Visual Assistance Camera (VAC)

By

NC Adeel Ahmed

NC Izaz Khan

NC Saad Gillani

NC Zain ul Abedeen

Supervisor: Dr. Naima Altaf

Submitted to Faculty of Department of Computer Software Engineering NationalUniversity of Sciences and Technology, Islamabad in partial fulfilment for the requirements of a B.E Degree in Computer Software Engineering, June 2019

In the name of ALLAH,

CERTIFICATE FOR CORRECTNESS AND APPROVAL

It is certified that work contained in the thesis – Visual Assistance Camera (VAC) carried out by Adeel Ahmed, Izaz Khan, Saad Gillani and Zain-ul-Abedeen under supervision of Dr. Naima Altaf for the fulfilment of Bachelors of Software Engineering Degree is verified and approved.

Approved By

Dr. Naima Altaf

Dept. of CSE, MCS

Dated: ________________ ________________

ABSTRACT

Visual Assistance Camera (VAC)

The advancement made in machine learning has allowed us to use neural networks in practical scenarios. We use convolutional neural networks and long short term memory units to make out the features in the image and then generate a caption of thatimage. The product is a project that uses deep neural networks to firstly give dense image descriptions and then give its output to the visually impaired user in audio format. This product will facilitate the visually impaired person so that he can have better sense of what is in front of him. Moreover, all the functionalities of the model will be incorporated in a web based mobile application that user will be able to use readily and easily. We use image data along with their annotations that is used to train the model on the dataset, the accuracy for the model depends on the range of images we have as well as the number of them and a combination of other factors.

.

DECLARATION

No portion of the work presented in this dissertation has been submitted in support of another award or qualification either at this institution or elsewhere.

DEDICATION

To our parents, without whose support and cooperation, a work of this magnitude would not have been possible. To our supervisor, Lec. Naima Altaf,To the great Scientists and Researchers of the history whose accomplishments have greatly contributed to the progress of this modern world and who are of great inspiration to us to excel in the fields of Science and Technology.

Chapter. 1 Introduction

1.1. Overview

Visual Assistance Camera (VAC) will facilitate the visually impaired person so that he can have better sense of what is in front of him. Moreover, all the functionalities of the model will be incorporated in a web based mobile application that user will be able to use readily and easily. System will provide a model that can capture an image of a certain quality and then give the description of the image in one-line statement in audio format. This application will be targeted specifically for the visually impaired i.e, those who have their vision partially damaged.

1.2. Problem Statement

The advancement of Deep Learning and Neural Networks has allowed for a lot of applications in different fields. People who are Visually Impaired have vision that is ‘impaired’, for people such as, a type of system that can help them give an idea of what is in front of them. A system that has standard accessibility features that allows the user with the click of a button to know what is in front of them.

1.3. Approach

It’s a purely Software based system. Model is trained usingkeras, numpy in python. The app is in full stack node, backend is completely in express and front end is in React-Native. An android application is implemented for the use of the visually impaired, subject will have to wear a headset which will enhance accessibility and will interact with the application through the buttons on the headset for takingphotos and receiving the inference.

1.4. Scope

This project is intended to provide a deep neural network model that strives to take a step in describing the content of images and reason of how to represent them in natural language. A complete software mobile application will then be built by using this model that takes an image and the gives it description through this model, convert it into audio format and give it to user. The targeted audience for this application are people that are visually impaired and who cannot see properly/completely.

1.5. Objectives

The main objective of this Visual Assistance Camera (VAC) is to provide a system that isable to take a picture and provide a caption that gives a context of the image. Following are the objectives that are kept in mind: -

- To develop ansystem that has a camera and a headset and can give one-line context of image back in audio through the earpiece.
- To develop an mobile app for android that implements this system.
- To employ machine learning techniques to generate captions of the images taken through the camera.

1.6. Deliverables

Sr .	Tasks	Deliverables
1	Literature Review	Literature Survey
2	Requirements Specification	Software Requirements Specification document (SRS)
3	Detailed Design	Software Design Specification document (SDS)

4	Implementation	Project demonstration
5	Testing	Evaluation plan and test document
6	Training	Deployment plan
7	Deployment	Complete application with necessary documentation

Table 1-1 Deliverables

1.7. Overview of the Document

This document shows the complete working process of our application Visual Assistance Camera. It starts off with the literature review which shows past work done in similar field, requirement analysis of the system, system architecture which highlights the modules of the software and represents the system in the form of component diagram, Use Case Diagram, Sequence Diagram and general design of the system. Then it will move on to discuss the detailed Description of all the components involved. Further the dependencies of the system and its relationship with other products and the capacity of it to be reused will be discussed. At the end test cases and any future work proposal has been presented.

1.8. Document Conventions

Heading are prioritized in a numbered fashion, the highest priority heading having a single digit and subsequent headings having more numbers, according to their level. Font used is Times New Roman. All the main headings are of size 18 and bold. All the second level sub-headings are of size 16 and bold. All the further sub-headings are of size 14 and bold. All references in this document are provided where necessary, however where not present, the meaning is self-explanatory. All ambiguous terms have been clarified in the glossary at the end of this document.

1.9. Intended Audience and Reading suggestions

This design document contains design specific information about Visual Assistance Camera, use cases, functions, features and sequence diagrams etc. It describes in detail how Visual Assistance Camera should function and what it should do for a visually impaired person.System features with use cases and constraints are discussed in detail. System interfaces are also discussed in detail.

For better understanding, the document is divided into chapters

- In chapter 1 & 2, an overall description of Application is provided. First product perspective is presented with product features and main functions. A complete research in project's domain is documented as a part of chapter 2. Then follow user classes and characteristics, operating environments that Application supports as well as design and implementation constraints. After all that, user documentation is presented and will provide you with more details about each feature's technology.
- In chapter 3 & 4 most important features are presented with detailed description, use cases and requirements.
- In chapter 5 testing and evaluation of the parking system is documented with detailed test cases for unit and integration testing.

This document is intended for:

1. **Developers:** (Project Group)

In order to be sure that they are developing the right project that fulfills the requirements provided in this document.

2. **Testers:** (Project Group, Supervisor)

   In order to have an exact list of the features and functions that must respond according to requirements.

3. **Users:**

   In order to get familiar with the idea of the project and how to use/respond in failure situations and suggest other features that would make it even more functional.

4. **Documentation writers:** (Project Group)

To know what features and in what way they have to explain. What technologies are required, how the system will respond in each user's action, what possible system failures may happen and what are the solutions to all those failures etc.

5. **Project Supervisor:** (Dr.Naima Altaf)

   This document will be used by the project supervisor to check whether all the requirements have been understood and, in the end, whether the requirements have been implemented properly and completely.

6. **Project Evaluators:** (CSE Dept. MCS)

   In order to know the scope of the project and evaluate the project throughout the development for grading.

Chapter 2.Literature Review

Brief description of projects previously carried out in this context will be discussed in this section.

Smart Navigation Cane for Visually Impaired, which is a guidance cane that detects an object in the way and guides the user by buzzing itself.

Dip Based Indoor Guidance System for Blind, which has images stored in a database and the user captures images in real time to see where they are, images are compared by DIP techniques and the system tells if it recognizes the place.

Image Description Generator by Prof. Abdul Waqeel, which generates image captions and then puts those image captions in pre-defined grammatical structures of sentence for eg. “This is a _________” , Caption=”Cat”, Sentence = ”This is a Cat”

Profound Visual-Semantic Alignments for Generating Image Descriptions by Andrej Karpathy, it was an examination model that created characteristic language depictions of pictures and their locales. It was a blend of Convolutional Neural Networks over picture areas, bidirectional Recurrent Neural Networks over sentences.

Show and Tell: A Neural Image Caption Generator by Oriol Vinyals, Alexander Toshev, Dumitru Erhan and SamyBengio was an exploration paper, depicting a model that is prepared to augment the probability of the objective portrayal sentence given the preparation picture. Model, is put together start to finish with respect to a neural system comprising of a dream CNN pursued by a language creating RNN. It creates total sentences in characteristic language from an information picture.

Warren McCulloch and Walter Pitts (1943) made a computational model for neural systems dependent on arithmetic and calculations called edge rationale. This model made ready for neural system research to part into two methodologies. One methodology concentrated on natural procedures in the mind while the other concentrated on the utilization of neural systems to man-made reasoning.

Yann LeCun (1989) utilized back-proliferation to gain proficiency with the convolution portion coefficients straightforwardly from pictures of written by hand numbers. Learning was in this manner completely programmed, performed superior to anything manual coefficient plan, and was fit to a more extensive scope of picture acknowledgment issues and picture types. This methodology turned into an establishment of current PC vision.

LSTM was proposed in 1997 by Sepp Hochreiter and Jürgen Schmidhuber. LSTM manages the detonating and evaporating inclination issues. The underlying form of LSTM square included cells, information and yield doors.

'Profound Learning with Python' is a book by François Chollet which is a designer level book for AI composed by the writer of Keras library. It begins by utilizing straightforward profound neural systems to portray make expectations on MNIST dataset to separate numbers from 1-9 in penmanship. House Pricing expectation utilizing relapse. The book additionally examines utilizing CNNs for picture acknowledgment and Recurrent Neural Networks and LSTMs to create sentences. Various techniques to expand exactness are talked about for e.g dropout.

Visual weakness, otherwise called vision hindrance or vision misfortune, is a diminished capacity to see to a degree that causes issues not fixable by normal methods, Visual disability is regularly characterized as a best revised visual keenness of more regrettable than either 20/40 or 20/60. Visual disability may cause individuals challenges with ordinary day by day exercises, for example, driving, perusing, mingling, and strolling.

To help the outwardly hindered with giving them a superior thought of what is before them by the snap off a catch, utilizing a headset catch by utilizing availability controls on an android telephone.

Chapter 3. Software Req. Specification (SRS)

3.1. Introduction

The introduction of the Software Requirements Specification (SRS) provides an overview of the entire SRS with purpose, scope, definitions, acronyms, abbreviations and references. The aim of this document is to present detailed description of the Visual Assistance Camera (VAC) by defining the problem statement in detail. The detailed requirements of the Visual Assistance Camera (VAC) are provided in this document.

3.1.1. Purpose

System will provide a model that can capture an image of a certain quality and then give the description of the image in one-line statement in audio format. This application will be targeted specifically for the visually impaired i.e, those who have their vision partially damaged.This document specifies the detailed requirements of a mobile application that is being developed to provide a way of describing images to the visually impaired.

3.1.2. Document Conventions

- When writing this document, it was inherited that all requirements have the same priority.
- This document mainly addresses the requirements of application tasked by our Project Instructor. This document also fulfils the requirements for FYP, CSE Dept. MCS, NUST.
- First there is presented an overall view about Visual Assistance Camera and then all features and functions are analysed in detail.
- When writing this document, it was inherited that no System/Subsystem Specification documents (SSS) or any other contract document exists.

3.1.3. Intended Audience and Reading Suggestions

This requirements document contains general information about Visual Assistance Camera (VAC), use cases, functions, features and special technologies. It describes in detail all that a Visual Assistance Camera (VAC) needs to do for its audience. Functional and non-functional requirements are addressed separately. System features with use cases and constraints are discussed in detail. System interfaces are also discussed in detail.

For better understanding, the document is divided into sections:

- In section 2 an overall description of Application is provided. First product perspective is presented with product features and main functions. Then follow user classes and characteristics, operating environments that Application supports as well as design and implementation constraints. After all that, user documentation is presented and will provide you with more details about each feature's technology.
- In section 3 most important features are presented with detailed description, use cases and requirements.
- In section 4 user, hardware, software and communication interfaces are described.
- In section 5 requirements about, safety and performance are presented along with the software quality attributes of the Application.

This document is intended for:

- **Developers:** (Project Group)
  In order to be sure that they are developing the right project that fulfills the requirements provided in this document.
- **Testers:** (Project Group, Supervisor)
  In order to have an exact list of the features and functions that must respond according to requirements.
- **Users:** (Visually Impaired User)
  In order to get familiar with the idea of the project and how to use/respond in failure situations and suggest other features that would make it even more functional.
- **Documentation writers:** (Project Group)
  To know what features and in what way they have to explain. What technologies are required, how the system will respond in each user's action, what possible system failures may happened and what are the solutions to all those failures etc.
- **Project Supervisor:** (Dr. Naima Altaf)
  This document will be used by the project supervisor to check and guide the group about the understanding and implementation of the requirements properly and completely during the development lifecycle.
- **Project Evaluators:** (CSE Dept. MCS)
  In order to know the scope of the project and evaluate the project throughout the development for grading.

3.1.4. Project Scope

This project is intended to provide a deep neural network model that strives to take a step in describing the content of images and reason of how to represent them in natural language. A complete software mobile application will then be built by using this model that takes an image and the gives it description through this model, convert it into audio format and give it to user. The targeted audience for this application are people that are visually impaired and who cannot see properly/completely.

3.2. Overall Description

3.2.1. Product Perspective

The product is a project that will use deep neural network model to firstly give dense image descriptions and then give its output to the visually impaired user in audio format. This product will facilitate the visually impaired person so that he can have better sense of what is in front of him. Moreover, all the functionalities of the model will be incorporated in a web based mobile application that user will be able to use readily and easily.

3.2.2. Product Function:

Main features of the product are given below:

- Generating images captions
- Constructing sentences out of those captions
- Give description of an image captured through a camera
- Give description in audio format
- Easy to use interface

3.2.3. User Classes and Characteristics

3.2.3.1. Tester (Group members, Supervisor, occasional user)

Tester will use this project to check for bug finding. They will also use the project to check if it’s in accordance to the Software Requirements Specification document.

3.2.3.2.Project Supervisor (occasional user)

Project supervisor will also use the product to evaluate. They will use this product to find the accuracy and error in the output.

3.2.3.3. Visually Impaired Person (Regular user)

Visually impaired person will use this device to get better understanding of what is in front of them.

3.2.4. Operating environment

3.2.4.1. Hardware Requirements

Computer/Mobile device: Camera, Android Smartphone, NVidia Graphic Card(1080TI GPU), Earpiece.

3.2.4.2. Software Requirements

- Python, TensorFlow, React JS, NodeJS,Keras.

3.2.5. Design and Implementation Constraints

Constraints of the product are given below:

- VAC will only process image if camera is upright.
- Image should be of certain quality to give reasonably accurate.
- Android version should be at least 5.0 Lollipop or higher.
- Ram should be at least 2GB or higher.

3.2.6. User Documentation

A user manual will be provided to the users in which separate instructions will be given according to the particular user i.e. Visually impaired user and the, developers and testers. It will include the details of the system's working. Help documents will also be a part of the system.

The project report will also be available for the users which will highlight the system features, working and procedures.

3.2.7.Assumptions and Dependencies

- Overall performance of the product will depend on the hardware infrastructure.
- User must know the language and User Interface for the better performance of the product.
- Limitations of the product must be kept in mind by the user.

3.3. System Features

System features are organized by use cases and functional hierarchy so that the main functions of the system will be understandable. In the description of system features there are several references in various system interfaces. These interfaces are better explained in section 4.1 of this document.

3.3.1.Image Acquisition Module

This feature allows VAC to acquire an image.

3.3.1.1Description

This module takes an image and passes it to the Image describing model.

3.3.1.2Stimulus/Response Sequences

Data flow:

3.3.1.2.1Basic Data Flow

1. User opens application will switch on.
2. When the button is pressed camera takes an image.
3. Then this module sends the image to the Image Description Module.

3.3.1.2.2Alternative Data Flows

3.3.1.2.2.1Alternative Data Flow 1

1. System is unable to acquire image through the camera.
2. Error message is conveyed to the user in audio format.

3.3.1.3Functional Requirements

1. System shall be able to acquire image in real time.
2. Application is allowed to communicate by the operating system.

3.3.2. Image Description Module

This module shall be able to effectively give a reasonable description of the image given to it.

3.3.2.1Description

The Image Description Module will have a model that takes the image generates captions and then runs it through another model that generates the image description using a neural network.

3.3.2.2Stimulus/Response Sequences

Data flow:

3.3.2.2.1Basic Data Flow

1. Module will generate image captions.
2. Module will then generate image description from those captions.
3. Module will pass over the text description to the Text to Speech Converter module.

3.3.2.2.2Alternative Data Flows

3.3.2.2.2.1Alternative Data Flow 1

1. The Image acquired is not of the required quality.
2. Error message is conveyed to the user.

3.3.2.3Functional Requirements

1. Image description model shall be able to describe the image to reasonable accuracy.

3.3.3. Text to Speech Converter

This module will provide audio description to the user through the earpiece.

3.3.3.1Description

The module will receive the description in textual format and then convert it into audio format and then output it through the earpiece.

3.3.3.2Stimulus/Response Sequences

Data flow:

3.3.3.2.1Basic Data Flow

1. The module will receive textual description.
2. The module will convert the textual description into audio format.
3. The module will output the audio to the earpiece.

3.3.3.2.2Alternative Data Flows

3.3.3.2.2.1Alternative Data Flow 1

1. Module crashes unexpectedly.
2. Error message is conveyed to the user.

3.3.3.3Functional Requirements

1. The module shall be able to convert text into speech.

The module shall be able to output the audio to the ear piece.

3.4. External Interface Requirements

3.4.1.User Interfaces

Responsive graphical user interfaces must be provided to user to work with the extension.

Figure 3.4.1 – User Interface

3.4.2.Hardware Interfaces

- System should meet the minimum hardware requirements.
- Android application will open by using android interface on android smartphone.
- External button will be provided to capture a image.

3.4.3.Software Interfaces

- Application will run on android 5.0 or higher.
- Model will run using tensor flow, Keras.

3.4.4.Communications Interfaces

Our inference engine will take the image captured through a camera, the engine will return the description of the image and then the text to speech generation module will return the description in audio format to earpiece.

3.5. Other Non-functional Requirements

3.5.1.Performance Requirements

Application shall run on a minimal amount of memory and take up a small amount of disk space after installation. Depending on the performance of the user's mobile, the processing might slow down the Application.

3.5.2.Safety Requirements

This application is a fast and responsive program. However as mentioned before working with large data may lead Application to become unresponsive or even crash. A crash will also waste a lot of time and user may lose information about message transfer.

3.5.3.Security Requirements

Application should not need any additional information other than the collected data from the user already present data. There are no connection to third party devices or servers so no data will be send or received.

3.5.4.Software Quality Attributes

3.5.4.1.Reliability

Application should provide reliability to the user. The product will run stably with all the features mentioned above available and executing perfectly. It should be tested and debugged completely. All exceptions should be well handled.

3.5.4.2.User Friendliness/Simplicity

Application should have a graphical user interface with user friendly menu and options.

3.5.4.3.Availability

Application will be provided to user through a suggested and well known platform.

3.5.4.4.Ease Of Installation

The user shall be able to install the app easily through one click\tap.

3.6. Sequence Diagram(Abstract)

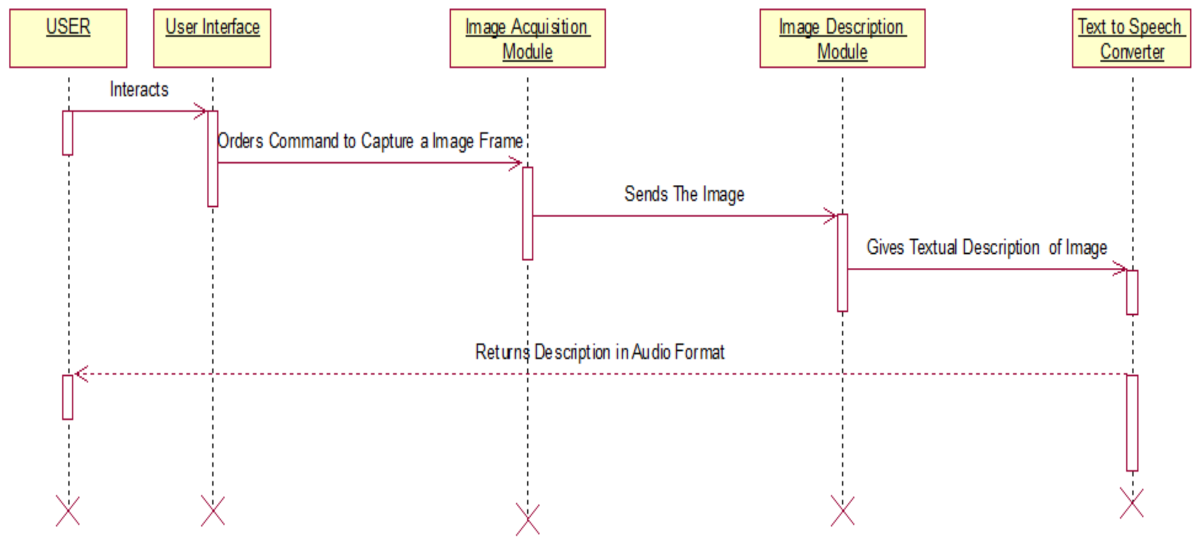


Figure 3.6.2 – Sequence Diagram(Abstract)

Chapter 4. Software Design & Implementation

4.1. Introduction

4.1.1. Purpose

This document includes software design for **VAC (Visual Assistance Camera)**, release number 1.0. System will provide a model that can capture an image of a certain quality and then give the description of the image in one-line statement in audio format. This application will be targeted specifically for the visually impaired i.e, those who have their vision partially damaged.

This document specifies the detailed design of a mobile application that is being developed to provide a way of describing images to the visually impaired. Document also focuses all the stakeholders involved in this project.

4.1.2. Project Scope

This project is intended to provide a deep neural network model that strives to take a step in describing the content of images and reason of how to represent them in natural language. A complete software mobile application will then be built by using this model that takes an image and the gives it description through this model, convert it into audio format and give it to user. The targeted audience for this application are people that are visually impaired and who cannot see properly/completely.

4.1.3. Definitions

SRS: Software Requirement Specification

SDS: Software Design Specification

CNNs: Convolutional Neural Networks

LSTM: Long Short Term Memory Units

4.1.4. Overview of Document

- When writing this document, it was inherited that all requirements have the same priority.
- This document mainly addresses the design of application tasked by our Project Instructor. This document also fulfils the design for FYP, CSE Dept. MCS, NUST.

- First there is presented an overall view about Visual Assistance Camera in design terms and then all features and functions are analysed in detail through diagrams, tables etc.
- When writing this document, it was inherited that no System Requirement Specification document (SRS) already exists.

4.2. Work Breakdown structure

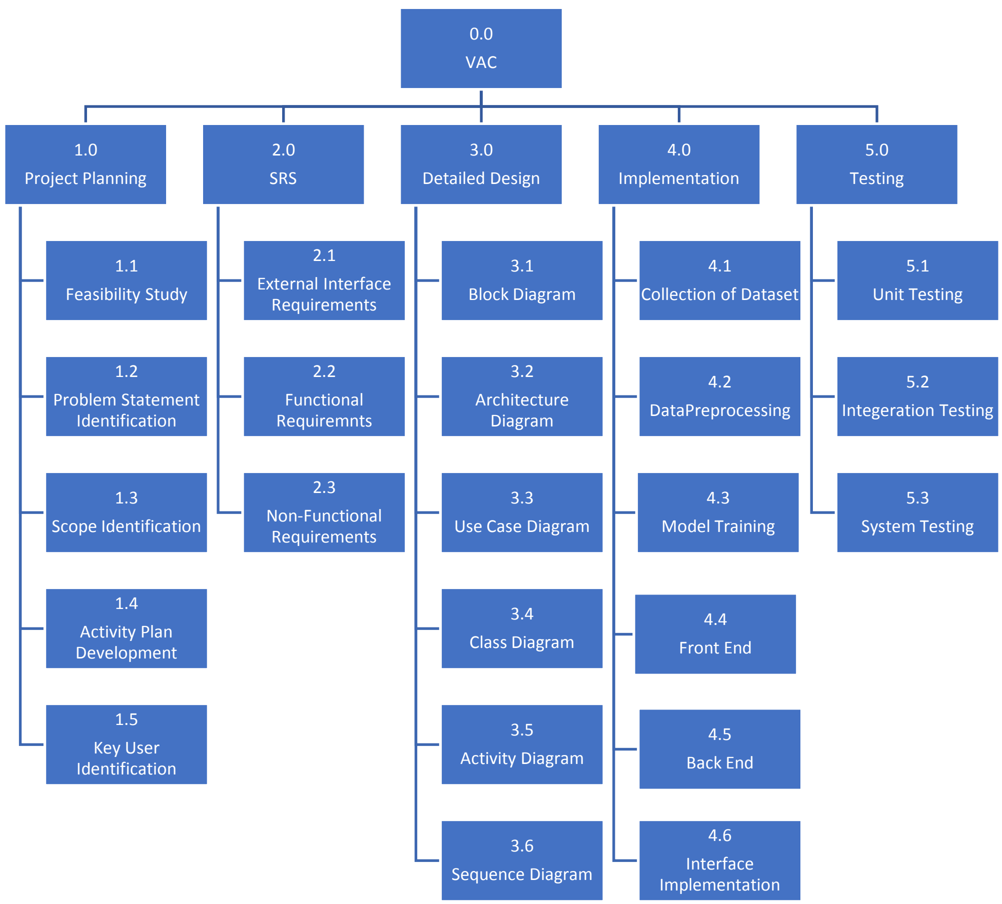


Figure 4.2.1. – Work Breakdown Structure

4.3. System Architecture Description

This section provides detailed system architecture of Hate Speech Detection. Overview of system modules, their structure and relationships are described in this section. User interfaces and related issues are also discussed.

4.3.1. Overview of Modules

- **4.3.1.1 User Interface**

  This is the module where the user interacts with the system, here the user gives command to take a picture.

- **4.3.1.2. Image Acquisition Module**

  This feature allows VAC to acquire an image. This module takes an image and passes it to the Image describing model. It uses the camera to take that image.

- **4.3.1.3. Image Description Module**

  This module shall be able to effectively give a reasonable one-line context of the image given to it. The Image Description Module will have a model that takes the image and uses conv-net to generate its features and then uses LSTMs to generate the image context.

- **4.3.1.4. Text to Speech Converter**

  This module will provide audio description to the user through the earpiece. The module will receive the description in textual format and then convert it into audio format and then output it through the earpiece.

4.3.2. Structure and Relationships

This section covers the overall technical description of VAC. It shows the working of application in perspective of different point-of-views and also shows relationships between different components.

4.3.2.1. System Block Diagram

This diagram shows the high-level description of the application. It shows all the modules of the system and their associations and flow of data between modules.

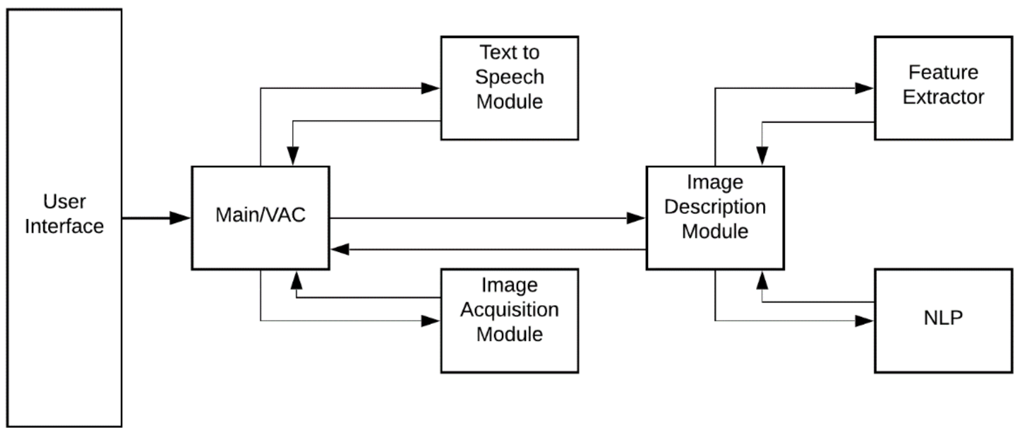


Figure 4.3.1. - System Block Diagram

User interacts with the system through the user interface, user gives a command to take image, Image Acquisition Module takes an image through the camera when button is pressed, then this module sends the image to the Image Description Module. Module will then generate one-line context of the image, Module will pass over the text to the Text to Speech Converter module. The module will convert the textual description into audio format return to the earpiece.

4.3.2.2. User View (Use case diagram)

Following diagram shows course of events that take place when an actor (user and other allowed interactions) interacts with system.

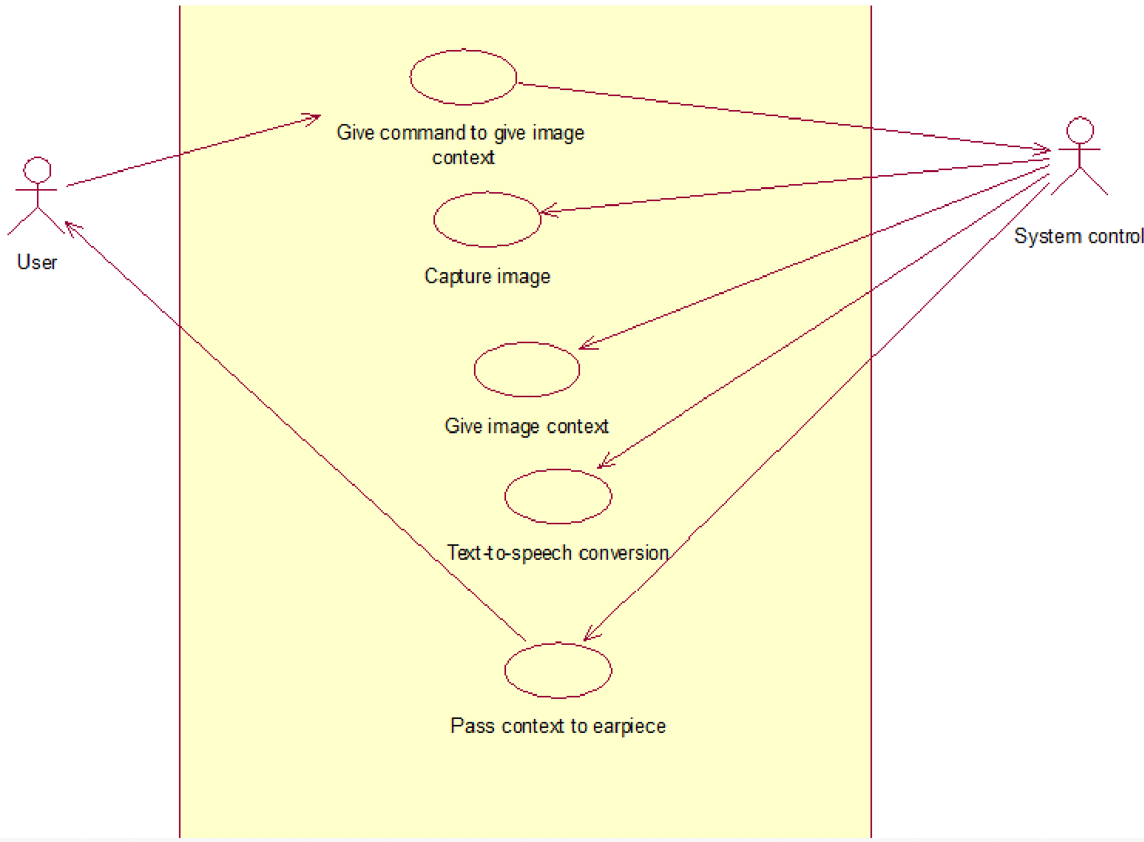
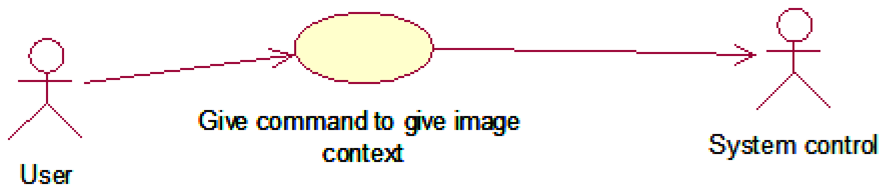


Figure 4.3.2. - Use case diagram

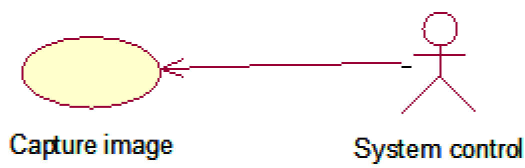
Use case 1



Use case name	Give command to give image context
Primary actor	User
Secondary actor	System Control
Normal course	- Send command - Initiate process
Alternate course	- Send command - Initiate process - Command failed - No process initiated
Pre-condition	System is running currently
Post-condition	Image Acquisition module receive command to take image
Extend	N/A
Include	N/A
Assumptions	App is running

Table 4-3-1 (Use Case 1)

Use case 2

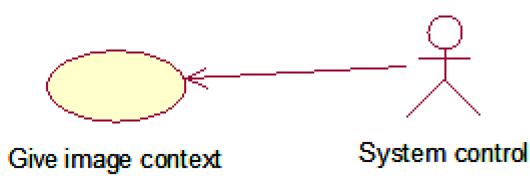


Use case name	Capture image
Primary actor	System Control
Secondary actor	N/A
Normal course	- Image Acquisition module receives command to take image - Uses the camera to take image - Passes the image to Image Description Module
Alternate course	- Image Acquisition module receives command to take image - Error occurs - Fails to acquire image - Send error message
Pre-condition	System has received command to take image.

Post-condition	Passes the image to Image Description Module
Extend	N/A
Include	N/A
Assumptions	App is running

Table 4-3-2 (Use Case 2)

Use case 3

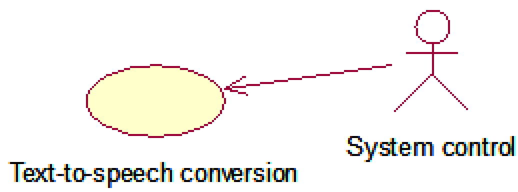


Use case name	Give image context
Primary actor	System Control
Secondary actor	N/A

Normal course	- Takes image - Passes it through the model - Model makes inference and gives context
Alternate course	- Takes image - Passes it through the model - Error occurs - Send error message
Pre-condition	Image Description Module has received command to make a inference.
Post-condition	Passes the textual description to Text-to-Speech module.
Extend	N/A
Include	N/A
Assumptions	App is running

Table 4-3-3 (Use Case 3)

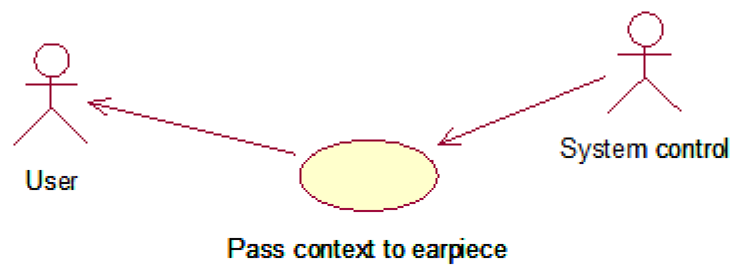
Use case 4



Use case name	Text-to-speech conversion
Primary actor	System Control
Secondary actor	N/A
Normal course	- Take texts - Converts to Audio from text
Alternate course	- Take texts - Error occurs - Send error message
Pre-condition	Text-to-speech module has received command to make a conversion.
Post-condition	Converts to Audio from text.
Extend	N/A
Include	N/A
Assumptions	App is running

Table 4-3-4 (Use Case 4)

Use case 5



Use case name	Pass context to earpiece
Primary actor	System Control
Secondary actor	User
Normal course	- Takes Audio - Outputs to user by ear piece
Alternate course	- Takes Audio - Error occurs - Send error message
Pre-condition	Audio has been generated and now needs to be outputted.
Post-condition	User hears the context through ear piece.
Extend	N/A
Include	N/A

Assumptions	App is running

Table 4-3-5 (Use Case 5)

4.3.2.3. Sequence Diagram

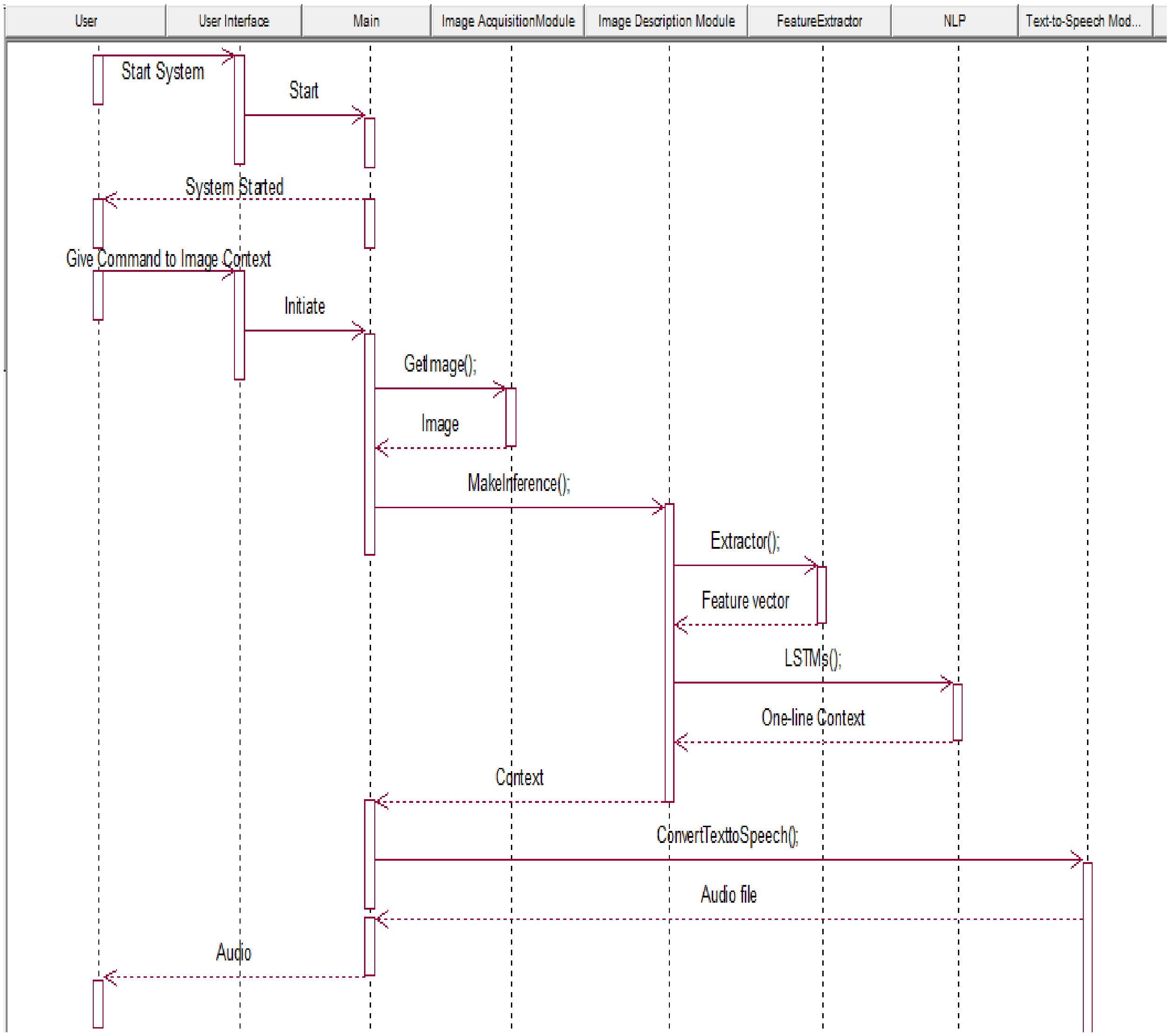


Figure 4.3.3. – Sequence Diagram

4.3.2.4. Implementation View (Class Diagram)

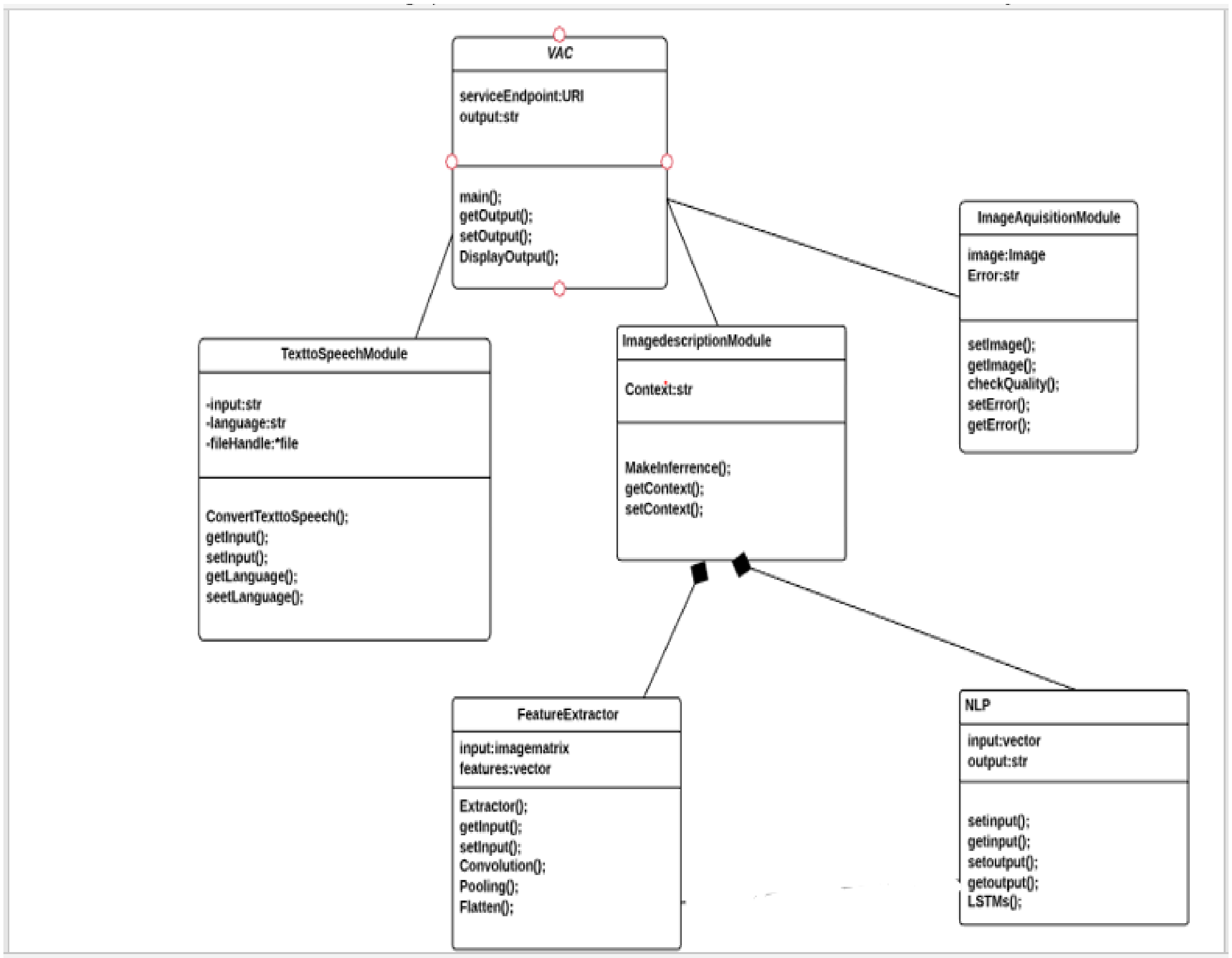


Figure 4.3.4. – Class Diagram

Class	Description
VAC	This class has all the methods and properties required to interact with all the main classes, control between different modules is maintained from here.
TexttoSpeech	This module is responsible to convert text to audio format. TexttoAudio() does this functionality.
ImageAcquisition	This class is responsible for taking the image from user, we get a image and check if it is of the right quality and we have a error message is displayed if the image is not of the right quality.
Inference	Inference is class where actual model. It takes objects from both feature extractor and NLP. It takes the features and generates one-line context.

Feature Extractor	Extracts features from image using CNN's by first applying convolutions, then applying pooling layer and at the end we flatten the result to gain a vector.
NLP	Generates a one-line context of the image by using the vector gained by the feature extractor passing to the LSTMs to generate the sentence.

4.3.2.5. Dynamic View (Activity Diagram)

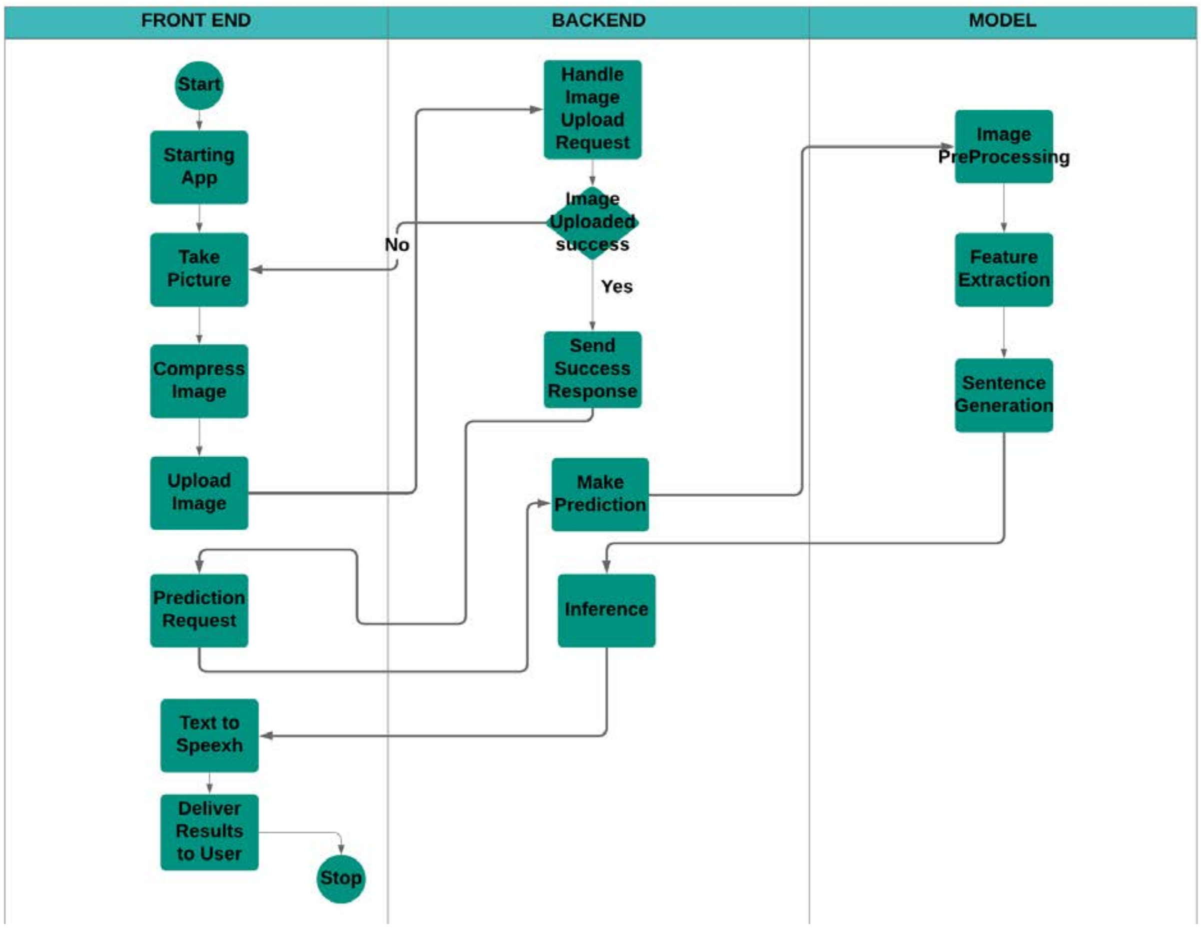


Figure 4.3.5. – Activity Diagram

4.3.3. User Interface

Figure 4.3.6. – User Interface

4.4. Detailed Description of Components

This section describes in detail all the modules of Visual assistance Camera (VAC).

4.4.1. User Interface

Identification	Name: User Interface Location: Presentation Layer
Type	Interface
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **4.1 User Interface** This feature allows a user to interact with the system. **Description:**Responsive graphical user interfaces must be provided to user to work with the application.
Function	A interface where the user will have a camera like interface where user will take a picture through a button and the one line context will be conveyed in audio format to the user through an ear piece
Subordinates	It has two subordinates: 1. User Interface: Requirement 4.1 in SRS
Dependencies	This component is the place user interacts with the system, so system must be running.
Interfaces	This component has following interfaces: • User

	• Control/Main
Resources	**Hardware:** RAM, Processor **Software:** JS, JS libraries, CSS, Html
Processing	Module realizes if command has been initiated by the user.
Data	This component uses following information of the application: 1. Data pertinent to the command made for image context.

Table 4-4-1 (User Interface)

4.4.2. Image Acquisition Module

Identification	Name: Image Acquisition Module Location: Presentation Layer
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.1 Image Acquisition Module** This feature will allow the system to acquire and a image through the camera. **Description:**This module takes an image through the camera and passes it to the Image describing model.

Function	When the button is pressed camera takes an image. Then this module sends the image to the Image Description Module.
Subordinates	This component has following subordinates:
	1. Image Acquisition Module: Requirement 3.1 in SRS
Dependencies	This component is dependent on the system to be running and a command issued by the user. Then this module sends the image to the Image Description Module
Interfaces	**N/A**
Resources	**Hardware:** RAM, Processor, Camera **Software:** JS, JS Node libraries
Processing	Command for the use of the camera to capture a image.
Data	This component uses following information of the application: 1.Image is the main data for this module.

Table 4-4-2 (Image Acquisition Module)

4.4.3. Image Description Module

Identification	Name: Image Description Module Location: Logic Layer

Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.2 Image Description Module** This module shall be able to effectively give a reasonable one line context of the image given to it. **Description:**The Image Description Module will have a model that takes the image generates image features and then runs it through LSTMs that generates the image context using a neural network.
Function	This module shall be able to effectively give a reasonable one line context of the image given to it.
Subordinates	This component has following subordinates:
	1. Image Description Module: Requirement 3.2 in SRS
Dependencies	This component is dependent on the system to be running and Image Acquisition Module needs to send the image to the Image Description Module
Interfaces	Communication with Image Acquisition Module and Text-to-speech module.
Resources	**Hardware:** RAM, Processor, GPU **Software:** Python, Tensorflow, Keras

Processing	Module will generate image features using ConvNets. Module will then generate image context from those features using LSTMs to generate a sentence.
Data	This component uses following information of the application: 1. Features, Image, the inference/one-line context.

Table 4-4-3 (Image Description Module)

4.4.4 Text to Speech Module

Identification	Name: Text to Speech Module Location: Logic Layer
Type	Component
Purpose	This component fulfils following requirement from Software Requirements Specification Document: **3.3 Text to SpeechModule** This module will provide audio description to the user through the earpiece. **Description:**The module will receive the description in textual format and then convert it into audio format and then output it through the earpiece. .
Function	The module will receive the description in textual format and then convert it into speech.
Subordinates	This component has following subordinates:

	1. Text to Speech Module: Requirement 3.3 in SRS
Dependencies	This component is dependent on the system to be running and Image Description Module needs to send the textual context of image to the Text-to-speech Module.
Interfaces	Communication with Image Description Module to send the textual context of image.
Resources	**Hardware:** RAM, Processor **Software:** JS, JS libraries, Flask
Processing	The module will receive textual description. The module will convert the textual description into audio format.
Data	This component uses following information of the application: 1. The inference/one-line context and its audio.

Table 4-4-4 (Text to Speech Module)

4.5. Reuse and Relationships to other Products

This application is meant to provide the visually impaired a better estimate of what is in-front of them by providing them the one-line context of the image and providing that context in speech form, so it is a highly targeted application for the visually impaired, the image context generated can be used in some other products that uses the context of the image.

4.6. Design Decisions and Trade-offs

This system mainly consists of three modules which are the basic functions of this system, the Image Acquisition module has the responsibility of getting the image in the required format. The context description is modularized into the Image Description module where the main CNNs and LSTMs are running to make the context. The conversion of the text to speech has been encapsulated in the Text to Speech module where text is to be converted into the speech and converted into audio format.

The application is a context-aware pervasive system. Interface of the system is distinct from the application logic. Layered architecture is used to isolate application logic from the user interface. It can be modelled using Multitier Layered Architecture consisting of three layers i.e.; Presentation Layer, Logic Layer, Control Layer. Presentation layer corresponds to elements of the user interface such as screens, textboxes, dialog boxes etc. Control Layer controls the communication of data between the Presentation Layer, Logic Layer and is the part where the control and communication between the module is controlled from. Logic Layer is the layer where the main functions of the system, the inference to make from the image and the converting to speech.

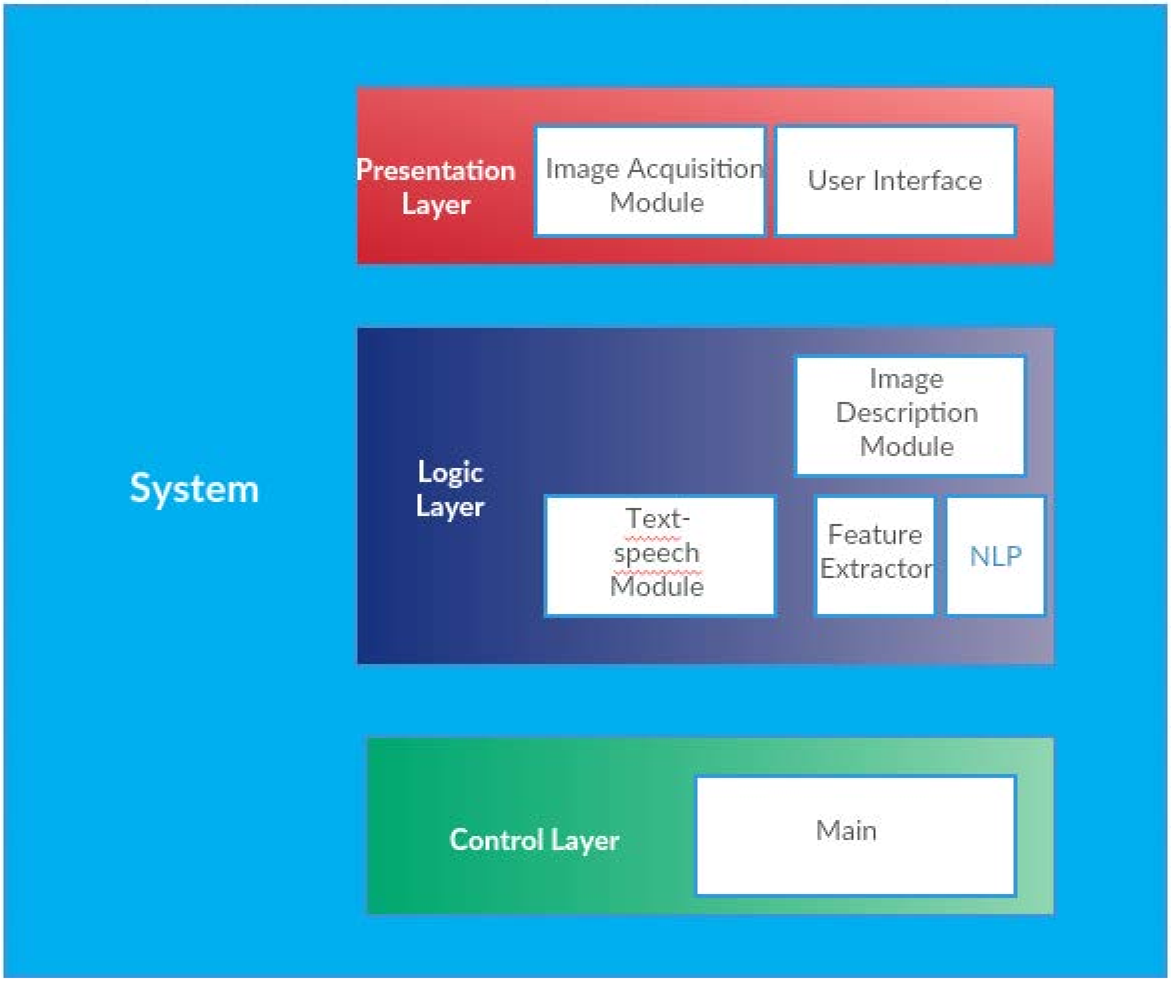


Figure 4.6.1. –System Architectural Diagram

Chapter 5. Project Test and Evaluation

5.1. Introduction

This test plan chapter describes the appropriate strategies, process and methodologies used to plan, execute and manage testing of the VAC (Visual Assistance Camera) Android application project. The test plan will ensure that the application meets the customer requirements at an accredited level.

Manual Testing will be followed which includes testing a software manually, i.e., without using any automated tool or any script. In this type, the tester takes over the role of an end-user and tests the software to identify any unexpected behaviour or bug. Each Unit will be tested separately and then will be integrated with other units, therefore Unit Testing and Integration testing will be followed. For each unit Black box Testing is done and for combined units Acceptance Testing is done.

The test scope includes the Testing of all functional, application performance and use cases requirements listed in the *requirement document*.

Software testing, depending on the testing method employed, can be implemented at any time in the development process. However, most of the test effort occurs after the requirements have been defined and the coding process has been completed.

This document includes the plan, scope, approach and procedure of VAC (Visual Assistance Camera) Application test. The pass/fail criteria of the test items are also defined. The VAC (Visual Assistance Camera) Software documents and tracks the necessary information required to effectively define the approach to be used in the testing of the product.

5.2. Test Items

Based on the VAC (Visual Assistance Camera) Application requirements and design description, application modules of mobile Android application and non-functional scenario will be tested. The Requirements Defined in Software Requirements Specification and the Design entities as explained in Software Design Document will be tested.

5.3. Features Tested

Following Features are Tested:

- Start App
- Start App (Failure)

- Accessibility
- Moving to Image capturing screen
- Image Capturing
- Image Capturing Cancelled
- Image Compression
- Back-end Testing Route Upload
- Back-end Testing Route Predict
- Text to Speech
- Image upload and inference
- App Integration

5.4. Approach

Acceptance test will be executed based on this acceptance test plan. And after all test cases are executed, a test report will be summarized to show the quality of VAC (Visual Assistance Camera) Application. Following test approaches will be used in test execution:

- **Unit test**. Developers are responsible for unit test as white-box testing. The implementation of each module and individual component will be verified separately.
- **Integration test**. After the unit test is passed above the defined quality threshold, testers will execute the integration test cases. After all the modules are integrated, it's crucial to test the product as a black-box. End-to-end scenarios will be tested to ensure the communication functionality.
- **Regression test**. After developers fix the bug in one feature, regression test will be executed by testers to ensure that the other functions are not affected.
- **Field test**. Firstly, untrained end users recreate one or more existing (but narrow) mass observation events in the VAC (Visual Assistance Camera) Android Application. A number of observers will be invited to help with evaluation. After that, post event questionnaires will be used to collect quantitative usage data as well as qualitative data and further improvement will be taken into consideration.

5.5. Item Pass/Fail Criteria

Details of the test cases are specified in section Test Deliverables. Following the principles outlined below, a test item would be judged as pass or fail.

- Preconditions are met
- Inputs are carried out as specified
- The result works as what specified in output => Pass
- The system doesn't work or not the same as output specification => Fail

5.6. Suspension Criteria and Resumption Requirements

Any bugs found can be fixed by developers quickly and no need to start the testing process from the beginning. However, when some major bugs occur, we will block some test cases as they are interdependent and the testing has to be paused. The test will restart from the very beginning until the major error is solved.

5.7. Test Deliverables

Following are the test cases:

Test Case Name	Start App
Test Case No	1
Description	Starting the app and to make sure that it is up and running
Preconditions	The user must have installed VAC (Visual Assistance Camera) Application in android operating system and android device must be connected with high speed internet.
Input Values	Selecting and clicking the app icon.

Valid Inputs	Press the icon one time.
Steps	First select the VAC (Visual Assistance Camera) android application installed in Android Operating System and click the app icon to launch the app in android device.
Expected Output	App is successfully launched and landed on home screen.
Actual Output	App is successfully launched and landed on home screen.

Table 5-7-1 (Test Case 1)

Test Case Name	Start App (Failure)
Test Case No	2
Description	Starting the app and to make sure that it is up and running
Preconditions	The user must have installed VAC (Visual Assistance Camera) Application in android operating system and android device must be connected with high speed internet.
Input Values	Selecting and clicking the app icon.
Valid Inputs	Press the icon one time.
Steps	First select the VAC (Visual Assistance Camera) android application installed in Android Operating System and click the app icon to launch the app in android device.
Expected Output	App not started
Actual Output	App not started

Table 5-7-2 (Test Case 2)

Test Case Name	Accessibility
Test Case No	3
Description	Accessibility options are open and working
Testing Technique Used	Unit Testing
Preconditions	The user must have installed and launched VAC (Visual Assistance Camera) Application in android operating system.
Input Values	Scroll over to the individual components through switch
Valid Inputs	Scroll over to the individual components through switch
Steps	From the home screen of the app press volume up switch of the audio headphone to select each component
Expected Output	A short accessibility description of each selected component is

	outputted through the speaker.
Actual Output	A short accessibility description of each selected component is outputted through the speaker.

Table 5-7-3 (Test Case 3)

Test Case Name	Moving to Image capturing screen
Test Case No	4
Description	Moving to the image capturing screen after pressing the button for image capturing
Testing Technique Used	Unit Testing
Preconditions	The user must have installed and launched VAC (Visual Assistance Camera) Application in android operating system and user must be on home screen.
Input Values	Press ‘Touch here’ button
Valid Inputs	Press ‘Touch here’ button
Steps	First select the VAC (Visual Assistance Camera) android application installed in Android Operating System then from the home screen press ‘Touch here’ button that in the background cause the handler for capturing the image
Expected Output	Move onto the image capturing screen
Actual Output	Move onto the image capturing screen, the actual image capturing module is launched

Table 5-7-4 (Test Case 4)

Test Case Name	Image capturing
Test Case No	5
Description	Testing image capturing unit
Testing Technique Used	Unit Testing
Preconditions	Image capturing screen is launched and user should be on image capturing screen
Input Values	Press the volume down button/the image capturing button, the image is captured and confirm screen is launched and ok button is pressed by first toggling to the ok button by clicking the volume up key and confirming by either clicking the ok button or using the headset hook to click the ok button
Valid Inputs	Press the volume down button/the image capturing button, the image is captured and confirm screen is launched and ok button is pressed by first toggling to the ok button by clicking the volume up key and confirming by either clicking the ok button or using

	the headset hook to click the ok button
Steps	Image capturing screen should be launched Press volume down/image capturing button Confirmation screen is launched Ok button is clicked either through the headset hook button or either clicking on the ok button on the screen
Expected Output	Image is captured and is path is saved.
Actual Output	Image is captured and is path is saved.

Table 5-7-5 (Test Case 5)

Test Case Name	Image capturing Cancelled
Test Case No	6
Description	Testing image capturing unit Cancelled feature
Testing Technique Used	Unit Testing
Preconditions	Image capturing screen is launched and user should be on image capturing screen
Input Values	Press the volume down button/the image capturing button, the image is captured and confirm screen is launched and ok button is pressed by first toggling to the cancel button by clicking the volume up key and confirming by either clicking the cancel button or using the headset hook to click the cancel button
Valid Inputs	Press the volume down button/the image capturing button, the image is captured and confirm screen is launched and ok button is pressed by first toggling to the cancel button by clicking the volume up key and confirming by either clicking the cancel button or using the headset hook to click the cancel button
Steps	Image capturing screen should be launched

	Press volume down/image capturing button Confirmation screen is launched Cancel button is clicked either through the headset hook button or either clicking on the Cancel button on the screen
Expected Output	Image is canceled and on the user interface the user returns to the image capturing screen.
Actual Output	Image is canceled and on the user interface the user returns to the image capturing screen.

Table 5-7-6 (Test Case 6)

Test Case Name	Image Compression
Test Case No	7
Description	Testing Image Compression feature
Testing Technique Used	Unit Testing
Preconditions	Application should be launched on the android device.
Input Values	The path to the captured image is inputted to image compression unit by entering the type of image to be returned(jpeg), and the compression ratio.
Valid Inputs	A valid path to the image residing currently in the context on the android image captured through.
Steps	Image path is captured through the previous image capturing screen. Image path in the form of json is passed to image compression module along with the ratio.
Expected Output	Compressed image path is returned from the module.
Actual Output	Compressed image path is returned from the module.

Table 5-7-7 (Test Case 7)

Test Case Name	Back-end Testing Route Upload
Test Case No	8
Description	Testing Image upload route.
Testing Technique Used	Unit Testing
Preconditions	Back-end server should be up and running.
Input Values	The upload route is to be hit with a post request to server with a form-data object containing path to the image.
Valid Inputs	Post request to server to upload image using form-data in the body of the post request.
Steps	Server is up and running. Post request to server through react fetch module.
Expected Output	Response is sent back to the client side with success message.
Actual Output	Success response is sent back to the client end.

Table 5-7-8 (Test Case 8)

Test Case Name	Back-end Testing Route Predict
Test Case No	9
Description	Testing Feature to get an inference from the model.
Testing Technique Used	Unit Testing
Preconditions	Back-end server should be up and running. Image should be uploaded to the server.
Input Values	The predict route is to be hit with a get request to the server.
Valid Inputs	The predict route is to be hit with a get request to the server if and only if the image is uploaded first to it.
Steps	Server is up and running. Get request to server through react fetch module.
Expected Output	Inference from the model is returned in response to the request by predict request.
Actual Output	Inference from the model is returned in response to the request by predict request.

Table 5-7-9 (Test Case 9)

Test Case Name	Text to Speech
Test Case No	10

Description	Testing text to speech model of application
Testing Technique Used	Unit Testing
Preconditions	Inference from the model needs to be already received from the model
Input Values	Text String to be spoken is entered as argument.
Valid Inputs	Text String to be spoken is entered as argument.
Steps	Inference is saved in a variable. Inference is passed to the speech model. Text is spoken through speaker/headset.
Expected Output	Inference is spoken.
Actual Output	Inference is spoken.

Table 5-7-10 (Test Case 10)

Test Case Name	Image upload and inference
Test Case No	11
Description	Testing image uploading and getting inference from the model
Testing Technique Used	Unit Testing
Preconditions	Back-end server should be up and running. Image should be captured, and path should be saved
Input Values	The path to the captured image is inputted to image compression unit by entering the type of image to be returned(jpeg), and the compression ratio. The upload route is to be hit with a post request to server with a form-data object containing path to the image. Post request to server to upload image using form-data in the body of the post request.
Valid Inputs	The path to the captured image is inputted to image compression unit by entering the type of image to be returned(jpeg), and the compression ratio. The upload route is to be hit with a post request to server with a form-data object containing path to the image. Post request to server to upload image using form-data in the body of the post request.
Steps	Image should be compressed with compression module, image

	should be uploaded to the server by sending post request of 'upload'. Prediction is received in response to the get request of predict back-end server
Expected Output	Inference is received from the model
Actual Output	Inference is received from the model

Table 5-7-11 (Test Case 11)

Test Case Name	App Integration
Test Case No	12
Description	Testing app as a whole
Testing Technique Used	Integration Testing
Preconditions	The user must have installed and launched VAC (Visual Assistance Camera) Application in android operating system.
Input Values	Scroll using key volume up button. Select the 'Touch Here', to speak the component description. Press the 'Touch Here' button.
Valid Inputs	Scroll using key volume up button. Select the 'Touch Here', to speak the component description. Press the 'Touch Here' button.
Steps	Scroll using key volume up button. Select the 'Touch Here', to speak the component description. Press the 'Touch Here' button. Image path is captured through the previous image capturing screen. Image path in the form of json is passed to image compression module along with the ratio. Image capturing screen should be launched Confirmation screen is launched cancel button is selected and clicked either through the headset hook button or either clicking on the ok button on the screen.

Expected Output	Inference is spoken.
Actual Output	Inference is spoken.

Table 5-7-12 (Test Case 12)

5.8. Environmental Needs

5.8.1. Hardware

- Mobile with Android platform.
- Computer (I7 6th Generation).
- 8GB RAM Minimum.

5.8.2. Software

- Python
- Node.js (along with dependencies.)
- TensorFlow
- Keras

5.9. Responsibilities, Staffing and Training Needs

5.9.1. Responsibilities

All developers of the project are responsible for the completion of all units testing and integration testing tasks.

5.9.2. Staffing and Training Needs

- Saad Gillani was responsible for Model, Zain ul Abedeen assisted him.
- Zain ul Abedeen was responsible for app development, Adeel Ahmed assisted with front end development, Izaz Khan assisted with back end development.
- Adeel Ahmed and Izaz Khan were responsible for documentation.

5.10. Risk and Contingencies

We have tried to test on various android platforms as much as possible, but it's impossible to test for all android platforms. What's more, mobile Android application is tested on Android devices and is tested on limited mobiles. platform etc.). Further investigation is required to verify and improve VAC (Visual Assistance Camera) Application.

Chapter 6. Future Work

The Project can be extended further by adding following modules:

1. Addition of Multiple Languages.
2. Addition of Voice Commands.
3. Making anWeb& IOS Applications of it.

Further improvements and accuracy of detecting model can be achieved by gathering and adding more and more diverse dataset in model with respect to time and retraining of model again and again with respect to time.

Chapter 7. References

1. Show and Tell: A Neural Image Caption Generator (2015) by Oriol Vinyals, Alexander Toshev, Dumitru Erhan and SamyBengio.
2. Deep Visual-Semantic Alignments for Generating Image Descriptions (2015) by Andrej Karpathy.
3. ‘Deep Learning with Python’ (2017) is a book by François Chollet.
4. Wikipedia (https://en.wikipedia.org/wiki/Artificial_neural_network)
5. Wikipedia (https://en.wikipedia.org/wiki/Convolutional_neural_network)
6. Wikipedia (https://en.wikipedia.org/wiki/Long_short-term_memory)