Software Requirements Specification

Project ID: 15J-004

Sri Lanka Institute of Information Technology

B.Sc. Special (Hons) Degree in Information Technology

Specialized in Information Technology

Date of Submission:

Title: Arithmetic Problem Solving System for Grade 5 Students using neural networks

Project ID: 15J-004

Author:

Student ID	Name	Signature
IT 12082148	A.M Raddella	

.....

Mr.Yashas Mallawarachchi

Supervisor

.....

Mr.Anupiya Nugaliyadde

Co-Supervisor

Declaration

I hereby declare that this proposal document or part of it is not a copy of a document done by any organization, university, institute or a previous student project group at SLIIT and is not copied from the Internet or other sources except where stated.

Student ID	Name	Signature
IT 12082148	A.M Raddella	

Contents

1.INTRODUCTION

1.1 Purpose

This document addresses different areas of the software under different sections and provides comprehensive information regarding the product. “Mahoshadha 2” is an intelligent system which can solve arithmetic mathematical problems in Sinhala language. It formally specifies the proposed system’s functional requirements, non-functional requirements, data requirements, quality requirements and constraints. This document will help system developers to understand the overall functionality of the system. By reading this document system users can get a clear idea of systems behavior and how to interact with the system.

This document mainly discuss the features under ‘Question Identification’ field.

The intended audiences of this System Requirement Specification are, the members of the research group, project supervisor, Mr. Yashas Mallawarachchi, project co – supervisor Mr. Anupiya Nugaliyadda , Project team-

1.2 Scope

The main characteristics and behavior of the system is firmed to attest a user-friendly, effective and efficient system. This system is intended to be used by grade 5 scholarship students who are not interesting to solve arithmetic problems while studying mathematics at the school. This APSS is capable of understanding a detailed mathematical sentence problem in Sinhala language.

1.3 Definitions, Acronyms, and Abbreviations

Term	Definition
NLP	Natural Language Processing
CRF	Conditional Random Fields
QA	Question Answering
APSS	Arithmetic Problem Solving System
NN	Neural Network

Table 1:Definitions, Acronyms, and Abbreviations

1.4 Overview

The main goal of "Mahoshadha 2" is solve the arithmetic mathematical problems in Sinhala language. This system works as a tutor for grade 1 to 5 students. In APPS understand the problem, then solve it and finally simulate the answer how it solved. Most of the QA systems are designed only for the information retrieval. But "Mahoshadha 2" System can solve Arithmetic mathematical problems automatically.
This system have mainly four parts.

1. Key Word Identification
2. Question Identification
3. Identifying the Mathematical Operation
4. Neural Network

Providing the user friendly interfaces is very important when deal with child. In "Mahoshadha 2" we provide the attractive interfaces. In this APSS we try to cover all the areas in scholarship syllabus deal with addition and subtraction.

2. Overall Description

2.1 Product perspective

According to this APSS, once the mathematical problem is inserted to the system, system breaks down the problem in to sentences. Key word identification is done for each sentence. Question identification is done in parallel to the key word identification. Identifying the required mathematical operation and performing it is done as the next step. Then the neural network will output the correct answer. Finally the answer will be displayed with simulations to the child. It is important to show the child that how the problem is solved in an attractive manner that will encourage the child to attempt in to this subject even more.

Most of the time students are considered mathematics is difficult subject. The intend of the "Mahoshadha2" system is to make it easy for them.
One of the most critical parts of this system is 'Question Identification' that is discussed under this document. Under question identification we do

- Chunk the question in to words.
- Tag each word.
- Identify the "question words" (what, how much,.. etc..)
- Pass the output to the neural network.

2.2 System Interfaces

To communicate with the system we have to use attractive interfaces

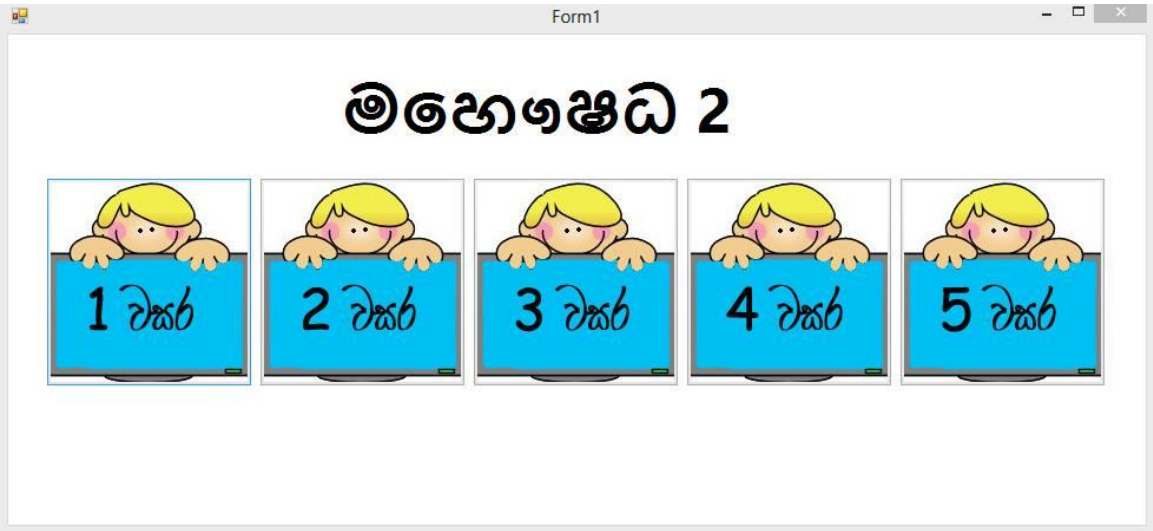

Figure 2:Main Interface

Figure 1: Interface2

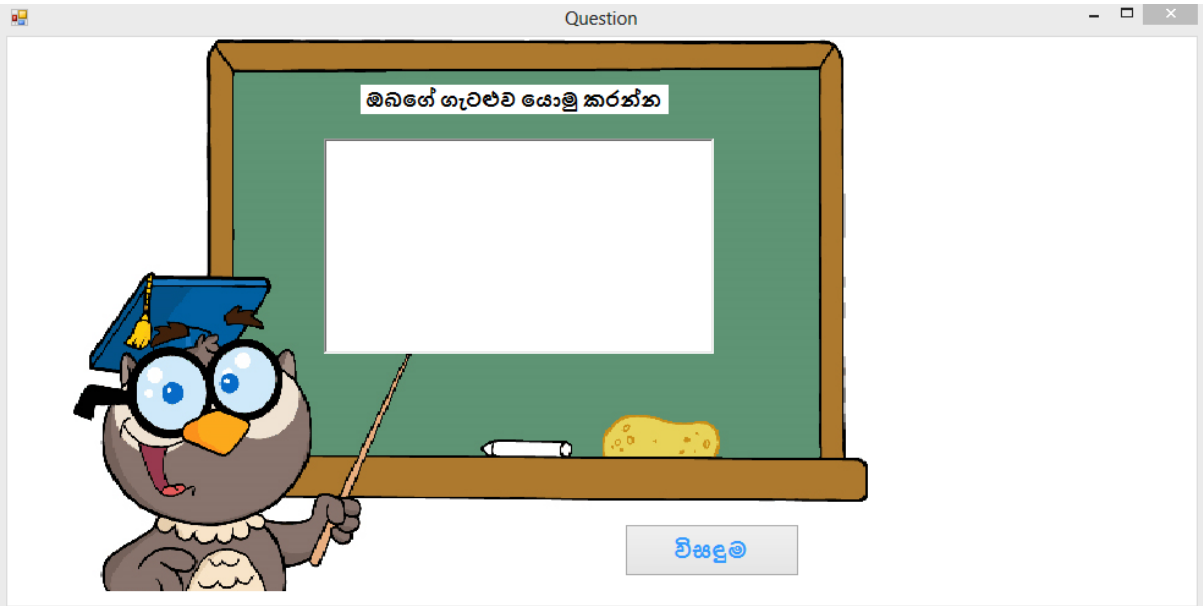

Figure 3:Interface 3

2.3 Product functions

- Identifying key words.
- Identifying the question.
- Generating the equation for a problem.
- Producing the final answer for a given problem.
- Checking the answers given by the user.
- Handle inputs through hidden layers.
- Passing output of a hidden layer to another layer.

2.4 Use case Scenarios

2.4.1 Taking inputs for a hidden layer

Use Case name	Handle inputs in hidden layers
Actor	User , system
Pre-Condition	System is working, outputs have been generated by other functions
Success Scenario	1. Use case scenario starts when the neural network receives keywords to the first hidden layer. 2. Weights are assigned. 3. Sigmoid activation function is applied to weights. 4. Threshold the results. 5. Get the values that exceed the threshold. 6. Outputs are sent to next hidden layer.
Exceptions	5a. if no input value pass through the threshold, layer takes next set of inputs
Post-Conditions	Output must be sent to the next layer.

Table 2:Taking inputs for a hidden layer

2.4.2 Insert a new question to the database

Use Case name	Insert a mathematical problem to the database
Actor	User , system database
Pre-Condition	System is working properly
Success Scenario	1. Use case scenario starts when the user press “add new question” button 2. User is given a text area to insert new question. 3. User enters the word problem. 4. User press enter button.
Exceptions	-
Post-Conditions	Update the database.

Table 3:Insert a new question to the database

2.4.3 Key word extraction.

Use Case name	Key word extraction
Actor	User , system
Pre-Condition	System is running, arithmetic question is received from the user
Success Scenario	1. The use case starts when the system received a arithmetic question 2. Token extraction from given question. 3. Extracting nouns, verbs, adjectives …. etc. 4. Calculate the properties of each candidate. 5. Select possible keywords.
Exceptions	4a. if system get different values for the calculation get the highest value among them. 4b. if system get negative values for the calculation omit that words.
Post-Conditions	System has successfully extracted required keywords

Table 4:Key word extraction.

2.4.4 Identifying Mathematical Operation and Generate the equation

Use Case name	Identifying Mathematical Operation and Generate the equation
Actor	User , system
Pre-Condition	System is running, key words are extract from the question, Question identification is done
Success Scenario	The use case starts when the system received extracted key words and question identification is done. 2. Forming the states. 3. Match the entities and categories. 4. Initializes or updates the values of the containers in the state. 5. Quantities of entities update or observe. 6. Using the learned verb categories find the relevant mathematical operation. 7. Forms an equation by comparing the quantities for containers matched between the two states.
Exceptions	3a. An entity/category is matched if it has the same head word and same set of attributes as an existing entity/category. 3b. If an entity or category cannot be matching, then a new state is created. 4a. For the matched entities, initializes or updates the values of the containers in the state.
Pre condition	

Table 5:Identifying Mathematical Operation and Generate the equation

2.4.5 Question Identification

Use Case name	Question Identification
Actor	User , system
Pre-Condition	System is running, arithmetic question is received from the user
Success Scenario	1. The use case starts when the system received an arithmetic question 2. Word chunking for the received question 3. Tagging the words 4. Identifying the question words(what, howmany,howmuch etc..) 5. Identify the question
Exceptions	
Post-Conditions	System has successfully identified the question

Table 6:Question Identification

2.5 Hardware interfaces

- Hard disk 2GB free space
- Dual Core processer or above
- Monitor
- Key board
- Mouse

2.6 Design Constraints

Under this section the constraints of the system design are discussed.

2.6.1 Nonfunctional Requirements

2.6.1.1 Performance Requirements

The only way in which systems will meet their performance targets is for them to be specified clearly and unambiguously. It is a simple fact that if performance is not a stated criterion of the system requirements then the system designers will generally not consider performance issues. The performance of the product must be to the highest level, since lot of mathematical questions are handled through the system. Since the system consists of lot sub modules coming up with a suitable design specifically concentrating on performance wise is very important.

2.6.1.2 Safety Requirements

Safety consistent with mission requirements, is designed into the software in a timely, cost effective manner. The system will always be confirmed to run properly and give the correct answer effectively, therefore there is no risk of any data losses. Further data retrieval is also efficient.

2.6.1.3 Security Requirements

Security is often an afterthought during software development. Realizing security early, especially in the requirement phase, is important so that security problems can be tackled early enough before going further in the process and avoid rework. According to the user privileges of the system, user can perform his /her relevant tasks which are assigned to them by the system.

References

[1] http://www.mathworks.com/matlabcentral/answers/42738-how-to-create-a-dataset-as-input-to-a-neural-network-for-a-character-recognition-system

[2] [2] "Learning to Solve Arithmetic Word Problems with Verb Categorization" (by Mohammad Javad Hosseini (University of Washington), Hannaneh Hajishirzi(University of Washington), Oren Etzioni(Allen Institute for AI), and Nate Kushman(Massachusetts Institute of Technology)

[3] Automatic Key phrase Extraction based on NLP and Statistical Methods

Martin Dostal and Karel Je_zek

Department of Computer Science and Engineering, Faculty of Applied Sciences

University of West Bohemia, Plze_n, Czech Republic, fmadostal, jezek kag@kiv.zcu.cz

[4] Automatic Keyword Extraction From Any Text Document Using N-gram Rigid Collocation
Bidyut Das, Subhajit Pal, Suman Kr. Mondal, Dipankar Dalui, Saikat Kumar Shome

International Journal of Soft Computing and Engineering (IJSCE) ISSN: 2231-2307, Volume-3, Issue-2

[5] Sinhala Ontology Generator for English to Sinhala Machine Translation

CONFERENCE PAPER · JANUARY 2014 Budditha Hettige General Sir John Kotelawala Defence Unive...George Rzevski The Open University (UK)