**Software Requirements Specification (SRS) Document**

|  |  |
| --- | --- |
|  | 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)

**Read the instructions below and fill in the table. Delete all the blue text turning it in.**

(This is a numbered list of use cases that are the features of the system to be implemented. Each use case is an operation that the user can perform on/with the system. For each use case, provide a description (2-3 sentences) so you know what to build and so you can write a test case to demonstrate that your system provides that feature. For each use case, you will identify (during release planning) the release in which it will be implemented: R1 or R2. Typically, your project will have 10-15 use cases, but feel free to add or delete table rows if you decide to use finer-grain or coarse-grain use cases).

|  |  |  |  |
| --- | --- | --- | --- |
| **No.** | **Use Case Name** | **Description** | **Release** |
| User Management | | | |
|  | Create account | Users can create account by adding email, password, and username | R2 |
|  | Login | Users can go into the page with their login credentials, to experience this app | R2 |
|  | Forgot password | User can change the password in case they forget | R2 |
| Home Page | | | |
|  | Displaying objects in home page | In the home page, items are being shown which are scanned to easily access them | R1 |
|  | Bookmark | For each item, bookmark button is present so that user can separate which are useful and which aren’t | R1 |
|  | Filter | Users can filter with respect to ID or date or status for finding the item easily | R1 |
|  | Description of object | For each item its properties also shown like its ID, created date and status of it whether it is detected or not | R1 |
|  | Add objects | In the home page, there is an ‘ + ’ button, on clicking user can add object which they want | R1 |
|  | Nav bar for the app | On every page, nav bar is shown with 5 items: Home page, search, scan, settings, profile | R1 |
|  | Search item | There is a search option in nav bar so that user can find item much efficiently than above | R2 |
|  | Settings | There is a setting page from which you can enhance your app responsive to you | R2 |
|  | Profile | Your information will be showed in this page like your name, email, account created on and logout | R2 |
|  | Scanning image | You can capture “image or video” and it will detect the items in the image in its corresponding item page | R1 |
| Order Description | | | |
|  | Identified Order(s) Table | Based on the CV Algorithm, display all the identified orders, and their detected quantities in a tabular format | R1 |
|  | Quantity Modification | User has the option to increase/decrease the required quantity of detected item(s) | R1 |
|  | Quantity Status | Based on the required and detected numbers generated, display a status message for the user to understand easily | R1 |
|  | Save Order Changes | If the user wants to make changes in any existing order details, and save them, app will save changes | R1 |
|  | Share Current Order | User can share the order on multiple platforms outside the app | R2 |
|  | Copy Current Order | User can create another copy of an existing order, and perform operations as required on the copy | R1 |
|  | Delete Current Order | User can delete existing order and displayed items in the list | R1 |
|  | Add Order Image(s) | User can scan more images for a particular order, and the newly scanned item(s) will also get appended in the existing order | R1 |
| Settings page | | | |
|  | Edit profile | User can change their profile, like their name, email etc. | R2 |
| 20 | Edit Items | User can edit any information about the item | R2 |
| 21 | Logout | User can logout from the page | R2 |
| 22 | Delete account | User can delete the contents which he/she thinks are unnecessary. | R2 |
| Camera Page | | | |
| 23 | Options | User can change the camera mode into photo/video | R1 |
| Backend | | | |
| 24 | User Login | Facilitating smooth user login and updating database accordingly | R2 |
| 25 | Addition of Support Set Images | Add support set images for computer vision algorithms to work on and produce results. | R2 |
| 26 | 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. | R2 |
| 27 | Database | Addition of user profile data, and order specifications in the database. | R2 |
| 28 | Deployment | Deploy the frontend and backend codebase on the cloud service provider (AWS). | R2 |

**Main Flow:**

1. User logins by entering their username and password.
2. User enters the Home Page, where user can see their scanned or unscanned orders.
3. If the user wants to search from the existing scanned/unscanned orders, then they can click on the search button in the navigation bar and type the required order number.
4. If the user wants to filter from the existing scanned/unscanned orders based on certain criteria, they can go to the filter bar, select the desired option, and get the filtered order(s).
5. If the user wants to scan a new order, they can click on the Add Button on the top right corner, and a camera option will appear, where the user chooses whether to take a video or click a picture for order processing.
6. After the scan is complete, the order gets added onto the home page, and the user can see all the data collected from the CV algorithm in a tabular format.
7. When the user clicks on a particular order on the Home Page, they get directed to the order details, where they 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.
8. User can also see the status of item recognized based on quantity detected vs required by looking at the Status Option next to every detected item.
9. User can copy the selected order, share the selected order, and save the changes made to the order by clicking on the respective buttons on the dropdown at the top right corner of the app.
10. From this page, user can be directed to adding a new image to the existing order, and appending the changes by clicking on the Add Item button at the centre of the navigation bar below.
11. User can also be directed back to Home Page, Settings Page and Profile Page by clicking on the respective buttons on the navigation bar.
12. User can also view their details in the Profile Page.
13. After performing the required tasks, user can log out by going to the Settings Page, and clicking the log out option.

**Use case diagram**

**Read the instructions below and fill in the table. Delete all the blue text before adding this to your repository or turning it in to your instructor.**

Draw the UML use case diagram for the system. Make sure the use cases shown in the diagram correspond to the use cases described in the previous section.

**Use case description**

**Delete all the blue text and fill-in the template before adding this to your repository or turning it in to your instructor.**

|  |  |
| --- | --- |
| **Use Case Number:** | UC-XX (Replace XX with a number) |
| **Use Case Name:** | Enter the name of Use Case |
| **Overview:** | Describe the purpose of the Use Case and give a 1-2 line description. This could be the same as the description provided in feature requirements section. |
| **Actors:** | List all actors that participate in this Use Case. |
| **Pre condition:** | Enter the condition that must be true before the main flow is executed. |
| **Flow:** | Main (success) Flow: Steps should be numbered. |
|  | Alternate Flows: Include the post condition for each alternate flow if different from the main flow. |
| **Post Condition:** | Enter the condition that must be true when the main flow is completed. |