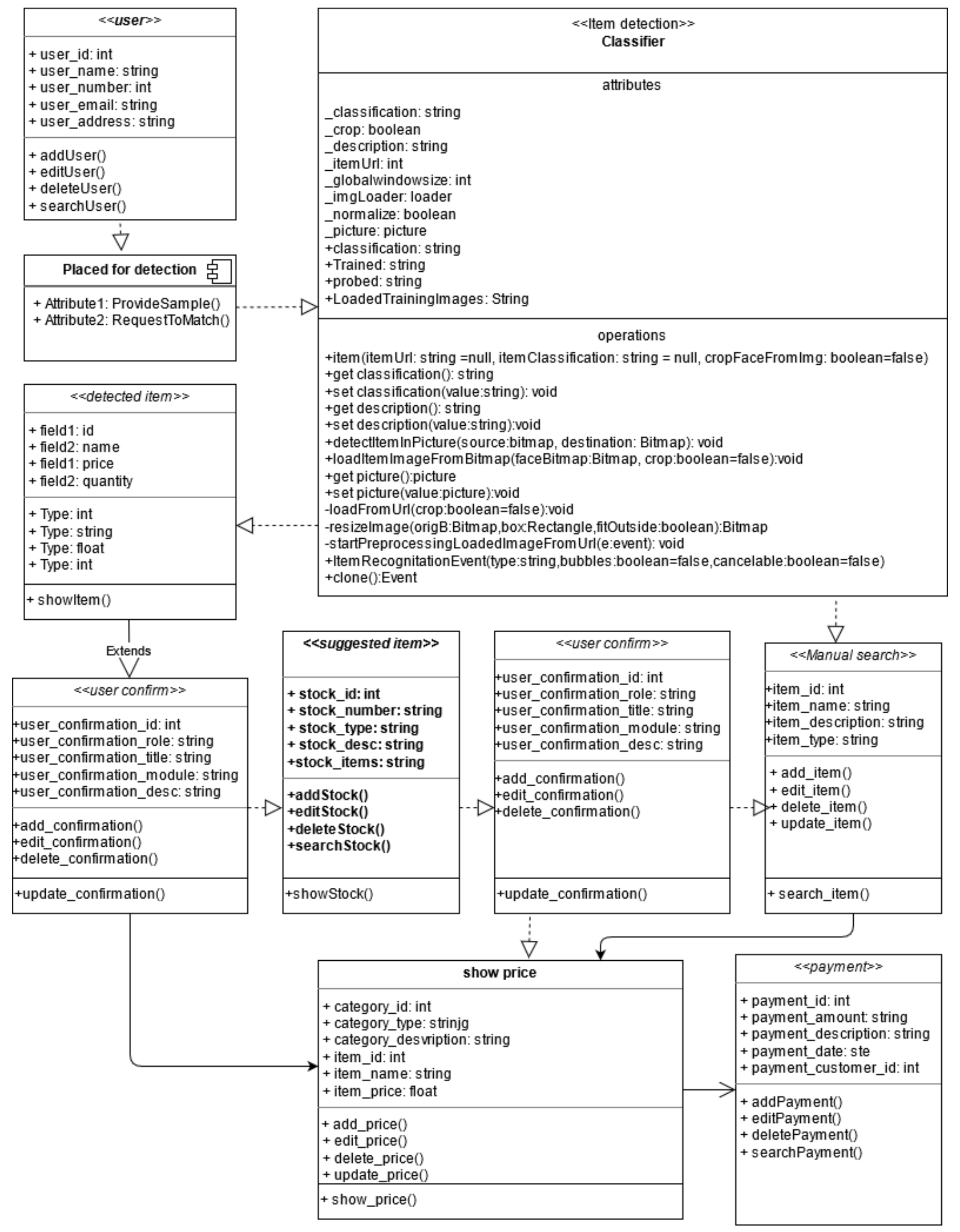
Sequence Diagram



Data Flow Diagram (DFD)



Class Diagram



Software System Component	Level of Complexity									SUM of CFP
	Low			Average			High			
	Count	Weighting Factor	Point	Count	Weighting Factor	Point	Count	Weighting Factor	Point	
	A	B	C=AXB	D	E	F=DXE	G	H	I-GXH	J = C + F + I
INPUT		3			4			6		
OUTPUT		4			5			7		
QUERY		3			4			6		
LOGIC FILE		7			10			15		
EXTERNAL INTERFACE		5			7			10		
SUM of CFP										

Crude Function Point Calculation

No.	Subject	Value					
1	The level of recovery relibility complexity	0	1	2	3	4	5
2	The level of data communication complexity	0	1	2	3	4	5
3	The level of distributed processing complexity	0	1	2	3	4	5
4	Level of need for performance complexity	0	1	2	3	4	5
5	The level of operating environment demand	0	1	2	3	4	5
6	The level of developer knowledge needs	0	1	2	3	4	5
7	The level of updating master file complexity	0	1	2	3	4	5
8	The level of installation complexity	0	1	2	3	4	5
9	The level of input, output, query and file application complexity	0	1	2	3	4	5
10	The level of data processing complexity	0	1	2	3	4	5
11	The improbability level of reuse code	0	1	2	3	4	5
12	The level of customar organization variation	0	1	2	3	4	5
13	The extent of possible change	0	1	2	3	4	5
14	Level of the ease of use demand	0	1	2	3	4	5
	Total = RCAF						

Relative Complexity Adjustment Factor Calculation

Function Point Analysis

1. The function point analysis approach can measure a software system's volume based on its complexity, both for an object-oriented and a structured model.
2. Practical expertise is required to implement the function point analysis method because of its highly subjective calculations.
3. Because of the enhanced number of calculations based on data processing representation, the function point analysis approach must also support additional data to improve the estimated volume of software systems.
4. The function points generated by the object-oriented and structured models are not significantly different. So, it can be assumed that object-oriented methods or structured methods are beneficial for providing an idea of the software size estimation.

Contact

Tanjimul Ahad Asif
tasif@student.concordia.ab.ca
Concordia University of Edmonton
Department of Mathematical and Physical Sciences
7128 Ada Boulevard, Edmonton, AB, T5B 4E4, CANADA

References

1. M. de Freitas Junior, M. Fantinato and V. Sun, "Improvements to the Function Point Analysis Method: A Systematic Literature Review," in IEEE Transactions on Engineering Management, vol. 62, no. 4, pp. 495-506, Nov. 2015.
2. G. Costagliola, F. Ferrucci, G. Tortora and G. Vitiello, "Class point: an approach for the size estimation of object-oriented systems," IEEE Transactions on Software Engineering, vol. 31, no. 1, pp. 52-74, Jan. 2005.
3. Y. Yavari, M. Afsharchi and M. Karami, "Software complexity level determination using software effort estimation use case points metrics," Malaysian Conference in Software Engineering, Johor Bahru, Malaysia, 2011, pp. 257-262, 2011.
4. T. Fetcke, A. Abran, and T. Nguyen, “Mapping The OO-Jacobsen Approach to Function Points”, Proceedings of Tools 23’97 – Technology of Object-Oriented Language and Systems, IEEE Computer Society Press, California, 1998
5. A. Zivkovic, R. Ivan, and H. Marjan, “Automated Software Size Estimation based on Function Points using UML Models”, Information and Software Technology, Elseiver, 2005