

Software Requirements Specification (SRS) Document v2

Computer Vision Application for Real-time Multi-modal Product Detection

Team 33

Aanvik Bhatnagar, Chetan Mahipal, Badarla Rohan Naidu

Rohan Rathee, Rohan Shridhar

Brief problem statement

The current application has technical complexities and lacks a user-friendly interface for product detection using CV algorithms. The problem deals with creating a premium **user interface** which is readily deployable and shareable. To address this, the usability and robustness of the app must be enhanced. The solution we propose is an updated version of the app with self-explanatory steps of usage, which would be error-free, readily demonstrable, and flexible to future changes in code and design. The scope of development also includes optimizing the user interface, ensuring a seamless and responsive experience. Leveraging cloud computing, the app would provide swift results to users parallel and concurrently. The primary objective is to make this powerful CV capabilities accessible through a user-friendly mobile platform. We also aim to create extensive documentation for future reference.

System requirements

Frontend

- React-Native
- Node.js
- NPM
- Javascript

Backend

- Python 3.8
- Pip or Pip3
- Flask
- Pymongo
- Numpy
- Opencv-python
- Pillow
- Sk-video
- Matplotlib
- Torchvision
- Yacs
- Scikit-learn scipy matplotlib
- Scikit-image
- Flask-Cors

Rendering on Phone

- ExpoGo
- Apk

Cloud Service

- DigitalOcean
 - AWS
- The client has not decided yet which to use.

Minimum JDK of 21.0 in mobile phones.

Users profile

The Users can be divided into 4 categories which are:

1. **Potential Client:** The very first user of the application will be the clients of our clients, to whom they are pitching their project. The mode will be for demonstration with explaining all the functionalities. This user will be deemed as one that does not have any technical background, thus will be an ordinary person who is interested in basic functionalities.
2. **Retail Stores:** The next category that the client is aiming at is retail stores. This app will act as an inventory management system which helps in keeping the stock updates and need for restocking when required. The main users in this category will be a few top managers of the stores, updating stock daily. These users are deemed the same as above with a little knowledge of software but are handy in using applications on mobile devices.
3. **Delivery System Companies:** This category of users are the officials who will keep a check at the quantity of the product delivered by delivery personnel. The delivery personnel will share the image of delivered items which will then be processed by the user to match the quantity with requirements. The users are ordinary people with little knowledge of the system behind the work. They are daily mobile users.
4. **E-commerce Websites:** The users will be the officials who are managers of each category of product on the website. This will be used for digital cataloging on the website. To update with range of products on the website. The users will have information about the systems and can use and modify app according to their needs.

The app's intended use is not for mass users. The app will be used by a few top officials. Apart from the above users, they all will be required to have 1-2 technical members for updating the support set images according to their use. So, the team should comprise at least one member with knowledge of the software systems used to make user-specific changes.

Feature requirements (described using use cases)

No.	Use Case Name	Description	Release*
A. Home Page			
A1.	Scan Order	In the home page, there is an '(-)' button, on clicking user can add object which they want. The user will be redirected to the camera page.	R1
B. Order Description			
B1.	Identified Order(s) Table	Based on the CV Algorithm, display all the identified orders, and their detected quantities in a tabular format	R1
B2.	Quantity Modification	User can increase/decrease the required quantity of detected item(s).	
B2.1	By Plus/Minus buttons	User can modify the quantity by a unit value from clicking the plus/minus buttons adjacent to the displayed quantity.	R1
B2.2	By Keyboard	User can modify the quantity by any amount that they wish, by typing the quantity in the box by double clicking on the number field of that column.	R2
B3.	Image View	User can initially see the overview of the image that they initially sent into the algorithm at top-center, on clicking the image, they can view the full screen image.	R1
C. Camera Page			
C1.	Camera Options	User can open the camera, and click a photo to be processed.	R1
C2.	Video Options	User can send a video and get the results displayed.	R2
C3.	Upload from Gallery	User can scan more images for a particular order, and the newly scanned item(s) will also get	R2

		appended in the existing order. User can add the image from the local device's gallery	
D. Backend			
D1.	Addition of Support Set Images	Add support set images for computer vision algorithms to work on and produce results.	Opt
D2.	Updating Images	Update support set images for better results and outcomes of the app or in-case of removal of that product, delete the images.	Opt
D3.	Database	Addition of user profile data, and order specifications in the database.	R2
D4.	Deployment	Deploy the front-end and backend codebase on the cloud service provider (AWS).	Opt

*R1: Release 1 (Mid-March); R2: Release 2(Mid-April); Opt: Optional

Changes Made:

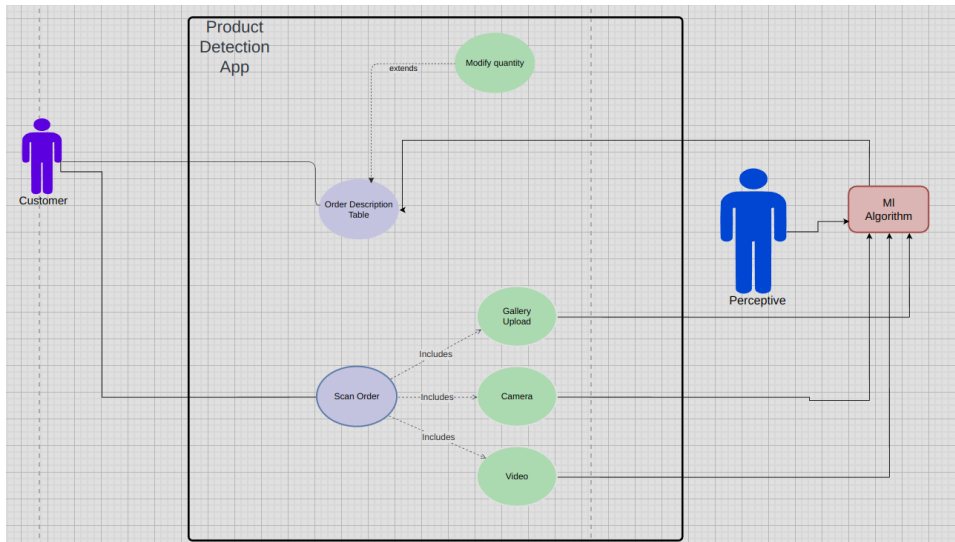
- a. v2, 6th March 2024: Removed the Login Page, Settings Page, and Profile Page functionalities as client required us to only focus on Home Page and Items Page, removed additional requirements from Home Page and only kept a scan button functionality in it. Added the upload from gallery feature to the Camera page.

Main Flow:

1. User enters the Home Page, where user can see their scanned or unscanned orders.
2. If the user wants to scan a new order, they can click on the Scan Button on the bottom center of the screen, and a camera option will appear, where the user chooses whether to take a video or click a picture for order processing, or upload from gallery.
3. After the scan is complete, the user can see all the data collected from the CV algorithm in a tabular format.
4. The user can see the options to adjust the required quantity of any item by either clicking on the arrows, or by entering the quantity by keyboard.

Use case diagram

Link to UML diagram: https://drive.google.com/file/d/1cgaGXLaf3N-Q9KOMdkcoBrsFAT_vms8b/view?usp=sharing



Use case description

Use Case Number:	UC-A: UC-A1
Use Case Name:	Home Page: - Scan Order
Overview:	It is the basic page for the whole app. User can access the scan functionality of the app by clicking on the scan button.
Actors:	Any Customer
Pre-condition:	You must be operating the app on your phone after proper set up.
Flow:	Main Flow: <ol style="list-style-type: none"> 1. Users enters the app, they see the Welcome Page logo, along with the scan button at bottom center. 2. Users can go to Camera page by clicking on the Scan button.
	Alternate Flows: N.A.
Post Condition:	Now you can go to the items page, capture images, anything you want can be done from here. In other words, it's a primary page.

Use Case Number:	UC-B: UC-B1 to UC-B3
Use Case Name:	Order Description: <ul style="list-style-type: none"> - Identified Order(s) Table - Quantity Modification (Plus/Minus or Keyboard) - Image View

Overview:	Details of a particular scanned item are displayed here. Users can modify the quantities of detected items from the CV Algorithm, and accordingly update the quantity of order.
Actors:	Returning Customer, Backend Developer (Perceptive Analytics)
Pre-condition:	Successful login, camera permission given to the app on local device, support set images provided by the Backend Developer for CV Algorithm to operate on. App should also be given permission to access local device's gallery.
Flow:	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. When the user scans a particular order, the result gets displayed on the Items page from the CV Algorithm. 2. User can see the quantities of items detected, along with the picture that was scanned.
	<p>Alternate Flows:</p> <ol style="list-style-type: none"> 1. Quantity Modification <ol style="list-style-type: none"> 1.1. The user wants to change the quantity required by a few increment/decrements; they can use the arrow keys on the side of the required number. 1.2. The user wants to change the required quantity by a huge margin, so they double click on the number field, and a form box gets opened, where user can input natural number values. 2. Image View: <ol style="list-style-type: none"> 2.1. The user clicks on the image/video that is displayed above the table. 2.2. The image/video is displayed in a full screen to the user. 2.3. The user presses back button on their phone, and goes back to the table.
Post Condition:	User's order gets displayed and modified as per the wish of the user.

Use Case Number:	UC-C: UC-C1 to UC-C3
Use Case Name:	<p>Camera Page:</p> <ul style="list-style-type: none"> - Camera Options - Video Options - Add Order Image(s) from Gallery
Overview:	In this page, users can capture the photo or video and can upload it and Items page will do its formalities
Actors:	Any Customers who want to add items can access this page

Pre-condition:	You will enter this page from the home page.
Flow:	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. There will be three options – photo/video/upload. 2. You can choose one of photo or video. 3. After that you can upload the image if you want so
	<p>Alternate Flows:</p> <ol style="list-style-type: none"> 1. Camera Mode <ol style="list-style-type: none"> 1.1. The user wants to switch to Video Mode. 1.2. The user has to long press the Camera Button to go into Video Mode. 1.3. Now user can take the video and release the button when done. 2. Upload Image: <ol style="list-style-type: none"> 2.1. The user can click on the upload button in the bottom right part of the Camera Page. 2.2. If the user wants to go back to Camera Page, they have to click back button on the phone.
Post Condition:	Now for this image, the ML code will access this image and will give the description of its properties. You can see all its info for its corresponding item page

Use Case Number:	UC-D: UC-D1 to UC-D4
Use Case Name:	<p>Backend:</p> <ul style="list-style-type: none"> - Addition of support set images - Updating Images - Database - Deployment
Overview:	It involves updating the database quantities like user and order specifications. Addition and updating of support set images for the working of the computer vision algorithms also comes under the backend. It also involves deploying the front-end and back-end on a cloud service (AWS).
Actors:	Backend Developers (Perceptive Analytics)
Pre-condition:	The CV algorithms work properly, the company has a database on the cloud service.
Flow:	<p>Main Flow:</p> <ol style="list-style-type: none"> 1. App user uploads an image 2. Developer accesses the database

	<p>3. Developer makes changes to the database through a database management system (DBMS)</p> <p>4. The CV algorithm runs on the image and the results are stored</p> <p>4. The changes are reflected in the cloud server.</p>
	<p>Alternate Flows:</p> <p>1. Developer checks the support set images used to train the CV Algorithm, and update it as needed.</p> <p>2. Results of the comparison are shown to the user.</p>
Post Condition:	CV algorithms use the new data to increase accuracy.