**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 a new 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

*Replace this text and the instructions below with your statement in black.*  
(Identify who will be using the system, in what mode, and their profile in terms of familiarity with using computers and such software).

# 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.** | **User Case Name** | **Description** | **Release** |
| User Management | | | |
|  | Create account | Users can create account by adding email, password and username | R1 |
|  | Login | Users can go into the page with their login credentials, to experience this app | R1 |
|  | 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 you can add object |  |
|  |  |  |  |
|  | 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 | R1.5 |
|  | 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 |

**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. |