
Software Requirements Specification

for

Automated Retail Checkout System using YOLOv5 in Computer Vision

Version 1.0 approved

**Prepared by: Nandita Yadav (2000290120100),
Kshiteesh Kumar (2000290120088),
Kumari Bhavya Chaubey (2000290120089)**

KIET Group of Institutions

25/04/2023

Table of Contents

Table of Contents	ii
Revision History	ii
1. Introduction.....	1
1.1 Purpose.....	1
1.2 Document Conventions.....	1
1.3 Intended Audience and Reading Suggestions	1
1.4 Project Scope	1
2. Overall Description	2
2.1 Product Perspective.....	2
2.2 Product Features	2
2.3 User Classes and Characteristics	2
2.4 Operating Environment.....	2
2.5 Design and Implementation Constraints	2
2.6 User Documentation	2
2.7 Assumptions and Dependencies	2
3. System Features	3
3.1 System Feature 1	
3.2 System Feature 2 (and so on).....	
4. External Interface Requirements	3
4.1 User Interfaces	3
4.2 Hardware Interfaces	4
4.3 Software Interfaces	4
4.4 Communications Interfaces	4
5. Other Nonfunctional Requirements	4
5.1 Performance Requirements	4
5.2 Safety Requirements	4
5.3 Security Requirements	4
5.4 Software Quality Attributes	4
6. Other Requirements	4
Appendix A: Glossary.....	4
Appendix B: Analysis Models	5

Revision History

Name	Date	Reason For Changes	Version
Nandita Yadav	25/04/23	Initial	1.1.0

1. Introduction

1.1 Purpose

The purpose of this document is to provide a detailed description of the requirements for the development of an automated retail checkout system utilizing the YOLOv5 model in computer vision. This system aims to enhance the efficiency and accuracy of the checkout process in retail environments.

1.2 Document Conventions

Every requirement statement described in this document has its own priority.

1.3 Intended Audience and Reading Suggestions

This document is intended to developers, project managers, users, testers, and documentation writers. The document contains all the necessary requirements that are must to develop and run the software.

The document should be read in sequence as defined in the table of contents and every section is pertinent to every reader type.

1.4 Project Scope

The system will focus on automating the item recognition and checkout process in retail stores, leveraging the YOLOv5 model for real-time object detection and classification. This system can enhance the process of checkout at small and large grocery or retail stores. It is a novel idea that can help the retailers to reduce the costs required to install expensive machines to smoothen the process of automated checkout. This system is cost friendly and can be easily used by both the customer and the retailers to calculate the costs of the purchased items. Thefts and loss of items can be largely reduced.

The benefits of this model are:

- Efficient product recognition as YOLOv5 can recognize multiple products simultaneously and efficiently with high accuracy and speed.*
- Establishment of seamless checkout experience for customers by eliminating manual labor and longer queues.*
- Reduce in the costs of manual labors as well as expensive machines for retailers.*
- Reduced errors in scanning the items and preventing thefts in stores.*
- Can track inventory in real-time, helping retailers monitor stock levels and improve supply chain management*

An objective of the present disclosure is:

- To pre-process the dataset and convert it into required CSV format.*
- To prepare YOLOv5 model using Roboflow.*
- To empirically evaluate the outcomes of the proposed model.*
- To assess the outcomes of the proposed model with previous models*

2. Overall Description

2.1 Product Perspective

This product is new and self-contained.

2.2 Product Features

The major feature of the product is that the end user can upload the video or image of the bought products and an automated bill is generated. The bill contains the name of the individual product, its quantity, and its price. The total amount of the products bought gets calculated.

2.3 User Classes and Characteristics

The user classes are retailers and customers. The retailers can update the prices of the products and upload the images or video to the software. The customers can get the result as it gets displayed by the system.

2.4 Operating Environment

This system can work on any operating system such as Windows, Linux, etc. It requires a camera to capture the picture or video of the products.

2.5 Design and Implementation Constraints

- *Accuracy and Precision:*

Constraint: The system must achieve high accuracy and precision in object detection to correctly identify and classify items at the checkout.

Implementation: Fine-tune the YOLOv5 model on a dataset representative of retail products to enhance its accuracy in detecting and classifying items.

- *Real-time Processing:*

Constraint: The checkout system should operate in real-time to maintain efficiency during peak hours.

Implementation: Optimize the YOLOv5 model for real-time processing by considering model size, hardware acceleration, and parallelization techniques.

2.6 User Documentation

User manuals, on-line help, and tutorials will be delivered along with the software.

2.7 Assumptions and Dependencies

2.7.1 Assumptions

It is assumed that the retail environment will provide a stable internet connection and standard hardware for system deployment.

2.7.2 Dependencies

The system's functionality depends on the availability and proper functioning of the YOLOv5 model and UI.

3. System Features

3.1 Real-time object detection and classification using YOLOv5

3.1.1 Description and Priority

The system shall utilize YOLOv5 for real-time object detection and classification of retail products. The priority of this task is high and should be done on priority basis,

3.1.2 Stimulus/Response Sequences

The user needs to upload the picture or the video of the bought products in order for the system to detect and classify the products.>

3.1.3 Functional Requirements

REQ-1: Availability of Dataset used.

REQ-2: Camera is required to click the pictures.

3.2 User-friendly UI for customers and store personnel.

3.2.1 Description and Priority

The system shall provide an intuitive UI with features including item recognition confirmation and error handling for customers and store personnel.

3.2.2 Stimulus/Response Sequences

The products get recognized.

3.2.3 Functional Requirements

REQ-1: Object Detection

REQ-2: Object Classification

4. External Interface Requirements

4.1 User Interfaces

The system shall provide an intuitive UI with features including item recognition confirmation and error handling for customers and store personnel. The user interface used here is Tkinter which is a standard GUI library for Python, to provide the user interface. The users can upload the images using this interface and the results get displayed.

4.2 Hardware Interfaces

No hardware interface is required.

4.3 Software Interfaces

No software interface is required.

4.4 Communications Interfaces

No communication interface is required.

5. Other Nonfunctional Requirements

5.1 Performance Requirements

The system shall process a minimum of 30 transactions per minute with a detection accuracy of 95% or higher

5.2 Safety Requirements

The system must be maintained and it is required that the computer on which the software is to be run is free from any malicious attacks and is always available.

5.3 Security Requirements

The information of the customer should be secured.

5.4 Software Quality Attributes

The system should have attributes such as adaptability, availability, correctness, flexibility, interoperability, maintainability, and portability.

6. Other Requirements

The system should only be used by the retailers.

Appendix A: Glossary

- *SRS: Software Requirements Specification*
- *YOLOv5: You Only Look Once version 5*
- *UI: User Interface*
- *UX: User Experience*
- *API: Application Programming Interface*

Appendix B: Analysis Models

Entity Relationship Diagram:

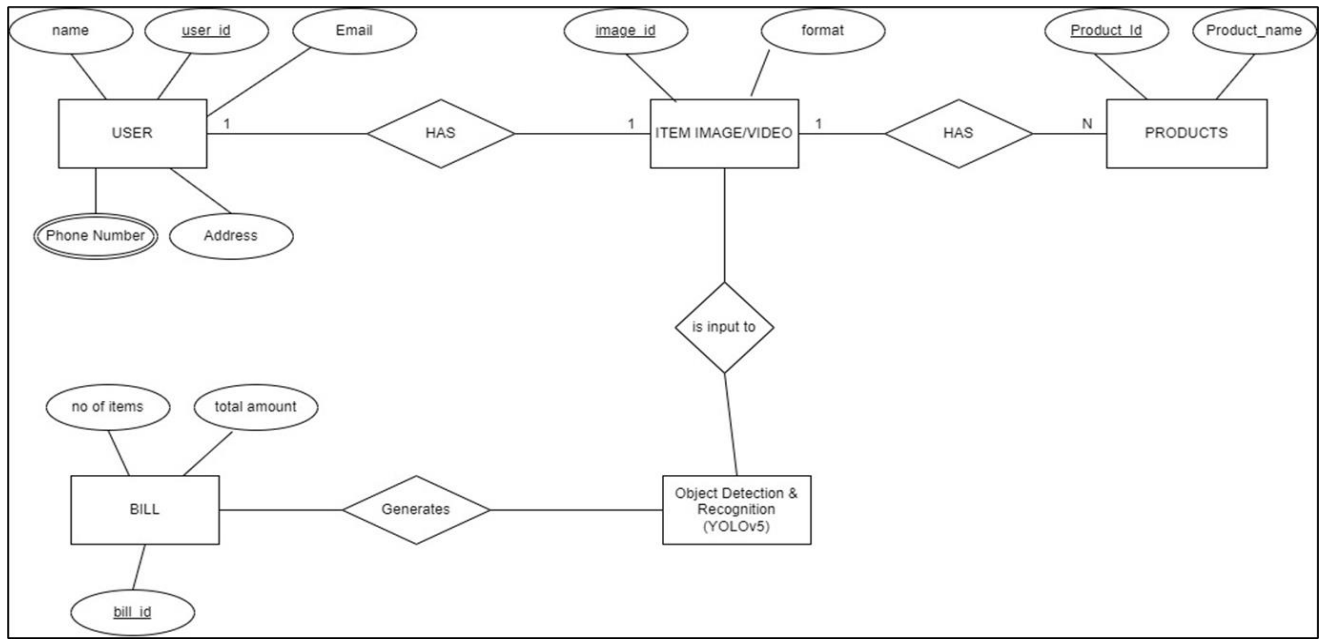


Figure 1 - ER Diagram for checkout process

Data Flow Diagrams:

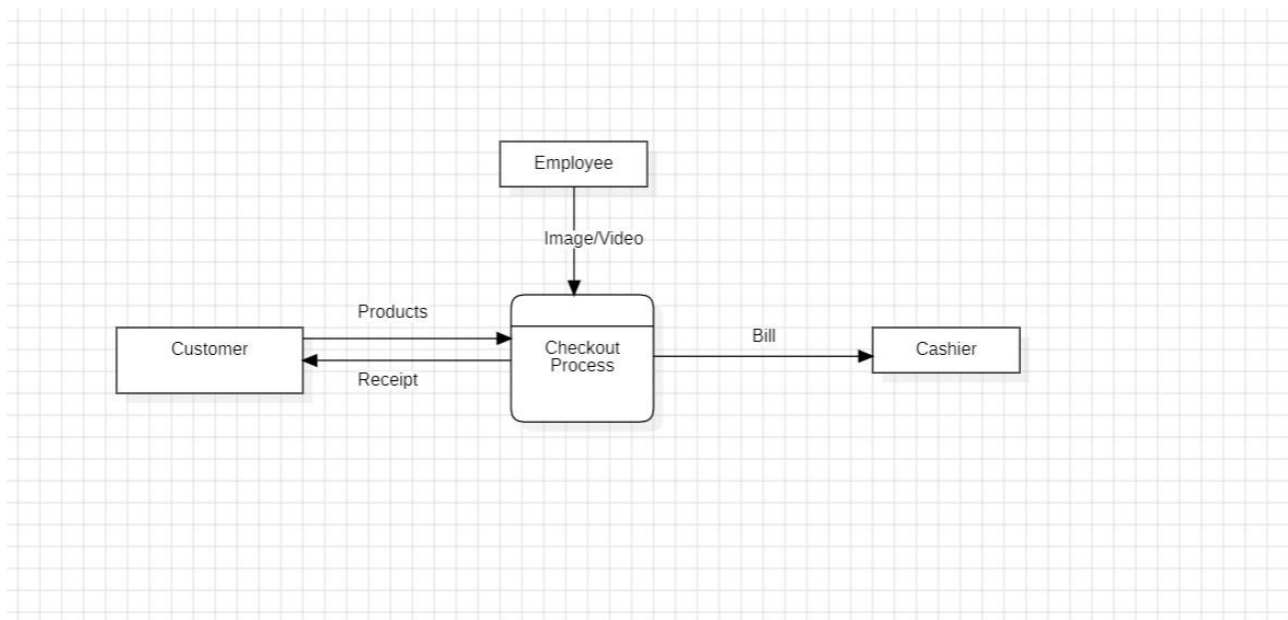


Figure 2 - 0 Level DFD

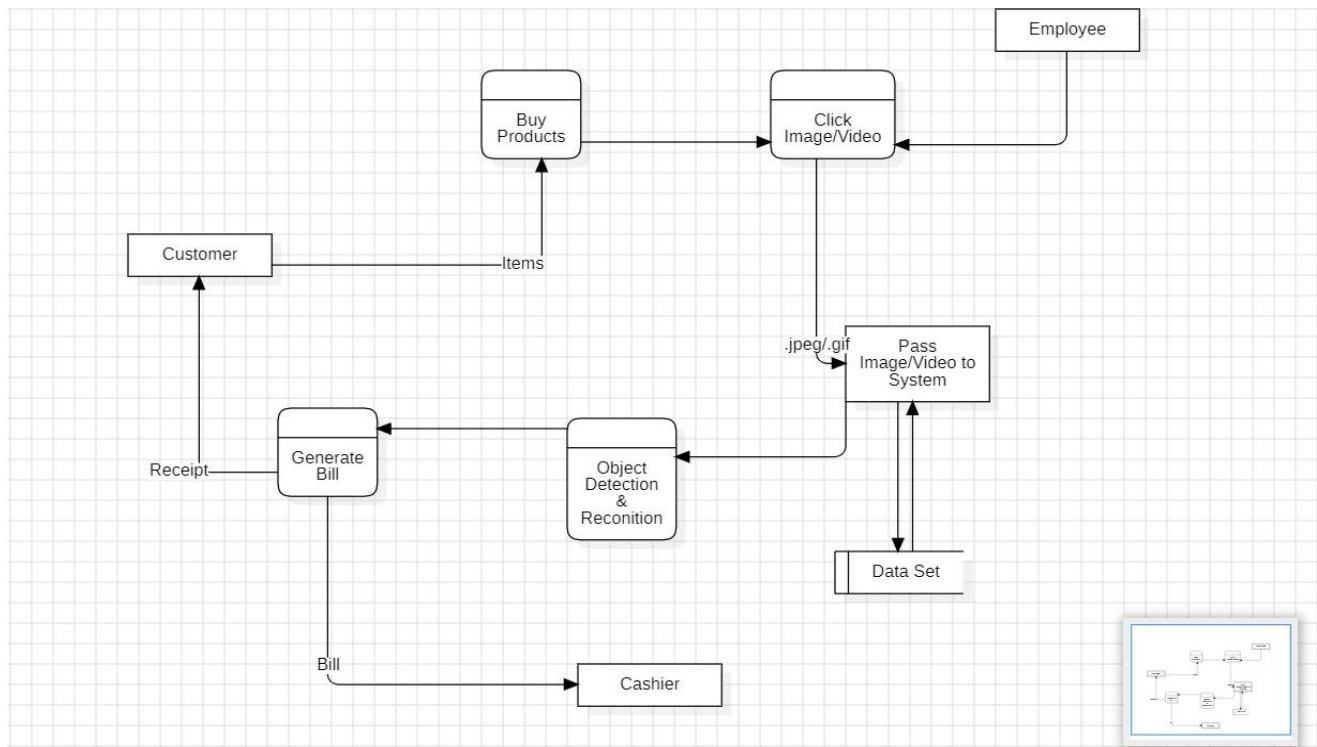


Figure 3 - 1 Level DFD

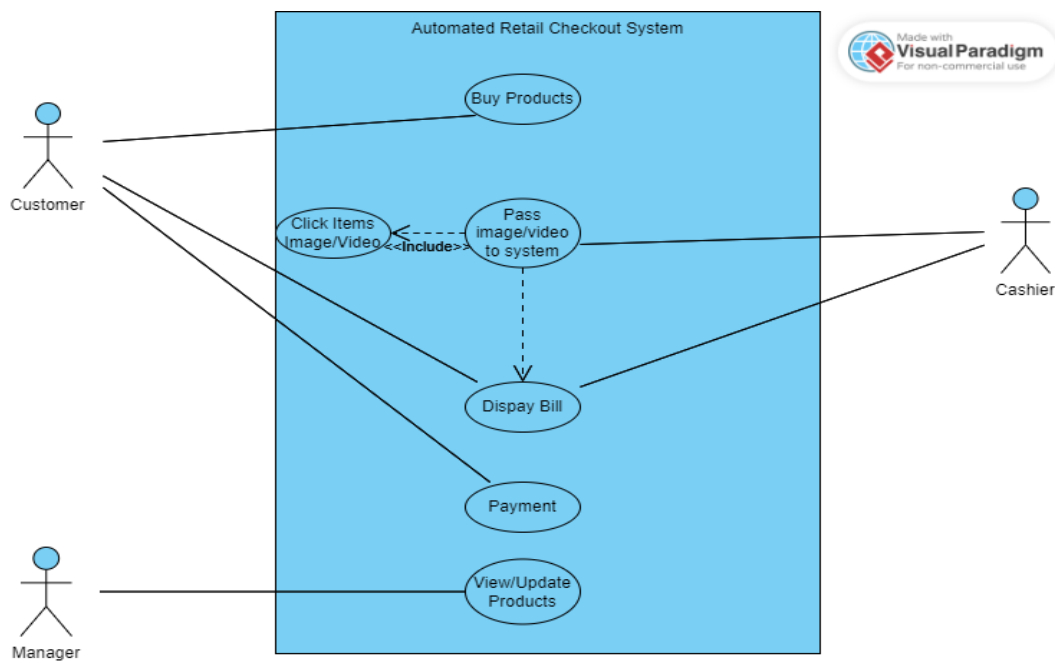
Use-Case Diagram:

Figure 4 - Use Case Diagram