# Software Project Lab – II

## Software Requirements Specification and Analysis

# Bengali Braille to Text Translator

Submitted by

**Sadikul Haque(bsse1003)**
**Md. Arman Hossain(bsse1029)**
**Samiha Sarkar(bsse1042)**

Supervised by

**Dr. Mohammad Shoyaib**

Professor

**Institute of Information Technology**

**University of Dhaka**

22 – October – 2020

# Letter of Transmittal

22th October, 2020.
BSSE 3rd Year SPL Committee
Institute of Information Technology
University of Dhaka

Sir,
We have prepared the report on Software Requirements Specification of Bengali Braille to Text Translator.
The primary purpose of this report is to summarize our findings that we have gathered during the requirements specification process. The report also includes details of each step we have followed while collecting the requirements.

Sincerely Yours,

Sadikul Haque Sadi
Md. Arman Hossain
Samiha Sarkar
Institute of Information Technology
University of Dhaka

Enclosure: Software Requirements Specification Report

# Executive Summary

Braille is a specialized writing system for visually impaired people, where raised dots on embossed paper are used as tactile alphabets. The tool Bengali Braille to Text Translator will take in a scanned image of Bengali Braille writing, apply pattern recognition, and translate it to text. The user does not need much theoretical or technical skill to run this software.

# Acknowledgement

By the grace of Almighty Allah We have completed our report on Software Requirements Specification of Bengali Braille to Text Translator. We are grateful to our supervisor Dr. Mohammad Shoyaib for his direction throughout the working time. It was almost impossible for us to complete this SRS document without him.

We are also thankful to the teachers and students of The Institute of Education and Research, University of Dhaka. They greatly helped us in collecting information among all businesses.

# Chapter 1

# Introduction

This chapter is a part of our Software Requirement Specification and Analysis for the project “Bengali Braille to Text Translator”. In this chapter, the center of our activity is the intended audience for this project.

## 1.1 Purpose

This document is the simple outline of the Software Requirement Specification and Analysis of our project for Software Project Lab – 2 titled as “Bengali Braille to Text Translator”. It contains functional, non – functional and supporting requirements and establishes a requirement baseline for developing the system. The Software Requirement Specification holds the requirements that were collected from student of the Institute of Education and Research (IER), University of Dhaka premises. The Software Requirement Specification serves as a platform to forward user requirements to us and provides a common reference point for us, our supervisor and student of IER.

## 1.2 Intended Audiences

Our Software Requirement Specification (SRS) is pinned for several audiences including students of IER as well as our project supervisor, SPL–2 coordinators and me.

- Teachers and students of IER, University of Dhaka will use this SRS to verify that we have developed a product that the required
- Our supervisor will use this SRS to plan milestones and ensure that we are on the right track when developing the system
- We will use this SRS as a basis for creating the system design. We will continually refer back to this SRS to ensure that the system we are designing, will fulfill the requirements of the teacher and students of IER.

- We will also use this SRS as a basis for developing the system functionality and link the requirements defined in this SRS to the software that we will create to ensure that we have created a software that will fulfill all the requirements

## 1.3 Conclusion

We wish, this analysis of the audience will help us to focus on the users who will be using our analysis. This document will help each and every person related to this project to perceive the subject matter of the project.

# Chapter 2

# Inception

In this chapter, we succinctly discuss the Inception part of the SRS for our "Bengali Braille to Text Translator".

## 2.1 Introduction

Requirements Engineering starts with Inception phase. Its goal is to identify parallel needs and conflicting requirements among the stakeholders of a project. The foundation was established by following the subsequent factors-

### 2.1.1 Identifying Stakeholders

Any person, group, or organization which will affect or be affected by the system directly or indirectly is a stakeholder. It includes both project developers and end-users. Here only client-side stakeholders will be focused.

Although, we intend to develop Bengali Braille to Text Translator for public use, we are currently building it for using only in the Institute of Education and Research (IER), University of Dhaka premises. For this reason, we have selected the stakeholders from the scope of IER only. We have identified following stakeholders for our project:

i. **Teachers of IER:** Teachers working with visually impaired student at IER are our primary client-side stakeholders. They will directly interact with the system.

ii. **Visually Impaired Students of IER:** Although the visually impaired students will not interact with the system directly, but they are the biggest group affected by the system. The software will process their writing and convert it to text.

iii. **General Students of IER:** Visually normal students will also use the software. But they will not be using it for the same reason as the teachers.

### 2.1.2 Recognizing Multiple Viewpoints

We have collected these view points by discussing with the teachers, visually impaired students, and general students of IER.

**1. Teachers of IER:**

- ➢ Proper braille syntax
- ➢ Intuitive user interface
- ➢ Ability to see intermediate results

**2. Visually Impaired Students of IER:**

- ➢ Error free solution
- ➢ Process double sided writing
- ➢ Special character recognition ability
- ➢ Fast processing

**3. General Students of IER:**

- ➢ Easy to use
- ➢ User friendly interface
- ➢ Easy to install
- ➢ Work with any scanner
- ➢ No need of theoretical or technical knowledge

### 2.1.3 Working towards Collaboration

Every stakeholder has his own set of requirements from his point of view. We followed following steps to merge these requirements:

- ➢ Identify the common and conflicting requirements
- ➢ Categorize the requirements
- ➢ Take priority points for each requirement from stakeholders and on the basis of this voting prioritize the requirements
- ➢ Make final decision about the requirements

**Common Requirements:**

- User friendly interface
- Easy to use

**Conflicting Requirements:**

- Low cost and process double sided writing
- Fast and minimum error rate
- Full control of the system and no need of theoretical or technical knowledge

**Final Requirements:** We have finalized following requirements for the system through categorization and prioritization process:

- Easy to use
- User friendly interface
- Users with theoretical and technical knowledge will be able to fully control the system
- Users with no technical knowledge will be able to use by reading tips and help for running the system
- Minimize error rate
- Can work with any scanner

# Chapter 3

# Elicitation

## 3.1 Introduction

Elicitation enables the client more specifically define the necessity. This stage faces many issues such as scope issues, volatility issues, and understanding issues. We worked in an organized and systematic way to solve these issues.

## 3.2 Eliciting Requirements

Unlike the beginning, Elicitation uses a requirements format that incorporates problem solving, preparation, negotiations and specification components, in which questions were answered. A group of end-users and developers must cooperate in order to generate the demands.

## 3.3 Collaborative Requirements Gathering

There are many different approaches to collaborative requirements gathering. Each approach makes use of a slightly different scenario. We followed the subsequent steps to do it:

I. Meetings were conducted with teachers and students of IER, DU. They were questioned about their requirements and expectations from the tool.

II. They were asked about the problems they are facing with exam papers written in Braille. We also inquired regarding the efficiency of the current process.

III. At last we selected our final requirement list from these meetings.

## 3.4 Quality Function Deployment

The technique which translates the needs of the customer into technical requirements for software is called Quality Function Deployment (QFD).
QFD concentrates on maximizing customer satisfaction from the Software Engineering process. With respect to our project the following requirements are identified by QFD-

### 3.4.1 Normal Requirements

Normal requirements refer to the objectives and the goals that are stated for the product during the meeting with the stakeholders. The presence of these requirements ensures the satisfaction of the customers. The normal requirements for the project are stated below.

- ➢ Users will be able to run the tool and get output with only a single 'run' button.
- ➢ User can view result from each level of Braille to text translation process.
- ➢ System will be able to translate braille images written with any scale of braille device.
- ➢ System will be connected with a printer and a scanner machine
- ➢ System will be able to translate any bangla character(joint letters, numbers, punctuations)

### 3.4.2 Expected Requirements

The requirements that are implicit to the system might not be brought up during the meeting because of their fundamental nature. Despite being not explicitly mentioned their presence must be ensured. Otherwise, the product will leave customers dissatisfied. These requirements are called expected requirements and these are stated below.

- ➢ The system will be able to support all popular image formats like- JPG, JPEG, PNG.
- ➢ User will be able to process multiple images of Braille writing.
- ➢ The user interface of the system shall be easy to use. It will make use of drop-down boxes, radio buttons, and other selectable fields

wherever possible instead of fields that require the user to type in data.

- System will be faster

### 3.4.3 Exciting Features

The factors that go beyond the customer's expectations and prove to be satisfying when present are called exciting features. The exciting features are the so called 'wow factor' for our project.

- The user interface should provide appropriate error messages for invalid input as well as tool-tips and help.
- User will be able to view each processing step separately.
- User will be able to save the output in a ".txt" file using this system if wishes
- User will be able to cancel processing at the middle of processing step without system crash.

## 3.5 Background Studies

This part of this document contains necessary terms which be helpful to understand the next Usage Scenario and Methodology of this project.

### 3.5.1 Image

an image can be defined by a two-dimensional array specifically arranged in rows and columns. Digital Image is composed of a finite number of elements, each of which elements have a particular value at a particular location which is called pixel [1]. Each pixel has three values of RGB (Red, Green, Blue) in between 0-255 or we can say that colors here are of the 24-bit format, that means each color has 8 bits of red, 8 bits of green, 8 bits of blue, in it. Each color has three different portions.

### 3.5.2 Image Acquisition (RGB to Gray)

It is converting a digital image from a given one processing each pixel with some operation. On this project the RGB to Gray conversion will be used.

Average method is the simplest one. You just have to take the average of three colors. Since it’s an RGB image, so it means that you have add r with g with b and then divide it by 3 to get your desired grayscale image.

It’s done in this way.

Grayscale = (R + G + B ) / 3

### 3.5.3 Noise filtering in Digital Image Processing

Noise is always presents in digital images during image acquisition, coding, transmission, and processing steps. noise is abrupt change in pixel values in an image. So when it comes to filtering of images, the first intuition that comes is to replace the value of each pixel with average of pixel around it. This process smooths the image. To reduce the noise from the image mean, median and/or gaussian filter will be used based on image quality.

- ➢ In image processing, a Gaussian blur (also known as Gaussian smoothing) is the result of blurring an image by a Gaussian function (named after mathematician and scientist Carl Friedrich Gauss). It is a widely used effect in graphics software, typically to reduce image noise and reduce detail.

- ➢ Mean filter is a simple sliding window that replace the center value with the average of all pixel values in the window. The window or kernel is usually a square but it can be of any shape.

- ➢ Median filter is a simple sliding window that replace the center value with the Median of all pixel values in the window. The window or kernel is usually a square but it can be of any shape.

### 3.5.4 Otsu's Thresholding

Image thresholding is used to binarize the image based on pixel intensities. The input to such thresholding algorithm is usually a grayscale image and a threshold. The output is a binary image.

If the intensity of a pixel in the input image is greater than a threshold, the corresponding output pixel is marked as white (foreground), and if the input pixel intensity intensity is less than or equal to the threshold, the output pixel location is marked black (background).

Otsu’s global thresholding algorithms usually have following steps.

1. Process the input image
2. Obtain image histogram (distribution of pixels)
3. Compute the threshold value T
4. Replace image pixels into white in those regions, where saturation is greater than T and into the black in the opposite cases

### 3.5.5 Morphological Dilation and Erosion

The most basic morphological operations are dilation and erosion. Dilation adds pixels to the boundaries of objects in an image, while erosion removes pixels on object boundaries. The number of pixels added or removed from the objects in an image depends on the size and shape of the structuring element used to process the image. In the morphological dilation and erosion operations, the state of any given pixel in the output image is
determined by applying a rule to the corresponding pixel and its neighbors in the input image. The rule used to process the pixels defines the operation as a dilation or an erosion .

## 3.6 Usage Scenario

Bengali Braille Character Recognizer will be a desktop application for Windows operating system. It will be a tool that will take a scanned image of Bengali Braille writing as input. The system will be able to support all popular image formats like- JPG, JPEG, PNG. The input will go through different types of image-preprocessing techniques, translate Braille to text through pattern recognition, and apply text correction procedures for final output. The methodology of whole process is discussed below. Then it will provide the Bengali text that is written in the scanned input image. If the user want to save the text file in the local directory of the computer our system will provide the user to save it.

### 3.6.1 Methodology

Braille is a reading and writing system which can only be read with the sense of touch. It is used by blind and visually impaired people who cannot access print materials. Braille is not a language. Rather, it is a code by which many languages can be written and read. It uses raised dots to represent the letters of the print alphabet. It also includes symbols to represent punctuation. A Braille character is made using a combination of 6 dots which is arranged in two columns and three rows.

At first, the user will select scanned images of Braille writing. Then the following methodology will be applied for final output.

#### 3.6.1.1 Image Preprocessing:

Different types of image enhancement processes will be run. Then Image preprocessing is an essential step during which errors that occurred while the images were taken are eliminated. Errors include noise, deformation, bad illumination or blurring. Image preprocessing can be used for image enhancement by reducing noise.

1. Grayscale conversion: In any image, each of the pixels should have a variation of RGB values for three different colors Red, Green, and Blue. But for having the same value of each three colors for every pixel it requires to grayscale conversion for this image.

2. Noise reduction: For noise reduction median filter, gaussian elimination will be used.

3. Binary conversion: Then it will take a thresholded image consisting of couples of white and black spots, where each couple denotes a single Braille dot. To do this Otsu thresholding will be used.

4. Filling: For filling the dots Morphological Dilation will be used based on dots quality.

### 3.6.1.2 Line Identification:

After preprocessing the most important task is horizontal line identification. This will be done based on the dot frequency in the image. Each horizontal text line will consist of three-dot lines.

### 3.6.1.3 Braille Cells and Dots Framing:

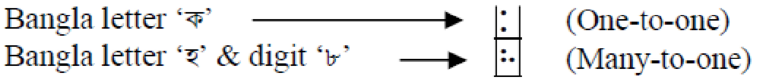
The framing process to determine the cells, words, and lines in the scanned image based on some statistical measurement of area of dot,distance between two dot,distance between two characters and distance between two lines.

### 3.6.1.4 Decimal Braille Code Generation:

This stage is a core stage of the system and it was done by testing each dot in Braille cell, if it active the position of this dot take digit one and if inactivated the position of this dot takes digit zero. The recognizing process of active or inactive dots was depended on taking summation of dot frame, if the summation was been one's digits, that means this dot is activated, else that dot will be inactive.

### 3.6.1.5 Braille letter recognition and transcription:

Braille is not a language. It is a code for mapping character sets of various languages to its 64 fixed permutations. Mapping of character sets of a language to Braille symbols is called character mapping. Different countries developed their own standard character mapping for their language. This character mapping is not necessarily one-to-one because some languages may have more than 64 letters in their alphabet. So possible character mapping can be one-to-one and many-to-one. For example



Each letter of Bangla alphabet, numeric and punctuation has corresponding Braille representation. Bangla letters and their corresponding presentations are given in Table I, II and III.

TABLE I. ONE TO ONE CHARACTER MAPPING

| Bangla | Braille | Bangla | Braille | Bangla | Braille |
|---|---|---|---|---|---|
| ক | ⠅ | ধ | ⠮ | ড় | ⠻ |
| খ | ⠭ | ন | ⠝ | ঢ় | ⠷ |
| ঘ | ⠣ | প | ⠏ | য় | ⠢ |
| ঙ | ⠬ | ম | ⠍ | ৎ | ⠂⠞ |
| ছ | ⠡ | য | ⠽ | ঁ | ⠄ |
| ঝ | ⠵ | র | ⠗ | ং | ⠰ |
| ট | ⠾ | ল | ⠇ | ঃ | ⠠ |
| ঠ | ⠺ | শ | ⠩ | ক্ষ | ⠟ |
| ড | ⠫ | ষ | ⠯ | জ্ঞ | ⠱ |
| ঢ | ⠿ | স | ⠎ | ; | ⠆ |
| ত | ⠞ | । | ⠲ | ! | ⠖ |
| থ | ⠹ | ‘ | ⠠⠦ | = | ⠰⠶ |
| ভ | ⠧ | ’ | ⠴⠄ | * | ⠔⠔ |
| - | ⠤ | [ | ⠠⠶ | ] | ⠶⠄ |

TABLE II. TWO TO ONE CHARACTER MAPPING

| Bangla | | Braille | Bangla | | Braille |
|---|---|---|---|---|---|
| অ | ১ | ⠁ | চ | ৩ | ⠉ |
| আ | া | ⠜ | জ | ০ | ⠚ |
| ঈ | ী | ⠔ | ঞ | : | ⠒ |
| উ | ু | ⠥ | ণ | Number prefix | ⠼ |
| ঊ | ূ | ⠳ | দ | ৪ | ⠙ |
| ঋ | ৃ | ⠐⠗ | ফ | ৬ | ⠋ |
| ও | ো | ⠕ | ব | ২ | ⠃ |
| ঔ | ৌ | ⠪ | হ | ৮ | ⠓ |
| গ | ৭ | ⠛ | ্ | ” | ⠴ |
| ( | ) | ⠶ | ? | “ | ⠦ |

TABLE III. THREE TO ONE CHARACTER MAPPING

| Bangla | | | Braille |
|---|---|---|---|
| ই | ি | ৯ | ⠊ |
| এ | ে | ৫ | ⠑ |
| ঐ | ৈ | / | ⠌ |
| , | . | '(lop) | ⠂ |

In this stage of the project, the Braille letter was recognized using a matching algorithm to match each of the input decimal Braille code from an input processed image with codes of each Bengali letter. After the recognition process implemented the recognized letter transcript into equivalent text. The following represent the Bengali Braille Characters.

## 3.6.2 Braille word recognition and transcription

### 3.6.2.1 Grammatical conversion rules

In conversion of Bangla text to Braille, one needs to deal with conjunctions, consonants, dependent and independent vowels, punctuations and numbers. Here all grammatical conversion rules are discussed with examples.

1) Replacing rule: In replacing rule each Bangla character is replaced by its corresponding Braille cell. Some uses of consonants, vowels (dependent and independent) and punctuations are given in Table IV.

TABLE IV. USES OF CONSONANTS, VOWELS AND PUNCTUATIONS

| Bangla Word | Distribution | Braille Representation |
|---|---|---|
| বল | ব ল | ⠃ ⠇ |
| উৎস | উ ৎ স | ⠥ ⠂ ⠞ ⠎ |
| কাঠ | ক া ঠ | ⠅ ⠜ ⠺ |
| দৃঢ় | দ ৃ ঢ় | ⠙ ⠐ ⠗ ⠷ |
| কমা | , | ⠂ |
| সেমি কোলন | ; | ⠆ |

2) Inserting rule: In inserting rule a Braille cell is inserted as prefix. Other characters are replaced according to replace rule.

- If there is "i", "u" or "o" after consonant and if "a" is pronounced there then Bangla "a" equivalent Braille cell dot 1 is inserted after consonant in Braille [14]. Examples are shown in Table V.

TABLE V. INSERTION OF "অ"

| Bangla Word | Distribution | Braille Representation |
|---|---|---|
| বই | ব ই | ⠃⠁⠒ |
| রওনা | র ও না | ⠗⠁⠕⠝⠜ |

- Braille cell dot 4 is inserted before conjunctions having combination of two letters. Examples are shown in Table VI.

TABLE VI. CONJUNCTION OF 2 LETTERS

| Bangla Word | Conjunct | Distribution | Braille Representation |
|---|---|---|---|
| গ্রাম | গ্র | গ ্ র | ⠈ ⠛ ⠗ ⠜ ⠍ |
| পূর্ব | র্ব | র ্ ব | ⠏ ⠳ ⠈ ⠗ ⠃ |

- Braille cell dot 4, 6 is inserted before conjunctions having combination of three letters or four letters. Examples are shown in Table VII.

TABLE VII. CONJUNCTION OF 3AND 4 LETTERS

| Bangla Word | Conjunct | Distribution | Braille Representation |
|---|---|---|---|
| রাষ্ট্র | ষ্ট্র | ষ ্ ট ্ র | ⠗⠜⠨⠯⠾⠗ |
| স্বাতন্ত্র্য | ন্ত্র্য | ন ্ ত ্ র ্ য | ⠈⠎⠃⠜⠞⠨⠝⠞⠗⠽ |

- Two Bangla conjunctions have direct representation in Bangla Braille. So, there is no need to use Braille cell dot 4 before them. They are given in Table VIII.

TABLE VIII. TWO CONJUNCTIONS WITHOUT PREFIX

| Bangla Word | Conjunct | Distribution | Braille Representation |
|---|---|---|---|
| কক্ষ | ক্ষ | ক ্ ষ | ⠅ ⠟ |
| জ্ঞান | জ্ঞ | জ ্ ঞ | ⠱ ⠜ ⠝ |

This is the last stage in implementation, the word recognition process flow through letter recognition and fill Braille the decimal array of the word, to apply matching process with stored addressed text and voices files of words, then to run equivalent files from the addressed word database.

# Chapter 4

# Scenario Based Modeling

This chapter contains the Scenario Based Model for our project "Bengali Braille to Text Translator".

## 4.1 Introduction

For developing our software, we are giving the highest priority to user satisfaction. To identify the requirements to establish meaningful analysis and design model we determine how users want to interact with the system. Thus, our requirements modeling begins with scenario generation in the form of use cases, activity diagrams.

## 4.2 Use Case

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions that some system or sub-systems can perform in collaboration with one or more external users of the system.

The first step in writing a Use Case is to define that set of "actors" that will be involved in the story. Actors are the different people that use the system or product within the context of the function and behavior that is to be described. Actors represent the roles that people play as the system operators. Every user has one or more goals when using the system.

### 4.2.1 Primary Actor

Primary actors interact directly to achieve required system function and derive the intended benefit from the system. They work directly and frequently with the software. In our system both users the system both are primary actor.

## 4.3 Activity diagram

Activity diagrams are graphical representations of workflows of stepwise activities and actions with support for choice, iteration and concurrency. In this chapter we did try to provide each use case and its corresponding activity diagram together.

## 4.4 Use Case and Activity Diagram

### 4.4.1 Level 0 Use Case Diagram of Bengali Braille to Text Translator

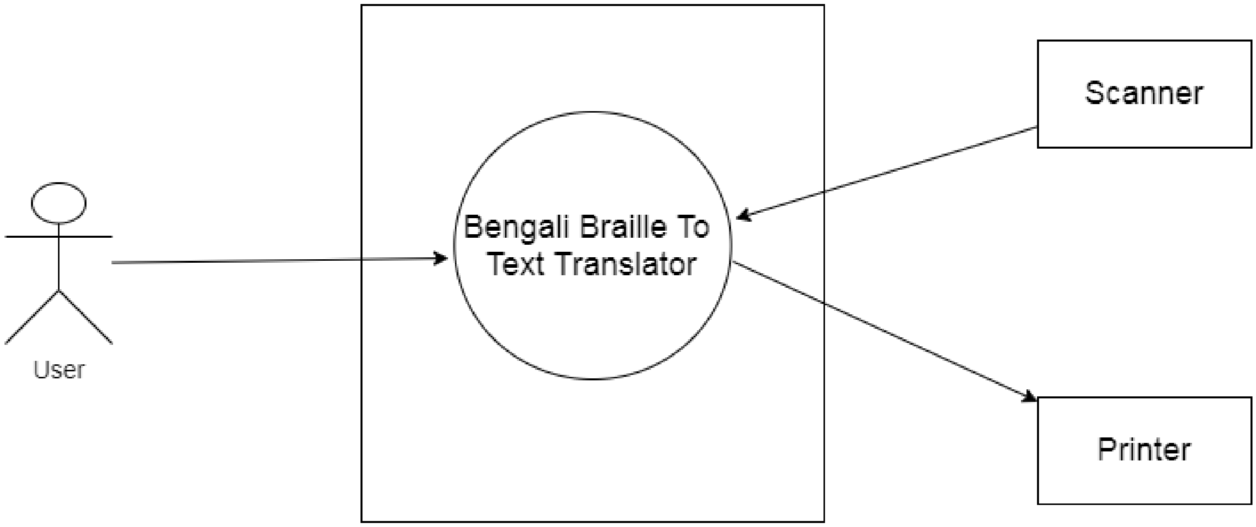


Figure 1: Level 0 use case diagram of Bengali Braille to Text Translator

Table 1: Information about level 0 use case diagram

| | |
|---|---|
| **Name:** | Bengali Braille to Text Translator |
| **ID:** | L-0 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

#### 4.4.1.1 Description of Level 0 Use Case Diagram

After analyzing usage scenario, we found that user interact with our system as primary actor.

### 4.4.2 Level 1 Use Case Diagram of Bengali Braille to Text Translator

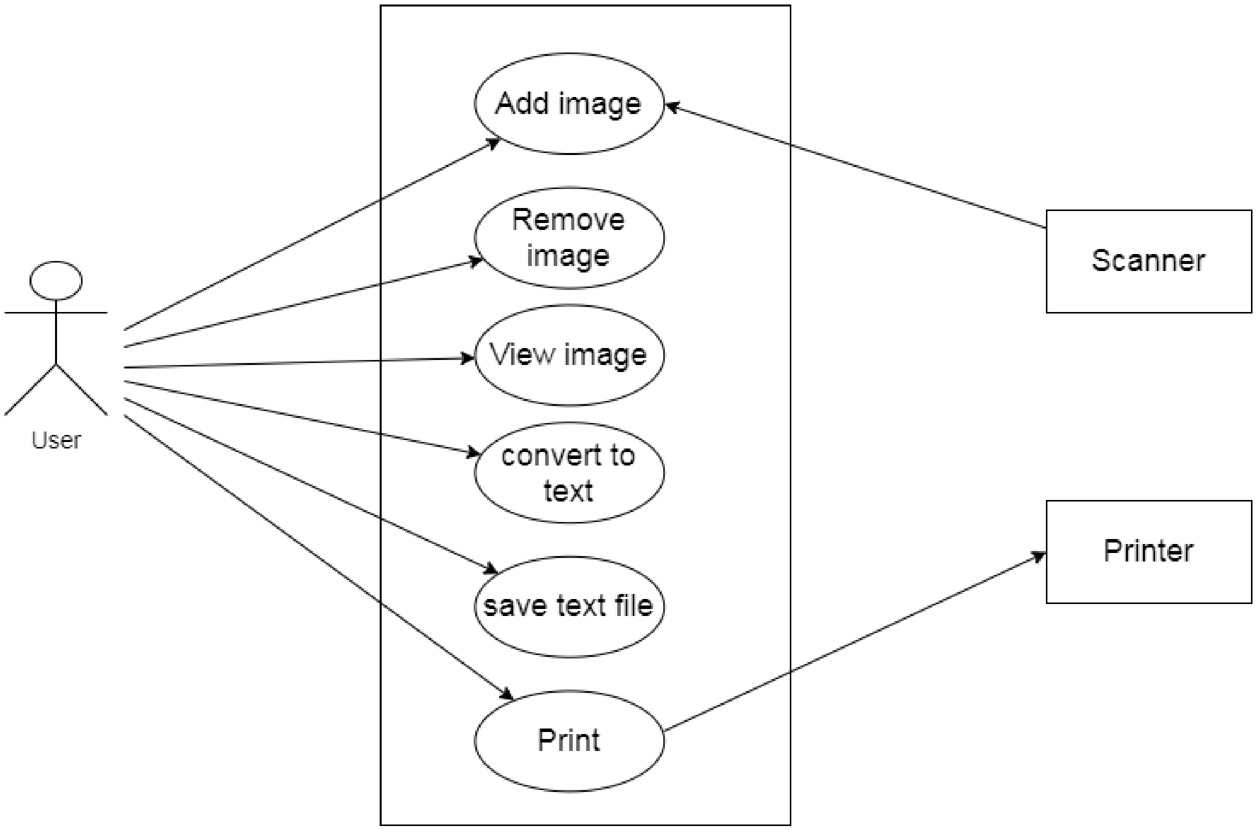


Figure 2: Level 1 use case diagram of Bengali Braille to Text Translator

Table 2: Information about level 1 use case diagram

| | |
|---|---|
| **Name:** | Bengali Braille to Text Translator |
| **ID:** | L-1 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

### 4.4.2.1 Description of Level 1 Use Case Diagram

In the usage scenario we separated our System into several modules. Here the user provides scanned image as input, then he/she can view the binary image of it, remove it or continue with it to get corresponding Bengali text. After this he/she can save the file and print it also.

**Action And Replay:**

Action-1: User will select one or more braille image
Reply-1: System will load this image and convert it to binary image

Action-2: User wants to view the image after uploading it
Reply-2: The binary image will be shown to the user

Action-3: User wants remove the uploaded image
Reply-3: The image will be removed from the system

Action-4: User converts the image to text
Reply-4: System will generate the corresponding text file

Action-5: User wants to save the text file
Reply-5: The generated text file will be saved to local drive

Action-6: User wants to print the output file
Reply-6: The system will print the output text file

### 4.4.3 Level 1.1 Use Case Diagram of Bengali Braille to Text Translator

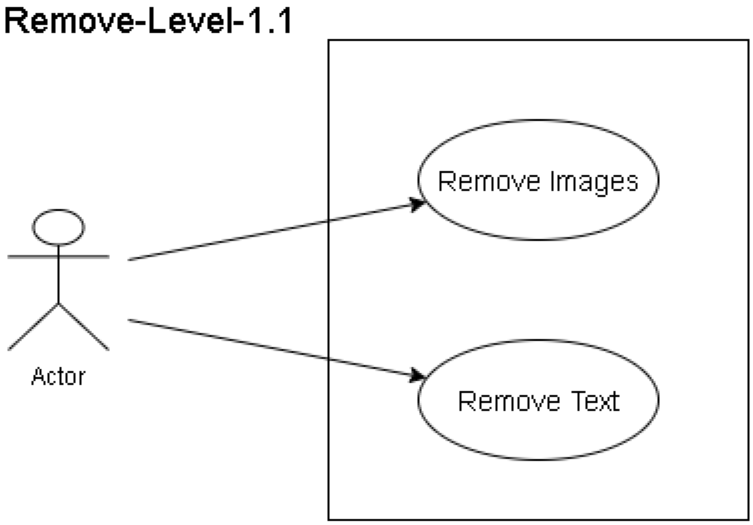


Figure 3: Level 1.1 use case diagram of Remove

Table 3: Information about level 1.1 use case diagram

| | |
|---|---|
| **Name:** | Remove |
| **ID:** | L-1 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

#### 4.4.3.1 Description of Level 1.1 Use Case Diagram

In this scenario users can remove the uploaded image and also the converted text of that image.

**Action And Replay:**

Action-1: User wants remove binary image
Reply-1: System will remove the image

Action-2: User wants to remove the text
Reply-2: System will remove the generated text

### 4.4.4 Level 1.2 Use Case Diagram of Bengali Braille to Text Translator

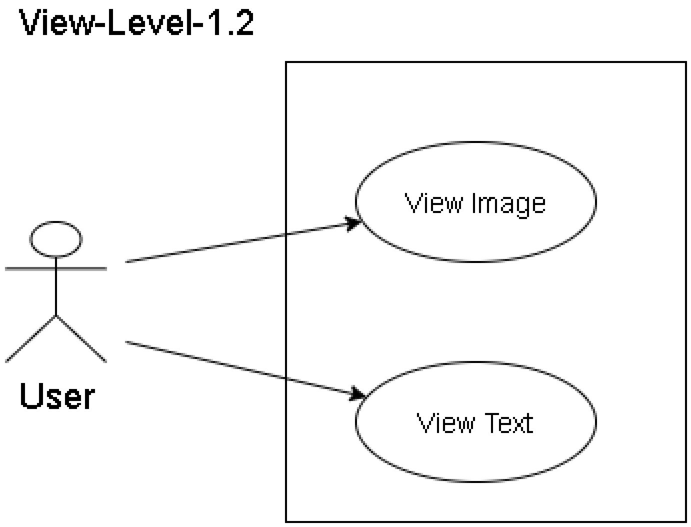


Figure 4: Level 1.2 use case diagram of View

Table 4: Information about level 1.2 use case diagram

| | |
|---|---|
| **Name:** | View |
| **ID:** | L-1.2 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

#### 4.4.4.1 Description of Level 1.2 Use Case Diagram

In this scenario user can view the binary image and also the converted text of that image.

**Action And Replay:**

Action-1: User wants view binary image
Reply-1: System will show the image

Action-2: User wants to view the text
Reply-2: System will show the generated text

## 4.4.5 Level 1.3 Use Case Diagram of Bengali Braille to Text Translator

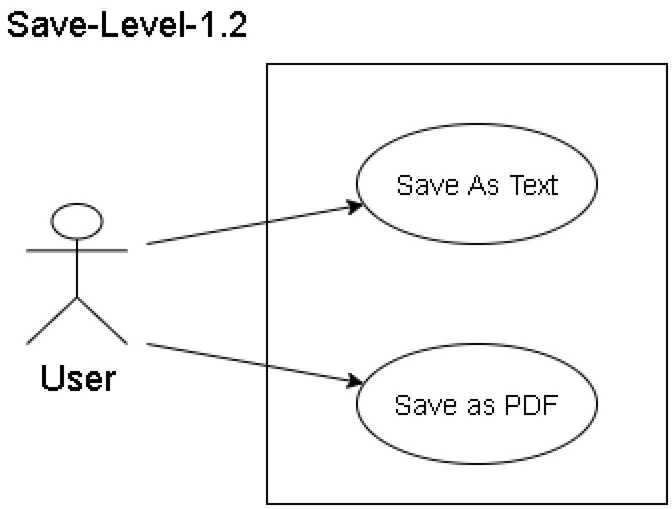


Figure 5: Level 1.3 use case diagram of Save

Table 5: Information about level 1.3 use case diagram

| | |
|---|---|
| **Name:** | Save |
| **ID:** | L-1.3 |
| **Primary Actor:** | User |
| **Secondary Actor:** | None |

### 4.4.5.1 Description of Level 1.3 Use Case Diagram

In this scenario user can save the generated text file as .txt file or .pdf file.

**Action And Replay:**

Action-1: User wants save txt file
Reply-1: System will save txt file in local drive

Action-2: User wants to save output as pdf file
Reply-2: System will convert txt to pdf file and will save it.

### 4.4.6 Level 1 Activity Diagram of Bengali Braille to Text Translator

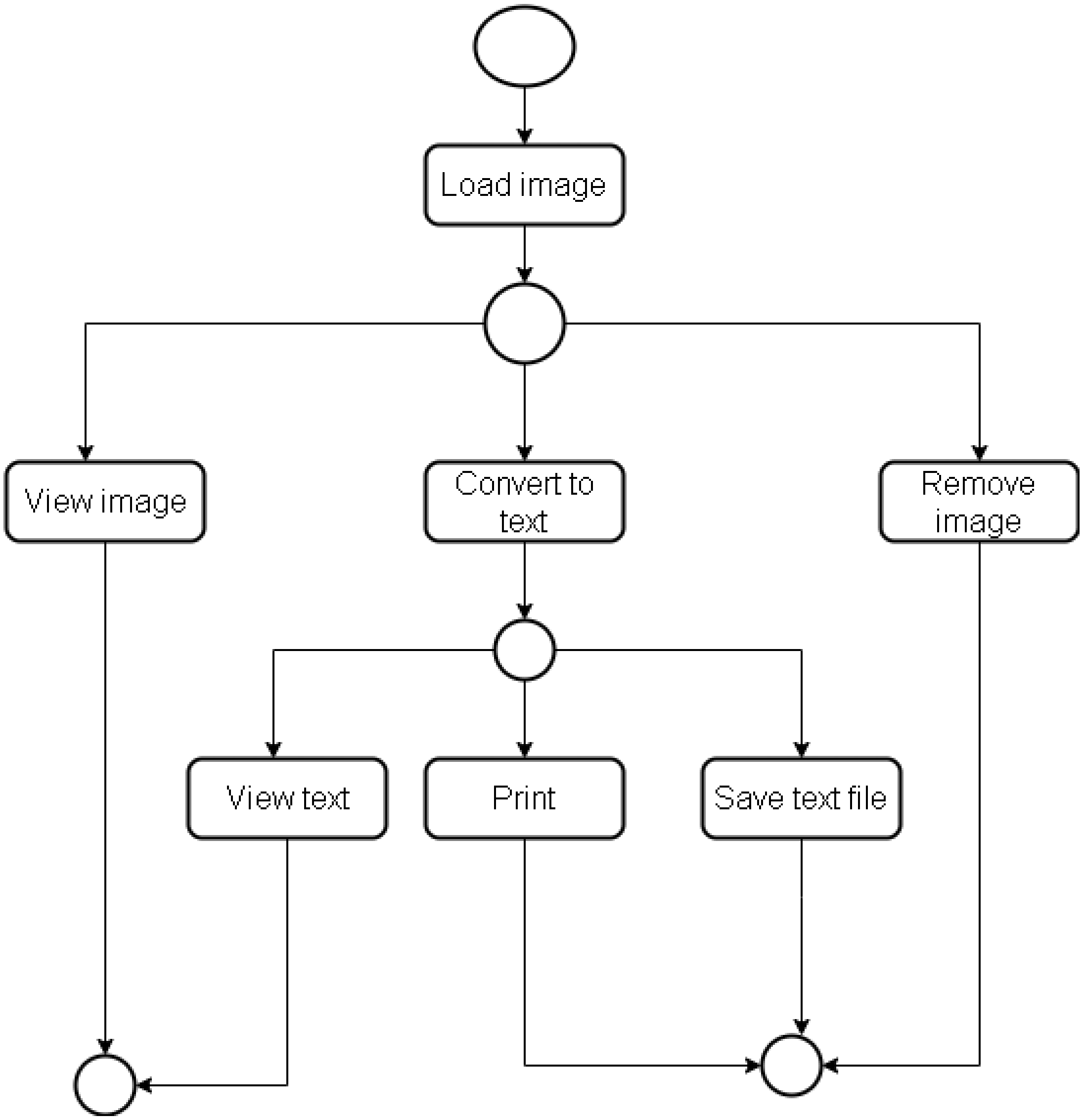


Figure 6: Level 1 activity diagram of Bengali Braille to Text Translator

### 4.4.7 Level 1.1 Activity Diagram of Bengali Braille to Text Translator

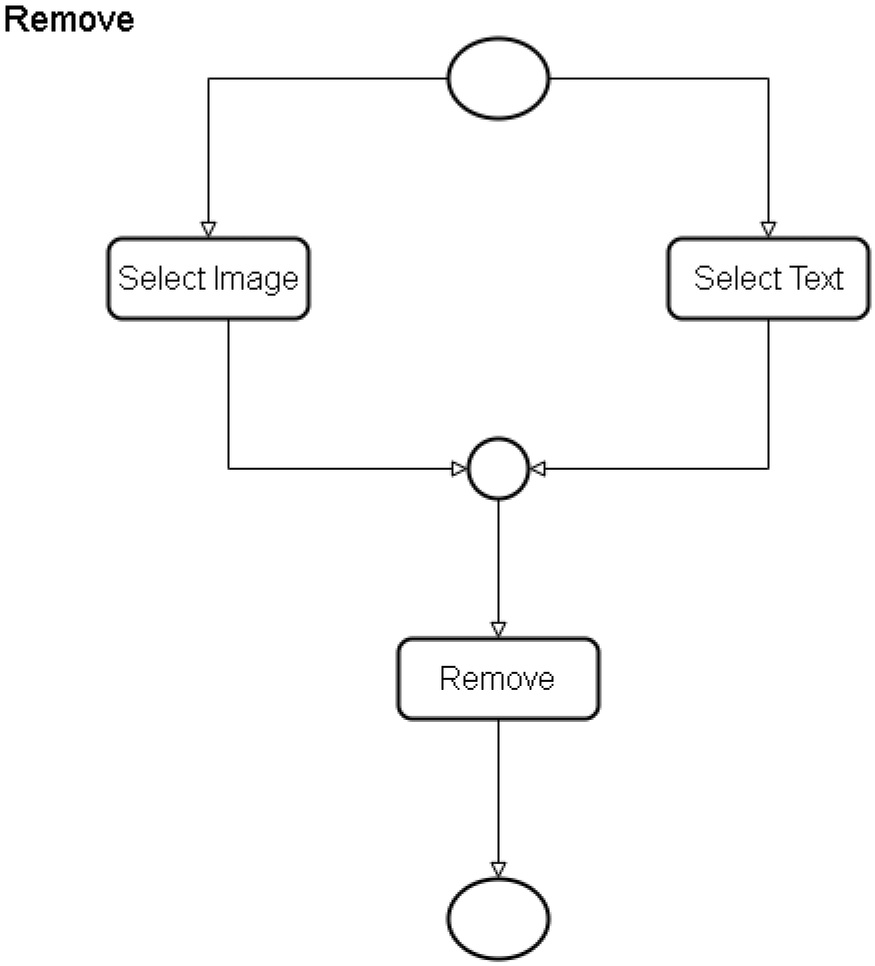


Figure 7: Level 1.1 activity diagram of Remove

### 4.4.8 Level 1.2 Activity Diagram of Bengali Braille to Text Translator

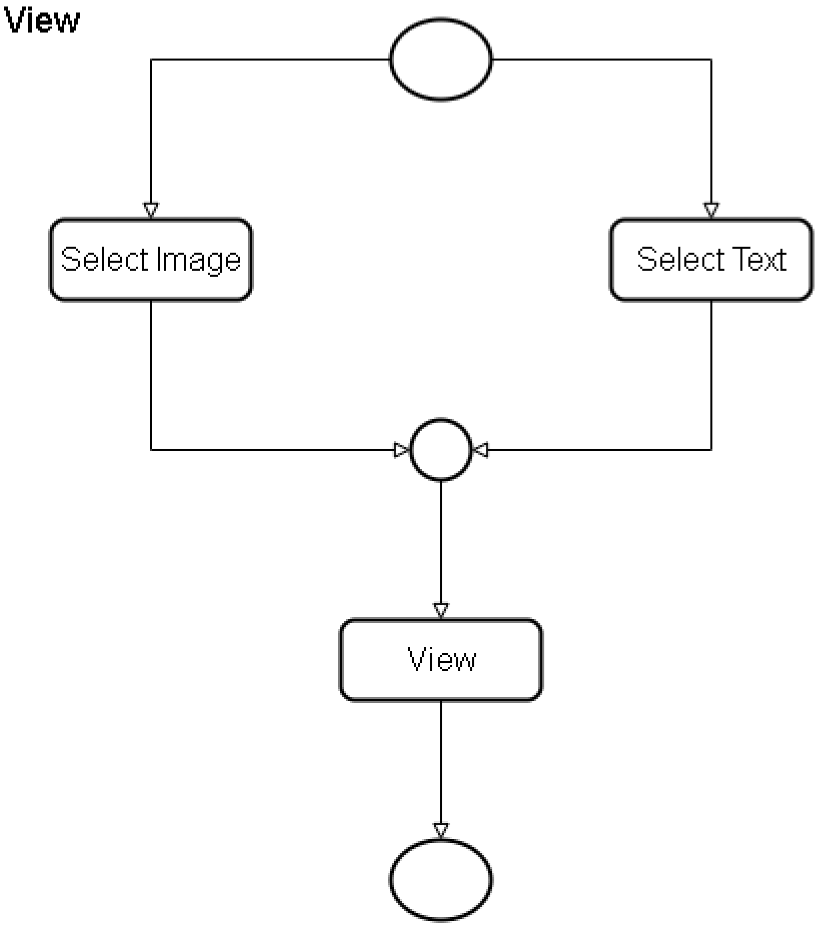


Figure 8: Level 1.2 activity diagram of View

### 4.4.9 Level 1.3 Activity Diagram of Bengali Braille to Text Translator

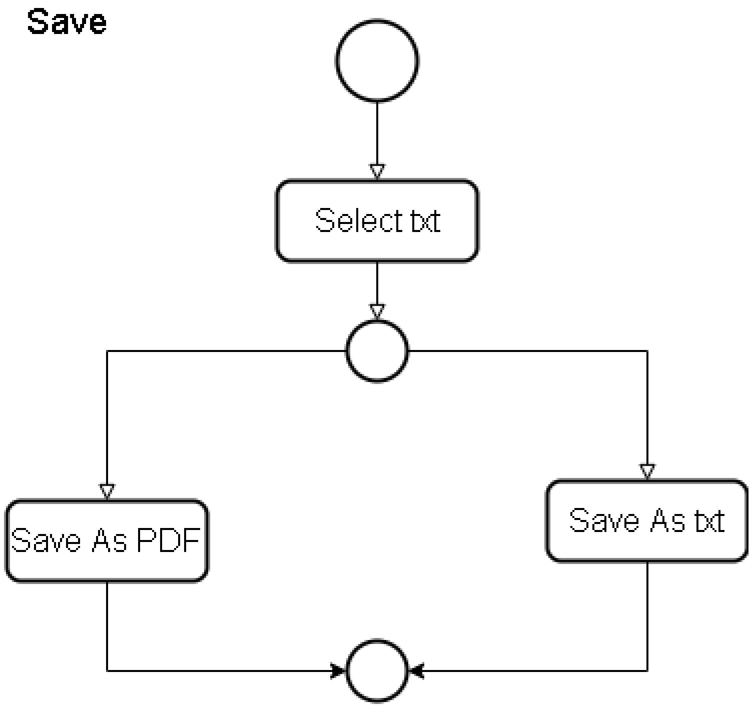


Figure 9: Level 1.3 activity diagram of Save

## 4.5 Swim lane Diagram

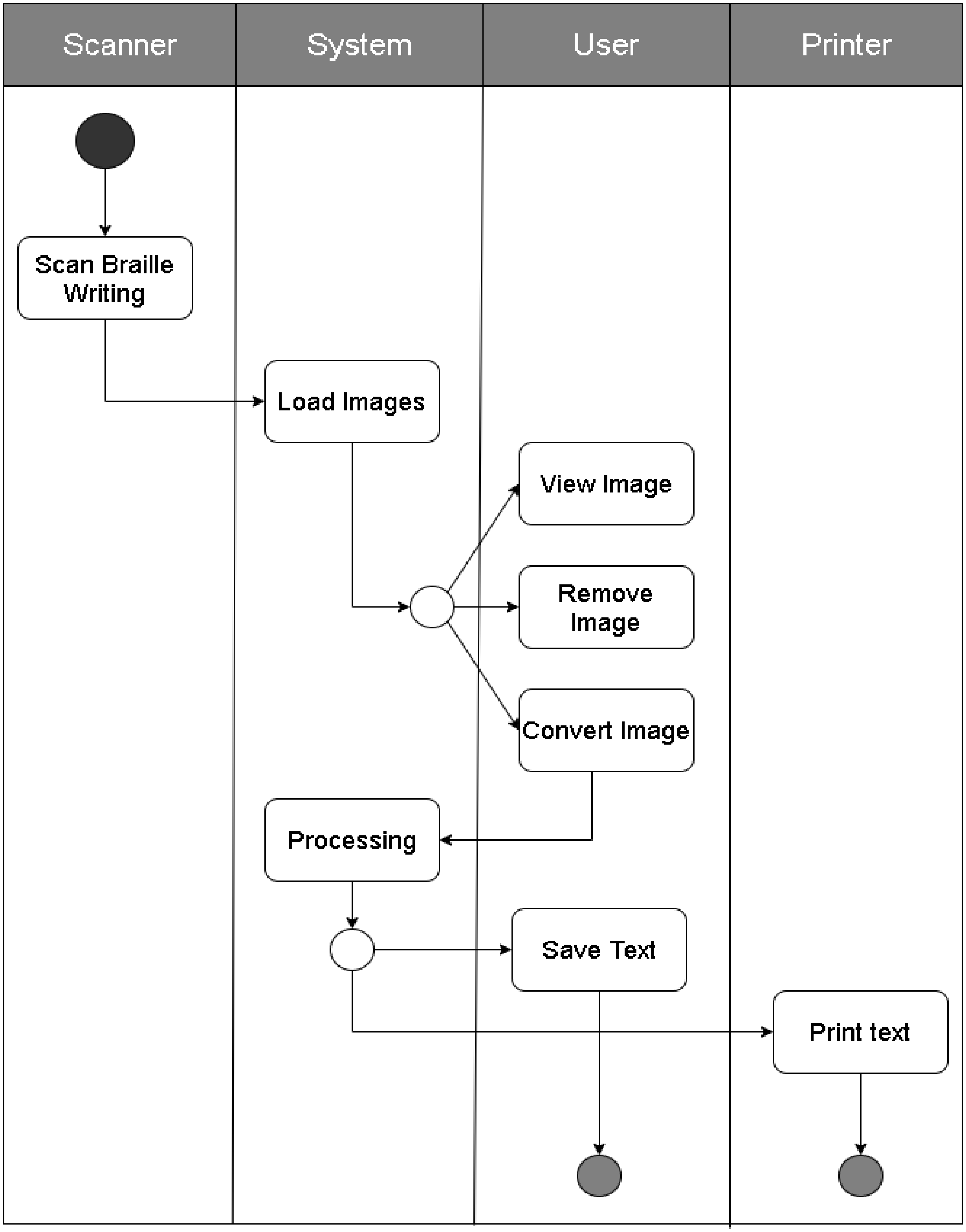


Figure 10: Level 1 swim lane diagram of braille to text translator

# Chapter 5

## Database Modeling

When software requires interfacing with a database or complex data structures need to be constructed and manipulated, data model is performed as part of overall requirements modelling.

### 5.1 Data Object Selection

Data objects are representation of information which has different attributes.The following table enlists all nouns from Usage Scenario and marks potential data objects:

Table 6: Data Object Selection Process

| Data Object Selection | | | | |
|---|---|---|---|---|
| **No** | **Noun** | **Attributes** | **Description** | **Verdict** |
| 1 | bengali braille character recognizer | | represents thee whole system | rejected |
| 2 | image | | alias of braille writing | rejected |
| 3 | braille writing | | input for the system | rejected |
| 4 | image preprocessing technique | | alias of algorithm | rejected |
| 5 | text | | alias of text file | rejected |
| 6 | pattern recognition | | alias of algorithm | rejected |
| 7 | final output | | output for the system | rejected |

| | | | | |
|---|---|---|---|---|
| 8 | braille | | alias of braille writing | rejected |
| 9 | system | | alias of bengali braille character recognizer | rejected |
| 10 | visually impaired people | | out of scope | rejected |
| 11 | raised dots | | out of scope | rejected |
| 12 | scanner | | a device, external entity | rejected |
| 13 | image files | | alias of braille writing | rejected |
| 14 | computer | | runs the whole system | rejected |
| 15 | scanned images | | alias of braille writing | rejected |
| 16 | image enhancement process | | alias of algorithm | rejected |
| 17 | section | | out of scope | rejected |
| 18 | braille pattern | | alias of braille writing | rejected |
| 19 | background | | out of scope | rejected |
| 20 | noise reduction algorithm | | alias of algorithm | rejected |
| 21 | artifact | | out of scope | rejected |
| 22 | pattern quality | | out of scope | rejected |
| 23 | braille dots | | alias of braille writing | rejected |
| 24 | connectivity improvement | | alias of algorithm | rejected |
| 25 | code | | an attribute of | rejected |

| | | | | |
|---|---|---|---|---|
| | | | braille pattern code | |
| 26 | **braille pattern code** | id, code | potential data object | **accepted** |
| 27 | braille character | | | rejected |
| 28 | **code to character map** | code, character | potential data object | **accepted** |
| 29 | hash table | | alias of braille code to character map | rejected |
| 30 | text file | | alias of final output | rejected |
| 31 | braille image | | alias of braille writing | rejected |
| 32 | stage | | alias of algorithm | rejected |
| 33 | spell checking | | alias of algorithm | rejected |
| 34 | cleaning procedure | | alias of algorithm | rejected |
| 35 | step | | alias of algorithm | rejected |
| 36 | setting | | an attribute of braille processing template | rejected |
| 37 | algorithm | | an attribute of braille processing template | rejected |
| 38 | parameter | | an attribute of braille processing template | rejected |
| 39 | pdf file | | alias of output file | rejected |
| 40 | braille processing system | algorithm, parameters | potential data object | **accepted** |
| 41 | template | | alias of braille | rejected |

| | | | | |
|---|---|---|---|---|
| | | | processing template | |
| 42 | user | | initializes system | rejected |
| 43 | system | | alias of braille writing | rejected |
| 44 | customized template | | alias of bengalii braille character recognizer | rejected |

## 5.2 Data Objects and Attributes

This is a brief view of all attributes we have found so far:

I. Braille Pattern Code : Id, Code
II. Code to Character Map : Code, Character
III. Braille processing System : Algorithm, parameters

## 5.3 Conclusion

Data objects found here have no relationship to each other. These data objects will be stored in a file system and will work independently. So there is no Data Object Relational Diagram here. For this same reason no E-R Diagram or Schema Diagram exists for this project.

# Chapter 6

## Class Based Modeling

We intended this chapter to describe class-based modeling for our "Complain Box".

### 6.1 Introduction

In this chapter, our designed class-based model represents the objects that our "Bengali Braille to Text Translator" will manipulate, the operation that will applied to the objects, relationships between and the collaboration that occur between the classes that are defined.

### 6.2 General Classification

Analysis classes can be marked by one of the following ways:

1. External Entity
2. Thing
3. Occurrence
4. Role
5. Organizational Unit
6. Place
7. Structure

Table 7: General Classification for Class Selection

| General Classification | | | |
|---|---|---|---|
| **No** | **Noun** | **Classification** | **Remark** |
| 1 | bengali braille character recognizer | 2 | problem space |
| 2 | image | 2, 7 | problem space |

| | | | |
|---|---|---|---|
| 3 | braille writing | | problem space |
| 4 | image preprocessing technique | 3, 7 | solution space |
| 5 | text | 2, 7 | problem space |
| 6 | pattern recognition | 3 | solution space |
| 7 | final output | 7 | problem space |
| 8 | braille | | problem space |
| 9 | system | 2 | problem space |
| 10 | visually impaired people | 1 | problem space |
| 11 | raised dots | 1 | problem space |
| 12 | scanner | 1,2 | problem space |
| 13 | image files | 1,2 | problem space |
| 14 | computer | 1,2,7 | problem space |
| 15 | scanned images | 1,2 | problem space |
| 16 | image enhancement process | 3 | solution space |
| 17 | braille to text | 4,6 | problem space |
| 18 | braille pattern | 1,2 | problem space |
| 19 | background | | problem space |
| 20 | noise reduction algorithm | 3 | solution space |
| 21 | artifact | | problem space |
| 22 | pattern quality | 3 | solution space |
| 23 | braille dots | 1,2 | problem space |
| 24 | connectivity | 3 | solution space |

| | | | |
|---|---|---|---|
| | improvement | | |
| 25 | code | | problem space |
| 26 | braille pattern code | 2 | solution space |
| 27 | braille character | | problem space |
| 28 | code to character map | 4,7 | solution space |
| 29 | hash table | 4,7 | solution space |
| 30 | text file | 1,2 | problem space |
| 31 | braille image | 1,2 | problem space |
| 32 | image processing | 3 | solution space |
| 33 | braille code identifier | 3,7 | solution space |
| 34 | text processing | 3 | solution space |
| 35 | step | | problem space |
| 36 | braille reader | 3,6 | solution space |
| 37 | algorithm | 3 | solution space |
| 38 | parameter | 7 | solution space |
| 39 | pdf file | 2 | solution space |
| 40 | braille processing system | 4,7 | solution space |
| 41 | word generator | 4,7 | solution space |
| 42 | user | 1,4 | solution space |
| 43 | system | 2 | problem space |
| 44 | customized template | 4,7 | solution space |
| 45 | code to text converter | 3, 7 | solution space |
| 46 | bangla dictionary | 2 | solution space |

## 6.3 Selection Characteristics

Coad and Yourdon suggest six selection characteristics that should be used to consider each potential class for inclusion in the analysis model:

1.Retained Information
2. Needed Services
3. Multiple Attributes
4. Common Attributes
5. Common Operations
6. Essential Requirements

Table 8: selection characteristics for selecting classes

| Selection Characteristics | | | |
|---|---|---|---|
| **No.** | **Potential Class** | **Characteristics** | **Verdict** |
| 1 | pattern recognition | 1,2,6 | No |
| 2 | image enhancement process | 1,2 | No |
| 3 | noise reduction algorithm | 1,2 | No |
| 4 | bangla dictionary | 2,6 | Yes |
| 5 | connectivity improvement | 6 | No |
| 6 | braille reader | 1,2,6 | Yes |
| 7 | braille to text | 1,2,3,4,5,6 | Yes |
| 8 | code to character map | 1,2,6 | No |
| 9 | image processing | 1,2,4,5,6 | Yes |
| 10 | spell checking | 1,2 | No |
| 11 | cleaning procedure | 1,2 | No |
| 12 | braille processing template | 1,2 | No |
| 13 | word generator | 1,2,4,6 | Yes |
| 14 | braille code identifier | 1,2,4 | Yes |
| 15 | code to text converter | 1,2,4,6 | Yes |

| | | | |
|---|---|---|---|
| 16 | text process | 1,2,4,6 | Yes |
| 17 | braille pattern | 6 | No |
| 18 | computer | 6 | No |

## 6.4 Attribute Selection

Here we find attributes for selected classes

Table 9: Attribute Selection Table for Classes

| **Attribute Selection** | | |
|---|---|---|
| **No.** | **Class** | **Attributes** |
| 1 | BrailleToText | scannedImage |
| 2 | ImageProcessor | scannedImage |
| 3 | BrailleReader | binaryImage |
| 4 | BrailleCodeIdentifier | binaryImage |
| 5 | CodeToTextConverter | braillePatternCode |
| 6 | TextProcess | braillePatternCode |
| 7 | WordGenerator | braillePatternCode |
| 8 | BanglaDictionary | NULL |

## 6.5 Defining Methods

In this section we find all the verbs from usage scenario and include necessary external verbs in a list. Then we select useful verbs as methods.

## 6.6 Verb List

Here we list all verbs from the usage scenario.

Table 10: Verb List from user scenario

| Verb List | | |
|---|---|---|
| No. | Verb | Remark |
| 1 | take scanned image | out of scope |
| 2 | loadImage | yes |
| 3 | translate braille to text | yes |
| 4 | view image | yes |
| 5 | scan paper | out of scope |
| 6 | save file | Yes |
| 7 | select image | out of scope |
| 8 | run image enhancement | yes |
| 9 | print | yes |
| 10 | remove unwanted portions | yes |
| 11 | apply noise reduction | yes |
| 12 | convert to binary image | yes |
| 13 | read braille | yes |
| 14 | identify line | yes |
| 15 | convert braille pattern to code | yes |
| 16 | map code to character | yes |
| 17 | generate plain text from code | yes |
| 18 | check valid braille character | yes |

|  |  |  |
|---|---|---|
| 19 | store word | out of scope |
| 20 | read line | yes |
| 21 | automate whole process | no need |
| 22 | check operator | yes |
| 23 | remove file | yes |
| 24 | check joint letter | yes |
| 25 | check number | yes |

## 6.7 Selected Methods

From the verb list above we have selected the following methods for classes.

Table 11: Class Card for Braille To Text

| **BrailleToText** |  |
|---|---|
| Interface<br>ImageFile | loadImage()<br>ViewImage()<br>Remove()<br>ConvertToText()<br>SaveFile()<br>Print() |
| **Responsibility** | **Collaborator** |
| loads an scanned image | User/Scanner |
| Creating and showing interface | System |
| converts to text file | CodeToTextConverter |
| prints output files | Printer |
| view or remove output files | NULL |

Table 12: Class Card for Braille Reader

| **Braille Reader** | |
|---|---|
| imageFile | readFullBraille()<br>readSingleLine() |
| **Responsibility** | **Collaborator** |
| scans full braille image and reads every lines | System |

Table 13: Class Card for Code To Text Converter

| **CodeToText Converter** | |
|---|---|
| text(braille pattern code) | generateText()<br>saveTextFile() |
| **Responsibility** | **Collaborator** |
| generates text file of braille | TextProcess |
| saves output text file | NULL |

Table 14: Class Card for Image Processor

| **Image Processor** | |
|---|---|
| Image | imageEnhancement()<br>noiseReduction()<br>removeUnwantedPortions()<br>convertToBinaryImage() |
| **Responsibility** | **Collaborator** |
| enhances image | NULL |
| filters and reduces noise from image | NULL |
| produces binary image | NULL |

Table 15: Class Card for Braille Code Identifier

| **Braille Code Identifier** | |
|---|---|
| text | readLine()<br>checkValidCharacter()<br>generateBrailleCodePattern() |
| **Responsibility** | **Collaborator** |
| identifies all lines in a braille text | NULL |
| generates braille pattern codes | NULL |

Table 16: Class Card for Text Process

| **Text Process** | |
|---|---|
| text(braille pattern code) | characterMapping()<br>generate_word()<br>generate_sentence() |
| **Responsibility** | **Collaborator** |
| generates bangla words and sentences | Word Generator |

Table 17: Class Card for Bangla Dictionary

| **Bangla Dictionary** | |
|---|---|
| text(braille pattern code) | getAlphabet()<br>getVowel()<br>getConsonant()<br>getPunctuation()<br>getNumber()<br>getOperator()<br>getKar() |
| **Responsibility** | **Collaborator** |

| | |
|---|---|
| matches the braille pattern code and returns the corresponding character | NULL |

Table 18: Class Card for Word Generator

| **WordGenerator** | |
|---|---|
| text | check_number()<br>check_operator()<br>check_jointLetter()<br>check_punctuation()<br>check_operator()<br>check_alphabet |
| **Responsibility** | **Collaborator** |
| generates bangla word | BanglaDictionary |

## 6.8 Class Diagram

In this stage we designed class diagram in the Unified Modeling Language. This is a type of static diagram to describe the structure of our system. Here we also designed two individual design for our two subsystems.

### 6.8.1 Class Diagram of Bengali Braille to Text Translator

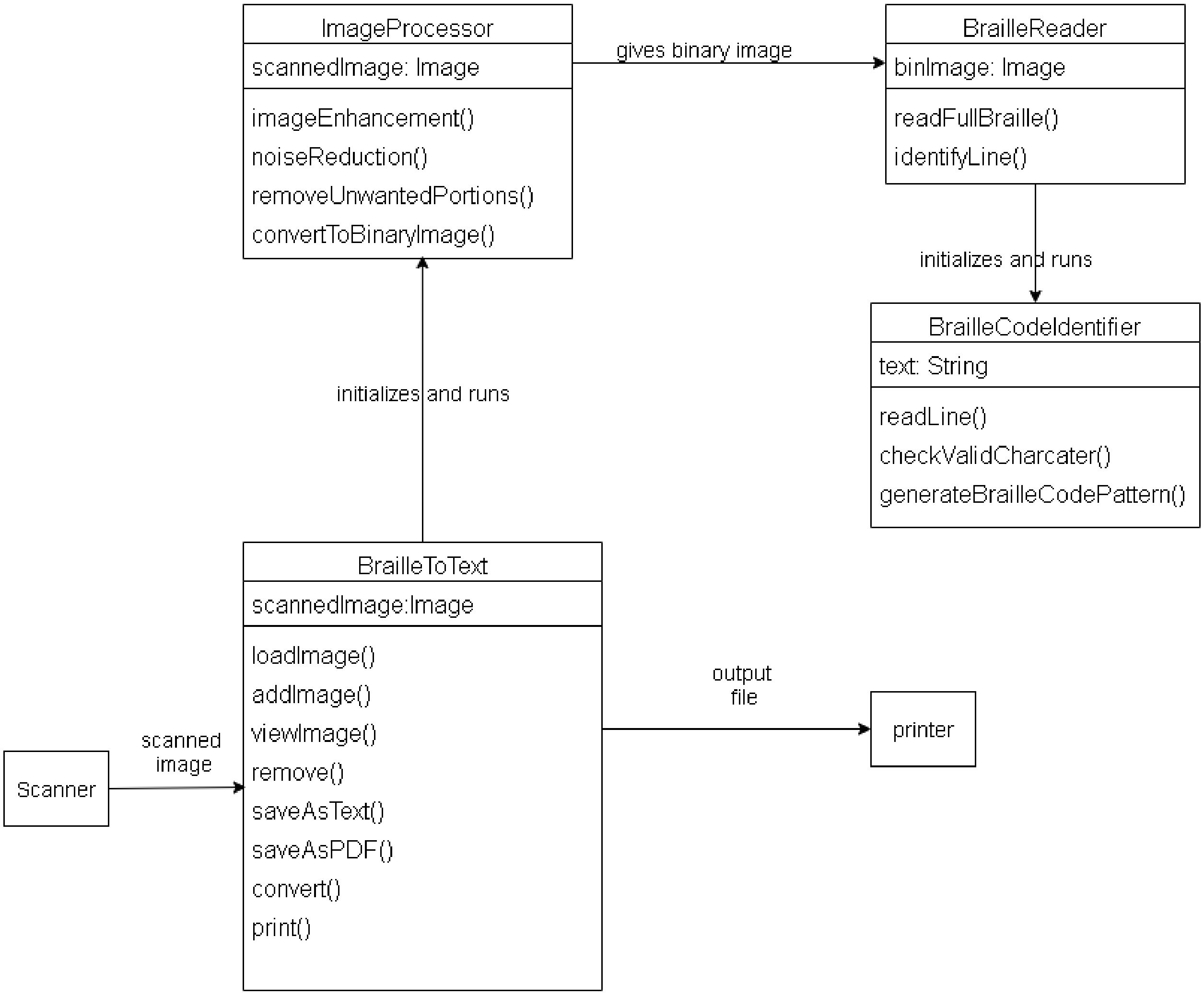


Figure 11: Class diagram of Bengali Braille to Text Translator

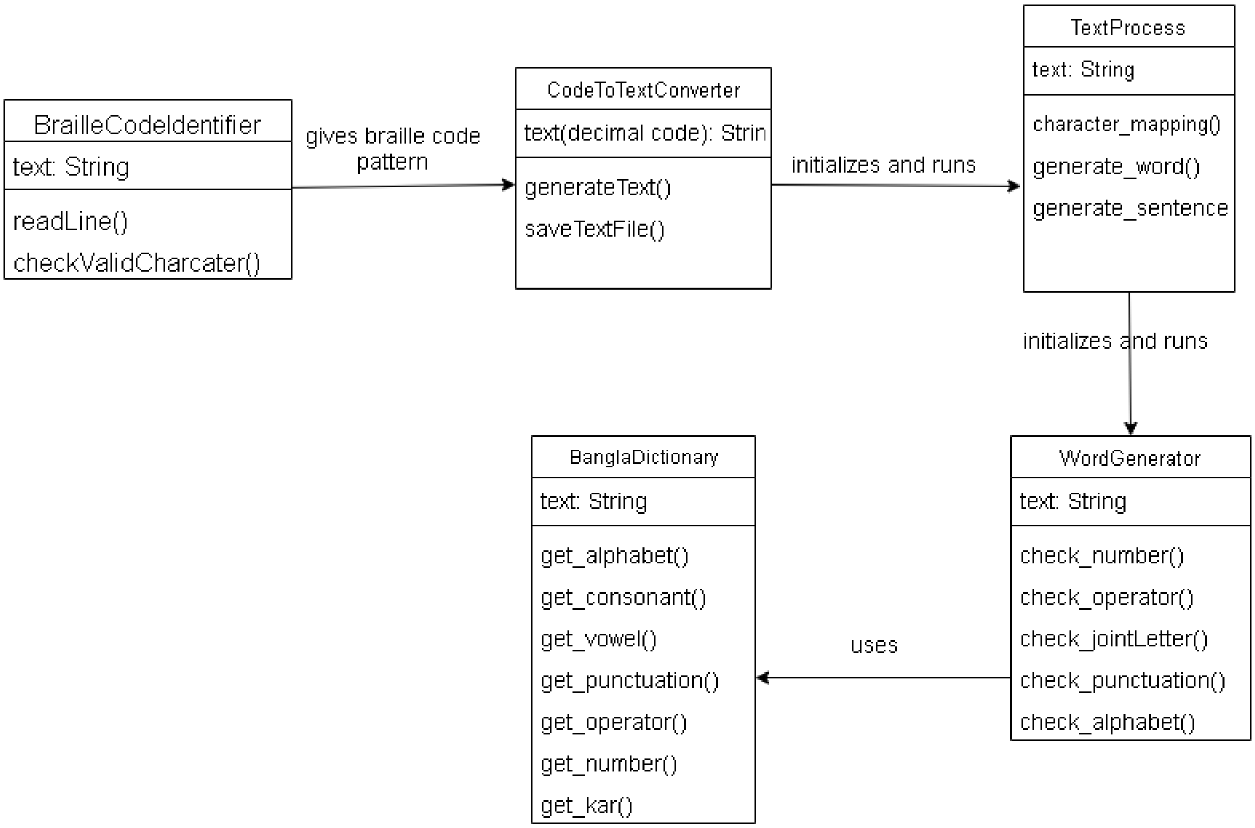


Figure 12: Class diagram of Bengali Braille to Text Translator

# Chapter 7

## Behavioral Model

### 7.1 Introduction

When the system is perceived in terms of states and transitions, it is called Behavior Modeling. It is also known as State Modeling, State Machines, or State Transition Matrix. This requires both identifying all of the interesting states of being that software or its components are likely to be in. And also, at a high level, abstracting what events are likely to cause software or its components to change between states of being.

### 7.2 Identifying Events

Here we have identified events from the Usage Scenario and listed their corresponding initiators & collaborators.

Table 19: List of Events

| Identifying Events | | |
|---|---|---|
| **Event** | **Initiator** | **Collaborator** |
| take scanned image of braille paper | User | Scanner |
| run different types of image processing technique | User | |
| translate braille to text through pattern recognition | User | Braille Pattern Code,<br>Code To Character Map |
| apply text correction procedures | User | Bangla Dictionary |
| run image | User | |

| | | |
|---|---|---|
| enhancement | | |
| apply noise reduction | User | |
| improve connectivity | User | |
| convert image to binary image | User | Image Pre Processor |
| run post processing technique | User | |
| check joint letter | User | |
| extract braille lines | User | |
| convert braille to code | User | Braille Pattern Code |
| map code to character | User | Bangla Dictionary |
| store result | User | System |
| remove unwanted portions from image | User | |
| remove image | User | |
| remove output file | User | |
| print output file | User | Printer |

## 7.3 State Transition Diagram

State Transition Diagram represents active states for each class and the events (triggers) that cause changes between these active states.

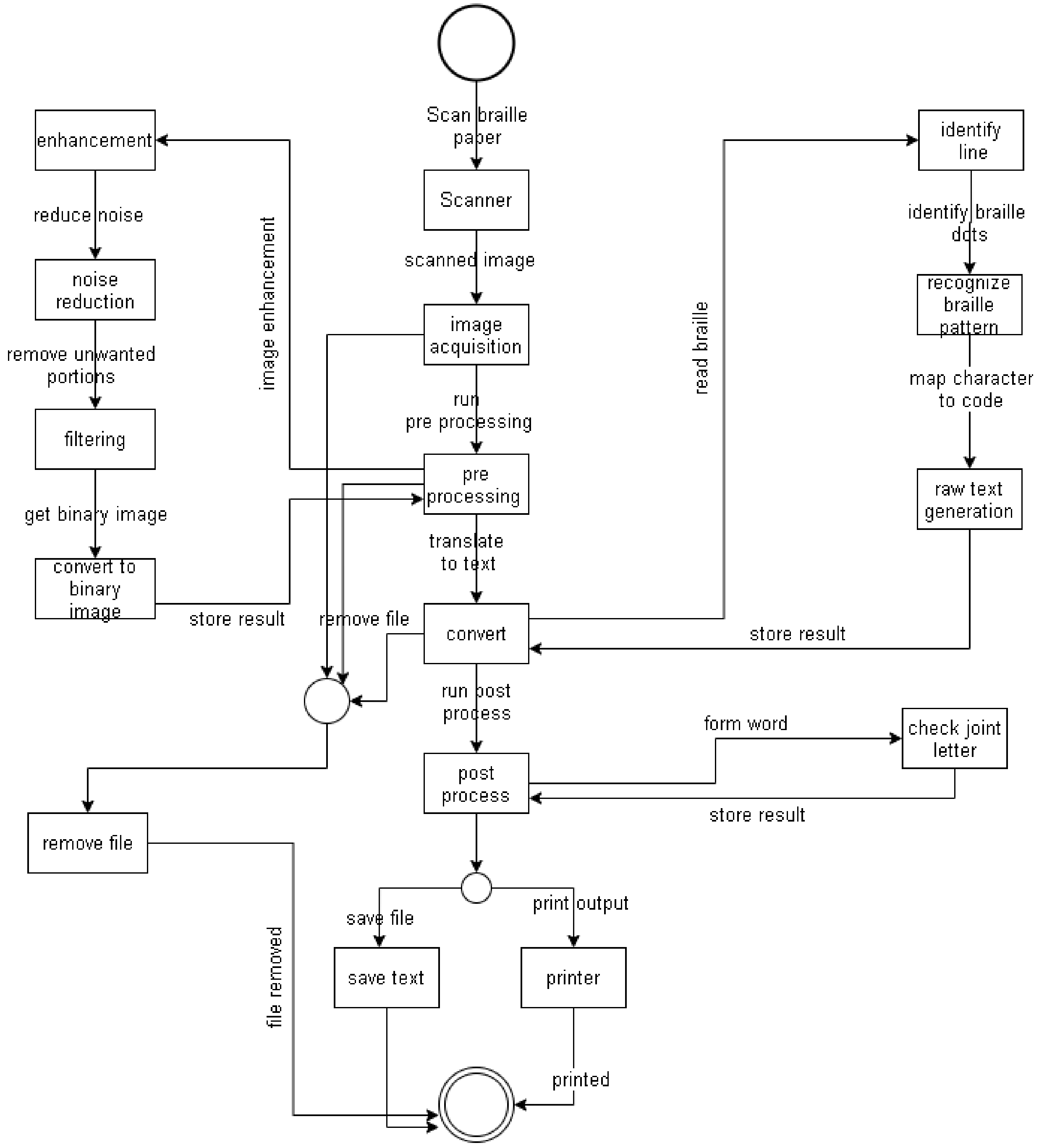


Figure 12: State transition diagram of Bengali Braille to Text Translator

## 7.4 Data Flow Modeling

We intended this chapter to describe data flow modeling for our “Complain Box”.

A data flow diagram is a graphical representation of the flow of data through an information system. We use data flow diagrams to diagrammatically represent the flow and exchange of information within our “Bengali Braille to Text Translator”. As in the previous chapter, we modeled our data flow diagram based on our two main sub systems.

## 7.5 Data Flow Diagram

We did try to go from the initial level to deep level in our system through our data flow diagram.

### 7.5.1 Level 0 Data Flow Diagram of Bengali Braille to Text Translator

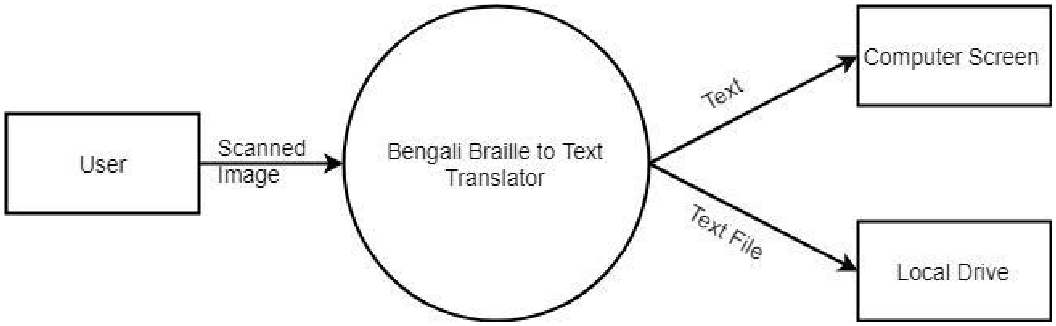


Figure 13: Level 0 data flow diagram of Bengali Braille to Text Translator

### 7.5.2 Level 1 Data Flow Diagram of Bengali Braille to Text Translator

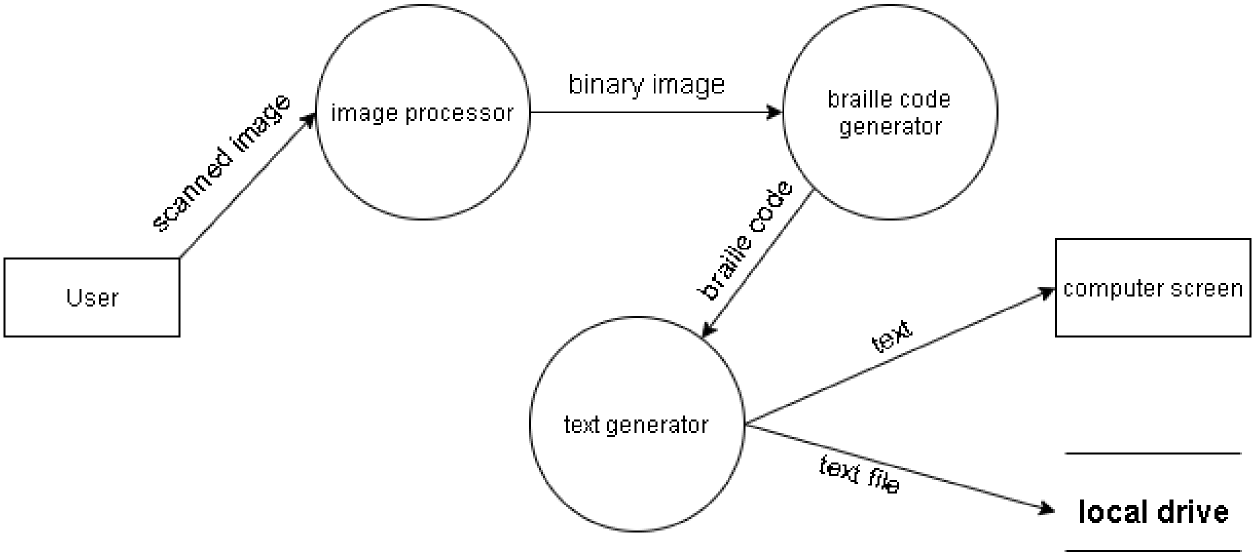


Figure 14: Level 1 data flow diagram of Bengali Braille to Text Translator

# Chapter 8

## Conclusion

We are pleased to submit the final SRS report on Bengali Braille to Text Translator. From this, the readers will get a clear and easy view of tactile writing system. They will also get a good understanding of the translation process.

This SRS document can be used effectively to maintain the software development cycle for the project. We have presented a detailed description of the total system. It will be much easy to conduct the whole project using this SRS. It will also help me to determine the pitfalls that may come ahead. Hopefully, this document can also help other software engineering students as well as practitioners.

We have tried our best to make effective and fully designed SRS in a short time. We wish, the readers will find it in order.

# Bibliography

[1] "Digital Image Processing Basics," 20 October 2020. [Online]. Available: https://www.geeksforgeeks.org/digital-image-processing-basics/.

[2] "Grayscale to RGB Conversion," tutorialspoint, [Online]. Available: https://www.tutorialspoint.com/dip/grayscale_to_rgb_conversion.htm. [Accessed 20 October 2020].

[3] “Otsu’s Thresholding”[Online]. Available: https://www.learnopencv.com/otsu-thresholding-with-opencv/ [Accessed 20 October 2020].

[4] "Braille letter recognition and transcription" [Online]. Available: https://www.researchgate.net/publication/291331547_Computational_model_for_Bangla_text_to_Braille_translation. [Accessed 20 October 2020].