



Urdu Language Recognition System



Submitted By

Afshan Wain	18321519-001
Eza Tehreem Asjad	18321519-003
Azbha Aleem	18321519-042

Supervised By
Dr. Umar Shoaib

BS (Computer Science)

Session 2018 – 2022

FACULTY OF COMPUTING AND INFORMATION TECHNOLOGY
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF GUJRAT

STATEMENT OF SUBMISSION

This is to certify that **Afshan Wain** Roll No. 18321519-001, **Eza Tehreem Asjad** Roll No. 18321519-003 and **Azbha Aleem** Roll No. 18321519-042 has successfully completed the final year project named as **Urdu Language Recognition System** at the Department of Computer Science, University of Gujrat, to fulfill the requirement of the degree of BSCS in Computer Science

Project Supervisor

Dr. Umar Shoaib

Department of Computer Science,
Faculty of Computing & Information Technology,
University of Gujrat, Punjab, Pakistan.
Email: umar.shoaib@uog.edu.pk

Dated: 08-26-2022

Abstract

Language Recognition Systems have received a significant improvement in performance recently, mainly due to the increase in capabilities of artificial intelligence algorithms. However, this advancement is not evenly distributed over all languages. Urdu is among the languages which did not receive much attention, especially in the font independent perspective. There exists no automated system that can reliably recognize printed Urdu text in images across different fonts. To help bridge this gap, we have developed Urdu Language Recognition System. The aim of this project is to develop a system for Android based mobile devices. The purpose of this application is to recognize Urdu text for different fonts in scanned text documents, text images, and any picture taken by an Android based device in order to reuse it later. This application will allow its users to perform many actions in few minutes, such as copy text from these aforementioned documents and modify it, instead of wasting time on retyping it. We have also developed Convolutional Neural Network (CNN) based classification models which can recognize Urdu ligatures. We believe that this application can serve as the basis for further improvement in the performance of Urdu language recognition systems.

TABLE OF CONTENTS

CHAPTER 1: PROJECT FEASIBILITY REPORT.....	6
.....	6
1.1. INTRODUCTION	7
1.2. PROJECT/PRODUCT FEASIBILITY REPORT.....	7
1.2.1. <i>Technical Feasibility</i>	7
1.2.2. <i>Operational Feasibility</i>	7
1.2.3. <i>Economic Feasibility</i>	7
1.2.4. <i>Schedule Feasibility</i>	8
1.2.5. <i>Specification Feasibility</i>	8
1.2.6. <i>Information Feasibility</i>	8
1.2.7. <i>Motivational Feasibility</i>	8
1.2.8. <i>Legal & Ethical Feasibility</i>	8
1.3. PROJECT/PRODUCT SCOPE	8
1.4. PROJECT/PRODUCT COSTING	9
1.4.1. <i>Project Cost Estimation by Function Point Analysis</i>	9
1.4.2. <i>Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)</i> ..	9
1.4.3. <i>Activity Based Costing</i>	10
1.5. TASK DEPENDENCY TABLE	10
1.6. CPM - CRITICAL PATH METHOD.....	11
<i>Sequence of Activities:</i>	11
<i>Network Diagram:</i>	11
1.7. GANTT CHART	12
1.8. INTRODUCTION TO TEAM MEMBER AND THEIR SKILL SET	12
1.9. TASK AND MEMBER ASSIGNMENT TABLE	12
1.10. TOOLS AND TECHNOLOGY WITH REASONING	14
1.11. VISION DOCUMENT	14
1.12. RISK LIST	14
<i>Development level risks:</i>	14
<i>Technical risks:</i>	14
<i>Operational risks:</i>	14
<i>Maintenance and design level risks:</i>	15
1.13. PRODUCT FEATURES/ PRODUCT DECOMPOSITION.....	15
SECOND DELIVERABLE FOR OBJECT ORIENTED APPROACH	16
CHAPTER 2 SYSTEM REQUIREMENT SPECIFICATION	16
2.1 INTRODUCTION.....	17
2.1.1 <i>Systems Specifications</i>	17
2.1.2. <i>Identifying External Entities</i>	19
2.1.3. <i>Context Level Data Flow Diagram</i>	19
2.1.4. <i>Capture "shall" Statements</i>	20
2.1.5. <i>Allocate Requirements</i>	20
2.1.6. <i>Prioritize Requirements</i>	20

2.1.7. Requirements Trace-ability Matrix.....	21
2.2.10. High Level Use Case Diagram	22
2.2.11. Analysis Level Use Case Diagram.....	23
2.2.12. Use Case Description	23
CHAPTER 3 DESIGN DOCUMENT	27
3.1. INTRODUCTION.....	28
3.2. DOMAIN MODEL	28
3.3. SYSTEM SEQUENCE DIAGRAM	29
3.4. SEQUENCE DIAGRAM	30
3.5. COLLABORATION DIAGRAM.....	30
3.6. OPERATION CONTRACTS.....	31
3.7. DESIGN CLASS DIAGRAM.....	32
3.8. STATE CHART DIAGRAM	33
CHAPTER 4: USER INTERFACE DESIGN	34
4.1. INTRODUCTION.....	35
4.2. SITE MAPS	35
4.3. STORY BOARDS	36
4.4. NAVIGATIONAL MAPS:	38
4.4.1. Scan Image button Result.....	38
4.5 TRACE-ABILITY MATRIX.....	40
CHAPTER 5: SOFTWARE TESTING	41
5.1 INTRODUCTION:	42
5.2. TEST PLAN.....	42
7.2.1. Purpose	42
7.2.2. Outline.....	42
5.3. TEST DESIGN SPECIFICATION	46
5.3.1. Purpose	46
5.3.2. Outline.....	47
5.4. TEST CASE SPECIFICATION.....	49
5.4.1. Purpose	49
5.4.2. Outline.....	49
5.5. TEST PROCEDURE SPECIFICATION	52
5.5.1. Purpose	52
5.5.2 Outline.....	52
5.6. TEST ITEM TRANSMITTAL REPORT	54
5.6.1. Purpose	54
5.6.2. Outline.....	54
5.7. TEST LOG	55
5.7.1. Purpose	55
5.7.2. Outline.....	55
5.8. TEST INCIDENT REPORT	57
5.8.1. Purpose	57
5.8.2. Outline.....	57
5.9. TEST SUMMARY REPORT.....	60

5.9.1. Purpose	60
5.9.2. Outline.....	60
CHAPTER 6: USER MANUAL	63
6.1. GENERAL INFORMATION	64
6.2 SYSTEM SUMMARY	64
6.3. GETTING STARTED	64
6.3.1 Main Screen and Image Scanner	65
6.3.2 Edit and Crop Image.....	65
6.3.3 Output Text Screen.....	66
APPENDIX 1	67
APPENDIX 1	68

Chapter 1: Project Feasibility Report

1.1. Introduction

Our project “Urdu Language Recognition System” is an android-based application. This project focuses on font-independent recognition of printed text in Urdu. This system has the ability to grab text from image including both actual photos you have taken and screenshots of machine printed text you have captured. User will open the application and select image file for recognition. The purpose of this application is to recognize text in scanned text documents, text images, and any picture taken by an Android based device in order to reuse it later. This application will allow its users to perform many actions in few minutes, such as copy text from these aforementioned documents and modify it, instead of wasting time on retyping it.

1.2. Project/Product Feasibility Report

1.2.1. Technical Feasibility

The proposed system can definitely be developed as there are resources available such as the font-independent Urdu Dataset. My teammates and I possess the technical expertise to develop such a system. However, the computational cost for training such kind of models is very high. In practice very few networks are trained from scratch since it requires relatively large data set, huge computation power and it is also harder to train a complex network from scratch e.g., modern CNN architectures require weeks of training on high-end GPU clusters.

1.2.2. Operational Feasibility

Nowadays, there is an enormous demand in storing any information available on papers, such as books or newspapers in mobile phones. There is an existing way to store information by scanning the desired text, but it will be stored as an image that won't help for further processing. For instance, if we store scanned text images, we can't read the text word by word, or line by line. the text in these scanned images can't be reused unless we rewrite the whole content by ourselves. Scanned documents are great. They let us archive stacks of paper into folder, taking up far less space and being infinitely easier to organize, move, and copy. What's not so great is finding content stored away inside one of our hundreds of scanned documents. By default, they're little more than a picture of our document and if we want to find info inside them, we will have to open each one and read it for ourselves. Or, we could let our device do the heavy lifting for us, by turning your image into text and letting us search through our scanned documents as easily as we search through any other documents. Proposed system will be able to solve this problem for Urdu language and will help many people and organizations that work with Urdu language.

1.2.3. Economic Feasibility

Cost Estimates:

We will not have any charges for the development of our application. The reason for this is that we have front end developer and a Backend developer in our team. However, for the development of training model the costs of computers, memory and GPU takes into account.

Benefit Estimates:

Our first and foremost benefit is that we are a working in a team and has the capacity for building initial phase of the project which will reduce a lot of money and this will indirectly increase our revenue.

1.2.4. Schedule Feasibility

The completion of a project with the available staff and resources within time is very essential. So, we will divide the project into tasks and sub tasks according to the durations and dependencies to complete those tasks. Including CPM also provide the feasible solution to our project.

1.2.5. Specification Feasibility

To make our project feasible and to satisfy end user interviews and meeting with staff or some end user is arranged so we can say proposed system is feasible. Knowing requirements and constraints of system will make work a lot easier during development of system it will also be of great for end user as he will already know what system is capable of or what not.

1.2.6. Information Feasibility

For our system to be developed properly training of dataset is done so by the end result given by the system is accurate and efficient.

1.2.7. Motivational Feasibility

“Urdu language recognition system” its aim is to make it easier to process Urdu documents and to provide relevant online content in local languages in order to more effectively utilize data for the benefit of the people of Pakistan.

With help of this system a lot of work related to all Urdu documents that have Urdu text in images will be processed faster that will also help in understanding our language internationally

1.2.8. Legal & Ethical Feasibility

The company and the developer’s perspective test the ethical feasibility of our project and include the maintenance of security rules and regulations.

1.3. Project/Product Scope

Urdu Language Recognition system focuses on the following scope of work:

1. The Urdu Language Recognition System will be an android application that will able to recognize 18,569 Urdu ligatures in different fonts.
2. The system will be able to only recognize the printed text and will not include recognition of handwritten text.
3. The images with plain background will be recognized by the system. Recognition of text in images with noisy background is not included in the scope of this project.
4. The system will output plain Urdu text in Unicode format which can be copied, selected and modified by the user.
5. More work, which includes real time Urdu Recognition and multilingual language recognition is not handled in this project.

1.4. Project/Product Costing

1.4.1. Project Cost Estimation by Function Point Analysis

Type of Component		Complexity of component		
	Low	Average	High	Total
External Input	1*3=6	2*4=8	10*6=60	71
External Output	1*4=4	1*5=5	1*7=7	16
External Logical Files	1*7=7	5*10=50	5*15=75	132
External Interface Files	5*5=25	6*7=42	8*10=80	147
Total UFP				312

Number	Complexity Weighting Factor	Value
1	Backup and Recovery	1
2	Data Communication	8
3	Distributed Processing	7
4	Performance critical	6
5	Existing operating environment	3
6	Online data entry	2
7	Input transaction over multiple screens	1
8	Master files updated online	3
9	Information domain value complex	4
10	Internal processing complex	7
11	Code designed for reuse	7
12	Conversion/ Installation in design	4
13	Multiple installations	3
14	Application designed for change	2
	Total complexity adjustment value	58

To compute function points (FP), the following relationship is used:

$$\begin{aligned}
 \text{FP est.} &= \text{Count Total} * [0.65 + 0.01 * (Fi)] \\
 &= 312 * [0.65 + 0.01 * (58)] \\
 &= 383.76
 \end{aligned}$$

Finally, Total Project Cost and Total Project Effort are calculated given the average productivity parameter for the system.

The formulae are given as follows:

Cost / FP = labor rate / productivity parameter

Total Project Cost = FP est. * (cost / FP)

Total Estimated Effort = FP est. / productivity parameter

1.4.2. Project Cost Estimation by using COCOMO'81 (Constructive Cost Model)

Our project is based on Semi-detached Type which is an intermediate stage between

organic and embedded types.

Basic COCOMO

Type	Effort	Schedule
Semi-Detached	PM= 3.0 (4.5) ^{1.12}	TD=2.5(20.24) ^{0.35} =7.16

PM= person-month (effort)

KLOC= lines of code, in thousands

TD= number of months estimated for software development (duration)

The schedule is determined using the Basic COCOMO schedule equations.

$$\begin{aligned}
 \text{People Required} &= \text{Effort} / \text{Duration} \\
 &= 20.24 / 7.16 \\
 &= 3 \text{ persons}
 \end{aligned}$$

1.4.3. Activity Based Costing

Activities	Resources	Cost	Duration
Dataset Collection	Kaggle	1k	15 days
Designing	Adobe Illustrator	2k	1 month
Implement	Deep Learning Library	10k	1 month
Android Development	Android Studio	10k	2-3 months
Testing	Self-Testing	0	15 Days

1.5. Task Dependency Table

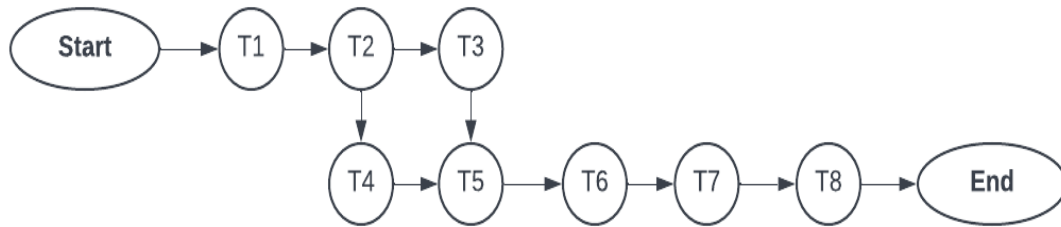
Tasks	Tasks details	Duration in weeks	Dependencies
T1	Idea and Proposal	2	-
T2	D2 document	2	T1
T3	Dataset collection	2	T2, T1
T4	Design	4	T2, T3
T5	Coding	3	T3, T4
T6	Training	3	T3, T5
T7	Testing	4	T5, T6
T8	Implementation	4	T4, T5, T6

1.6. CPM - Critical Path Method

Sequence of Activities:

Activity	Activity details	Duration in weeks	Immediate predecessor
T1	Idea and Proposal	2	-
T2	D2 document	2	T1
T3	Dataset collection	2	T2, T1
T4	Design	4	T2, T3
T5	Coding	3	T3, T4
T6	Training	3	T3, T5
T7	Testing	4	T5, T6
T8	Implementation	4	T4, T5, T6

Network Diagram:



Estimate Activity Completion Time:

Activity	Duration	ES	EF	LS	LF	TS	FS
T1	2	0	2	0	2	0	0
T2	2	2	4	2	4	0	0
T3	2	4	6	6	8	2	2
T4	4	4	8	4	8	0	0
T5	3	8	11	8	11	0	0
T6	3	11	14	11	14	0	0
T7	4	14	18	14	18	0	0
T8	4	18	22	18	22	0	0

Identify the Critical Path

From above table we can find critical path

The critical path is:

T1, T2, T4, T5, T6, T7, T8

1.7. Gantt chart

Task Name	Start	End	Duration	Mar	Apr	May	June	July	Aug	Sep
Proposal	24/2/2022	7/3/2022	16 days	■						
Data Set Collection	8/3/2022	23/3/2022	15 days		■					
Design	24/3/2022	24/4/2022	1 month			■				
Coding	25/4/2022	25/6/2022	2 months				■	■		
Testing	26/6/2022	26/7/2022	1 month						■	
Implementation	22/7/2022	21/8/2022	1 month							■

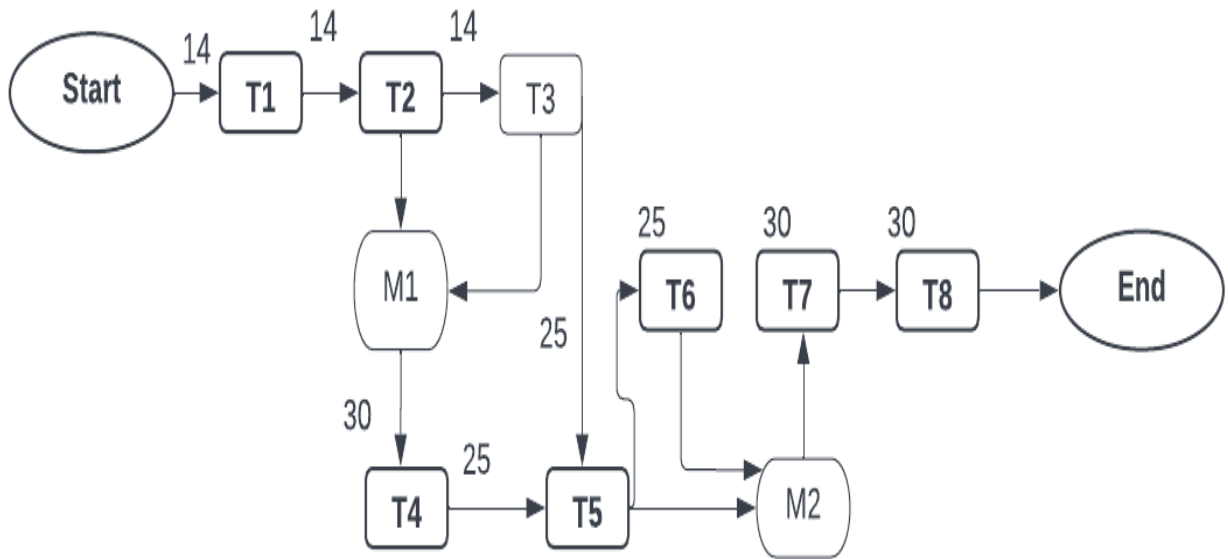
1.8. Introduction to Team member and their skill set

Team Member Name	Skill set
Afshan Wain	Python Developer, Android Developer, Researching
Eza Tehreem Asjad	Dataset Collection, Designing, Researching
Azbha Aleem	Android Developer, Designing

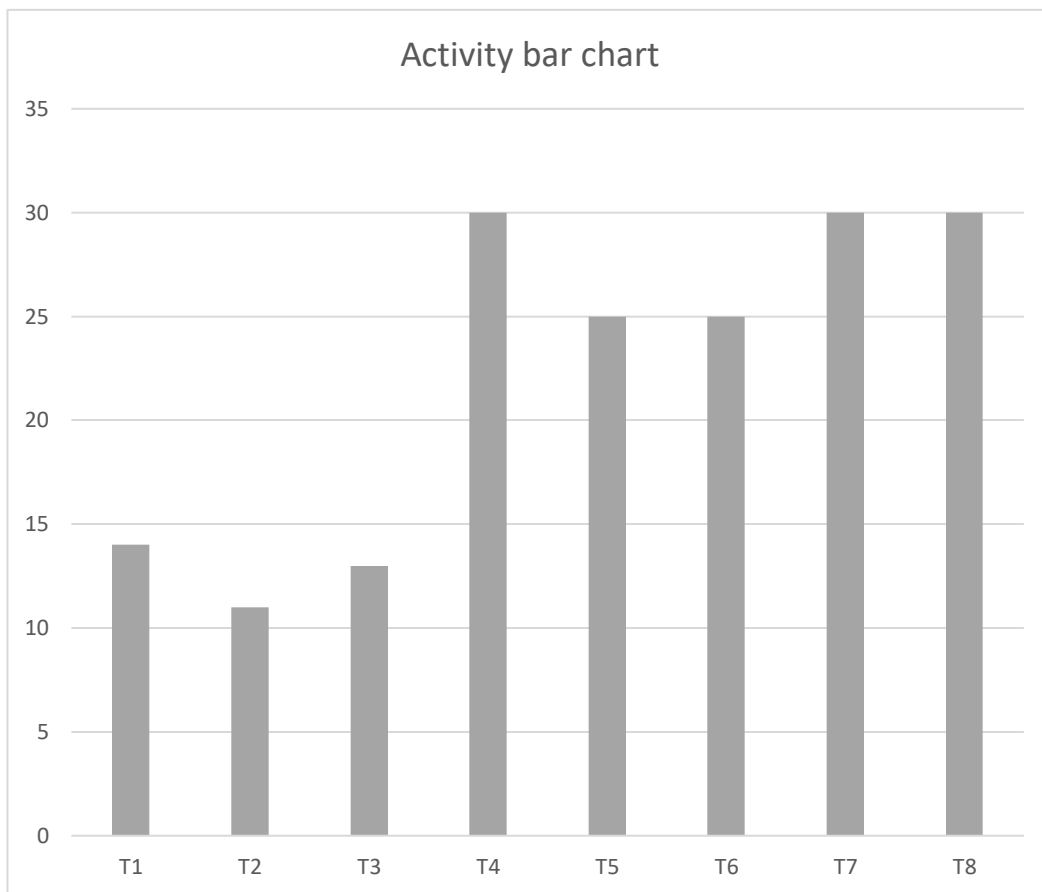
1.9. Task and Member Assignment Table

Task	Duration (days)	Dependencies
T1	14	-
T2	14	T1
T3	14	T1, T2
T4	30	T2, T3(M1)
T5	25	T3, T4
T6	25	T3, T5
T7	30	T5, T6 (M2)
T8	30	T4, T5, T6

Task durations and dependencies



Activity Bar Chart



1.10. Tools and Technology with reasoning

Tools	
Google Colab	For training and testing models
Draw.io	For creating simple activity and class diagrams
Adobe Illustrator	For designing diagrams for documentation
Android Studio	To build Application
Technologies	
Pandas & Numpy	For performing operations on data
Matplotlib	For visualization of data
Seaborn	For more advance visualization of data
Pytorch	For creating neural network models
NLTK	For text processing

1.11. Vision Document

The vision of the project is to create an android application on recognition of printed text in Urdu independent of the font style. It will make it easier to process Urdu documents and to provide relevant online content in local languages in order to more effectively utilize data for the benefit of the public in Pakistan.

1.12. Risk List

Development level risks:

- Time risks may occur because limited time is available to complete this project as the limited time can be delayed or exceed from the required time.
- The finance risk may occur because sufficient resources are available for the development and maintenance of our project but more resources can be used to efficiently resolve our problems which can be occur.

Technical risks:

- High complexity in implementation
- Result with less accurate performance than expected
- Improper integration of modules
- Less number of skilled employees

Operational risks:

- Conflict between tasks and employee
- No proper planning about project
- Lack of clarity in roles and responsibilities
- Lack of communication and cooperation
- Improper management of tasks

Maintenance and design level risks:

Estimation: The uncertainty about the parameters of work time, cash flow process and the rate of defects, repair, and duration can occur. Therefore, the development team can go wrong in calculating estimation about these factors.

Reusability: The functionality of our project can be affected due to the reusability of the modules.

Scope: The total features requested may be beyond what the development team can deliver in the time available.

1.13. Product Features/ Product Decomposition

1. Training of model for font-independent Urdu Recognition
2. Recognition of Urdu text from input image
3. Processing of recognized text i.e., select, edit, search, save

Second Deliverable for Object Oriented Approach

Chapter 2 System Requirement Specification

2.1 Introduction

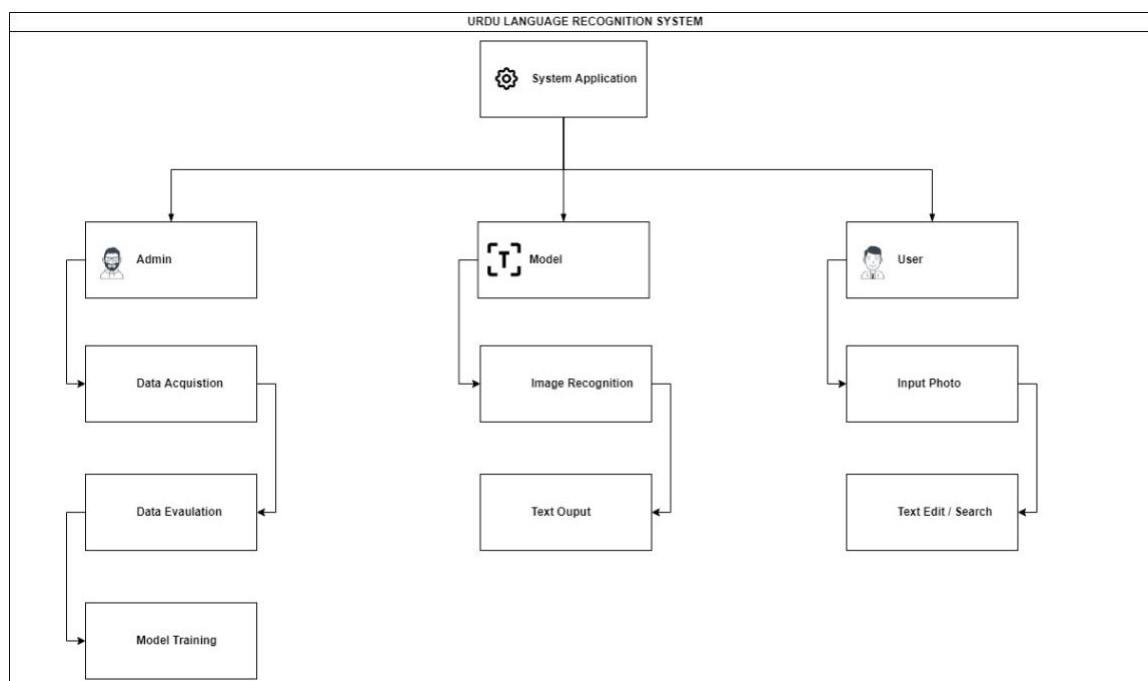
Mobile applications grew in less than two decades to achieve the status of the largest information repository in human history. By providing efficient, fast, consistent and authentic tools in the form of internet and mobile applications, information technology is penetrating human life and is playing an important role in changing lives of so many people around the globe. Today many traditional industrial firms are moving towards utilizing information technology including mobile applications. Mobile applications run banking transactions, air traffic, and emergency room equipment. Nowadays, there is an enormous demand in storing any information available on papers, such as books or newspapers in mobile phones. There is an existing way to store information by scanning the desired text, but it will be stored as an image that won't help for further processing. For instance, if we store scanned text images, we can't read the text word by word, or line by line. The text in these scanned images can't be reused unless we rewrite the whole content by ourselves. For this reason, we need a software for the recognition of text. Scanned documents are great. They let us archive stacks of paper into folder, taking up far less space and being infinitely easier to organize, move, and copy. What's not so great is finding content stored away inside one of our hundreds of scanned documents. By default, they're little more than a picture of our document and if we want to find information inside them, we will have to open each one and read it for ourselves or we could let our device do the heavy lifting for us, by turning your image into text and letting us search through our scanned documents as easily as we search through any other documents.

2.1.1 Systems Specifications

Existing System

A few commercial Language Recognition applications are available for Arabic script-based languages but they lack in performance. Urdu is the super set of Arabic and Persian languages with some additional characters. Due to these additional characters, designing a Language Recognition system for Urdu is even more complex than Arabic and Persian. Among the fonts used to print Urdu text, Nastaliq and Naskh are the most common and there have been some successful attempts to build LRS specific to these fonts. However, there are more than 400 fonts registered for Urdu and to the best of our knowledge there does not exist any Language Recognition system that can recognize Urdu text written in all these fonts. One of the major reasons for that is huge inter and intra-class variability found in Urdu characters across fonts. Unlike English and other Latin-based languages, letters in Urdu are not constrained to only a single shape. Also, Urdu is a bidirectional language where normal text is written from right to left, while numbers are written from left to right. All these challenges make it hard to build a robust Language Recognition system and that is the reason why Urdu Language Recognition systems are less mature in performance than other languages.

Organizational Chart



Scope of the System

Urdu Language Recognition system focuses on the following scope of work:

1. The Urdu Language Recognition System will be an android application that will be able to recognize 18,569 Urdu ligatures in different fonts.
2. The system will be able to only recognize the printed text and will not include recognition of handwritten text.
3. The images with plain background will be recognized by the system. Recognition of text in images with noisy background is not included in the scope of this project.
4. The system will output plain Urdu text in Unicode format which can be copied, selected and modified by the user.
5. More work, which includes real time Urdu Recognition and multilingual language recognition is not handled in this project.

Summary of Requirements (Initial Requirements)

The purposed system must fulfill following requirements as follow:

1. Taking/ Choosing the Desired Text Image

The most important thing here is the use of an Android mobile phone and its camera. The user can take a picture of a text image or choose one from the mobile's directory. The user must use a camera of typical resolution and take a picture of a text image or choose one from existing ones in his phone.

2. Recognition of the Text

The application must be able to recognize the Urdu text from the chosen image.

3. Copying the Text for Different Uses.

Once the text is recognized and ready to be used, the user will be able to copy, edit, and modify it. He/she may also be able to retrieve the data from the image and store it directly on the phone.

2.1.2. Identifying External Entities

The Identification of External Entities is done in two phases.

a. Over Specify Entities from Abstract

On the basis of the abstract, one might identify the following entities:

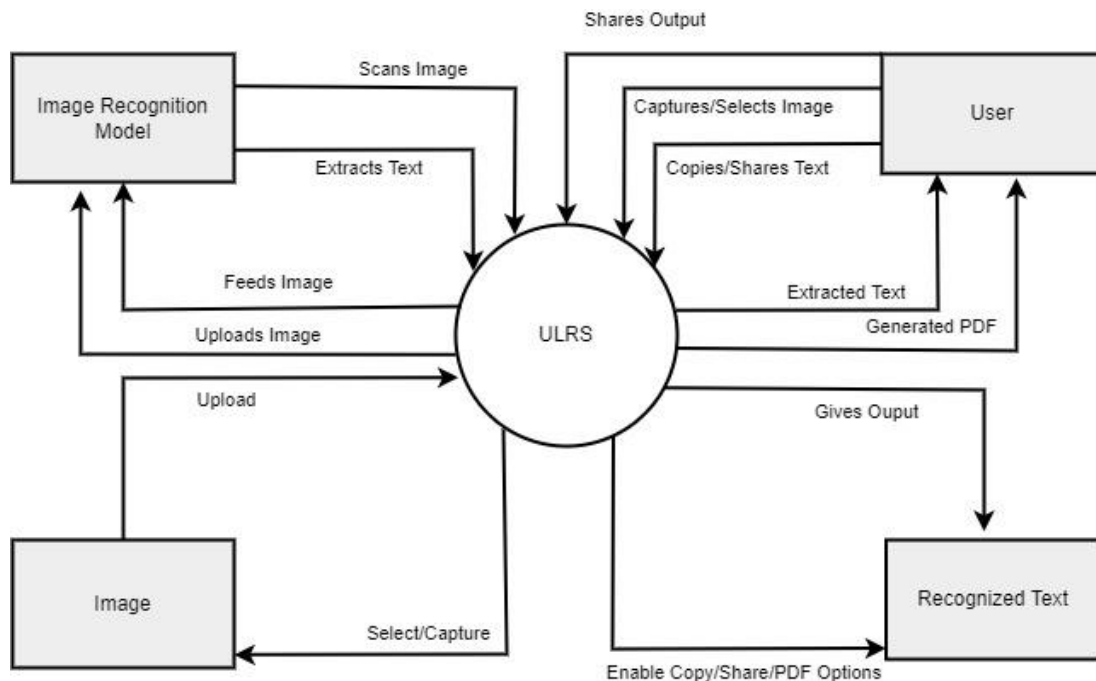
1. Recognition System
2. Classification model
3. User

b. Perform Refinement

After over specifying the entities, you have to refine them on the basis of your Business Logic. For example, in this example we found the following entities more related to our Business Logic:

1. Urdu Language Recognition System
2. Classification model
3. User

2.1.3. Context Level Data Flow Diagram



2.1.4. Capture "shall" Statements

Para #	Initial Requirements
1.0	A user “shall” take a photo of image with printed Urdu text
1.0	A user “shall” pick a photo with printed Urdu text from gallery
1.0	An administrator “shall” train the system on Urdu text with different fonts
1.0	System “shall” recognize the Urdu text from the image provided by the user
2.0	A user “shall” select the recognized Urdu text
2.0	A user “shall” copy the recognized Urdu text
2.0	A user “shall” edit the recognized Urdu text
2.0	A user “shall” search the recognized Urdu text

2.1.5. Allocate Requirements

Para #	Initial Requirements	Use Case Name
1.0	A user “shall” take a photo of image with printed Urdu text	UC_Take_Image
1.0	A user “shall” pick a photo with printed Urdu text from gallery	UC_Pick_Image
1.0	An administrator “shall” train the system on Urdu text with different fonts	UC_Train_System
1.0	System “shall” recognize the Urdu text from the image provided by the user	UC_Recognize_Image
2.0	A user “shall” select the recognized Urdu text	UC_Select
2.0	A user “shall” copy the recognized Urdu text	UC_Copy
2.0	A user “shall” edit the recognized Urdu text	UC_Edit
2.0	A user “shall” search the recognized Urdu text	UC_Search

2.1.6. Prioritize Requirements

Para #	Rank	Initial Requirements	Use Case ID	Use Case Name
1.0	Highest	System “shall” recognize the Urdu text from the image provided by the user	UC_1	UC_Recognize_Image
1.0	Highest	An administrator “shall” train the system on Urdu text with different fonts	UC_2	UC_Train_System

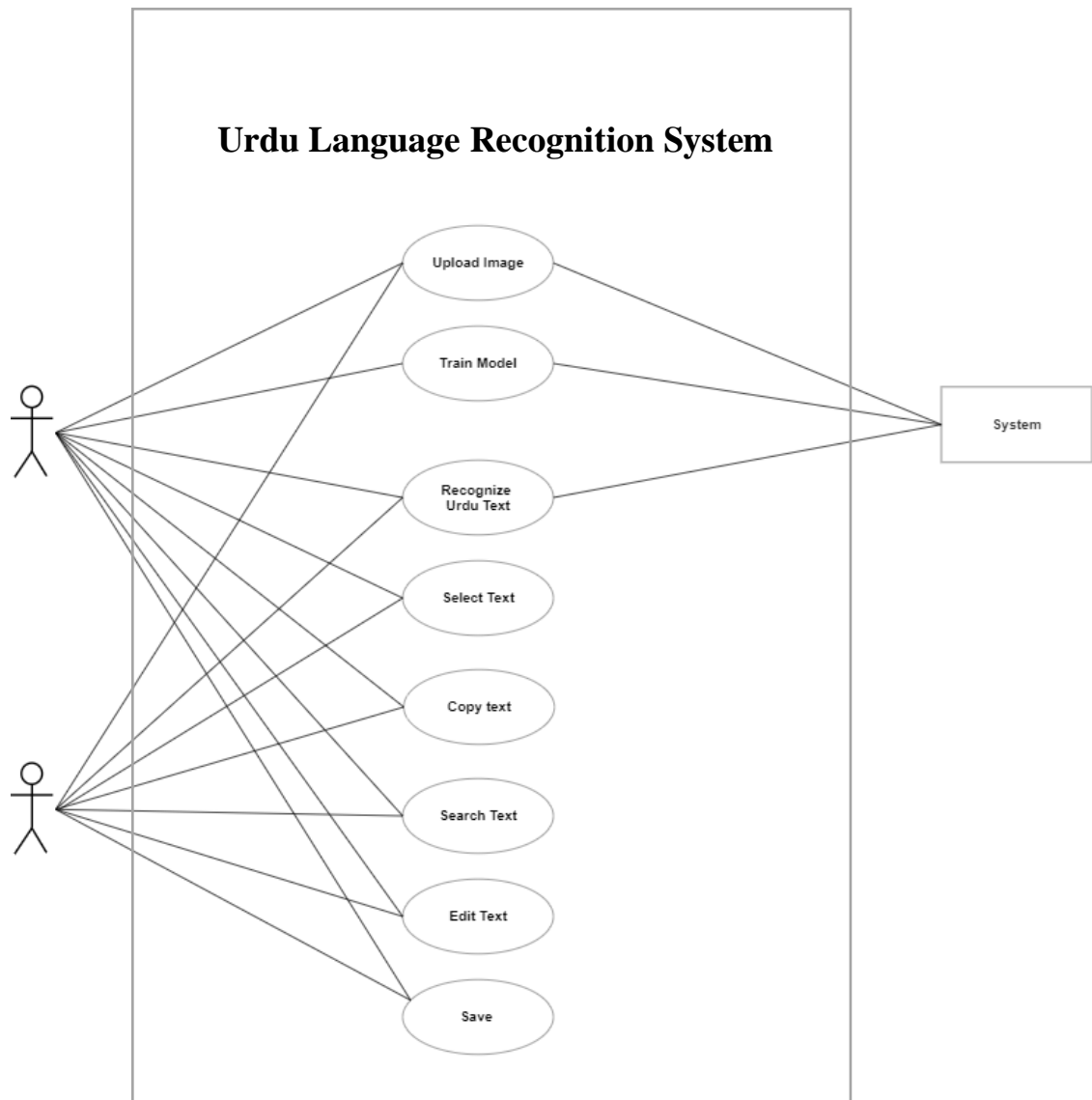
2.0	Highest	A user “shall” edit the recognized Urdu text	UC_3	UC_Edit
1.0	Highest	A user “shall” take a photo of image with printed Urdu text	UC_4	UC_Take_Image
1.0	Highest	A user “shall” pick a photo with printed Urdu text from gallery	UC_5	UC_Pick_Image
2.0	Medium	A user “shall” select the recognized Urdu text	UC_6	UC_Select
2.0	Medium	A user “shall” copy the recognized Urdu text	UC_7	UC_Copy
2.0	Medium	A user “shall” search the recognized Urdu text	UC_8	UC_Search

2.1.7. Requirements Trace-ability Matrix

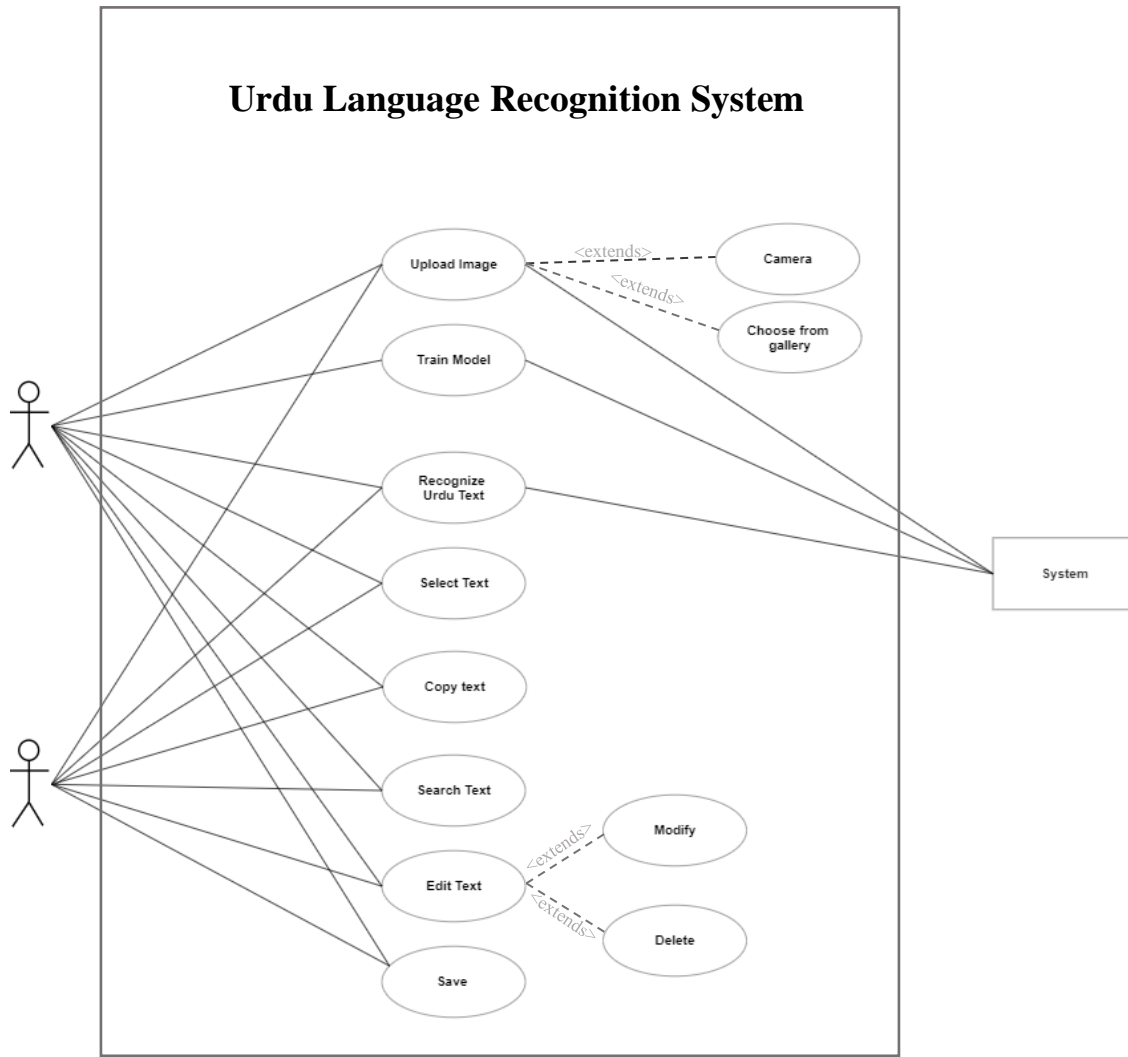
Sr#	Para #	System Specification Text	Build	Use Case Name	Category
1	1.0	System “shall” recognize the Urdu text from the image provided by the user	B1	UC_Recognize_Image	Technical
2	1.0	An administrator “shall” train the system on Urdu text with different fonts	B1	UC_Train_System	Technical
3	2.0	A user “shall” edit the recognized Urdu text	B1	UC_Edit	Technical
4	1.0	A user “shall” take a photo of image with printed Urdu text	B1	UC_Take_Image	Technical
5	1.0	A user “shall” pick a photo with printed Urdu text from gallery	B1	UC_Pick_Image	Technical
6	2.0	A user “shall” select the recognized Urdu text	B1	UC_Select	Technical

7	2.0	A user “shall” copy the recognized Urdu text	B1	UC_Copy	Technical
8	2.0	A user “shall” search the recognized Urdu text	B1	UC_Search	Technical

2.2.10. High Level Use Case Diagram



2.2.11. Analysis Level Use Case Diagram



2.2.12. Use Case Description

UC_Recognize_Image (UC_1):

Use Case Description	
Use Case ID	UC_1
Use Case Name	UC_Recognize_Image
Description	The Administrator trains the system according to the given symbols or alphabets. Then, the characters are recognized from the image given by user after the system is trained.
Primary Actors	Administrator, User
Secondary Actors	System
Preconditions	Before trying to recognize the characters, the system should be trained first with the font characteristics and the font size.

Basic Flow	<ol style="list-style-type: none"> 1. The administrator trains the system to recognize the characters. 2. User provides the image for recognition 3. The trained system then recognizes the characters and return the text to the user
Postconditions	The characters are recognized and are displayed on screen as text.
Extension	

UC_Train_System (UC_2):

Use Case Description	
Use Case ID	UC_2
Use Case Name	UC_Train_System
Description	The administrator trains the system for the recognition of Urdu text from the images independent of the font.
Primary Actors	Administrator
Secondary Actors	System
Preconditions	Administrator has the required dataset of characters and ligatures for the independent recognition of Urdu text.
Basic Flow	<ol style="list-style-type: none"> 1. Administrator inputs the specific dataset in order to train the system. 2. After entering it, the application uploads the dataset in memory. 3. Administrator defines the parameters and finally trains the system. 4. The system gets trained and become ready to be used.
Alternate Flow	<ol style="list-style-type: none"> 4a. The system gets trained but shows low accuracy 4b. Administrator tunes the parameters and trains the system again to achieve the required accuracy.
Extension	

UC_Edit (UC_3):

Use Case Description	
Use Case ID	UC_3
Use Case Name	UC_Edit
Description	The user chooses to edit the recognized Urdu text
Primary Actors	User of the System
Secondary Actors	System
Preconditions	The application has processed the image and extracted its text to that can be edited.
Basic Flow	<ol style="list-style-type: none"> 1. The User has selected the text or the part of the text that he wants to edit. 2. The user edits the text or part of the text.

	3. The application updates the text and stores it in the system memory.
Postconditions	The text is updated
Extension	

UC_Take_Image (UC_4):

Use Case Description	
Use Case ID	UC_4
Use Case Name	Take a Photo
Description	The user takes a valid photo of the text that needs to be digitalized
Primary Actors	User of the application
Secondary Actors	System
Preconditions	Application is waiting for the user either to choose a photo from the gallery or to take one
Basic Flow	<ol style="list-style-type: none"> 1. User chooses the take photo icon 2. User takes a photo
Postconditions	The photo is taken and is ready to be used.
Extension	

UC_Pick_Image (UC_5):

Use Case Description	
Use Case ID	UC_5
Use Case Name	UC_Pick_Image
Description	The user chooses a photo that contains the text from the directory to be converted.
Primary Actors	User of the application
Secondary Actors	System
Preconditions	Application is waiting for the user either to choose a photo from the directory or to take one.
Basic Flow	<ol style="list-style-type: none"> 1. User chooses the take photo icon. 2. User chooses the desired image with Urdu text
Postconditions	The photo is taken and is ready to be used.
Extension	

UC_Select (UC_6):

Use Case Description	
Use Case ID	UC_6
Use Case Name	Select Text
Description	The user selects the text that was converted from the image.
Primary Actors	User of the application
Secondary Actors	System

Preconditions	The application has processed the image and extracted its text to that can be selected.
Basic Flow	1. The application displays the recognized Urdu text on the screen 2. The user gets the text from the application and selects it.
Postconditions	User is able to select the text from the image.
Extension	

UC_Copy (UC_7):

Use Case Description	
Use Case ID	UC_7
Use Case Name	Copy Text
Description	The user copies the text that was converted from the image
Primary Actors	User of the application
Secondary Actors	System
Preconditions	The application has recognized the text and is displayed on the screen
Basic Flow	1. The User has selected the text or the part of the text that wants to copy. 2. The user copies the text or part of the text that he\she needs
Postconditions	The text from the photo is copied and is ready to be used.
Extension	

UC_Search (UC_8):

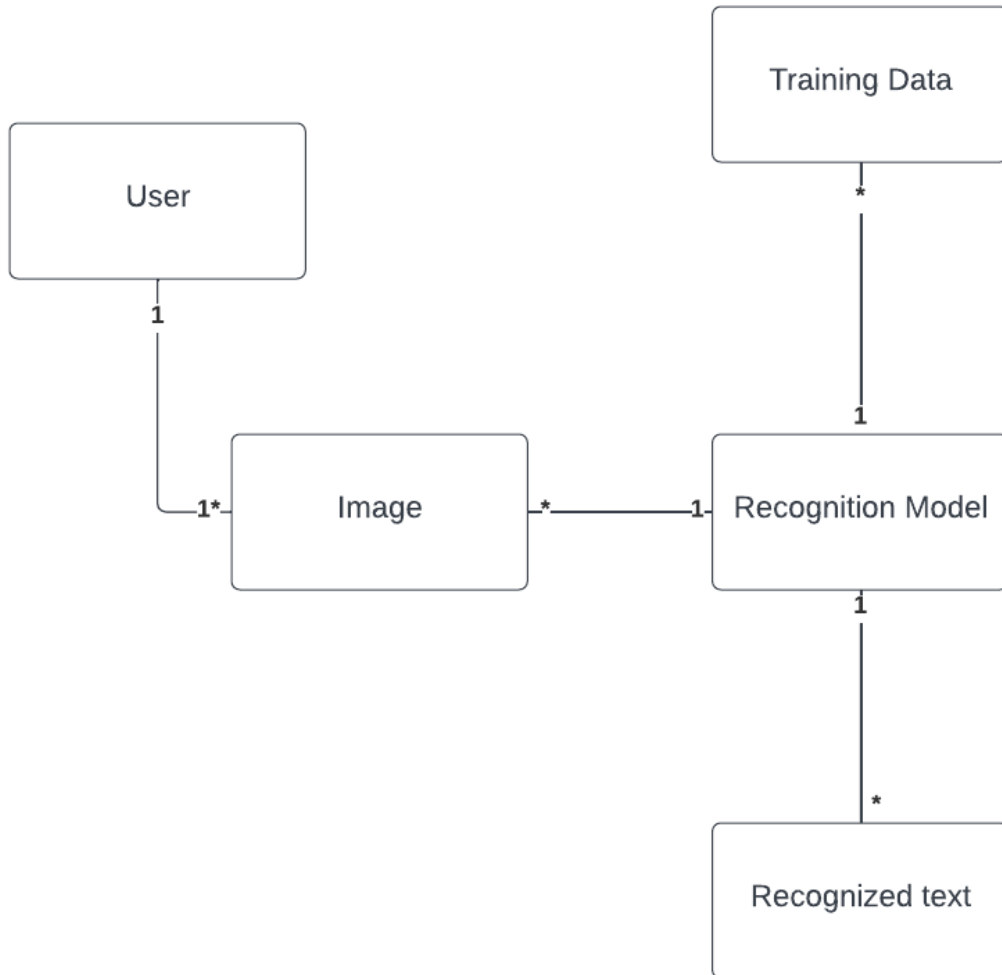
Use Case Description	
Use Case ID	UC_8
Use Case Name	UC_Search
Description	The user looks for the recognized text in the search engine
Primary Actors	User of the application
Secondary Actors	System
Preconditions	The application has recognized the text and is displayed on the screen
Basic Flow	1. The User has selected the text or the part of the text that wants to search. 2. The user clicks the search icon 3. The application redirects the user to the search engine with the searched text.
Postconditions	
Extension	

Chapter 3 Design Document

3.1. Introduction

Third deliverable is all about the software design. In the previous deliverable, analysis of the system is completed. So, we understand the current situation of the problem domain. Now we are ready to strive for a solution for the problem domain by using object-oriented approach.

3.2. Domain Model



3.3. System Sequence Diagram

Model Training

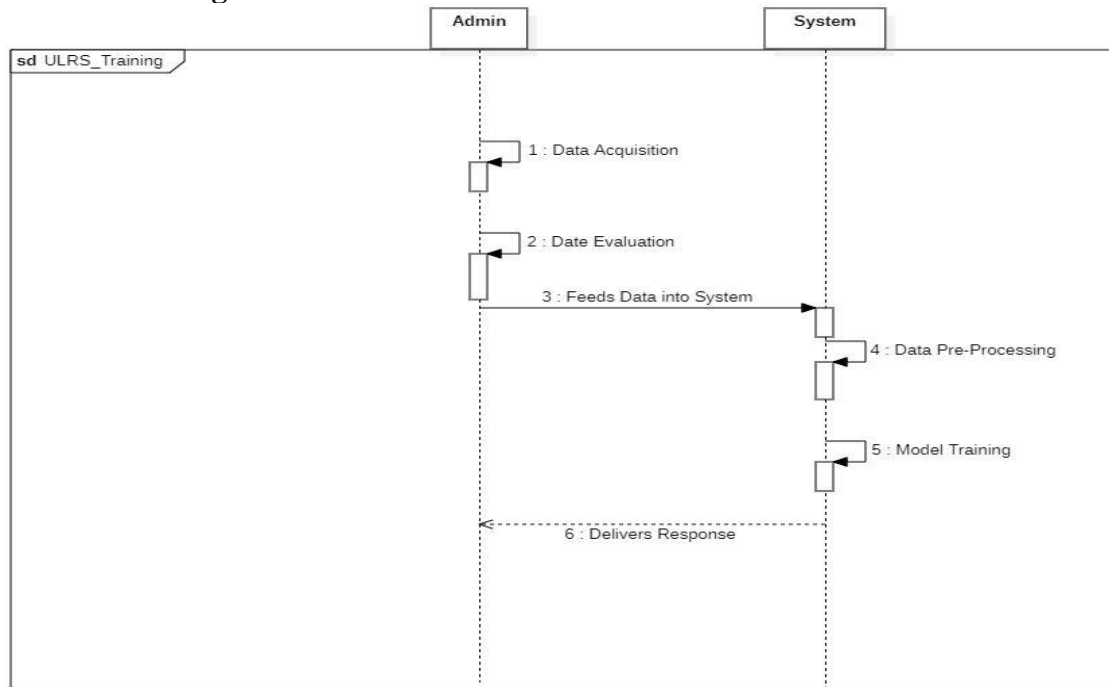
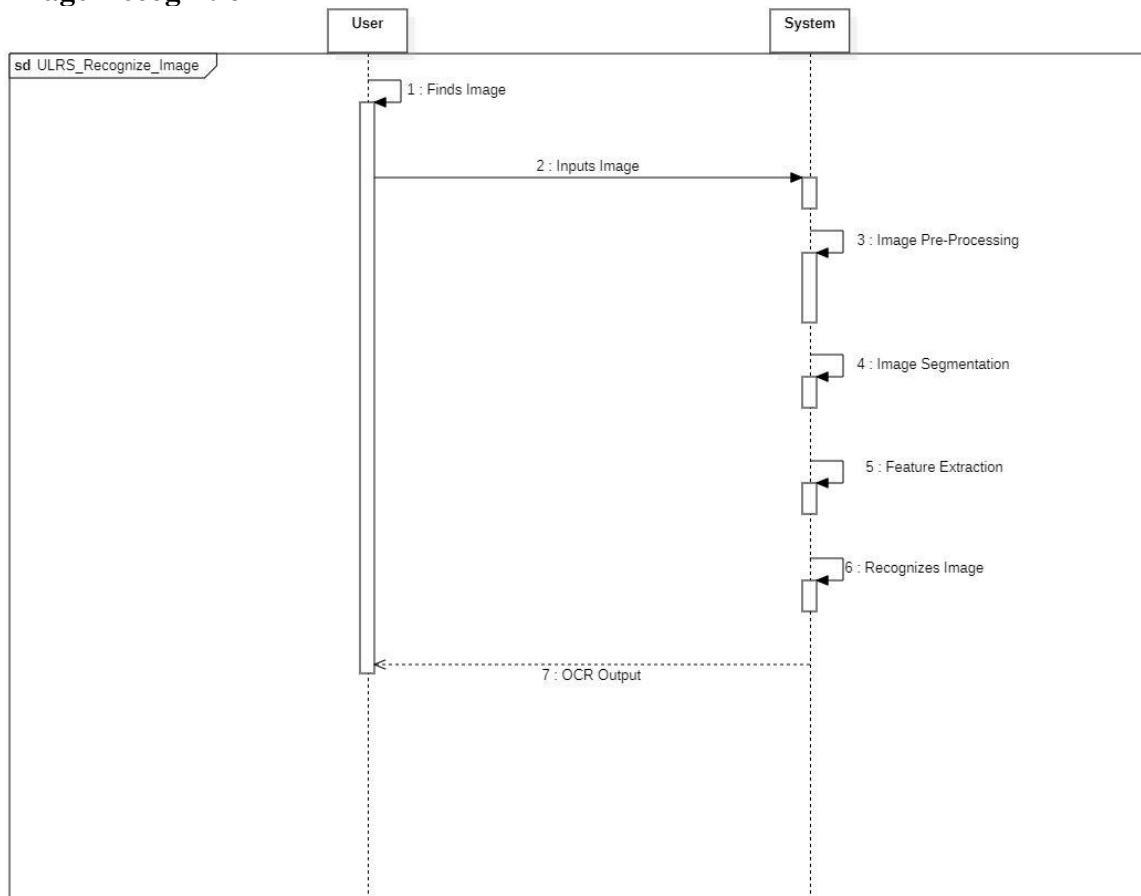
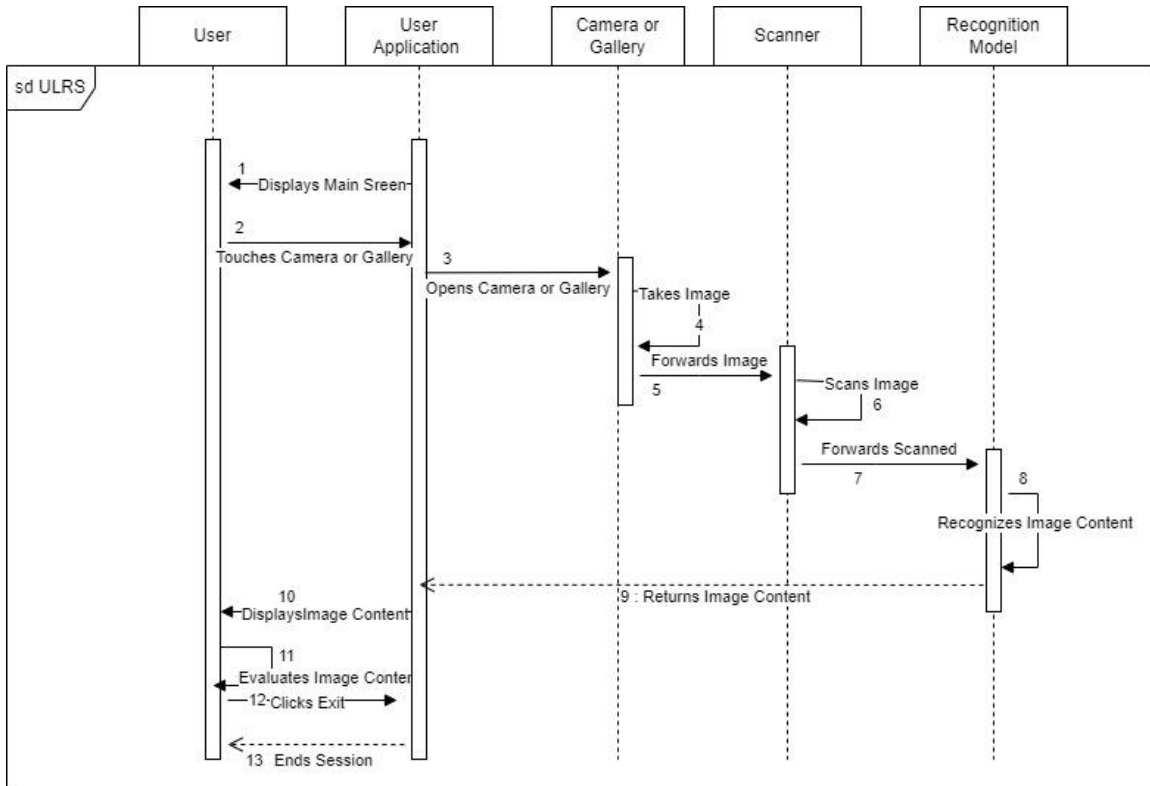


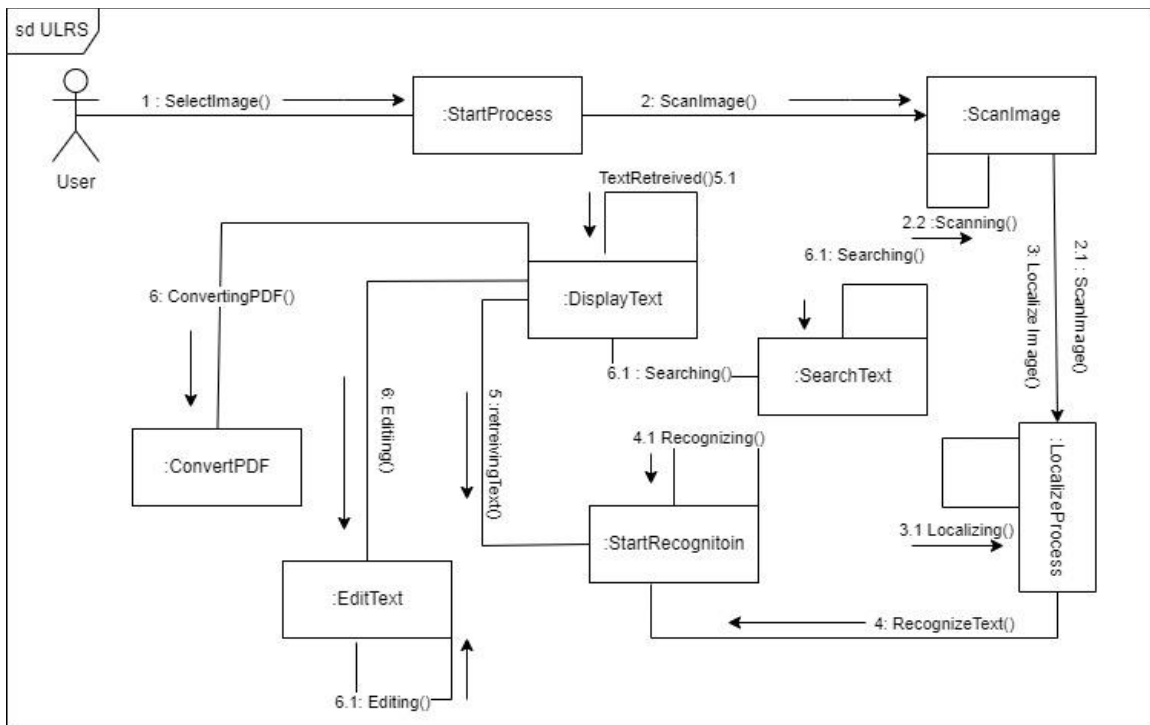
Image Recognition



3.4. Sequence Diagram



3.5. Collaboration Diagram



3.6. Operation Contracts

Name	InputDataset
Responsibilities	User inputs the dataset for model training
Cross Reference	UC_2
Exceptions	None
Preconditions	The dataset is acquired and evaluated
Postconditions	The dataset is successfully uploaded into the memory and given to model for further processing

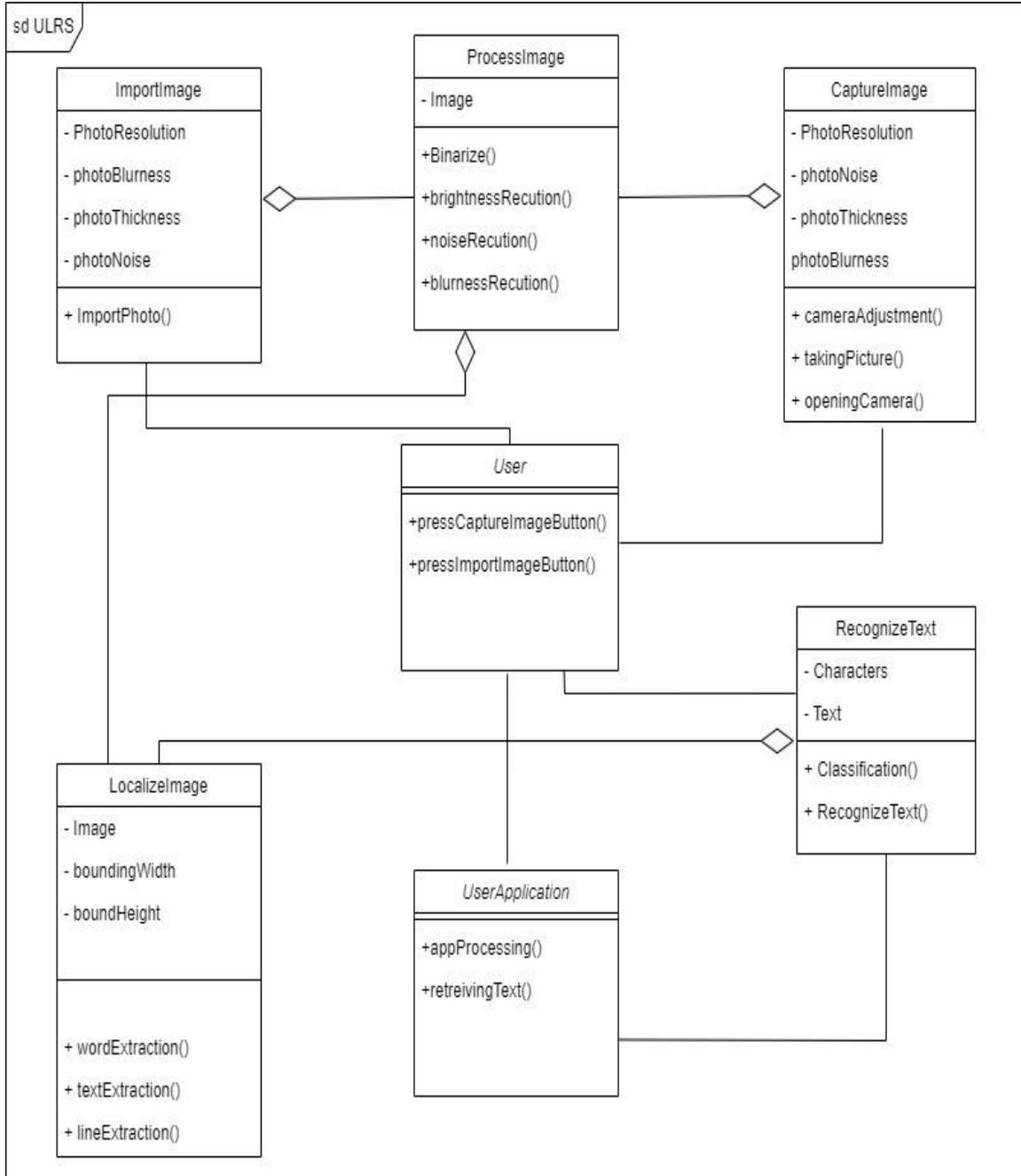
Name	TrainSystem
Responsibilities	TrainSystem trains the recognition model
Cross Reference	UC_1, UC_2
Exceptions	None
Preconditions	User provides the required dataset for training
Postconditions	Model is trained and ready to be used for recognition

Name	InputImage
Responsibilities	User inputs image for extracting text
Cross Reference	UC_1
Exceptions	None
Preconditions	The text in image is in printed form not in handwritten form and is in Urdu language
Postconditions	Image is fed to model for the extraction of text.

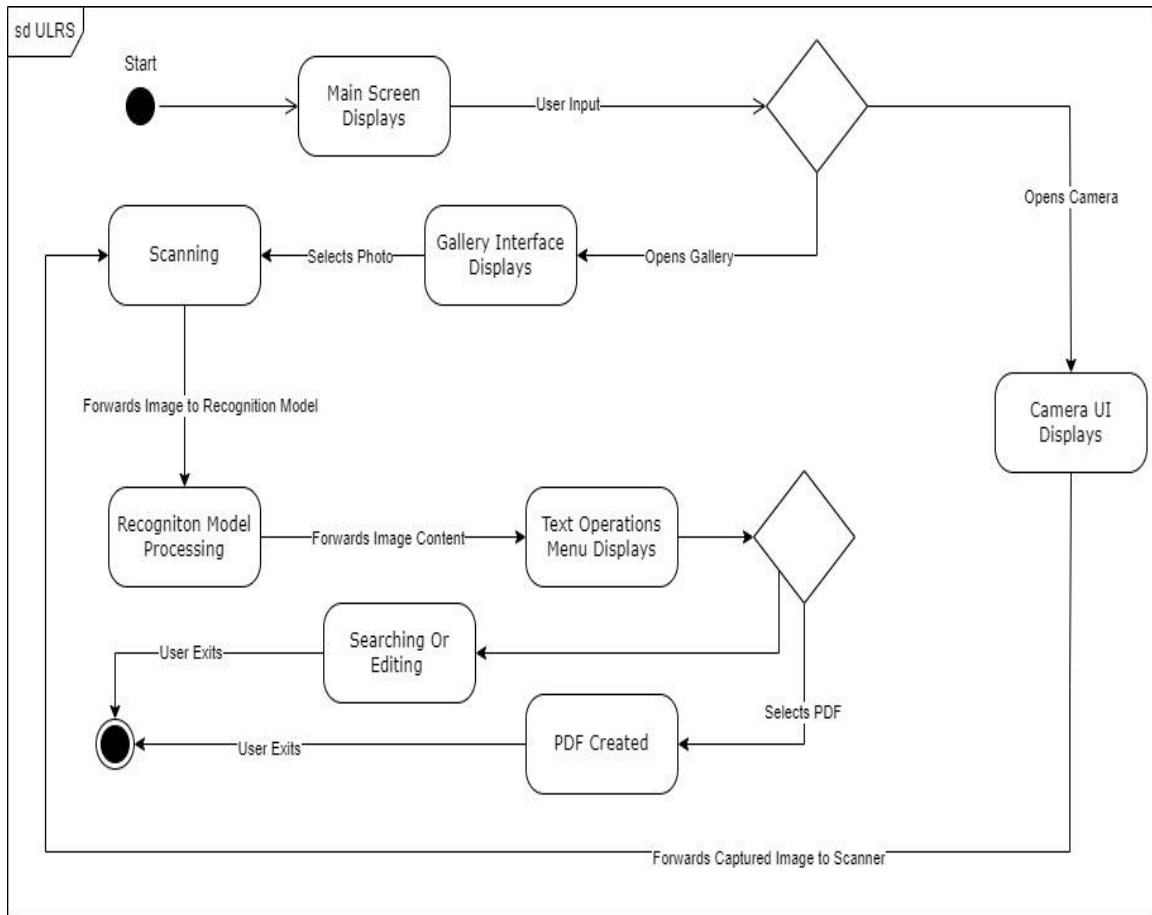
Name	SearchText
Responsibilities	User chooses to search for the text for relevant data

Cross Reference	UC_8
Exceptions	None
Preconditions	The text has been extracted from the model
Postconditions	The application redirects the user to the browser

3.7. Design Class Diagram



3.8. State chart diagram



Chapter 4: User Interface Design

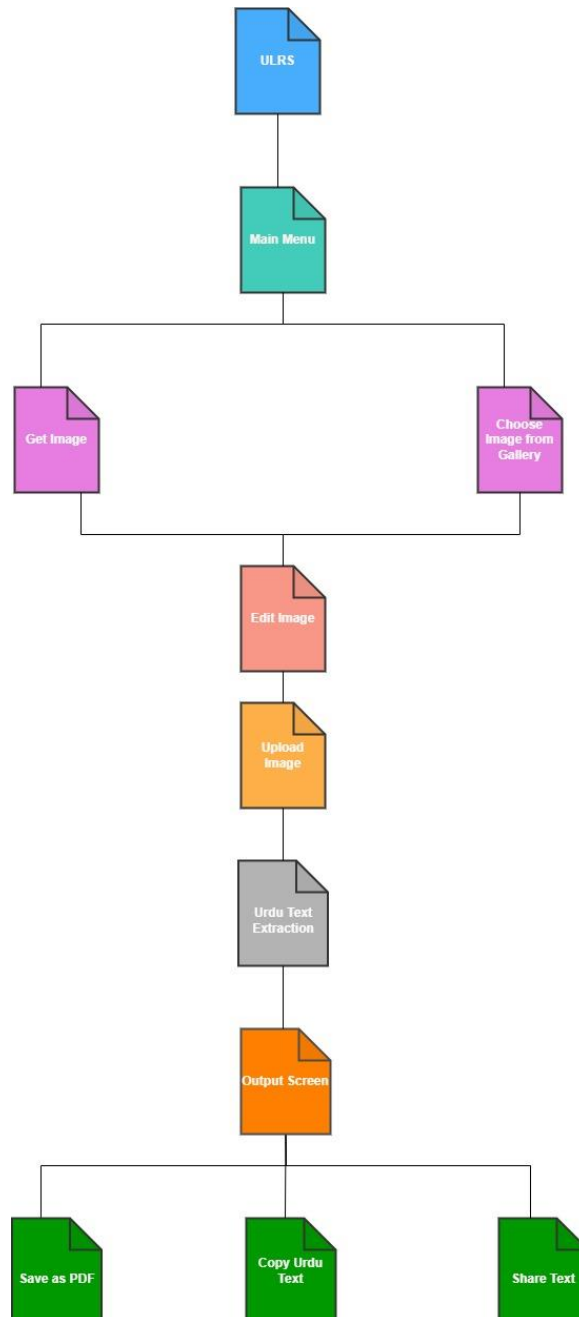
4.1. Introduction

Urdu Language Recognition System page elements are visualized on paper before building them in the computer. The purpose of this chapter is to help user for better understanding of smart analyzer interface.

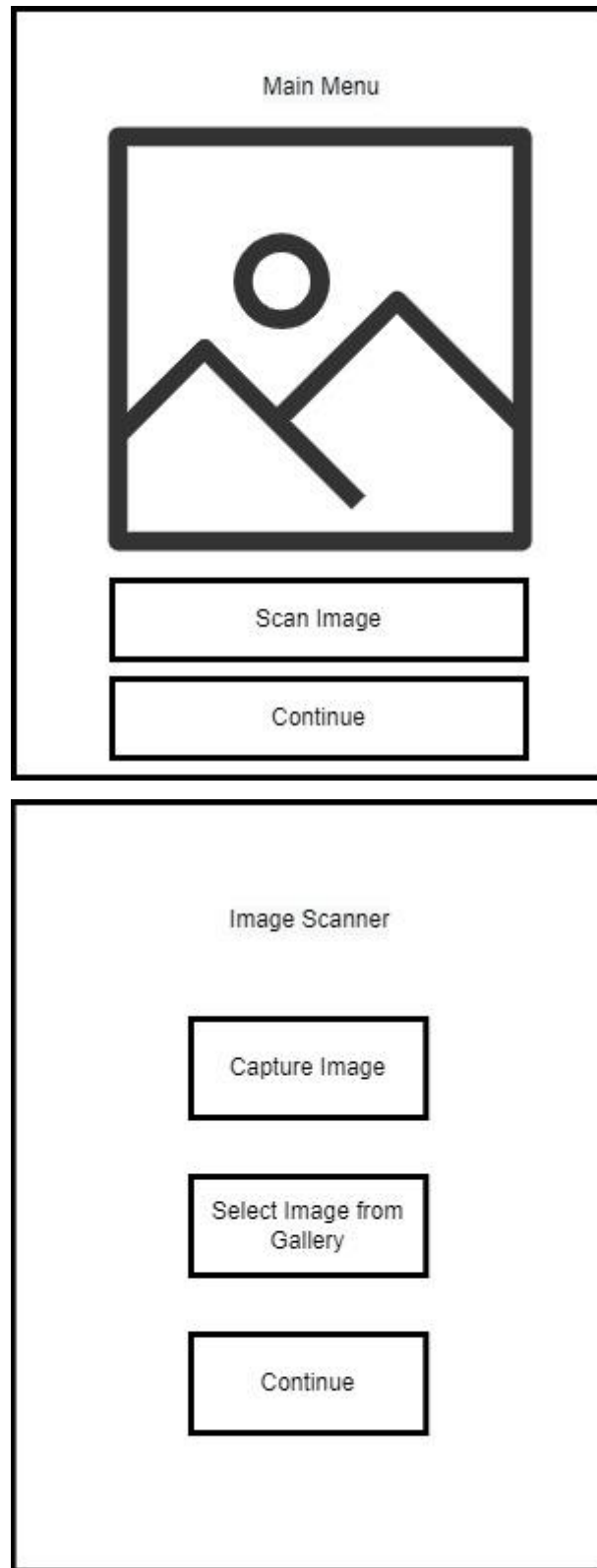
Following artifacts has been included in the 3rd deliverable.

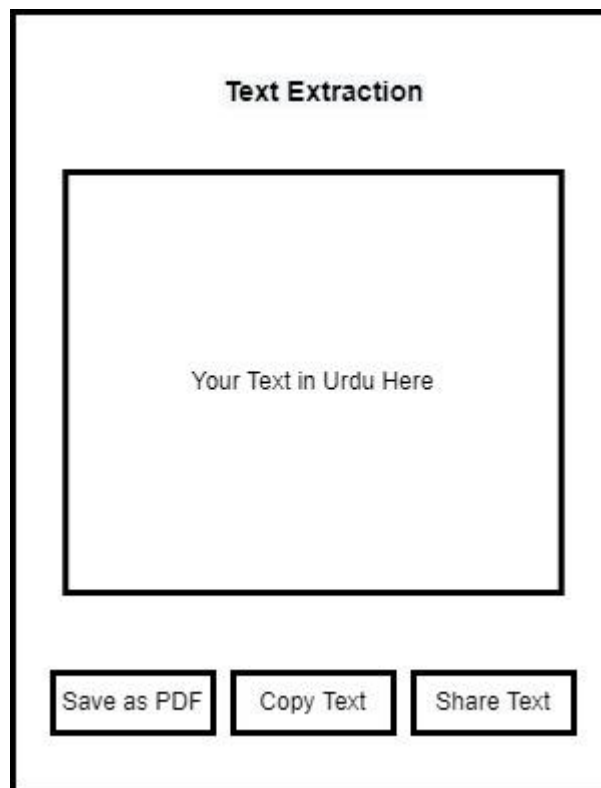
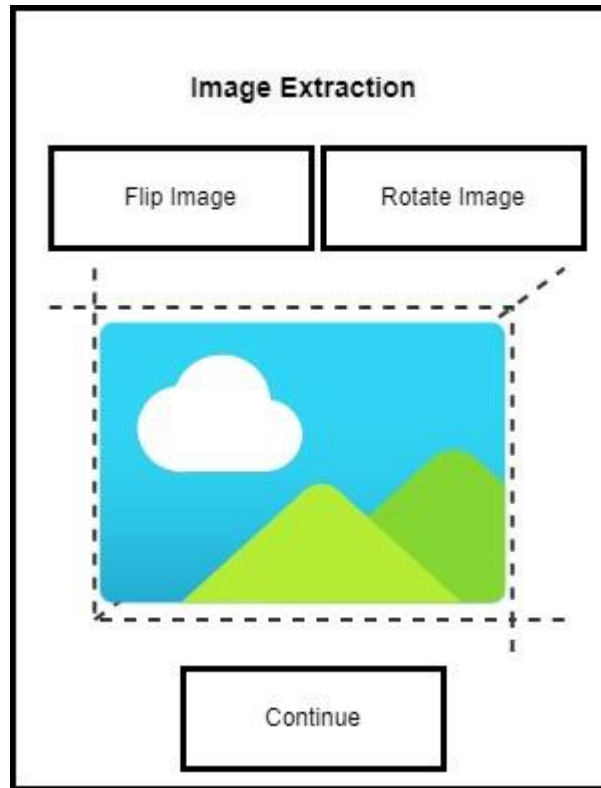
1. Site maps
2. Storyboards
3. Navigational maps
4. Traceability Matrix

4.2. Site Maps



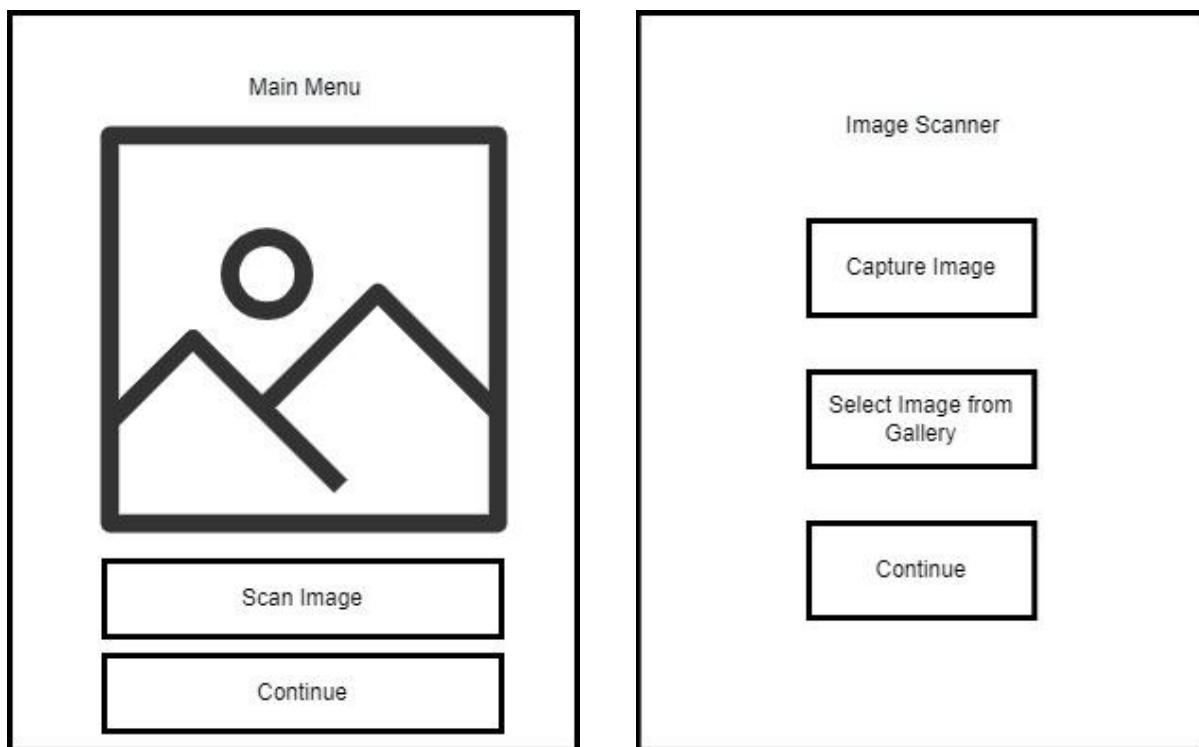
4.3. Story boards



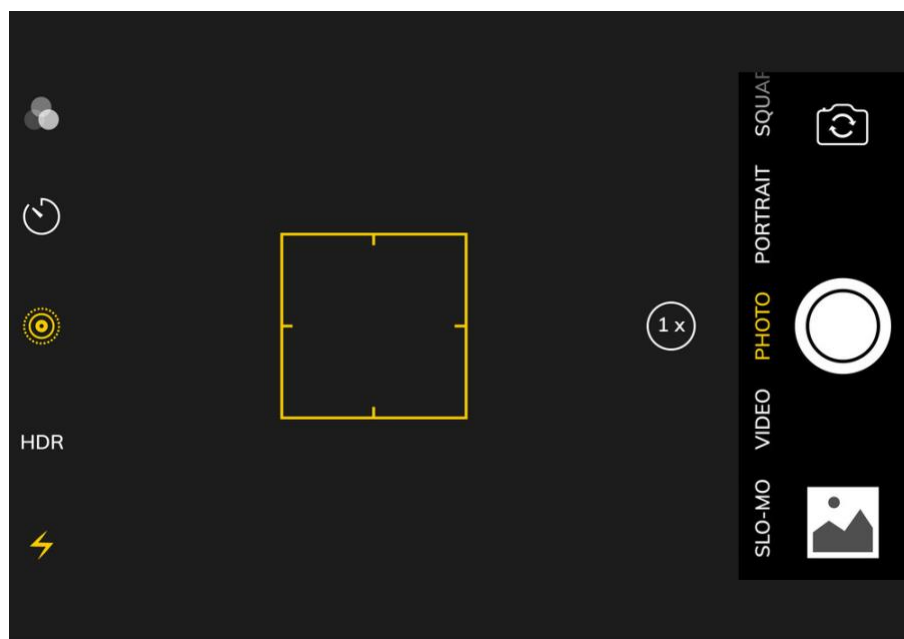


4.4. Navigational maps:

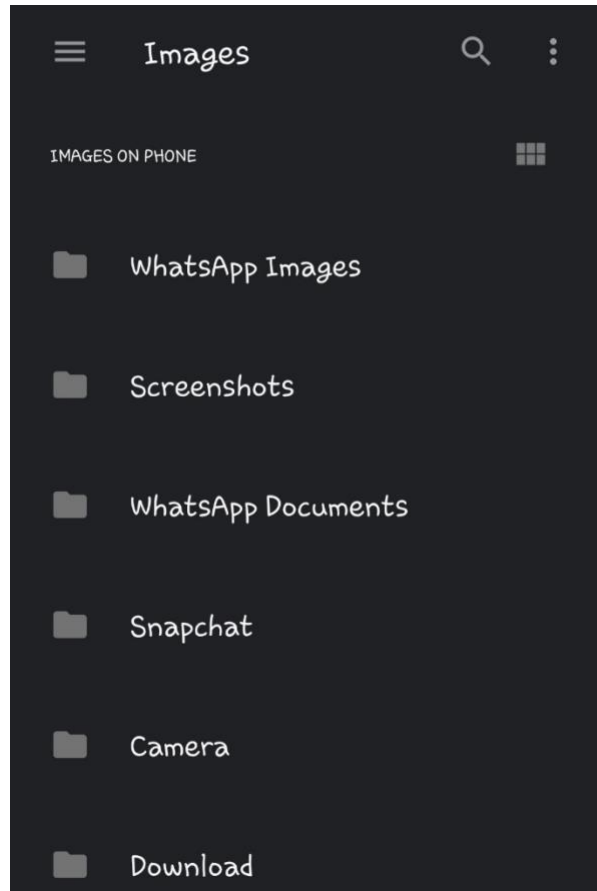
4.4.1. Scan Image button Result



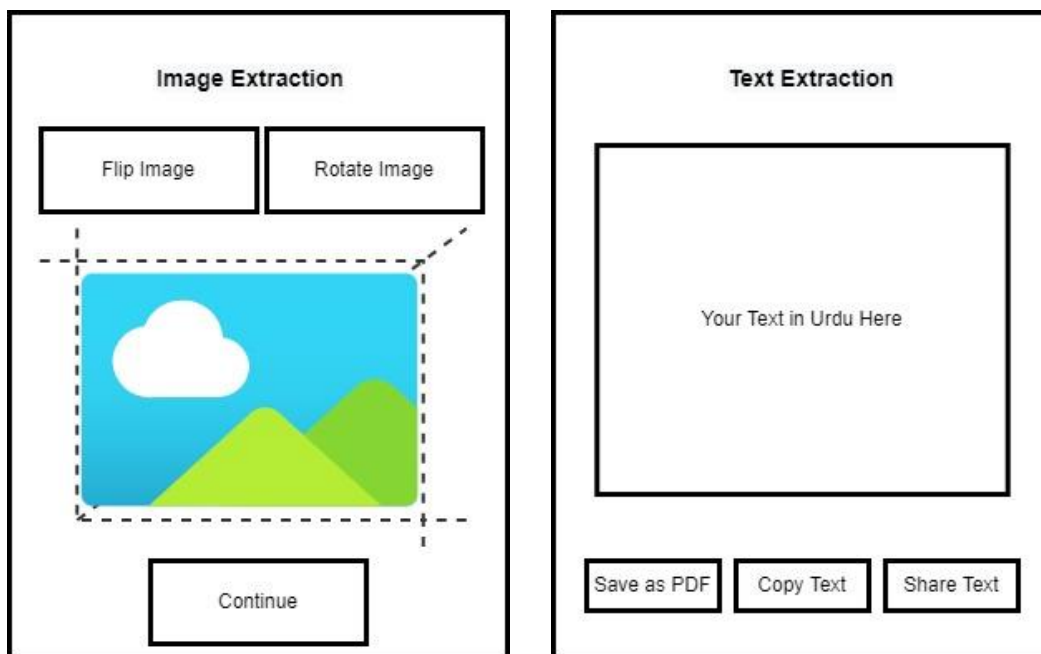
4.4.2 Camera Button Result:



4.4.3 Select Image from Gallery Button Result:



4.4.4 Continue Button Result:



4.5 Trace-ability Matrix

ID	Requirement Description	Business Needs, Justification	Use Case ID	Category	Priority	Specification	Design
1	Recognize Image	Recognizing image is a step towards further navigation	UC_1	Recognition Model	High	Finished	Finished
2	Train System	System must be trained for better accuracy	UC_2	Recognition Model	High	Finished	Finished
3	Edit Image	Image should be editable according to the need of client for better User Experience	UC_3	Application Content	Medium	Finished	Finished
4	Capture Image	By allowing user to capture an image will enhance the functionality	UC_4	Application Content	Medium	Finished	Finished
5	Pick Image	User must be able to pick existing images from gallery	UC_5	Application Content	High	Finished	Finished
6	Select Text	Required to enhance User Experience	UC_6	Application Content	Low	Finished	Finished
7	Copy Text	Further enhances User Experience	UC_7	Application Content	Low	Finished	Finished

Chapter 5: Software Testing

5.1 Introduction:

Since the beginning of software engineering, different software testing schemes and testing templates have been invented. These testing templates are used to test both the UI and individual units of the system. Every credible software house developed their testing documentation for the purpose of developing efficient systems.

In our testing module, we are going to test different features and modules of the system based on the IEEE standard of software. The IEEE developed the 829 Standard for Software Test Documentation for any type of software testing, including User Acceptance Testing. Applicability is not restricted by the size, complexity, or criticality of the software. However, the standard does not specify any class of software to which it must be applied. The standard addresses the documentation of both initial development testing and the testing of subsequent software releases.

Following are standard artifacts, which must be included in this deliverable:

1. Test Plan
2. Test Design Specification
3. Test Case Specification
4. Test Procedure Specification
5. Test Item Transmittal Report
6. Test Log
7. Test Incident Report
8. Test Summary Report

5.2. Test plan

5.2.1. Purpose

To prescribe the scope, approach, resources, and schedule of the testing activities. To identify the items being tested, the features to be tested, the testing tasks to be performed, the personnel responsible for each task, and the risks associated with this plan.

5.2.2. Outline

A test plan shall have the following structure:

- a. Test plan identifier
- b. Introduction
- c. Test items
- d. Features to be tested
- e. Features not to be tested
- f. Approach
- g. Item pass/fail criteria
- h. Suspension criteria and resumption requirements
- i. Test deliverables

- j. Testing tasks
- k. Environmental needs
- l. Responsibilities
- m. Staffing and training needs
- n. Schedule
- o. Risks and contingencies

5.2.2.1. Test plan identifier

The identifier for test plan is ULRS-1 The abbreviation for the identifier ULRS-1 is as: Urdu Language Recognition System version 1

5.2.2.2. Introduction

The test plan is for testing the whole system including modules and their features. The testing of application features will be conducted manually within 18-21 days. The testing will also be conducted in the environment populated with test data.

Test planning phases defines and elaborate following points:

1. What tester has to perform in testing
2. Standards of the quality to use in testing
3. Resources to be employed for testing
4. Schedule and time scale for the testing phase
5. Describes all the risks and contingencies which involved in testing and how they will be overcome.

5.2.2.3. Test items

Following are the testing items which we are going to test in this plan:

1. Urdu Recognition Model
2. API for the Recognition Model
3. Deployed website for API
4. Internal connection of API with android app
5. Image Scanner
6. Image Cropper
7. Output Result Display
8. Saved PDF file
9. Copied text
10. Sharing result

All these are the generalized items which will in fact contain different tests to remove anomalies and defects from the Project.

5.2.2.4. Features to be tested

Following is the list of all the features which will be tested:

1. Scan Image Button working as desired
2. Application asking for permission to access camera and storage as desired

3. Camera Button allowing users to take picture within app as desired
4. Select from gallery Button allowing users to select picture within app as desired
5. Image Scanned successfully
6. Image can be edited
7. Image cropper working properly
8. Connection successful with server
9. Image sent to server successfully
10. Image uploaded to the bucket successfully by the server
11. Text recognized by server successfully
12. Text data retrieved from server
13. Data scanned successfully
14. Text displayed as expected
15. Save Button working as desired
16. Copy Button working as desired
17. Share Button working as desired

5.2.2.5. Features not to be tested

As all the testing is going to be perform on every feature, so it will make sure that nothing left without testing.

5.2.2.6. Approach

A test approach is the test strategy implementation of a project. It defines how testing would be carried out. Test will be conducted as per documented test cases. This project will involve mostly independent testing for each member. Each member will perform testing manually. Each member will test each feature and mark each case as Pass/Fail. Each tester will note the actual result and all relevant details. Once all tests will complete, the test manager will review the test report to the team accordingly. Following are the testing approaches that will be followed:

1. Unit testing
2. Instrumental and Integrational testing

Unit testing covers almost 70 percent of test program. So, we will be using unit test extensively to test each and every component of our project possibly. Moreover, instrumentation tests are used to test the UI of mobile application. It covers 30 percent of total test coverage.

5.2.2.7. Item pass/fail criteria

There will be different criteria for the Unit testing and Instrumentational testing for the passing and failure of testing. Moreover, we will be using test driven approach in our development phase of the application. It means that we are going to first write test and execute it. After the failure of test, we will write the code required to pass the last test. After that, we will refactor and test again until we pass the entire test.

Pass/Fail Criteria for Unit Testing

1. A unit test will be considered as completed if it evaluates that the functionality which is tested is also there in the project.
2. A unit test will be considered as fail if the time out happens during testing.

Pass/Fail Criteria for Instrumentation Testing

1. In UI testing everything will be tested by manual performing operations on the UI of application and a test will be consider pass if it will perform the desired operation on UI and the result in the UI is displayed as expected.
2. Now the entire test which will not be able to follow the above criteria will be deemed as flaky test and also considered as fail test.

5.2.2.8. Suspension criteria and resumption requirements

The criteria for the suspension and resumption of testing shall be as follow

1. If the number or types of defects reach a point where the follow-on testing has no value, it will make no sense to continue the testing. In this case, I will be sending tests for further reviews and development.
2. After completion of development and reviews and refactoring, testing shall resume.

5.2.2.9. Test deliverables

The following deliverables will be provided after the testing efforts ends

- a. Test plan;
- b. Test design specifications;
- c. Test case specifications;
- d. Test procedure specifications;
- e. Test item transmittal reports;
- f. Test logs;
- g. Test incident reports;
- h. Test summary reports.

5.2.2.10. Testing tasks

The following activities must be completed:

- a. Functional specifications written and delivered to the testing team.
- b. Environment should be ready for testing (test data, test logins etc.)
- c. Design and plan test in order to write and execute them.

5.2.2.11. Environmental needs

The test site must be populated with test data including a variety of images with Urdu text.

Hardware Specifications:

- a. Android environment with internet connectivity
- b. Working camera

Operating System:

- a. Android version 7 or later

Software Specification:

- a. Android Studio Emulator / mobile phone

5.2.2.12. Responsibilities

The Test Manager is responsible for facilitating the testing project, coordinating availability and schedule of testers and training them as needed. Each tester should understand the expectations on completion date and level of quality. The Test Manager should also communicate any risks to the team. The main person which is responsible for the testing execution and management is called test manager. In our documentation and project, the testing is performed by **Ms. Afshan** and **Ms. Eza Tehreem Asjad**.

Ms. Afshan is responsible for Test Planning, Test case development.

Ms. Eza Tehreem Asjad is Responsible for Test Specifications, Test writing.

Test Execution is performed by all of us

5.2.2.13 Staffing and training needs

- a. To apply systematic testing methods, test team require a complete understanding of software testing.
- b. To apply test in android development and to understand them a person requires complete knowledge of testing components provided by android testing teams.
- c. Training for the team members can be organized to get understanding of testing framework.

5.2.2.14. Schedule

Testing will take place 4 weeks prior to the launch date. The first round of testing should be completed in 3 weeks.

5.2.2.15. Risks and contingencies

If the first round of testing is not completed within 2 weeks, it could delay bug fixes and final testing. If this happens, UAT would be pushed back and eventual affect the launch date. If the testers don't have a basic understanding the android platform, testing could be delayed or not conducted properly.

5.3. Test design specification

5.3.1. Purpose

Test design specification deals with the designing and predicting about the features which we are going to test. In this phase we also prioritize the test cases in their order of importance. This is quite important document and is missed by some organization. This is the biggest problem that people try to execute and develop test without first designing them.

5.3.2. Outline

Test plan shall have the following structure:

- a. Test plan identifier;
- b. Introduction;
- c. Test items;
- d. Features to be tested;
- e. Approach Refinement;
- f. Test Identification
- g. Features pass/fail criteria;

5.3.2.1 Test plan identifier

The identifier for test plan is **ULRS-1** The abbreviation for the identifier ULRS-1 is as: Urdu Language Recognition System version 1

5.3.2.2. Introduction

The test plan is for testing the whole package including their features. This document basically describes the plan for testing the design of application.

- a. Identify our project information and its components that should be tested.
- b. List the recommended test requirements (high level).
- c. Recommend and describe the testing strategies.
- d. Identify the required resources, provide an estimate of the test efforts, and detail testing schedule.
- e. List the deliverable elements of the test activities.

5.3.2.3. Test items

Following are the testing items which we are going to test in this plan:

1. Urdu Recognition Model
2. API for the Recognition Model
3. Deployed website for API
4. Internal connection of API with android app
5. Image Scanner
6. Image Cropper
7. Output Result Display
8. Saved PDF file
9. Copied text
10. Sharing result

All these are the generalized items which will in fact contain different tests to remove anomalies and defects from the Project.

5.3.2.4. Features to be tested

Following is the list of all the features which will be tested:

1. Scan Image Button working as desired
2. Application asking for permission to access camera and storage as desired
3. Camera Button allowing users to take picture within app as desired
4. Select from gallery Button allowing users to select picture within app as desired
5. Image Scanned successfully
6. Image can be edited
7. Image cropper working properly
8. Connection successful with server
9. Image sent to server successfully
10. Image uploaded to the bucket successfully by the server
11. Text recognized by server successfully
12. Text data retrieved from server
13. Data scanned successfully
14. Text displayed as expected
15. Save Button working as desired
16. Copy Button working as desired
17. Share Button working as desired

5.3.2.5. Approach Refinement

We will be using the same strategy and approach as mentioned in the test plan phase.

5.3.2.6. Testing Identification

In this phase we make a list of testing features which can be tested and can be taken as a test case in the future steps of testing. In our testing strategies we are taking all the test items as test cases.

5.3.2.7. Feature pass/fail criteria

There will be different criteria for the Unit testing and Instrumentational testing for the passing and failure of testing. Moreover, we will be using test driven approach in our development phase of the application. It means that we are going to first write test and execute it. After the failure of test, we will write the code required to pass the last test. After that, we will refactor and test again until we pass the entire test.

Pass/Fail Criteria for Unit Testing

1. A unit test will be considered as completed if it evaluates that the functionality which is tested is also there in the project.
2. A unit test will be considered as fail if the time out happens during testing.

Pass/Fail Criteria for Instrumentation Testing

1. In UI testing everything will be tested by manual performing operations on the UI of application and a test will be consider pass if it will perform the desired operation on UI and the result in the UI is displayed as expected.
2. Now the entire test which will not be able to follow the above criteria will be deemed as flaky test and also considered as fail test.

5.4. Test Case Specification

5.4.1. Purpose

One of the deliverables offered to the client, test case specification is a document that delivers a detailed summary of what scenarios will be tested in a software during the software testing life cycle (STLC). This document specifies the main objective of a specific test and identifies the required inputs as well as expected results/outputs. Moreover, it acts as a guide for executing the procedure of testing and outline the pass & fail criteria for determining acceptance.

5.4.2. Outline

Following are the contents of test case specification:

- a. Test case specification identifier
- b. Test items
- c. Input specifications
- d. Output specifications
- e. Environmental needs
- f. Special procedural requirements
- g. Inter case dependencies

5.4.2.1. Test case specification identifier

Following are the test cases identifiers that will be used in our test:

Serial No.	Test Feature	Test Case Identifier
1.	Scan Image Button working as desired	TC1
2.	Application asking for permission to access camera and storage as desired	TC2
3.	Camera Button allowing users to take picture within app as desired	TC3
4.	Select from gallery Button allowing users to select picture within app as desired	TC4
5.	Image Scanned successfully	TC5
6.	Image can be edited	TC6
7.	Image cropper working properly	TC7
8.	Connection successful with server	TC8
9.	Image sent to server successfully	TC9
10.	Image uploaded to the bucket successfully by the server	TC10
11.	Text recognized by server successfully	TC11
12.	Text data retrieved from server	TC12
13.	Data scanned successfully	TC13
14.	Output Text displayed as expected	TC14

15.	Save Button working as desired	TC15
16.	Copy Button working as desired	TC16
17.	Share Button working as desired	TC17

5.4.2.2 Test items

Following is the list of all the features which will be tested:

1. Scan Image Button working as desired
2. Application asking for permission to access camera and storage as desired
3. Camera Button allowing users to take picture within app as desired
4. Select from gallery Button allowing users to select picture within app as desired
5. Image Scanned successfully
6. Image can be edited
7. Image cropper working properly
8. Connection successful with server
9. Image sent to server successfully
10. Image uploaded to the bucket successfully by the server
11. Text recognized by server successfully
12. Text data retrieved from server
13. Data scanned successfully
14. Text displayed as expected
15. Save Button working as desired
16. Copy Button working as desired
17. Share Button working as desired

5.4.2.3. Input specifications

Here we will specify the input for every test case:

Serial No.	Test Case Input	Test Case Identifier
1.	Input Image with Urdu Text from gallery	TC1
2.	Input from User to grant permission to resources	TC2
3.	Camera input	TC3
4.	Connection String	TC4

5.4.2.4. Output specifications

Here we will specify the output for every case:

Serial No.	Test Case Input	Test Case Identifier
1.	Scanned Image	TC1
2.	Edited Image	TC2
3.	Retrieved Text	TC3
4.	Specify whether request to server failed	TC4
5.	Copied Text	TC5
6.	PDF file	TC6

5.4.2.5. Environmental needs

5.4.2.5.1. Hardware

Following test cases need hardware for their successful execution:

Serial No.	Hardware Required	Test Case Identifier
1.	Mobile	TC1
2.	Mobile	TC2
3.	Mobile	TC3
4.	Mobile	TC4
5.	Mobile	TC5
15.	Mobile, Emulator, Storage Space	TC15

5.4.2.5.2. Software

Specify the system and application software required to execute this test case. This may include system software such as operating systems, compilers, simulators, and test tools. In addition, the test item may interact with application software.

Following test cases need hardware for their successful execution:

Serial No.	Software Required	Test Case Identifier
1.	Emulator, Android 7 or above, android studio	TC1
2.	Emulator, Android 7 or above, android studio	TC2
3.	Emulator, Android 7 or above, android studio	TC3
4.	Emulator, Android 7 or above, android studio	TC4
5.	Emulator, Android 7 or above, android studio	TC5
6.	Emulator, Android 7 or above, android studio	TC6
7.	Emulator, Android 7 or above, android studio	TC7
8.	Emulator, Android 7 or above, android studio	TC8
9.	Emulator, Android 7 or above, android studio	TC9
10.	Configured Server, storage container on cloud	TC10
11.	Configured server and website/flask/postman	TC11
12.	Configured server and website/flask/postman	TC12
13.	Emulator, Android 7 or above, android studio	TC13
14.	Emulator, Android 7 or above, android studio	TC14
15.	Emulator, Android 7 or above, android studio	TC15
16.	Emulator, Android 7 or above, android studio	TC16
17.	Emulator, Android 7 or above, android studio	TC17

5.4.2.6. Special procedural requirements

There are no other special procedural requirements for the execution of these tests cases.

5.4.2.7. Inter case dependencies

Serial No.	Test Feature	Test Case Identifier	Test Case Dependency
1.	Camera Button allowing users to take picture within app as desired	TC3	TC2, TC1
2.	Select from gallery Button allowing users to select picture within app as desired	TC4	TC2, TC1
3.	Image Scanned successfully	TC5	TC1, TC2, TC3, TC4
4.	Image can be edited	TC6	TC1, TC2, TC3, TC4, TC5, TC7
5.	Image cropper working properly	TC7	TC1, TC2, TC3, TC4, TC5
6.	Image sent to server successfully	TC9	TC8
7.	Image uploaded to the bucket successfully by the server	TC10	TC8, TC9
8.	Text recognized by server successfully	TC11	TC8, TC9, TC10
9.	Text data retrieved from server	TC12	TC8, TC9, TC10, TC11
10.	Data scanned successfully	TC13	TC12
11.	Output Text displayed as expected	TC14	TC13
12.	Save Button working as desired	TC15	TC14, TC2
13.	Copy Button working as desired	TC16	TC14
14.	Share Button working as desired	TC17	TC14

5.5. Test procedure specification

5.5.1. Purpose

To specify the steps for executing a set of test cases or, more generally, the steps used to analyze a software item in order to evaluate a set of features.

5.5.2 Outline

A test procedure specification shall have the following structure:

- a. Test procedure specification identifier
- b. Purpose
- c. Special requirements
- d. Procedure steps

5.5.2.1. Test procedure specification identifier

The identifier for the test procedure specification is ULRS TPSI

5.5.2.2. Special requirements

To proceed with the testing of API one should have knowledge of following

1. Unit testing
2. Flask Framework
3. Postman
4. Azure Cloud Platform
5. Azure Storage Blob
6. Android Studio

5.5.2.3. Procedure steps

5.5.2.3.1. Log

Logging in the android studio can be performed using following techniques:

1. Manual Logging
2. Debug Logging

We will be using both kind of logging depending upon the scenario.

5.5.2.3.2. Set up

1. Install Android Studio
2. Install all dependencies
3. Make sure the deployed API is working
4. Make sure the storage container is there and correctly configured.
5. Enable Mobile's debugging mode
5. Install App on emulator and mobile

5.5.2.3.3. Start

We need to plan and design the test before executing them. Once we write all the test we will Run.

5.5.2.3.4. Proceed

We have to execute test cases in the order of their dependency. Test case with no dependencies or solved dependency first.

5.5.2.3.5. Measure

The measurement whether a test is correct or not is solely based on the human observation. Logs will be displayed in android studio which will display that whether a test is successful or not

5.5.2.3.6. Shut down

To shut down the testing we have to stop testing forcefully using android studio.

5.5.2.3.7. Restart

To restart the test simply write start the app again in android studio

5.5.2.3.8. Stop

To stop the test orderly, first we will have to wait for the completion of execution. After that, we will stop the execution from testing within the android studio software.

5.5.2.3.9. Wrap up

Now to test the code again, we will reinstall the app in android studio.

5.5.2.3.10. Contingencies

Poor speed of internet may cause application to not work properly and the required results cannot be obtained or altered.

5.6. Test item transmittal report

5.6.1. Purpose

To identify the test items being transmitted for testing. It includes the person responsible for each item, its physical location, and its status. Any variations from the current item requirements and designs are noted in this report.

5.6.2. Outline

A test item transmittal report shall have the following structure:

- a. Transmittal report identifier
- b. Transmitted items
- c. Location
- d. Status

5.6.2.1. Transmittal report identifier

The identifier for the transmittal report is ULRS TRL.

5.6.2.2. Transmitted items

All the items which are mentioned in the Test Plan are part of the test transmittal report. We should also provide all those tests which are failed and again revised and developed for refactoring for further testing. As we are a small two of three members working in this project, so, **Eza Tehreem Asjad** and **Afshan Wain** are responsible for test planning and specification development as indicated in Testing Responsibilities. Whereas these test case specifications then will be transmitted to for successful running of the test.

5.6.2.3. Location

All the test documents including most important test case specification will be placed in a PDF file and will be transmitted to test manager and test executor by email. After receiving email tester will be able to convert test case specifications into actual written test cases.

5.6.2.4. Status

At this phase all the test cases which are provided are not tested prior so their status in unresolved or so we can say false. After the testing will be done then if there will be any problem in code, a request will be made to the developers to again correct and refactor the code.

5.7. Test log

5.7.1. Purpose

To provide a chronological record of relevant details about the execution of tests.

5.7.2. Outline

Following will be the content of test log.

- a. Test log identifier;
- b. Description;
- c. Activity and event entries.

5.7.2.1. Test log identifier

The identifier for test log is ULRS TLI

5.7.2.2. Description

Serial No.	Test Feature	Test Case Identifier
1.	Scan Image Button working as desired	TC1
2.	Application asking for permission to access camera and storage as desired	TC2
3.	Camera Button allowing users to take picture within app as desired	TC3
4.	Select from gallery Button allowing users to select picture within app as desired	TC4
5.	Image Scanned successfully	TC5
6.	Image can be edited	TC6
7.	Image cropper working properly	TC7
8.	Connection successful with server	TC8
9.	Image sent to server successfully	TC9
10.	Image uploaded to the bucket successfully by the server	TC10
11.	Text recognized by server successfully	TC11
12.	Text data retrieved from server	TC12
13.	Data scanned successfully	TC13
14.	Output Text displayed as expected	TC14
15.	Save Button working as desired	TC15
16.	Copy Button working as desired	TC16
17.	Share Button working as desired	TC17

Software Requirements

Software requirements which are included are as:

1. Android studio
2. Android 7 or above
3. Utility libraries

Hardware Requirements

Hardware requirements which are included are as:

1. Android 7 or above
2. Storage Space for storing pdf files

5.7.2.3. Activity and event entries

For each event, including the beginning and end of activities, record the occurrence date and time along with the identity of the author. The information in 9.2.3.1 through 9.2.3.5 should be considered.

5.7.2.3.1. Execution description

While executing the tests whole team was present so that all of the team get through understanding of testing process.

5.7.2.3.2. Procedure results

Following is the complete result of our test execution based on the pass/fail criteria already mentioned in planning phase:

Serial No.	Test Case Identifier	Test Case Revision	Test Case Result	Test Status
1.	TC1	1 st	Specify whether it lets user choose from camera / gallery or not	Pass
2.	TC2	1 st	Specify if application asks for permission or not	Pass
3.	TC3	1 st	Specify if camera app opens or displays no permission message	Pass
4.	TC4	1 st	Specify if gallery opens or not	Pass
5.	TC5	1 st	Specify if image is displayed on screen or not	Pass
6.	TC6	1 st	Specify if the image is displayed and edit operations can be applied or not	Pass
7.	TC7	1 st	Specify if crop guidelines are there and allows users to crop image or not	Pass
10.	TC10	1 st	Specify if image is present in storage container or not	Pass
11.	TC11	1 st	Specify if the image text is recognized or not	Pass

12.	TC12	1 st	Specify the output is received or not	Pass
13.	TC13	1 st	Specify if text is readable or not	Pass
14.	TC14	1 st	Specify is text is properly displayed on screen or not	Pass
15.	TC15	1 st	Specify if saved feedback is given and output pdf is generated or not	Pass
16.	TC16	1 st	Specify if text is copied on clipboard or not	Pass
17.	TC17	1 st	Specify if share screen pops up or not	Pass

5.7.2.3.3. Environmental information

The test site must be populated with test data including a variety of plants.

Hardware Specifications:

- Android environment with internet connectivity
- Working camera

Operating System:

- Android version 7 or Later

Software Specification:

- Mobile or emulator

5.7.2.3.4. Anomalous events

Our tests are successful passed at the moment and no anomaly occurred except in one case which will be later investigated. But once developer start writing more code, problems can occur at any time. So, it is necessary to use test driven approach.

5.8. Test incident report

5.8.1. Purpose

To document any event that occurs during the testing process that requires investigation.

5.8.2. Outline

A test incident report shall have the following structure:

- a. Test incident report identifier
- b. Summary
- c. Incident description
- d. Impact

5.8.2.1. Test incident report identifier

The identifier for incident report is ULRS IRI.

5.8.2.2. Summary

During the execution of test, we have faced an anomaly when one of the tests get failed. TC14 got failed. In this section, we will analyze and investigate the reason behind failure of test case.

5.8.2.3. Incident description

Here is the complete description of the anomalies occurred during the testing phase. Also, a complete solution and refactoring is suggested in the description.

5.8.3.1 Test Case Specification to save file as PDF

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC15	Generated PDF file	Click on “save” button	Asking for permission if not granted	Asked for permission if not granted		Pass
			Permission granted notification appears	Permission granted notification appears		Pass
			PDF file generated msg appears			Failed
			PDF file saved	No PDF		Failed

5.8.3.2 Test Case Specification to Scan Image Button

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC1	Scan Image Button	Click on “Scan image” button	Asking for permission if not granted	Asked for permission if not granted		Pass
			Displays a screen to let user choose from camera or gallery	Displayed a screen to let user choose from camera or gallery		Pass

5.8.3.3 Test Case Specification to take picture

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC3	Camera button	Click on “Camera” button	Opens the camera of mobile app	Opened the camera of mobile app		Pass
			Image can be taken	Image was taken		Pass

5.8.3.4 Test Case Specification to select picture

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC4	Gallery button	Click on “Select from gallery” button	Opens the gallery of mobile	Opened the gallery of mobile		Pass
		Image	Image can be selected	Image was selected		Pass

5.8.3.5 Test Case Specification to Crop Image

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC7	Cropped Image	Select “crop guidelines”	Image can be cropped in the Edit Window	Image can be cropped in the Edit Window		Pass

5.8.3.6 Test Case Specification to recognize text

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC14	Generated output text	Click on “continue” button	Dialogue box to wait appears	Dialogue box to wait appeared		Pass

			Screen with output text appears	Screen with output text appeared		Pass
--	--	--	---------------------------------	----------------------------------	--	------

5.8.3.7 Test Case Specification to copy text

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC16	Copy Button	Click on “copy” button	Copied to clipboard notification appears	Copied to clipboard notification appeared		Pass

5.8.3.8 Test Case Specification to share text

Test Case ID	Test Item	Input	Expected Output	Actual Output	Environmental Needs	Status
TC17	Share Button	Click on “share” button	Screen to share text on various apps appears	Screen to share text on various apps appeared		Pass

5.8.2.4. Impact

The incident will have only impact on the test case specification. We will need to rewrite test case specification for the failed test case after sorting out the issue.

5.9. Test summary report

5.9.1. Purpose

To summarize the results of the designated testing activities and to provide evaluations based on these results.

5.9.2. Outline

A test summary report shall have the following structure:

- Test summary report identifier
- Summary
- Variances
- Comprehensive assessment
- Summary of results
- Evaluation
- Summary of activities

h. Approvals

5.9.2.1. Test summary report identifier

Test summary report identifier is as ULRS SRI

5.9.2.2. Summary

All the items described in the test plan are tested and after attestation we figured out that one of them get failed. Except one test case all others test cases passed. We determine passing and failure of test cases according to the criteria mentioned above in test planning phase. After figuring out that one of the test cases is failed we prepared its incidental report so that it can be again refactor by the developer and send again. After refactoring test cases with defects, we again tested it and it passed the testing criteria.

5.9.2.3. Variances

Every test is executed according to the test plan, test specification and test case specification. We follow the testing approach and strategy stated in the test planning phase. Test were executed according to test procedure specification and they are deemed passed and failed according to testing criteria mentioned.

5.9.2.4. Comprehensiveness assessment

All the features described in the test planning phase are tested and no test case is left out of testing due to any sort of reason. Our test coverage report is more than 80 percent.

5.9.2.5. Summary of results

Summary Assessment	Total Number of Test Cases
Test Cases Planned	21
Test Cases Run	21
Test Cases Reviewed	3
Test Cases Passed	60
Test Cases Failed	3

5.9.2.6. Evaluation

We have covered almost 85 percent code with unit and UI testing. Moreover, we use test driven approach in our code. Out of sixteen test cases only one test case gets failed which was again refactored. So, our team including test planner, test designer and test manager work in great harmony to execute all phases of testing.

5.9.2.7. Summary of activities

Activity	Date
API working properly on local computer	20-07-22
Cloud Account and storage container configured and created	21-07-22
API deployed Successfully	23-07-22
API Working Correctly with app	25-07-22

Scan Image Button working as desired	26-07-22
Application asking for permission to access camera and storage as desired	26-07-22
Camera Button allowing users to take picture within app as desired	26-07-22
Select from gallery Button allowing users to select picture within app as desired	26-07-22
Image Scanned successfully	26-07-22
Image can be edited	29-07-22
Image cropper working properly	29-07-22
Connection successful with server	5-08-22
Image sent to server successfully	05-08-22
Image uploaded to the bucket successfully by the server	06-08-22
Text recognized by server successfully	10-08-22
Text data retrieved from server	10-08-22
Data scanned successfully	10-08-22
Output Text displayed as expected	10-08-22
Save Button working as desired	12-08-22
Copy Button working as desired	13-08-22
Share Button working as desired	15-08-22

5.9.2.8. Approvals

Name	Title	Signature
Dr. Umar Shoaib	Supervisor	

Chapter 6: User Manual

6.1. General Information

Urdu Language Recognition System is developed to help people to extract text from the images they have captured in real time or residing in their gallery, The proposed system will be android-based application that will let us digitize images containing Urdu Language text.

6.1.1 Summary of activities

This project will open a new portal towards Urdu Language Recognition, Urdu language has been neglected so far in the online world and the work done on digitizing Urdu language is negligible.

The Proposed System ULRS is equipped with all the tools and technologies needed in order to accurately guess different fonts of Urdu Language with considerable level of noise in pictures along with that it provides the user with the ability to convert that extracted text into PDF, Search the Urdu text or share it to other users on the internet, This will help increase the versatility of Urdu language in the online world and will further reduce the gap between the research done on Urdu language.

6.2 System Summary

System Summary section provides a general overview of the system. The summary outlines the uses of the system's hardware and software requirements, system's configuration, user access levels and system's behavior in case of any contingencies.

6.1.1 System Configuration

The system will be able to receive image as an input from the end user and after uploading the image to the server. It will process the image using our Recognition Model and Extract the Urdu Language Text from the image as a result. The image text will be digitized and converted into editable text that then can be further Copied, Shared or Edited according to the need of user. It is a system that will help Urdu Language Recognition reach new heights.

6.2.2 User Access Levels

The user can access the system once it has been downloaded and installed into the android operating system. After this user will be able to use the app to scan the image and convert the scanned text into digital text, providing further options of making a PDF, Searching the text, Sharing the text or copying the text.

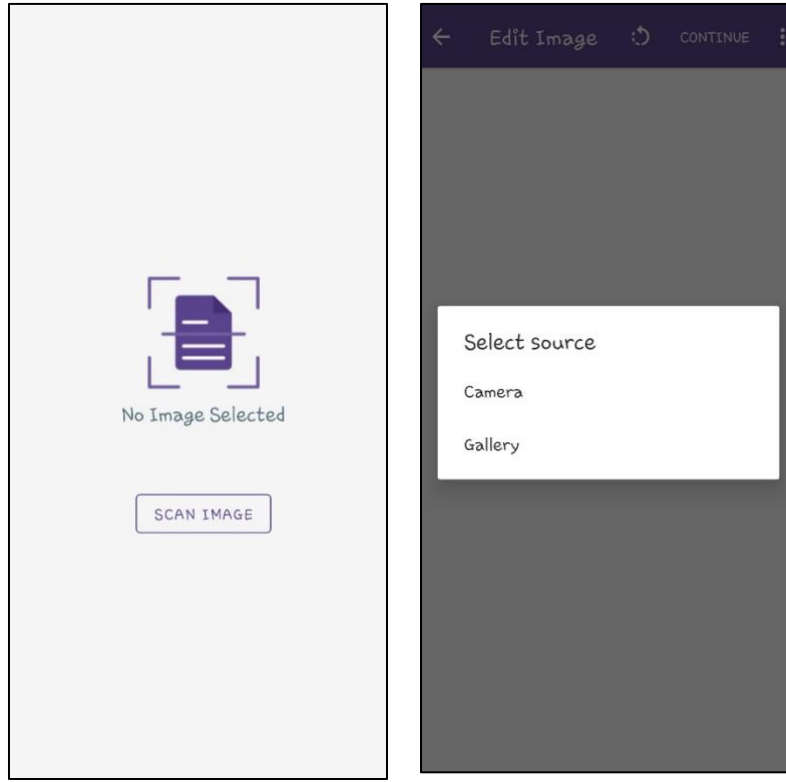
6.2.3 Contingencies

In case of unresponsiveness or mobile shutting down the image won't be able to process and complete text extraction process. In case of no Internet Connection available the mobile application will not be able to process the image.

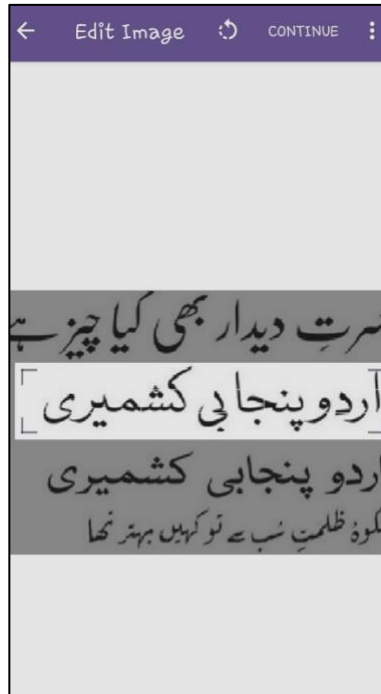
6.3. Getting Started

Getting Started section explains how to use the website. The section presents briefly how to get Urdu Text from the given Image:

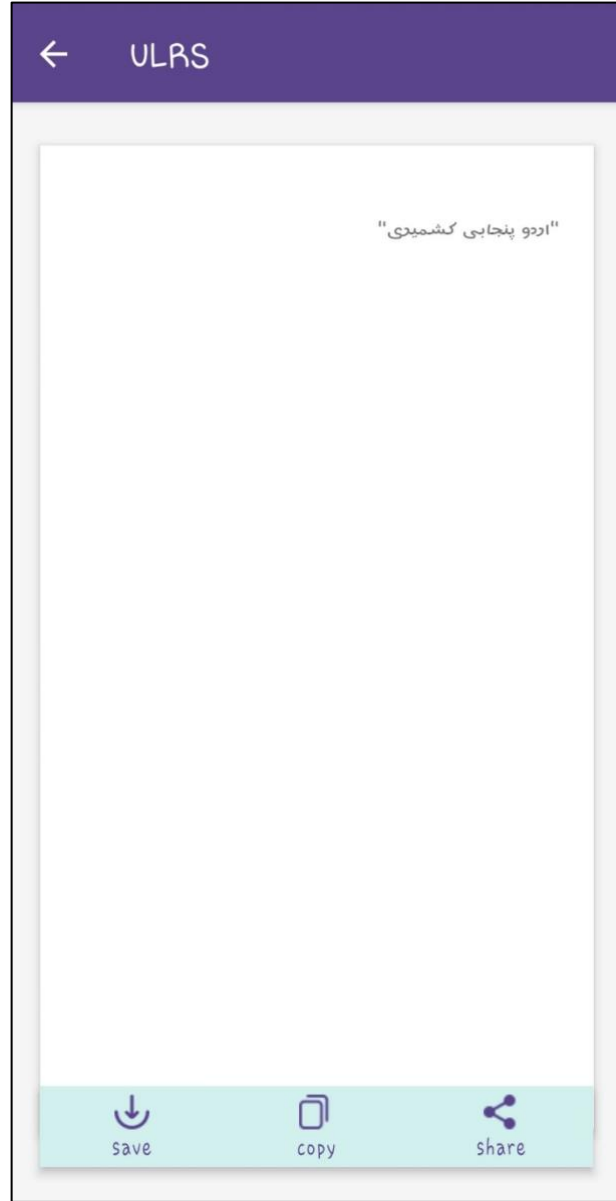
6.3.1 Main Screen and Image Scanner



6.3.2 Edit and Crop Image



6.3.3 Output Text Screen



Appendix 1

Appendix 1

What is Optical Character Recognition?

It is a technology that recognizes text within a digital image. It is commonly used to recognize text in scanned documents and images. An artificial Intelligence Image Recognition model is used to convert a physical paper document, or an image into an accessible electronic version with text.

What is Image Recognition Model?

Recognition Model is a computer vision task that works to identify and categorize various elements of images. Image recognition models are trained to take an image as input and output and process those images.

What is an API?

API is the acronym for Application Programming Interface, which is a software intermediary that allows two applications to talk to each other.

What is Google Colab?

Colaboratory, or “Colab” for short, is a product from Google Research. Colab allows anybody to write and execute arbitrary python code through the browser, and is especially well suited to machine learning, data analysis and education.

What is Python?

Python is an interpreted, high-level, general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace.

What is Android Studio?

Android Studio is the official integrated development environment for Google's Android operating system, built on JetBrains' IntelliJ IDEA software and designed specifically for Android development.

What is PyTorch?

PyTorch is an open-source machine learning (ML) framework based on the Python programming language and the Torch library. It is one of the preferred platforms for deep learning research.

What is Pandas and NumPy?

NumPy is a library for Python that adds support for large, multi-dimensional arrays and matrices, along with a large collection of high-level mathematical functions to operate on these arrays. Pandas is a high-level data manipulation tool that is built on the NumPy package.

What is Gantt Chart?

A Gantt chart is a type of bar chart that illustrates a project schedule, named after its inventor, Henry Gantt, who designed such a chart around the years 1910–1915. Modern Gantt charts also show the dependency relationships between activities and current schedule status.

What is the use of site map?

A sitemap is a file where you can list the web pages of your website to tell Google and other search engines about the organization of your site content. Search engine web crawlers like Googlebot read this file to crawl your site more intelligently.

What is Storyboard?

A storyboard is a graphic organizer in the form of illustrations or images displayed in sequence for the purpose of pre-visualizing a motion picture, animation, motion graphic or interactive media sequence.

What is Use case Diagram?

A use case diagram at its simplest is a representation of a user's interaction with the system that shows the relationship between the user and the different use cases in which the user is involved.