

**University Institute of Information Technology,**

**PMAS-Arid Agriculture University,**

**Rawalpindi Pakistan**

**VisionForBlinds**

***By***

**Anum Areej 19-ARID-785**

**Muhammad Komail Abbas Khan 19-ARID-832**

**Muhammad Rakib Nadeem 19-ARID-835**

***Supervisor*Mr. Suleman Khurram**

***Bachelor of Science in Computer Science (2019-2023)***

**The candidate confirms that the work submitted is their own and appropriate  
 credit has been given where reference has been made to the work of others**.

**DECLARATION**

We hereby declare that this software, neither whole nor as a part has been copied out from any source. It is further declared that we have developed this software documentation and accompanied report entirely on the basis of our personal efforts. If any part of this project is proved to be copied out from any source or found to be reproduction of some other. We will stand by the consequences. No Portion of the work presented has been submitted of any application for any other degree or qualification of this or any other university or institute of learning.

Anum Areej Muhammad Komail Abbas Khan Muhammad Rakib Nadeem

(19-ARID-785) (19-ARID-832) (19-ARID-835)

--------------------------- --------------------------- ---------------------------

**CERTIFICATE OF APPROVAL**

It is to certify that the final year project of **BS (CS)** **“VisionForBlinds”** was developed by **“Anum Areej (19-ARID-785)”**, **“Muhammad Komail Abbas Khan (19-ARID-832)”** and **“Muhammad Rakib Nadeem (19-ARID-835)”** under the supervision of **“Mr. Suleman Khurram”** and that in their opinion; it is fully adequate, in scope and quality for the degree of Bachelors of Science in Computer Science.

---------------------------------------

**(Mr. Suleman Khurram)**

**Supervisor**

---------------------------------------

**External Examiner**

---------------------------------------

**Administrator UIIT**

**Executive Summary**

People see their surroundings through the beautiful sense of sight that is gifted to them. But some people have lost this sense either because of any ailment or due to any other reasons. It becomes difficult for them to live their rest of lives with this disability. They cannot enjoy their lives and perform their daily life activities on their own. They need someone else’s aid to guide them in certain activities and tasks. Due to advancements in technology, the lives of such people have now been eased to some extent. These people can use their smartphones and perform various daily life activities and tasks easily. To create more ease for people having any kind of visual impairment, we have built a mobile application that has various utilities and modules. This mobile application titled **“VisionForBlinds”** will guide its visual impaired users in the best possible way.

Our mobile application **VisionForBlinds** is an Artificial Intelligence based project that would assist its blind and visual impaired users in various ways. Our application has used the technology of **React Native** for developing UI for mobile. Since it is an AI based mobile application, we have used various techniques of Artificial Intelligence, mostly Deep Learning techniques like **Convolutional Neural Networks (CNN)** and **Optical Character Recognition (OCR)** for its utilities and modules. Our mobile application has three important modules including **Image Captioning** and **OCR Detection**, **Currency Detection** and **Color Detection**. If the user wants to use these utilities, they will have to grant camera access to its mobile phone. The proposed solution has used various datasets for training and to predict the best possible outcomes for its users. Since we have developed this application for visually impaired people, we have incorporated the feature of **text-to-speech** with all the working modules.

**Acknowledgement**

All praise is to Almighty Allah who bestowed upon us a minute portion of His boundless knowledge by virtue of which we were able to accomplish this challenging task.

We are greatly indebted to our project supervisor **“Mr. Suleman Khurram”** for personal supervision, advice, valuable guidance and completion of this project. We are deeply indebted to him for encouragement and continual help during this work.

And we are also thankful to our parents and family who have been a constant source of encouragement for us and brought us the values of honesty & hard work.

Anum Areej Muhammad Komail Abbas Khan Muhammad Rakib Nadeem

(19-ARID-785) (19-ARID-832) (19-ARID-835)

--------------------------- --------------------------- ---------------------------

**Abbreviations**

|  |  |
| --- | --- |
| **FYP** | Final Year Project |
| **CNN** | Convolutional Neural Networks |
| **OCR** | Optical Character Recognition |
|  |  |
|  |  |

**Table of Contents**

**Introduction 1**

[1.1 Brief](#_Toc268523777) 1

[1.2 Relevance to Course Modules](#_Toc268523779) 2

[1.3 Project Background](#_Toc268523780) 3

[1.4 Literature Review](#_Toc268523780) 4

[1.5 Analysis from Literature Review](#_Toc268523780) 6

[1.6 Methodology and Software Life Cycle](#_Toc268523782) 6

[1.6.1 Rationale behind Selected Methodology](#_Toc268523787) 7

**Problem Definition** **9**

[2.1 Problem Statement](#_Toc268523795) 9

[2.2 Deliverables and Development Requirements](#_Toc268523796) 10

[2.3 Current Systems](#_Toc268523804) 12

**Requirement Analysis** 15

[3.1 Functional Requirments](#_Toc268523823) 15

[3.2 Non – Functional Requirments](#_Toc268523825) 16

[3.2.1 Usability](#_Toc268523787) 16

[3.2.2 Performance](#_Toc268523787) 16

[3.2.3 Reliability](#_Toc268523787) 16

[3.2.4 Availability](#_Toc268523787) 17

[3.2.5 Design Constraints](#_Toc268523787) 17

[3.3 Use cases](#_Toc268523823) 17

[3.3.1 Use Case Diagarm](#_Toc268523787) 17

[3.3.2 Use Case and Actors Description](#_Toc268523787) 18

**The Design**

[4.1 UML Structural Diagrams](#_Toc268523830)

[4.1.1 Component Diagram](#_Toc268523787)

[4.1.2 Package Diagram](#_Toc268523787)

[4.1.3 Deployment Diagram](#_Toc268523787)

[4.2 UML Behavioral Diagrams](#_Toc268523830)

[4.2.1 Activity Diagrams](#_Toc268523787)

[4.2.2 State Machine Diagrams](#_Toc268523787)

[4.3 UML Interaction Diagrams](#_Toc268523830)

[4.3.1 Sequence Diagrams](#_Toc268523787)

[4.4 Node Structure](#_Toc268523830)

[4.5 Communication Design Protocol](#_Toc268523830)

**Implementation**

[5.1 Communication Protocol Implementation](#_Toc268523830)

[5.2 PC Application Implementation](#_Toc268523830)

[5.3 Embedded Application Implementation](#_Toc268523830)

[5.4 Wireless Sensor Application Implementation](#_Toc268523830)

**Testing and Evaluation**

[6.1 Verification](#_Toc268523830)

[6.1.1 Functional Testing](#_Toc268523787)

[6.1.2 Static Testing](#_Toc268523787)

[6.2 Validation](#_Toc268523830)

[6.3 Usability Testing](#_Toc268523830)

[6.4 Unit Testing](#_Toc268523830)

* 1. [Integration Testing](#_Toc268523830)

6.6 [System Testing](#_Toc268523830)

**GUI**

**Future Work**

**References**

**List of Figures**

Fig 1.4.1 ColorBlindPal 5

Fig 1.4.2 Cash Reader 5

Fig 1.4.3 KNFB Reader 5

Fig 1.6.1 Agile Methodology Process Diagram 7

Fig 1.6.2 Agile Values 8

Fig 2.3.1 Current System-1 12

Fig 2.3.2 Current System-2 13

Fig 2.3.3 Current System-3 13

Fig 3.3.1 Use Case Diagram 17

**List of Tables**

Table 2.2.1 Deliverables 10

Table 2.2.2 Development Requirements 11

Table 2.3.1 Current Systems Summary 14

Table 3.1.1 Choose Module……………………………………………………………………...18

Table 3.1.2 Input Image………………………………………………………………………….18

Table 3.1.3 Open Camera………………………………………………………………..............18

Table 3.1.4 Caption Image……………………………………………………………………….19

Table 3.1.5 Detect OCR……………………………………………………………….................19

Table 3.1.6 Detect Currency Note……………………………………………………………….20

Table 3.1.7 Detect Color………………………………………………………………................21

Table 3.1.8 Generate Output……………………………………………………………………..22

# Chapter 3: Requirement Analysis

In our daily lives, we encounter several different new products and services that are created in response to the customers/ user’s needs. However, despite spending tremendous time and resources on development, there can be a mismatch between the desired product and the final product. Hence, there is a need for focused and detailed requirement analysis in the early stages of any project to avoid any major problems in the future development. Requirement Analysis or Requirement Engineering is one of the most critical steps in software development.

In this chapter, we discussed all the functional and non-functional requirements of our proposed solution. We also discussed about the use cases of our system and how the system will respond to different use cases.

# Functional Requirements

Functional requirements are what a system is supposed to accomplish. It tells which functions need to be provided to its target users.

The functional requirements of our proposed solution are mentioned below:

1. **Image Captioning**

The user will have to place camera of its mobile to any image and will show the captions about what’s happening and what’s shown in the image through the feature of text-to-speech. This functionality will use both CNN and RNN based models by restructuring VGG-16 model. Our image captioning model will be trained using different datasets like Flickr-8k and then it will predict the expected output for any image.

1. **OCR Detection**

The user just has to point its camera to any image, billboard or any other surrounding having any text written on it. The OCR functionality will extract text from that image and will speak out the output through text-to-speech feature. The mobile application will use this functionality after training the model using different datasets like MNIST. After successful training of datasets, our model will predict the output and will extract the text from any image.

1. **Currency Note Detection**

The user will also have to use camera to point towards the currency note (Pakistani currency note). Firstly, its amount will be detected using training datasets based on notes of PKR 10, 50, 100, 500, 1000 and 5000. The trained model will detect the amount of Pakistani currency note and will speak out the amount using text-to-speech. This functionality will be used after training dataset for Pakistani currency notes.

1. **Color Detection**

The user will point its camera to detect color from whatever image they want to. For this purpose, datasets are trained using different colors along with their color names for different images. For this purpose, we will be using datasets having RGB values mapped with their color names. After complete training of our model, the user will feed images to the mobile application and its expected color will be predicted and its output will be spoken to the user.

# Non-Functional Requirements

Non-functional requirements specify the criteria in the operation and the architecture of the system. Non-functional requirements are often called quality attributes of a system. Other terms for non-functional requirements are qualities, quality goals, and quality of service requirement, constraints, non-behavioral requirements, or technical requirements.

1. **Usability**

This software will be easy to use for its users with minimal instructions. 100% of the languages on the graphical user interface (GUI) shall be intuitive and understandable by non-technical users. The visually impaired persons just have to point their mobile’s camera to get a particular output about their surroundings. The ease with which the user can operate the mobile application and interpret outputs through interaction with a system. The system should be user friendly for every person whether they are from technical background or not. User Interface issues such as accessibilities, aesthetics, and consistency this will include that our project or system is easily accessible to the desired target audience.

1. **Performance**

This software shall minimize the number of calculations needed to perform image captioning, OCR Detection, Currency Detection and Color Detection. Performance measures the throughput, response time, recovery time, and start-up time. In case of failure how quickly it will recover and start working again. Defines how fast a software system or its piece responds to certain users’ actions under certain workload. Minimum execution time when the user holds the mobile’s camera in order to detect images and ending when the image has been detected successfully and showing suitable output like captions, currency note detection and color detection through text-to-speech feature.

1. **Reliability**

Our mobile application will be operable in all lighting conditions. Regardless of the brightness level in user’s operating environment, the program shall always detect user’s input image and show appropriate outputs. The extent to which the software system consistently performs the specified functions without failure.

1. **Availability**

As our platform is a mobile application, the users just have to download it first and then it will be available 24/7 to them. Our mobile application shall be available 27/7 of time to handle concurrent requests of its intended users.

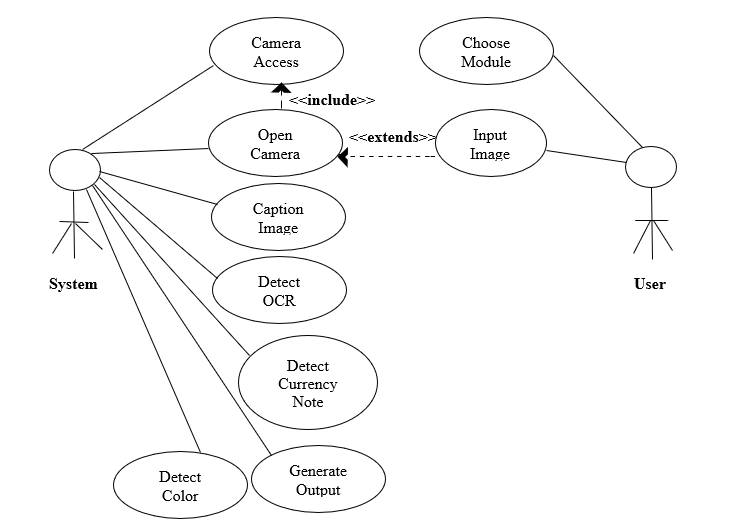
1. **Design Constraints**

* The application must run on both Android and iOS devices.
* The application must meet accessibility standards.

# Use Cases

In Unified Modelling Language (UML), a use case diagram can summarize the details of our mobile application’s users (also called actors) and their interaction with the mobile application. Following are the use cases of our mobile application “VisionForBlinds”.

1. **Use Case Diagram**

****

**Figure 3.1.1: Use Case Diagram**

**3.3.2 Use Case and Actors Description**

**Table 3.1.1: Choose Module**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.1 |
| **Use Case Name:** | Choose Module |
| **Actors:** | User |
| **Description:** | User will open the mobile application to choose module. |
| **Preconditions:** | System must be connected to the network. |
| **Postconditions:** | After choosing module, the user will proceed further. |
| **Normal Flow:** | 1. User opens the system/ mobile application. 2. User choose module from Home Screen. |

**Table 3.1.2: Input Image**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.2 |
| **Use Case Name:** | Input Image |
| **Actors:** | User |
| **Description:** | User provides input image to the system for its further processing. |
| **Preconditions:** | The user must have to choose one of modules shown on Home Screen in order to proceed further with processing. |
| **Postconditions:** | When the user provides input image from gallery, the system will start applying different models according to suitable module selected. |
| **Normal Flow:** | User provides input image from gallery and the system will start processing the input image. |
| **Exceptions:** | The input image provided by the user cannot be found or unable to fetch. |

**Table 3.1.3: Open Camera**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.3 |
| **Use Case Name:** | Open Camera |
| **Actors:** | User |
| **Description:** | User must open the mobile camera to capture image. |
| **Preconditions:** | The user must have to choose one of modules shown on Home Screen in order to proceed further with processing. |
| **Postconditions:** | When the user opens mobile camera, the system will start applying different models according to suitable module selected. |
| **Exceptions:** | The user is unable to provide image to the application. |

**Table 3.1.4: Caption Image**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.4 |
| **Use Case Name:** | Caption Image |
| **Actors:** | System |
| **Description:** | When user provides input image or open mobile camera, the system will process the input by the user by applying image captioning model. |
| **Trigger:** | User provides input image or open mobile camera. |
| **Preconditions:** | User opens mobile camera or provide input image, then the system will start captioning the image. |
| **Postconditions:** | After applying image captioning model, the system will generate a spoken output using Text-To-Speech feature. |
| **Normal Flow:** | 1. User chooses the module. 2. User opens mobile camera or provide input image. 3. System applies image captioning model based on CNN and RNN. 4. System will generate spoken output using Text-To-Speech. |
| **Alternative Flow:** | 1. User chooses module. 2. User unable to provide input image from gallery because of no image file or unable to open mobile camera. 3. System unable to apply image captioning model due to no input image. 4. User have to again try opening the mobile camera or try providing an input image again. |
| **Exceptions:** | If user unable to provide input image or open mobile camera, there will be a message showing “No input image provided” or “Unable to open mobile camera”. |
| **Assumptions:** | 1 User providing inputs correctly or mobile camera working  properly.  2 System correctly applying model and predicting the expected  output. |

**Table 3.1.5: Detect OCR**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.5 |
| **Use Case Name:** | Detect OCR |
| **Actors:** | System |
| **Description:** | When user provides input image or open mobile camera, the system will process the input by the user by applying suitable OCR model trained on a dataset. |
| **Trigger:** | User provides input image or open mobile camera. |
| **Preconditions:** | User opens mobile camera or provide input image, then the system will start applying OCR detection model trained on a dataset. |
| **Postconditions:** | After applying trained model for OCR detection, the system will generate a spoken output using Text-To-Speech feature. |
| **Normal Flow:** | 1 User chooses the module.  2 User opens mobile camera or provide input image.  3 System applies model trained for OCR detection.  4 System will extract text from any image.   1. System will generate spoken output using Text-To-Speech. |
| **Alternative Flow:** | 1. User chooses module. 2. User unable to provide input image from gallery because of no image file or unable to open mobile camera. 3. System unable to apply model trained on any dataset for OCR detection due to no input image. 4. User have to again try opening the mobile camera or try providing an input image again. |
| **Exceptions:** | 1 If user unable to provide input image or open mobile camera,  the system will display a message showing “No input image  provided” or “Unable to open mobile camera”.   1. System unable to extract text from input image due to any error. |
| **Assumptions:** | 1 User providing inputs correctly or mobile camera working  properly.  2 System correctly applying model and predicting the expected  output. |

**Table 3.1.6: Detect Currency Notes**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.6 |
| **Use Case Name:** | Detect Currency Notes |
| **Actors:** | System |
| **Description:** | When user provides input image or open mobile camera, the system will process the input by the user by applying suitable deep learning model trained on a dataset for Pakistani currency notes of PKR 10, 50, 75, 100, 500, 1000 and 5000. |
| **Trigger:** | User provides input image or open mobile camera. |
| **Preconditions:** | User opens mobile camera or provide input image, then the system will start applying model on a self-trained dataset for Pakistani currency notes. |
| **Postconditions:** | After applying trained model, the system will generate a spoken output using Text-To-Speech feature. |
| **Normal Flow:** | 1 User choose the module.  2 User opens mobile camera or provide input image.   1. System applies model trained to detect amount of currency   notes.  4 System will provide spoken output to the user. |
| **Alternative Flow:** | 1. User chooses module. 2. User unable to provide input image from gallery because of no image file or unable to open mobile camera. 3. System unable to apply model trained on any dataset for currency detection due to no input image provided or due to any other error. 4. User have to again try opening the mobile camera or try providing an input image again. |
| **Exceptions:** | 1 If user unable to provide input image or open mobile camera,  the system will display a message showing “No input image  provided” or “Unable to open mobile camera”.   1. System unable to detect correct amount of currency notes from input image due to any error. |
| **Assumptions:** | 1 User providing inputs correctly or mobile camera working  properly.  2 System applying model correctly and predicting the expected  output. |

**Table 3.1.7: Detect Color**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.7 |
| **Use Case Name:** | Detect Color |
| **Actors:** | System |
| **Description:** | When user provides input image or open mobile camera, the system will process the input by the user by applying model trained on a dataset by mapping color names with their RGB values. |
| **Trigger:** | User provides input image or open mobile camera. |
| **Preconditions:** | User opens mobile camera or provide input image, then the system will start applying suitable model trained on a dataset for color detection. |
| **Postconditions:** | After applying trained model for color detection, the system will generate a spoken output using Text-To-Speech feature. |
| **Normal Flow:** | 1 User choose the module.  2 User opens mobile camera or provide input image.  3 System applies model trained for color detection.  4 System will predict the color name from any image.   1. System will generate spoken output using Text-To-Speech. |
| **Alternative Flow:** | 1. User choose module. 2. User unable to provide input image from gallery because of no image file or unable to open mobile camera. 3. System unable to apply model trained on any dataset for color detection due to no input image or due to any error. 4. User have to again try opening the mobile camera or try providing an input image again. |
| **Exceptions:** | 1 If user unable to provide input image or open mobile camera,  the system will display a message showing “No input image  provided” or “Unable to open mobile camera”.  2 System unable to correctly predict color from input image. |
| **Assumptions:** | 1 User providing inputs correctly or mobile camera working  properly.  2 System applying model correctly and predicting the expected  output. |

**Table 3.1.8: Generate Output**

|  |  |
| --- | --- |
| **Use Case ID:** | UC-1.8 |
| **Use Case Name:** | Generate Output |
| **Actors:** | System |
| **Description:** | When user provides input image or open mobile camera, the system will generate output according to the module selected by the user. |
| **Trigger:** | User provides input image or open mobile camera and system will process the suitable model as selected by the user. |
| **Preconditions:** | System applies suitable model according to the module selected by the user. |
| **Postconditions:** | System will predict the output correctly and generate a spoken output using Text-To-Speech feature. |
| **Normal Flow:** | 1 User chooses the module.   1. User opens mobile camera or provide input image. 2. System applies model trained according to module selected by the user. 3. System will generate spoken output using Text-To-Speech. |
| **Alternative Flow:** | 1. User chooses module. 2. User unable to provide input image from gallery because of no image file or unable to open mobile camera. 3. System unable to apply model trained on any dataset due to no input image. 4. User have to again try opening the mobile camera or try providing an input image again. |