Software Requirement Specification

Bengali Braille to Text Translator

Course: SE – 801 Software Project Lab – 3

Submitted by

Atiq Ahammed BSSE0817
Institute of Information Technology
University of Dhaka

Supervised by

Dr. Ahmedul Kabir
Assistant Professor
Institute of Information Technology
University of Dhaka

Submitted To

SPL – 3 Program Committee
Institute of Information Technology
University of Dhaka

Document Version: 1.0
Date: 01 – 09 – 2019

Institute of Information Technology

University of Dhaka.

Letter of Transmittal

01-09-2019
Institute of Information Technology
University of Dhaka.

Subject: Submission of software requirements specifications of Software Project Lab 2.

Dear Sir,

I have prepared the report on Software Requirements Specification of Bengali Braille to Text Translator.

The primary purpose of this report is to summaries my findings that I have gathered during the requirements specification process. This report also includes details of each step I have followed while collecting the requirements.

Although this report may have short comings, we did try our level best to produce an acceptable software requirement specification. We would be highly obliged if you overlooked our mistakes and accepted our effort we put in this SRS.

Sincerely yours,

____

Atiq Ahammed
BSSE 0817
Institute of Information Technology
University of Dhaka

____

Dr. Ahmedul Kabir
Assistant Professor
Institute of Information Technology
University of Dhaka

Acknowledgement

By the grace of Almighty Allah, I have completed my report on Software Requirements Specification of Bengali Braille to Text Translator.

I am grateful to my supervisor Dr. Ahmedul Kabir for his direction throughout the working time. It was almost impossible for me to complete this SRS document without him.

I am also thankful to the teachers and students of The Institute of Education and Research, University of Dhaka. They greatly helped me in collecting information among all business.

Abstract

This document contains the software requirements and specifications for the Software Project Lab – III. It contains a scenario-based model, data-based model, class-based model, data flow and behavioral model. Using this document as a guide, we are describing the requirements, necessary diagrams, procedures, design for database and working sequence of our project.

This is a project of a desktop application that will be develop to convert the braille code from a scanned image to corresponding Bengali text. Here I will discuss how I will identify the requirements, how to analyze them and how to present a recommended solution for the system.

This will help to make the software according to the demand of the stakeholders.

Contents

Tables

Figures

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 – 3 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 me, my 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 – 3 coordinators and ours.

- Teachers and students of will use this SRS to verify that we have developed a product that the required
- My supervisor will use this SRS to plan milestones and ensure that we are on the right track when developing the system
- I will use this SRS as a basis for creating the system design. I 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:

- Full control of the system
- Minimum error rate
- Handing batch of images
- 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 customize and control the system
- Users with no technical knowledge will be able to use templates (created by specialists) 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.

3.2.1 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.

3.2.2 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-

3.2.3.1 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.

- User can customize the work sequence, and store the configuration for future use.

- Will be able to run the tool and get output with only single ‘run’ button.
- User can view result from each level of Braille to text translation process.
- User will be able to run different types of image enhancement and other pre-processing algorithms to improve input image quality.
- There will be some default configuration templates available with the software.

3.2.3.2 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 both single image and batch of 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.

3.2.3.3 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.

3.2.4 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. Then it will provide the Bengali text that is written in the scanned input image.

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.

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.

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.

Braille Cells and Dots Framing:

The framing process to determine the cells, words, and lines in the scanned image based on standard dimensions of Braille documents. Mainly statistics measures of central tendency will be used for performing this task.

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.

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

Bangla letter ‘ক’ → ⠅ (One-to-one)
Bangla letter ‘হ’ & digit ‘৮’ → ⠓ (Many-to-one)

Each letter of Bangla alphabet, numeric and punctuation has corresponding Braille representation. Bangla letters and their corresponding Braille representations 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	8	⠙
ঋ	ৃ	⠐ ⠗	ফ	৬	⠋
ও	ো	⠕	ব	২	⠃
ঔ	ৌ	⠪	হ	৮	⠓
গ	৭	⠛	ৎ	”	⠴
(	)	⠶	?	“	⠦

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.

Braille word recognition and transcription

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.

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

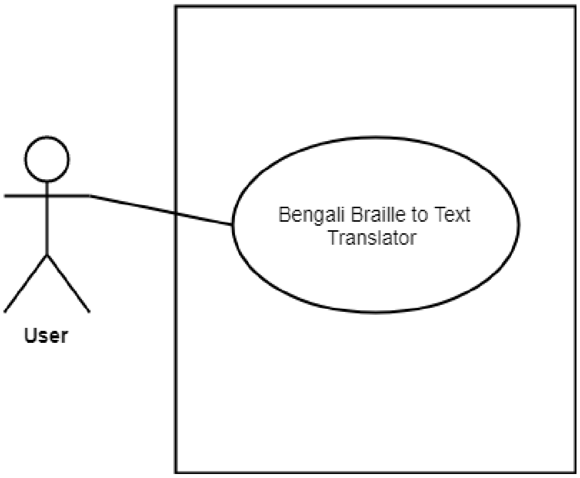

Figure 1: Level 0 use case diagram of Bengali Braille to Text Translator

Table 1: Information about level 0 use case diagram

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

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

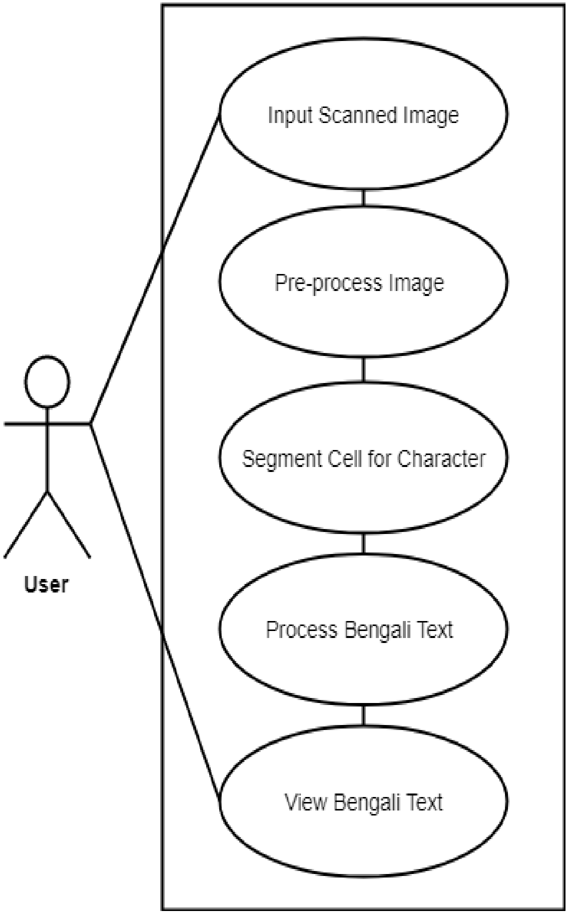

Figure 2: Level 1 use case diagram of Bengali Braille to Text Translator

Table 2: Information about level 1 use case diagram

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

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 and get corresponding Bengali text.

4.4.3 Level 1 Activity Diagram of Bengali Braille to Text Translator

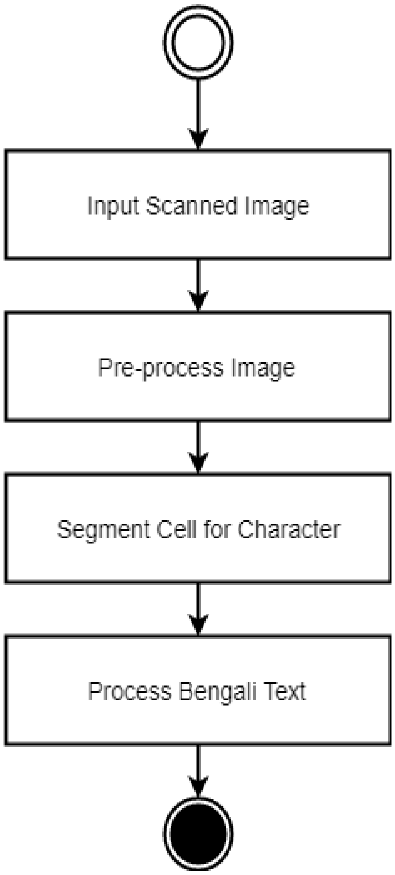

Figure 3: Level 1 activity diagram of Bengali Braille to Text Translator

Chapter 5: Class Based Modeling

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

5.1 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.

5.2 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.

5.2.1 Class Diagram of Bengali Braille to Text Translator

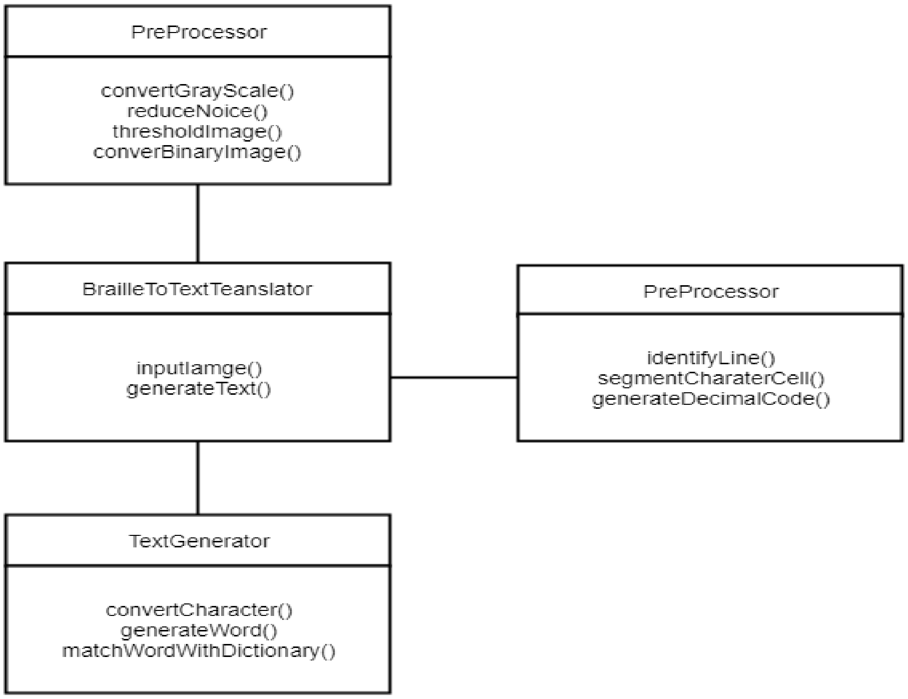

Figure 3: Class diagram of Bengali Braille to Text Translator

Chapter 6: Data Flow Modeling

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

6.1 Introduction

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

6.2 Data Flow Diagram

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

6.2.1 Level 0 Data Flow Diagram of Bengali Braille to Text Translator

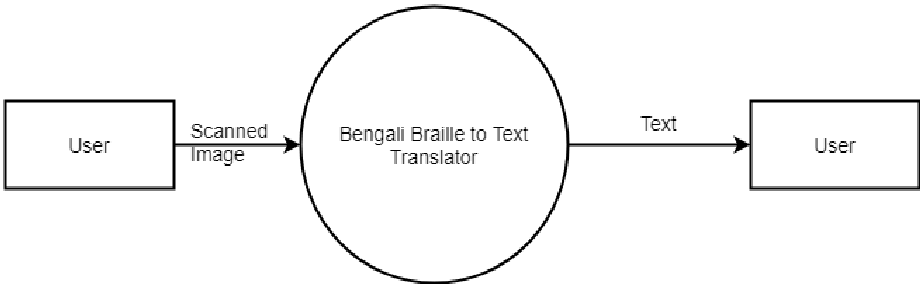

Figure 4: Level 0 data flow diagram of Bengali Braille to Text Translator

6.2.2 Level 1 Data Flow Diagram of Bengali Braille to Text Translator

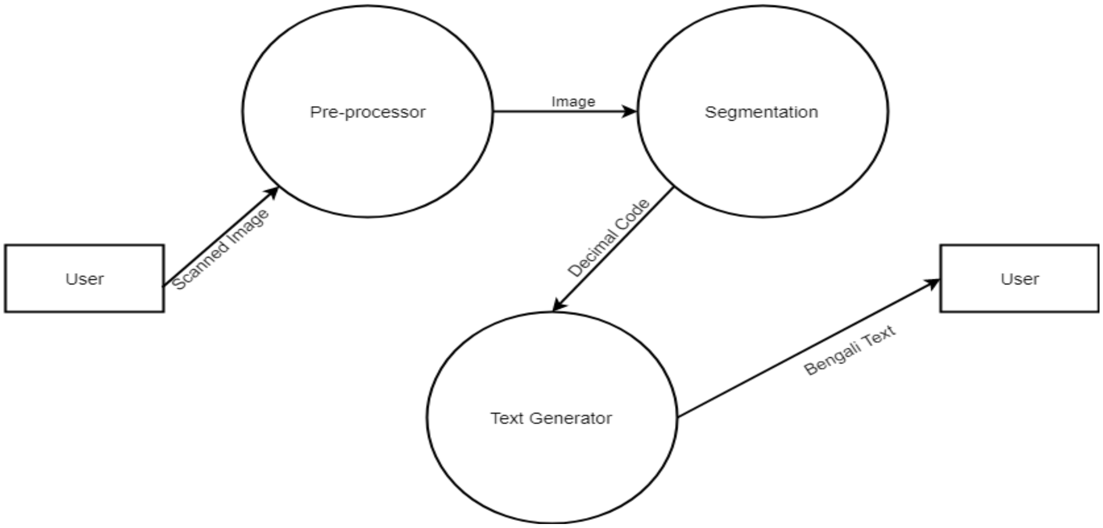

Figure 5: Level 1 data flow diagram of Bengali Braille to Text Translator