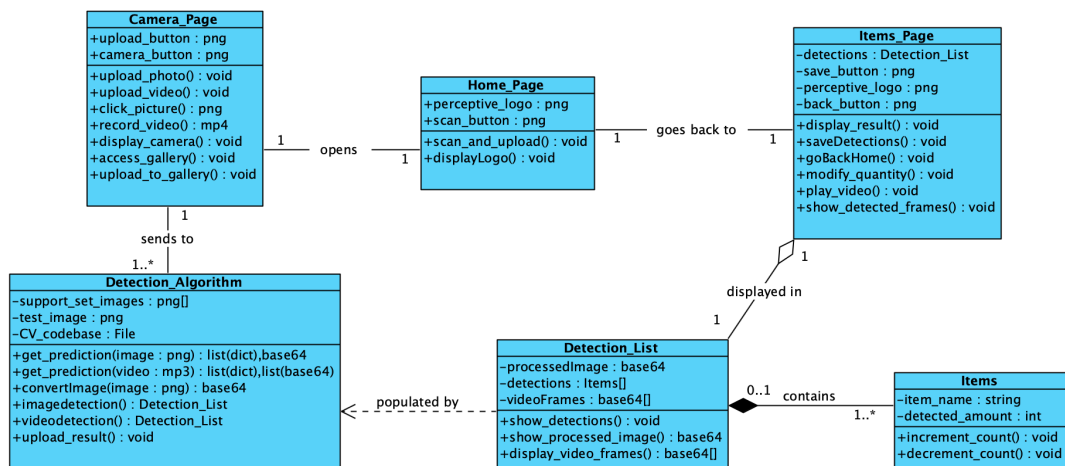


# Product Design v3

## Team 33: Multimodal Product Detection Mobile App

Aanvik Bhatnagar  
Chetan Mahipal  
Badarala Rohan Naidu  
Rohan Shridhar  
Rohan Rathee

### Design Model



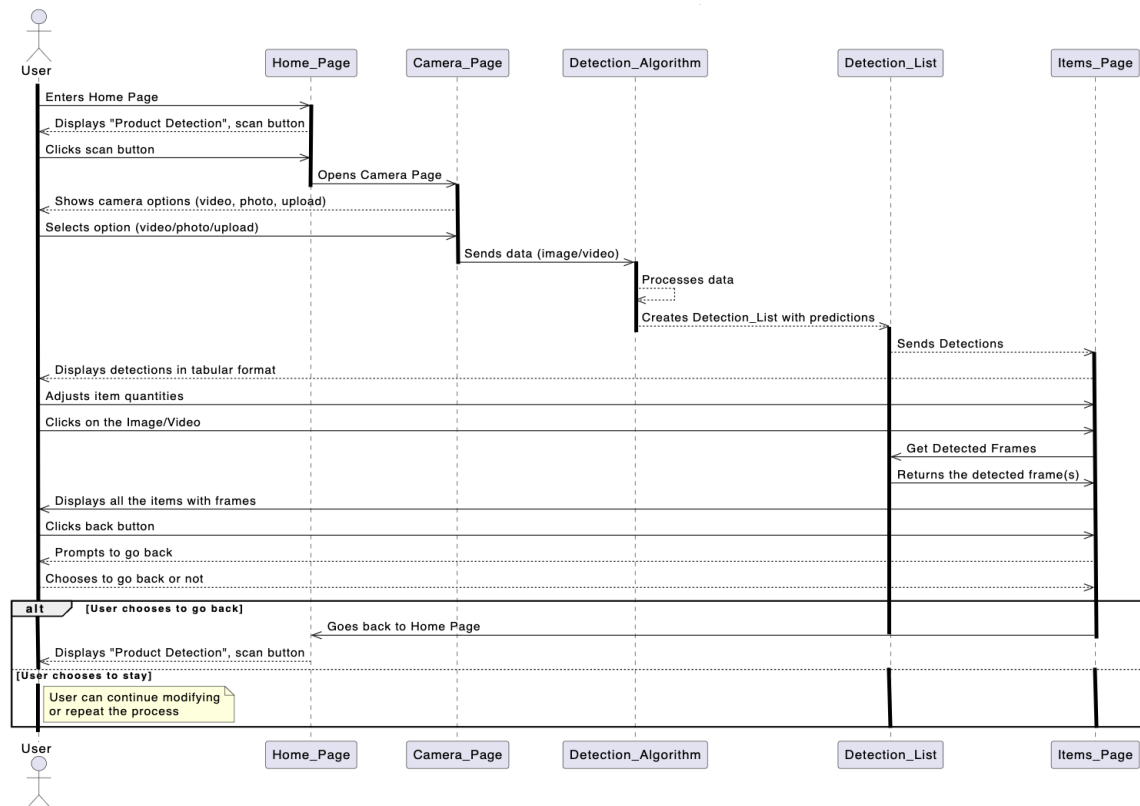
### Class Description

Home_Page	<p><b>Class state</b></p> <ul style="list-style-type: none"><li>The logo of the company: Perceptive Analytics</li><li>Image of the scan button (displayed at center bottom of the page)</li></ul> <p><b>Class behavior</b></p> <ul style="list-style-type: none"><li>On clicking the scan button, the camera page opens</li></ul>
Camera_Page	<p><b>Class state</b></p> <ul style="list-style-type: none"><li>Upload button for gallery upload</li><li>Camera button to take image/video</li></ul> <p><b>Class behavior</b></p> <ul style="list-style-type: none"><li>On long pressing the button at the bottom, a photo is taken.</li><li>On clicking the button once, the video recording starts, and on clicking it again, the video recording stops.</li></ul>

	<ul style="list-style-type: none"> <li>On clicking the upload button, the device gallery opens, and the user can select any image or video from the gallery, which will be uploaded and later processed.</li> </ul>
Detection_Algorithm	<p>Class state</p> <ul style="list-style-type: none"> <li>Support set images are associated with the detection algorithm.</li> <li>The test (or scanned) image is processed by the detection algorithm.</li> <li>The CV algorithm which is trained for product detection (provided by the client)</li> </ul> <p>Class behaviour</p> <ul style="list-style-type: none"> <li>The scanned image is retrieved by the algorithm for processing.</li> <li>The algorithm runs to identify different items in the image.</li> <li>The result of the detection is returned as properties of the identified items.</li> <li>Video is processed frame by frame, and images are converted into a suitable format (base64).</li> <li>Results are also uploaded to Detection_List class from here.</li> </ul>
Items	<p>Class state</p> <ul style="list-style-type: none"> <li>This class is identified by the name of the item.</li> <li>The scanned amount of the item is also an attribute of this class.</li> </ul> <p>Class behaviour</p> <ul style="list-style-type: none"> <li>On clicking the '+' button on the right of the item quantity, the quantity is increased by 1.</li> <li>On clicking the '-' button on the left of the item quantity, the quantity is decreased by 1.</li> </ul>
Detection_List	<p>Class state</p> <ul style="list-style-type: none"> <li>Images of scanned items are stored here.</li> <li>The scanned items names are stored here. Stored as a list of Items class.</li> <li>The detected quantities of scanned items are stored here, along with the list of frames from a video.</li> </ul> <p>Class behavior</p> <ul style="list-style-type: none"> <li>The items are queried, which is returned and displayed (on item's page).</li> </ul>
Items_Page	<p>Class state</p> <ul style="list-style-type: none"> <li>This class maintains the image and the items table, which consists of the items' names and scanned quantities.</li> <li>It also consists of back button and save button.</li> </ul> <p>Class behavior</p>

	<ul style="list-style-type: none"> <li>• Popping up a full screen view of the image, when clicked. This image shows the results of the detection algorithm, as grids around the detected objects.</li> <li>• Going back to the home page on clicking the back button, with a pop up appearing to confirm whether the user wants to go back or not.</li> <li>• Saving the detections with a pop up appearing, and then redirecting to home page.</li> </ul>
--	--

## Sequence Diagram(s)



## Design Rationale

### Camera Page

The first proposed design of the camera page has two options, both for video and image. It could be switched by just clicking on the desired option. But after deliberation with the client, we switched the design with a long press click for image capture and to record video press the capture button once (to start recording) and to finish recording press the capture button again. According to the client this was more user-friendly and intuitive.

## **Home Page**

The first proposed design contained a list of orders, with options for the creation of new orders and editing orders. But the purpose of the app was for giving demos to our client's client, so we switched to just the homepage with a scan button which was more intuitive and fulfilling requirement.

## **Items Page**

As the use-case was just for the demo purpose, we thought to implement and show the list of detected items only, with an ability of changing the quantity of the detected item for correcting additional scans (due to algorithm). We agreed to have a picture with scanned items for better usability and to get the user information about the location of scanned items. We thought to store the images and implement an option for adding images, but the client suggested making it a single database entry every time for better differentiation between different scans. The back button is placed at the bottom of the page because most people tend to operate the app with their thumb only and it can be accessed easily. Similar rationale applies for save button, and we have kept it in centre so that accidental clicking of save button is avoided while holding it from sides.

## **Overall UI**

The overall colour palette was chosen keeping the logo of the company and using complement colours for enhancement of the design. The circular design captures the logo in a beautiful manner.

## **Detection List**

The detection list consists of the results after scanning. All the items after getting scanned are returned with item's name, image of scanned (marks with detection in image) and the quantity. On loading item's page, these entries are displayed accordingly on the page.

## **Items List View**

We decided to use scrolling in the item's list because displaying 7-9 items and then changing pages takes time. Scrolling would be fast and efficient.