

A System to Suggest Meaningful Domain Names in the Domains.lk Search Bar

By R.A.D.W. De Silva 198747R

**A System to Suggest Meaningful
Domain Names in the
Domains.lk Search Bar**

R.A.D.W. De Silva
198747R

Faculty of Information Technology
University of Moratuwa
2022

Declaration

I declare that this thesis has been composed solely by myself and that it has not been submitted, in whole or in part, in any previous application for a degree. Except where stated otherwise by reference or acknowledgment, the work presented is entirely my own.

Name of the Student

Signature of the Student

UOM Verified Signature

R A D W De Silva

.....

Date:

Supervised by

Name of the Supervisor

Signature of the Supervisor

Dr. Chaman Wijesiriwardene

.....

Date:

Dedication

I dedicate my research work to my two daughters Sanuki and Shraddha. A special feeling of gratitude to my workplace, LK Domain Registry, for the initial idea of doing this research project and providing me with necessary support when it requires. I also would like to dedicate this research project to my family for allowing me to do this academic work.

Acknowledgements

First and foremost I am extremely grateful to my supervisor, Dr. Chaman Wijesiriwardhana for his invaluable advice, encouragement, continuous support, and patience during my research project. His academic experience in conducting such a technical research project was shared with me without any hesitation. I would also like to thank Prof. Gihan Dias and Ms. Roshini Peters for their continuous support on this project and timely guidance on conducting this project. Last but not the least I would like to thank Mr. Dinushika Dissanayake for his technical support given to this project throughout the period. I should also thank my family for all their support during this period to make this project a success. Without the tremendous understanding, support and encouragement of all the above-mentioned people in the past few months, I have been able to complete my study.

Abstract

For any given domain registry the domain searching feature is an integral part of the domain registration process. The Initial user interaction with the domain registry would start with an availability checkup using this domain searching feature. If that is not providing a satisfactory level of clarity about the availability of the domain or the other options a customer can select, the domain searching feature is a failure itself. Most global domain registrars have sophisticated domain searching features while CCTLD (Country Coded Top Level Domain) registries find it difficult to create ideal domain searching features as they are facing the native language problem. Since the domains are typed in using English characters so a transliteration will also be required to search the domain. This ambiguity is blocking the development of an ideal algorithm to give closest suggestions for domains in native language. In this project we are trying to find a solution for that while implementing it with the domain search.

Furthermore effective utilisation of customer searches is lacking with these CCTLD registry systems. Global giants in the domain registry business use this information effectively to change prices dynamically and earn lots of profits by selling premier domains to its customers. Since CCTLD is lacking such a feature in their domain search they had to sell all the domains under basic categorization and they potentially lose much profits by selling premier domains under wrong categories. Storing the domain searches customers have done, analysing them and then generating an algorithm to categorise them would be the best way to go ahead with developing such a feature in the domain search bar. All these functionalities would be associated with developing a sophisticated domain searching feature for a CCTLD.

Table of Contents

Chapter 1	1
Introduction	1
1.1 Prolegomena.....	1
1.2 Problem Statement	2
1.3 Aims and Objectives	4
1.4 Background and Motivation.....	4
1.5 Problem in brief	5
1.6 Proposed Solution.....	5
1.7 Structure of the thesis	6
Chapter 2	7
2.1 Literature Review	7
2.2 Reviews of Other's work.....	9
Chapter 3	12
Technology Adopted.....	12
3.1 Introduction	12
3.2 Programming Languages	12
3.3 Server Technologies	13
3.4 Libraries	13
3.5 IDEs.....	14
3.6 High Level Architecture of the Technologies used.....	14
3.7 Summary	15
Chapter 4	16
Novel Approach to generate domain name suggestions.....	16
4.1 Introduction	16
4.2 System.....	16
4.3 High Level Architecture of the System.....	16
4.4 Users	17
4.5 Inputs	18
4.6 Process.....	20
4.7 Outputs	21
4.8 Summary	22
Chapter 5	23

Analysis and Design	23
5.1 Introduction	23
5.2 High-level Architecture of the Overall System	23
5.3 High-level Architecture of the Individual Modules	24
5.4 Summary	31
Chapter 6	32
Implementation	32
6.1 Introduction	32
6.2 Caching Process	32
6.2 Implementation of Singlish Translator	33
6.3 Dictionary	38
6.4 Similar Domain Names Generator	38
6.5 Feedback Module	39
6.6 Summary	39
Chapter 7	40
Evaluation	40
7.1 Introduction	40
Chapter 8	41
Conclusion and Further Work	41
8.1 Introduction	41
8.2 Limitations	42
8.3 Future Developments	42
8.4 Summary	43
References	43

List of Figures

FIGURE 3.6 HIGH LEVEL VIEW OF THE TECHNOLOGIES	15
FIGURE 4.3 APPROACH FOR ENGLISH WORDS.....	17
FIGURE 4.3 APPROACH FOR SINGLISH WORDS	17
FIGURE 4.4 PREDEFINED DOMAIN NAME SUGGESTIONS	18
FIGURE 4.5 USER INTERFACE	19
FIGURE 4.7 FINAL OUTPUT USER INTERFACE.....	21
FIGURE 5.2 OVERALL SYSTEM HIGH-LEVEL ARCHITECTURE	24
FIGURE 5.3.1.2 NEW ALGORITHM.....	27
FIGURE 5.3.2 SINHALA ENGLISH API.....	29
FIGURE 5.3.3 SIMILAR DOMAIN NAMES GENERATOR	30
FIGURE 5.3.4 FEEDBACK MODULE.....	31

List of Tables

TABLE 4.5 MAIN COMPONENTS	19
TABLE 4.7 MAIN MODULES.....	22
TABLE 5.2 OVERALL SYSTEM FINAL OUTPUT.....	23
TABLE 7.1 JMETER RESPONSES	41

Chapter 1

Introduction

1.1 Prolegomena

A domain name is the address where Internet users can access your website. Computers use IP addresses, which are a series of numbers that are difficult for humans to remember strings of numbers. Because of this, domain name system was developed and used to identify entities on the Internet rather than using IP addresses.

A domain name is typically a unique combination of letters and numbers, and it can be used in combination of the various domain name extensions, such as .com, .net global domains and Country coded top level domains assigned to each country.

The LK Domain Registry is the sole administrator for domain name that ends in “.lk” in Sri Lanka. As the national-level domain name (Country Coded Top Level Domain for Sri Lanka), a “.lk” domain provides Sri Lankan organizations and individuals with their unique brand identity on the Internet. LK Domain Registry register a wide range of domain names including the top-level domains .lk, .com.lk, .org.lk and .edu.lk.

In addition to English, domain names can also be registered in the Sinhala and Tamil language top-level domains. Its clientele spans all countries, with a majority of it resident in Sri Lanka.

[1]

As an old domain registry in Asia Pacific region, LK Domain registry has very low number of domain registrations. So they are in pursuit of registering as many domains as possible. So enabling the customers to do domain search much easier and sophisticated manner would definitely help them to get more domain registrations. And giving meaningful suggestions to domain search ideally help the customer to buy a domain which matches to their requirement in case the original domain they searched for is not available. Basically the LK Domain Registry will achieve the goal of registering a domain even the customer cannot buy their first choice domain due to unavailability.

Sri Lanka is a multilingual country where the majority is using Sinhalese as their mother tongue and Tamil is the other main native language. With the boom in IT related technologies and the usage of new devices triggered the need of using native language in English characters. Even the term ‘Singlish’ been used to define this new usage. Singlish is the term used to describe the

mixture of Sinhala and English which is frequently used by bilingual speakers in Sri Lanka [2]. So with the domain names there is a trend now to buy domains which has Sinhala sounding words. Technically the domain is then registered as the transliteration of the Sinhala word. Transliteration is the act or process of writing words using a different alphabet: and it helps people speak a language by showing the pronunciation in the language they understand. [3] A good example is ikman.lk domain name. They wanted a domain which says about a quick search, so they find the Sinhala word for quick which goes as “ඉක්මන” and transliterated to ‘ikman’. That became a great success in Sri Lanka.

If LK domain registry can encourage to follow the same practice people and businesses in Sri Lanka would surely register more domain names.

In addition to this point, at the moment LK Domain registry is not making any use from the thousands of domains searches happening through their domain registration feature. Upon saving them, they will have a massive number of domain preferences requested by customers. And utilizing some mechanisms to identify the popular keywords, patterns of buying domains it would help LK Domain Registry to re-categorize available domains to premier domains. Also it would help to make sensible suggestions (synonyms, plurals, joined with other keywords) for domain buyers.

Ex: if a person requests ‘book’ as the domain name and if it is not available, alternative suggestions such as books, ebooks, mybook, goodbook, greatbook etc may encourage the customer to buy an alternative.

1.2 Problem Statement

One of the main goals of LK Domain Registry Company is to increase the number of registered .lk domains to 1 million. Currently they have less than 100,000 domain registrations. In order to reach that goal it is really important to encourage people to register more domains. And let them know the importance of protecting their brand names digitally by registering all the related domain names.

At the point of domain search if the customer can see some matching domains to their main domain, there is a high possibility that the customer might register those similar domains. Getting the most accurate domain suggestions to the customer's domain search panel is the challenge here. We can bring the most important questions as follows,

Giving the right domain names as the suggestions provided by the domain search bar

In the current domain search bar, the user has to come up with all the domain name suggestions related to any keyword.

If a user trying to protect his brand, he/she has to reserve all related domain names for the brand name. In the case where the brand name is a generic name there can be synonyms, different meanings in Sinhala language etc. At the moment user has to think everything by themselves. But with this proposed system domain search bar will give all the suggestions.

Ex: if the brand is ‘Kulu badu’ main domain can be kulubadu.lk and following can be domains they need to book in order to protect the brand.

English domains - Spices.lk , condiments.lk Flavorings.lk, Seasonings.lk, herbs.lk

Sinhala domains - Thunapaha.lk, 3-5.lk, rasakaraka.lk

Ambiguities and complexities in translation and transliteration

The process of translation and transliteration can become very time consuming and resource intensive [4]. Sometimes ideas, concepts, and feelings might not always translate exactly from one language to another. It is even possible that in some cases there is apparent contradiction between valuing meaning, on the one hand, and a desire to obtain conceptual equivalence, on the other, which might present a real challenge to the novice. In such cases the whole research team might wish to discuss a particular section of the tapes and transcripts to arrive at the closest meaning [5]

There is no common standardization for transliterations in English to Sinhala or Sinhala to English to get clarifications on ambiguities.

Ex: English to Sinhala transliteration

‘Pa’ can be ‘ප’ or ‘පැ’

‘Pe’ can be ‘පෙ’ or ‘පේ’

Ex: Sinhala to English transliteration

‘පැ’ can be ‘Pa’ or ‘Pe’

Because of these ambiguities it has been very difficult to build a system with a standard.

1.3 Aims and Objectives

Considering the main problems we address in this research project we have to come up with “Give valuable suggestions to users to make their domain name choice accurate. Also to protect their brand from potential competition” as the aim for this research project.

Focusing on the aim of the project we have derived a few objectives as the driving tasks for this project.

1. If user search for an English word, suggest synonyms, Sinhala meaning transliterations and similar Sinhala words.
 - a. Ex: book (books.lk, tome.lk, potha.lk, grantha.lk)
2. If searched for a sinhala word, suggest similar sinhala words, and matching English words.
 - a. Ex: poth (grantha.lk, books.lk)
3. Categorize trendy domains or domains with buzz words as premium domains
4. Implement a Feedback module for the suggestions

1.4 Background and Motivation.

With the spreading of Covid pandemic globally the conventional business models has fallen apart since the customers physically visiting to stores becomes a rare thing. So the businesses had to look for an alternative method to present themselves to the customers. Online selling or E-Commerce was the answer for this problem though it has not been so popular among general public. With the new normal scenario online shopping became the new norm.

Businesses had to established their online branding solidly in this trying times. So the rush to register new domains and launching new websites and apps become a trend.

LK Domain Registry being in the domain business for more than 25 years hasn't had such a hype in domain registration in their history. They had to improve their systems and make them user friendly for the customers.

Being the Sri Lankan Domain Registry which has to cater multiple languages (Sinhala, Tamil and English) they faced a uphill task giving domain suggestions to its customers when the certain popular names and generic names were not available. Lot of ambiguities were there when typing Sinhala and Tamiil words in English characters as a domain.

This infact gave us the idea to improve domain search bar of domains.lk to have a sophisticated method in making meaningful domain suggestions to the customers.

Apart from that LK Domain Registry was not having a mechanism to track the common domain terms in the search, so they could make those popular terms upgrade to premier categories and generate more revenue from that. That guide us to collect and study the inputs given by customers and identify valuable terms and patterns in domain searches done by the customers 24/7 and 365 days.

1.5 Problem in brief

Domains.lk was having a mere search bar without providing only a handful of domain suggestions based of hard coded structure. It was not interactive at all or was not supporting the multilinguistic nature of the domain search terms.

Also Domains.lk was not saving the user entered domain search terms for any further processing which was losing a huge opportunity to study their customers' genuine intentions.

1.6 Proposed Solution

In our solution all the changes will be applied to the core of domains.lk search bar and we are breaking the solution into five main modules in order to study and develop with much focus on each module. Each of these module provides a different value to the final Domain Name Suggestions. The five of the modules are as follows. The tasks of each module are described in the later part of the paper.

- i. Singlish Translator
- ii. Sinhala,English Dictionary
- iii. Similar Domain Names Generator
- iv. Trendy Domain Names Finder
- v. Feedback Module

Once all the above 5 modules developed properly and merge into one single domain search bar we would be able to achieve the expected goal.

1.7 Structure of the thesis

The overall thesis is structured as follows, First chapter gives an introduction to fill project with the objectives, background, problem, and solution and second chapter critically reviews the literature in the domains registry aspects, translation and transliteration definitions, method and algorithms. Third chapter is about technologies used in developing the proposed domain search bar with all the five modules. Forth chapter present our innovative approach with users inputs, Translation and transliteration algorithms used in the solution. In fifth chapter we have discussed the design and the highlevel architecture of the system and analysis part of the developed system. Chapter six is all about the implementation of the solution in practical usage and in chapter 7 we have discussed the evaluation of the overall solution. Further improvements and plans are proposed in chapter 8 in this thesis.

Chapter 2

2.1 Literature Review

2.1.1 Domain Name System (DNS)

DNS, which stands for Domain Name System, is an ordered naming system and an important component of the internet that allows people to access and locate resources. These resources may include and are not limited to servers, hosts, and online services. [6]

2.1.2 Transliteration

Transliteration is a method of conversion of a word from a certain language to another. This means that we use the pronunciation of a word in the source language and depict it using the alphabet of our target language. Transliteration provides some level of understanding and access to people who are unable to read or write words in other languages. However, languages still vary from one another in different areas despite having some common features. Spelling, the alphabet, and interpretation still depend on an individual. [13]

2.1.3 Translation

Translation is a complex example of human interaction that fuels globalization and immigration. Its growth is directly proportional to the development of trade, culture, technology, and science. It is a process that involves the expression of a source language, including its meaning and logic, into the target language, without losing any of its semantics.

The English language is a widely spread language in terms of use, but it comes with its own challenges and restrictions. The translation of Bangla and English can be approached by studying their differences and similarities. Software applications are now available to streamline and simplify the process of translation, but issues related to syntax (grammar), lexis/vocabulary (word), stylistics (style), phonology (sound), and usage can still prevail. [13]

2.1.4 Translation and Transliteration

Transliteration is the conversion of a term from one or language to another that involves the replacement of letters, while translation is to express the meaning of a written text language to

another. There are times when translation may not be possible due to the lack of an equivalent word or cultural differences which would give way to transliteration. In transliteration, representation of the sound of the original word is not the primary goal. Instead, representing the characters as accurately and directly as possible.

Transcription, on the other hand, focuses on the sound and not on the spelling. Transliteration and transcription can be very close to one another if the relationship between the sounds and the letters are similar in both languages. A rule used in transcription is that the IPA sound symbols are used instead of the regular alphabets. Translation allows us to understand the meaning of expressions used in other languages, while transliteration helps us pronounce the words. An example of the use of transliteration is when genealogists research people and places that use non-Roman alphabets, such as Greek, Hebrew, or Cyrillic, and the names of immigrants from those places. People of the past would transliterate their names so that Americans could spell and pronounce them, producing many variants in spelling. It is also useful when researching and the characters in the target language are unavailable. Translation offers the meaning of a text that is written while interpretation provides the meaning of a spoken expression. [13]

2.1.5 Translation processes and algorithms

There are instances when translation is not possible or is not necessary at all. Here, transliteration becomes the only option. Let's take this example. A 'visiting card' should be transliterated than translated. Transliteration is a type of conversion of a term or text from one script or language to another that involves swapping letters in order to pronounce the foreign word correctly, while translation is to give the meaning of a written text of one script or language to another. Transliteration does not focus on representing the sounds of the original but rather with representing the characters, ideally accurately and unambiguously.

Equally, transcription notes the sounds, but not necessarily the spelling (i.e. letters). A transliteration may be very close to a transcription if the relations between sounds and letters are similar in both languages. But as a rule, the IPA sound symbols is used in transcription, not the regular language alphabets. Transliteration helps us pronounce the words and doesn't tell us the meaning. Translation, on the other hand, tells us the meaning of words or expressions in another language. For genealogists, transliteration is used when researching people and places that use non-Roman alphabets, such as Hebrew, Cyrillic or Greek, and name changes of

immigrants from those places. In the olden times, our ancestors often would transliterate their names so that Americans could spell and pronounce them, producing many variant spellings. This is also helpful when we make research notes and our word processing software doesn't offer fonts with characters in the language of interest. However, a translation provides us with the same virtue as the original, and generates the same responses in its readers. While a translation gives the meaning of a written text, an interpretation conveys the meaning of an oral or a spoken text. [13]

2.1.6 Transliteration process and algorithms

In general, transliteration is done by transcribing a word or text written in one writing system into another. Similarly, Machine transliteration is a method to convert words in one language into phonetically equivalent ones in another language automatically. For example, the English word 'machine' is transliterated into Sinhala as මැෂින්. Currently, there are number of Machine transliteration approaches available. Examples are grapheme-based transliteration model, phoneme-based transliteration model, hybrid transliteration model and correspondence-based transliteration model [10]

2.1.7 Customer Feedback System

The digital economy necessitates services be created in nearly real time – while continuously addressing customer inputs. Managing and analyzing the data collected about products and customers become very critical. Successful companies must collect data regarding customer behavior in a sensible manner, understand their customers and engage in constant interaction with them. To utilize the data, they collect and analyses customer feedback quickly, companies require automation of customer feedback processing. [12].

2.2 Reviews of Other's work

Translation and transliteration are going hand in hand with the native language. So it is quite evident that not much research or scholarly attempts has done in this area of research. Because the certain topic this research is covering is all about Sinhala translation and transliterations with English letters. Similar research and approaches has been made in different other languages though.

In 2019 in ‘British Journal of English Linguistics’ Md Faruquzzaman Akan has published a paper with the title ‘Transliteration and Translation from Bangla into English: A Problem Solving Approach’. That is quite similar to what this particular research is trying to address, but the same mechanisms and algorithms cannot be used since those were created targeting Bangla and English translations and transliterations.

When a language is mixed with the others, those languages can be a great influence. For this reason, the use of English alphabet may cause some changes in the pronunciation of a Bangla word in transliteration. Some phonetic features of Bangla and English alphabet are discussed here.[13]

Since this is a process which cannot implement standards fully, researches has always had to use their own set of rules to transliterate accurately. These rules has been implemented as algorithms or regular expressions. The same kind of an exercise we also have used in our research as well.

There was another research happened on a similar topic in University of Stuttgart in the institute for NLP by students called Nadir Durrani, Hassan Sajad, Alexander Fraser and Helmut Schmid on the topic on machine translation from Hindi to Urdu using transliteration. They have given much of the details of their research for us to understand further on translations and transliteration.

They have separated the previous works in to 3 groups. The first group is purely for basic transliteration related work, and it has to be measured without any involvement in the context of translation. In this area of work, it uses grapheme based models or phoneme based models for these transliterate words lists. The job by Malik et al. covered transliteration from Hindi to Urdu using manually developed rules and a phonemic representation. One other factor about this work of Malik et al is that it ignores the context of translation.[14]

Another group, which is the second group deals with terms that aren't part of the normal lexicon for SMT systems built on a massive collection of text and consequently focuses on name transliteration, which is essentially independent of context. AlOnaizan and Knight (2002) use transliteration for converting Arabic NEs into English and score them against their respective translations employing a modified IBM Model 1. To resolve and avoid equivocation the choices are further re-categorized based on different measures including web counts and using grammatical relation. These re-ranking methodologies,

however, can't be performed in statistical machine translation at the decoding time. it'd be possible to have an efficient way to compute and re-rank the transliterations of name entities and integrate them at once, but these are impractical with the model because it considers transliterations of all input words and not just name entities. A log-linear block transliteration model is applied to out of vocabulary name entities in Arabic to English statistical machine translation by Zhao et al. (2007). This work is additionally transliterating only name entity's and not doing any disambiguation. the simplest method proposed by Kashani et al. (2007) integrates translations provided by external sources like transliteration or rule-base translation of numbers and dates, for a random number of entries within the input text. Their work is different from Kashani et al. (2007) therein their model compares transliterations with translations on the fly whereas transliterations in Kashani et al. (2007) don't compete with internal phrase tables and only compete amongst themselves during the second pass of decoding. Hermjakob et al. (2008) use a tagger to spot good candidates for transliteration (which are mostly name entity's) in the input text and add transliterations to the statistical machine translation phrase table dynamically such that they can directly compete with translations during decoding. this is often closer to their approach except that they use transliteration as an alternative to translation for all Hindi words. Their focus is disambiguation of Hindi homonyms whereas they're concentrating only on transliterating name entities.

Moreover, they're working with a large bitext so they can rely on their translation model and only need to transliterate name entities and out of vocabulary's. Their translation model is predicated on data that is both sparse and noisy. For this, they pit transliterations against translations for each input word. In 2009 with Sinha's work, he presented a rule-based machine translation system that uses Hindi as a rotation to translate from English to Urdu. This work also uses transliteration just for the translation of unknown words. Their work can't be used for direct translation from Hindi to Urdu (independently of English) "due to various ambiguous mappings that have to be resolved".[14]

Technology Adopted

3.1 Introduction

The technologies which were used for the system development were selected with the limitations of the existing technologies in LK Domain Registry. The domain name suggestion system was designed as an enhancement to the existing Registration System with new additions. New system had to be compatible with Silverstripe (PHP Framework) to be inter connected with the Registration system. This chapter describes the technologies used to implement the domain name suggestion system and why they are used.

3.2 Programming Languages

1. PHP

The Registration System of LK Domain Registry mainly uses PHP as their programming language. PHP stands for Hypertext Preprocessor. This technology is widely used in the world as a general purpose web developing programming language. PHP is at the core of the largest blogging content management system in the world which is Wordpress. The powerfulness of PHP can be determined by the usage of the technology world wide.

2. Javascript

Javascript can be described as a programming or scripting language which is used to implement simple or simple features in the web pages. In LK Domain Registry, Javascript is used for the development of the frontend of the Registration system. The same technology is used for the domain suggestion system to ensure the compatibility of the existing system.

3. CSS

CSS is used for styling purposes of the frontend of the Domain Name Suggestion system. CSS and HTML are in the core of any webpage and the user interface is developed using these technologies commonly in the web development industry.

3.3 Server Technologies

1. Apache Web Server

Apache is an open source and free technology which is used to deliver web content from the server to the internet browsers. Apache is the most used web server around the world and the level of the security it provides is very outstanding when compared to the other web servers. It is well known for its reliability and maturity. First version of Apache was released in 1995 and since then, it was developing rapidly as a web server technology.

2. MYSQL

MYSQL is used as the database management system to work with the application. MySQL is developed and distributed by Oracle Corporation and It is widely used around the world as a relational database management system. MySQL also a free and open source technology which provides industry standards and a high level of security. The reasons for choosing MySQL as the DBMS are high performance, Flexibility and convenience of use.

3.4 Libraries

1. Composer

Composer is a dependency management tool which is commonly used with PHP framework in the recent past. Silverstripe framework uses composer as a foundation provider for the application. Convenience it provides is very high as it offers a developer friendly environment.

3.5 IDEs

1. PHPStorm

PHPStorm is the IDE used for the development of the Domain Name Suggestion system. It supports following languages which are important as the Registration System is developed in multiple languages.

- b. PHP
- c. JavaScript
- d. HTML
- e. SQL
- f. CSS

3.6 High Level Architecture of the Technologies used

This diagram describes the high level view of the technologies and how they are implemented.

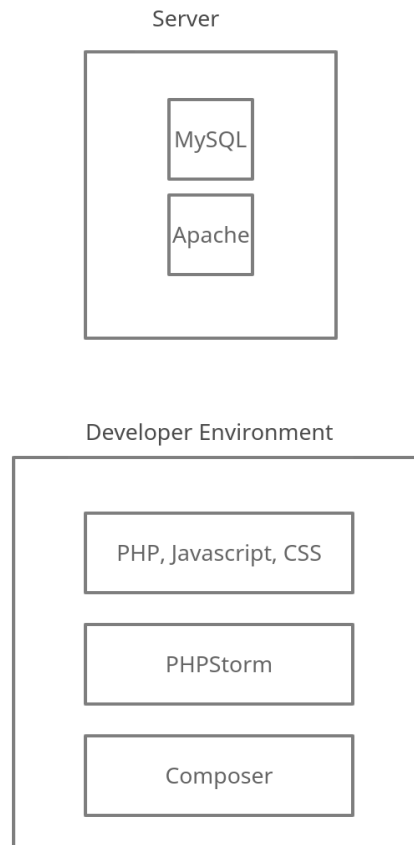


Figure 3.6 High Level View Of The technologies

3.7 Summary

This chapter gives an overview of the technologies that are used in the implementation of the system and why those technologies are chosen. Next chapter illustrates the approach of implementing the system.

Chapter 4

Novel Approach to generate domain name suggestions

4.1 Introduction

The solution I proposed consists of 5 components and this chapter addresses the solving path for the problem. I have described the overall outputs, processes and inputs and expected outcomes of the developed system. This chapter illustrates the overall approach and the design of each component's relationship with the system which also can be called as modules.

4.2 System

The developed system contains five main modules to be exact. Each module provides a different value to the final Domain Name Suggestions. The five of the modules are as follows. The tasks of each module are described in the later part of the paper.

- i. Singlish Translator
- ii. Sinhala,English Dictionary
- iii. Similar Domain Names Generator
- iv. Trendy Domain Names Finder
- v. Feedback Module

4.3 High Level Architecture of the System

The inputs of for the system can be categorised into two parts, which are,

1. Sinhala words typed in english (pansala, gasa, mala, potha)
2. English words (book, temple)

The high level view of the system is as following diagram and the architecture for the two categories are slightly different. Singlish words(Sinhala words typed in english) and English words have different road maps in the system.

Approach for English words

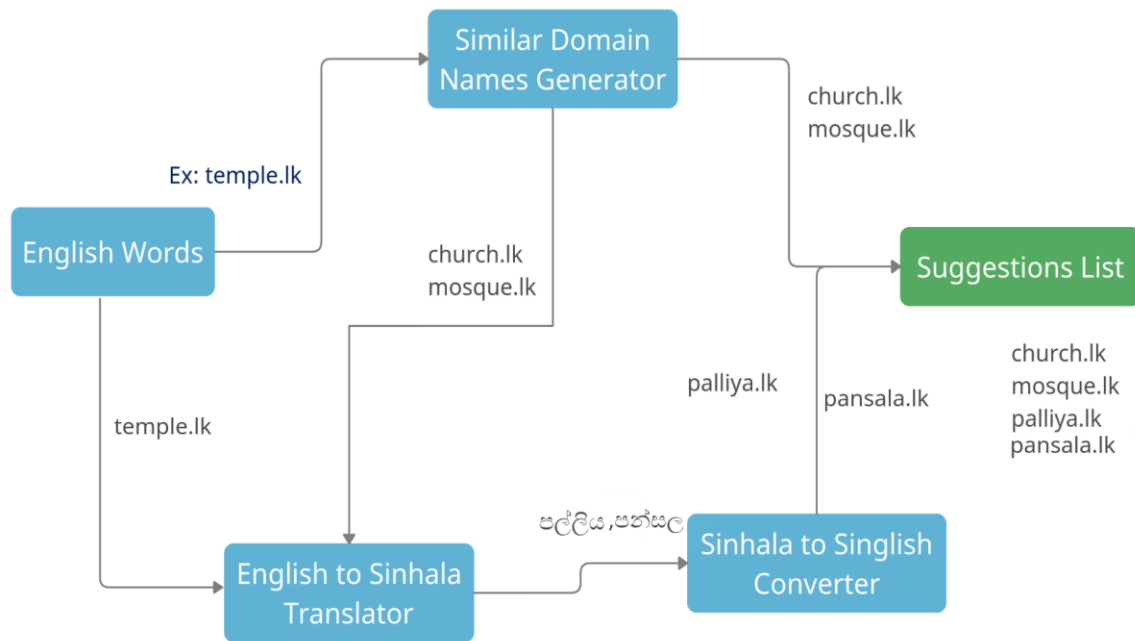


Figure 4.3 Approach for English words

Approach for Singlish words

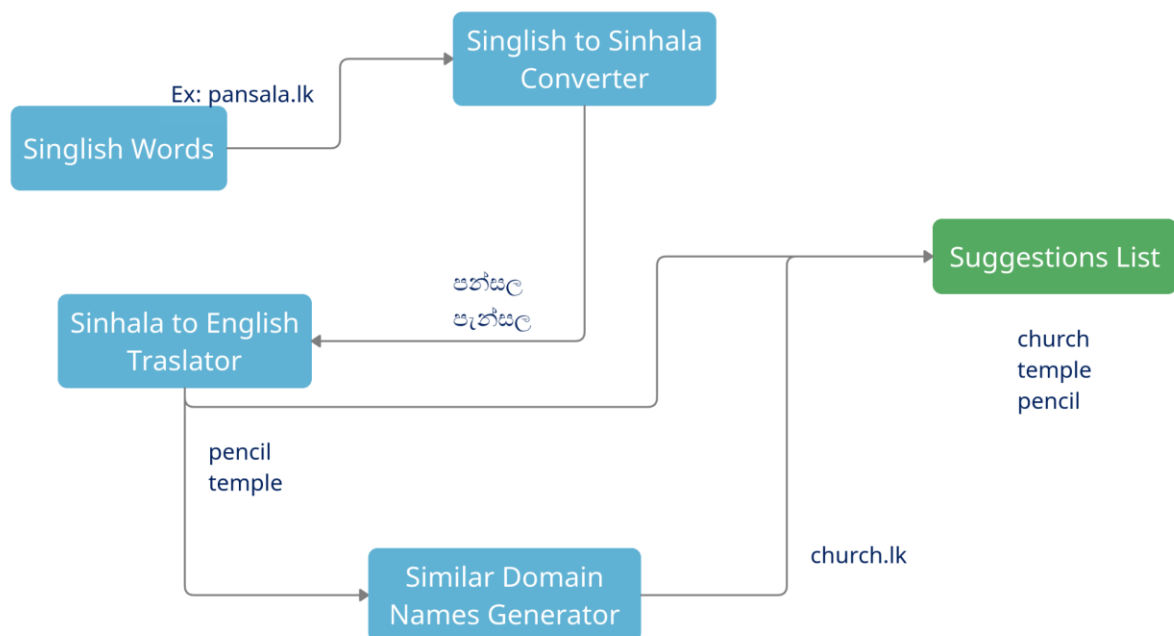


Figure 4.3 Approach for singlish words

4.4 Users

Large number of customers reach domains.lk to search their desired domain names. Currently they are given predefined domain name suggestions with prefixes to the searched domain name.(Ex: ibook.lk, ebook.lk). Therefore, customers are viewing only a fixed amount of suggestions and more often than not they are not interested in predefined suggestions. Following Image shows the predefined suggestions for the domain name : ‘vehicle’ .

Suggested Top Level Domains		
ivehicle.lk	LKR 3,000.00	Add To Cart 
evehicle.lk	LKR 3,000.00	Add To Cart 
myvehicle.lk	LKR 3,000.00	Add To Cart 
fastvehicle.lk	LKR 3,000.00	Add To Cart 
mvehicle.lk	LKR 3,000.00	Add To Cart 

Figure 4.4 Predefined domain name suggestions

Predefined Suggestions

Customers of LK Domain Registry are benefited from the developed Domain Name Suggestions System as they are given a variety of suggestions that they can choose from. And also the feedback regarding the suggestions are taken to develop the Domain Name Suggestions algorithm.

4.5 Inputs

Inputs are taken from the customer who is searching the domain. The customer is given a user interface to enter the inputs at the start of the road map. Some of the modules are taking other module’s outputs as inputs. User interface for the customers is plan and simple as shown in the image below, which allows the customer to search a domain name, add the domain name to the cart, rate the domain name suggestions. These are the inputs which are taken from the users of domains.lk. The user interface is as follows,

✔ 'pansala.lk' Domain is available for registration.

CAT1 - Full Domain Package

pansala.lk LKR 5,000.00

You automatically reserve following names

pansala.web.lk pansala.edu.lk

pansala.com.lk pansala.org.lk

pansala.hotel.lk

Add To Cart

CAT2 - Top Level Domain Only

pansala.lk LKR 3,000.00

Add To Cart

Other options related to your domain search

CAT3 - Second Level Domain only

pansala.edu.lk	LKR 1,000.00	Add To Cart
pansala.com.lk	LKR 1,000.00	Add To Cart
pansala.hotel.lk	LKR 1,000.00	Add To Cart
pansala.org.lk	LKR 1,000.00	Add To Cart
pansala.web.lk	LKR 1,000.00	Add To Cart

Suggested Top Level Domains (Rate the domain suggestions)

Sinhala Words Detected - සිංහල වචන හඳුනාගන්න

pencil.lk	★★★★★	LKR 3,000.00	Add To Cart
temple.lk	★★★★★	LKR 3,000.00	Add To Cart
monastery.lk	★★★★★	LKR 3,000.00	Add To Cart

Figure 4.5 User interface

The five components have different inputs from the system. Inputs are taken as follows in the table,

Module	Inputs
Singlish Translator	book.lk,pansala.lk
Sinhala,English Dictionary	Words
Similar Domain Names Generator	Book.lk, pencil.lk
Trendy Domain Names Finder	Database of searched domain names
Feedback Module	Customer can enter whether the suggestions are good or bad

Table 4.5 Main Components

4.6 Process

The 5 modules carry different tasks in order to get the final process and the output. A dictionary is used to translate English words to Sinhala words and Sinhala words to English words. The user user inputs are translated back and forth in order to get suggestions from english and sinhala. The Singlish translator is the most important part of the system and new algorithms were introduced as there were no existing methods to address the issues faced when using Singlish words. (Ex: pansala can be interpreted as both පන්සල and පැන්සල and also the Sri Lankans are writing පැන්සල as ‘pansala’ and ‘pensala’). The system had to address this issue in order to produce meaningful and complete suggestions for the name that the customers are searching. The singlish translator gives all the meaningful words that a search query can get.

Ex:

Search term : ‘kalaya’

- Singlish translator processes to get the outputs as කලය and කැලය so that the final suggestions can be given to user as,
 - Pot
 - Jungle
 - Forest

In this system customer searched queries are stored in the database to generate trendy domain names. Customers tend to buy trendy domain names to increase their exposure to the cyber world. For example, because of the currency situation in Sri Lanka, customers bought the domain names that are trending.

Ex: aragalaya.lk

The Domain Names generator is used to find the words related to the search term. The system uses an existing Application programming interface (API). An example of this module’s process is as follows,

Search term - temple

Output - mosque, church

Customer's feedback about the suggestions are taken into account as the system will not be hundred percent accurate. Customers can rate the suggestions and the next suggestions for the same domain name will consider the feedback for the previous occasion.

All the modules are related to each other and communicate to increase the accuracy and the meaningfulness of the suggestions that the system gives.

4.7 Outputs

Final outputs of the system are the suggestions that the customers are given. Suggestions are displayed in the user interface and the customer can add those suggested domain names to the cart if he or she is interested. Following image shows how the suggestions are displayed on the UI for the search query : 'pansala'.

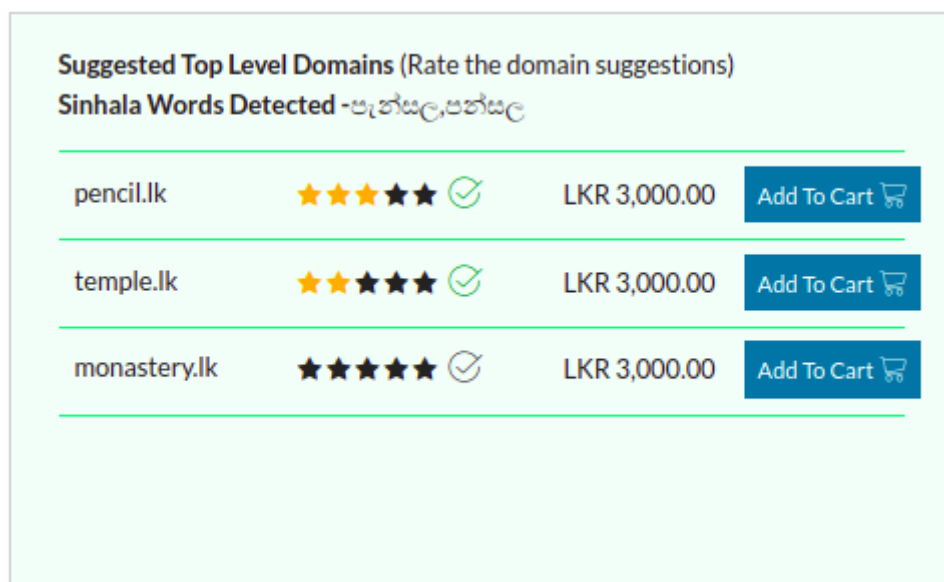


Figure 4.7 Final Output User Interface

Outputs of each module determine the final output. And also all the internal outputs are cached, therefore the system does not need to run through the algorithm repeatedly to generate the suggestions. Each module outputs are as follows(Outputs are based on the inputs declared previously),

Module	OutPuts
Singlish Translator	potha.lk,temple.lk
Sinhala,English Dictionary	Translated words
Similar Domain Names Generator	Trendy domain names
Similar Domain Names Generator	mybook.lk,ebook.lk
Feedback Module	Administrators can see those feedbacks and act on modifications to the algorithms used

Table 4.7 Main Modules

4.8 Summary

The proposed solution, module-by-module inputs, implementation flow, and module-by-module outputs were briefly discussed in this chapter. The system's design is illustrated in the next chapter, which includes a detailed explanation of each module's implementation.

Analysis and Design

5.1 Introduction

This chapter describes the design of the implemented system and the overall architecture of the solution. The main 5 modules are described in detail in this section and modules are as follows,

1. Singlish Translator
2. Sinhala English Dictionary
3. Similar Domain Names Generator
4. Trendy Domain Names Generator
5. Feedback Module

5.2 High-level Architecture of the Overall System

The following figure illustrates the high level architecture of the Domain Name Suggestions system and how 5 main modules are inter connected to each other. In the system, customer search terms are identified as a Singlish word or an English word. The process of identifying the word includes novel algorithms to address the issues in Singlish words. The final desired output should be as follows for the names that the customers are searching.

Search Term	Suggestions
pansala	pencil, temple, church, mosque
temple	pansala, palliya

Table 5.2 Overall System Final Output

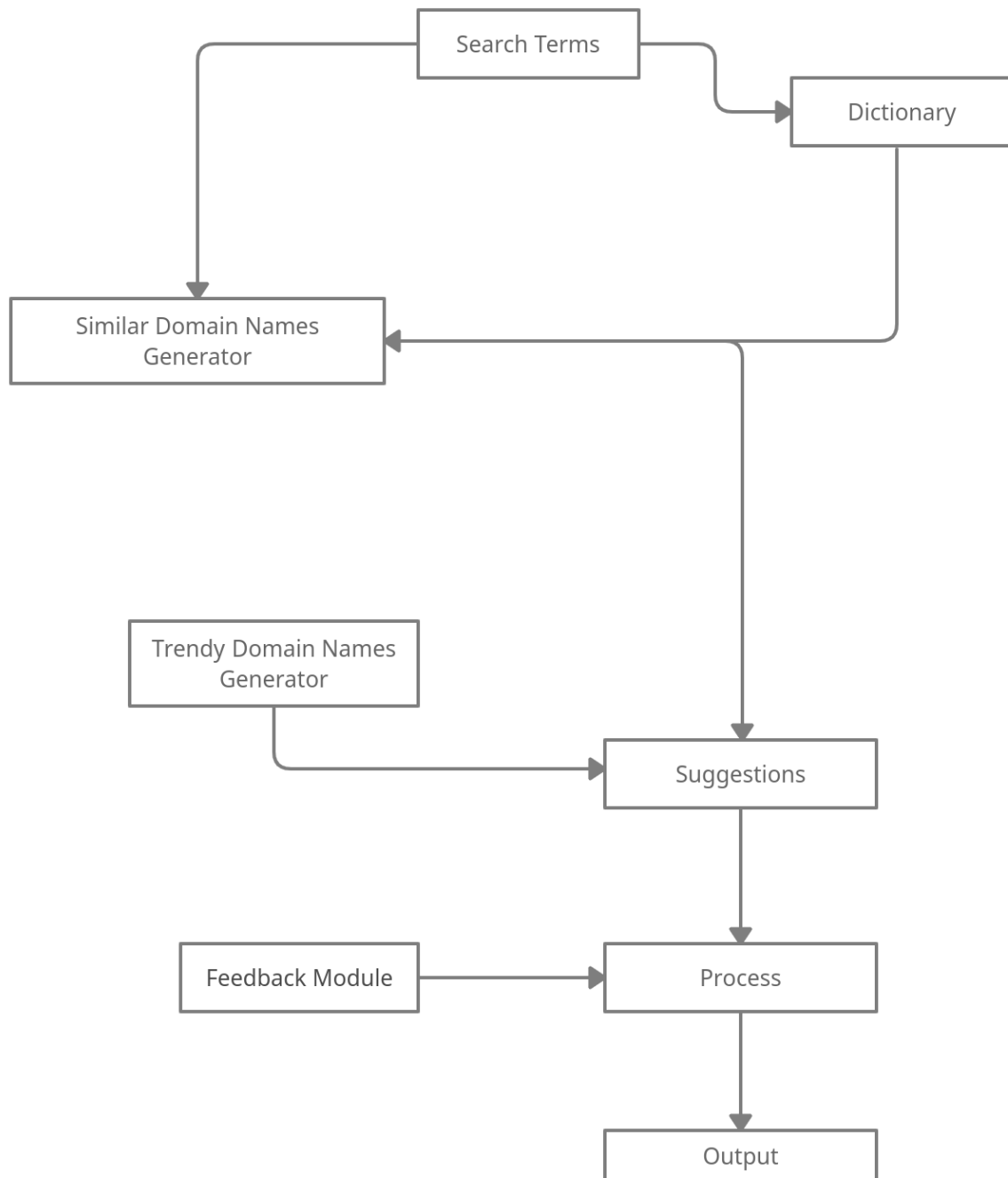


Figure 5.2 Overall System High-level Architecture

5.3 High-level Architecture of the Individual Modules

5.3.1 Singlish Translator

Singlish words are the words which are written in English letters and at the same time, have a sinhala meaning. (Ex: samanalaya, sapatthuwa) There are two types of conversions related to singlish words. The conversion can be happened from sinhala words to singlish words (Ex:

සමනලයා -> samanalaya) and the other way around (samanalaya -> සමනලයා). There are systems which can produce one of the conversion types which is singlish words to sinhala letter words. But the main issue with the existing system is that the same singlish word can be interpreted as two or more meanings.

Ex: 'gona' can be read as ගොනා and ගෝනා

So the solution that is provided in this system addresses this issue and finds for all the possibilities that a particular word can be interpreted. The suggestions are based on the possibilities which are detected from the novel algorithm.

On the other hand, the other type of the conversion which is sinhala words to singlish words, was also a requirement. The names which is searched in english are translated to sinhala and then converted to singlish words and given to the customer as suggestions

Ex: Search Term : scissor

1. Word is translate to sinhala (scissor -> කතුර)
2. Converted to singlish word (කතුර->kathura)

But, there are no existing methods or systems for this type of conversion. The solution that the system provides the ability to address this matter.

The types of words(Singlish and English) are addressed in different ways by the Singlish Translator.

5.3.1.1 Singlish Words

First step is to get the possibilities that a word can be read. The novel algorithm takes the word and generates all possible words in order to take the meaning full words. An example word can be taken as 'pansala'. The customer may be searched the term with idea of temple(පන්සල) or pencil(පැන්සල). Therefore this module converts the word to the possibilities considering the vowels of the word.

Ex: 'pansala' ->

be used again with the search term because this process can be time consuming. Following words can be taken as the meaningful words related to above terms.

Ex: 'pansala' -> පැන්සල,පන්සල
'gona' -> ගොනා,ගෝනා
'samanalaya' -> සමනලයා

Then these words are translated to English and suggested to the customers.

Ex: 'pansala' -> පැන්සල,පන්සල -> pencil, temple
'gona' -> ගොනා,ගෝනා -> bull, sambardeer
'samanalaya' -> සමනලයා -> butterfly

And also the english words are sent to similar domain names generators to generate more suggestions related to the word.

Temple -> church, mosque

5.3.1.2 English Words

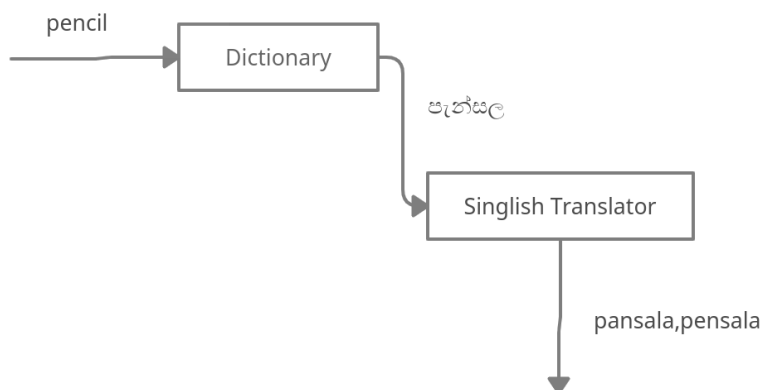


Figure 5.3.1.2 New algorithm

English words are translated to sinhala first and entered into the singlish translator to generate singlish words out of sinhala letter words afterwards. This procedure is novel as there are no systems currently. There are some issues with this conversion as there are differences in

preferences of using english vowels in each person. Some of the Sri Lankans prefer ‘a’ to the sound of ‘අ’ and at the same time some prefer letter ‘e’ for the same sound. Therefore the design was done to address this issue and designed the new algorithm.

The suggestions are given considering this by generating all possible preferences of the people.

Ex: pencil -> pansala, pensala
 Jungle -> kelaya, kalaya

These suggestions can be used to protect the brand names of the customers. For example if a customer has a business and he or she can buy the domain names related to the customer's business name, to ensure that no one is buying the names related to the business in the future.

5.3.1.3 English words which can be interpreted as singlish words

There are some words in the English dictionary which also can be a singlish word. These words are identified in the system and address for both scenarios. All the words are checked with the English dictionary and if there is a meaning, the word can be taken as an English word. And also, the same word is taken as a singlish word and the system generates all the possibilities of sinhala letter words. Then meaningful words are identified if there are any.

Ex: ‘those’

This can be identified as an English word. At the same time it can be read as ‘තෝස’.

This matter is addressed by considering every word as a both Singlish word and English word. The tasks of the Singlish Translator Module as above.

5.3.2 Sinhala English Dictionary

Sinhala English dictionary is the module which is used to get the meanings of Sinhala and English words. This module is built using an existing Application Programming

Interface(API). The API is developed by <http://www.deranadictionary.com> and the response is processed according to the requirement of the domain name suggestions system.

The meaningful Sinhala words are also detected using this Dictionary module for the use of Singlish Translator. The API calls are also cached in the system, therefore there will be zero API calls for the second time with the same domain name. This allows the system to increase the performance of the process.

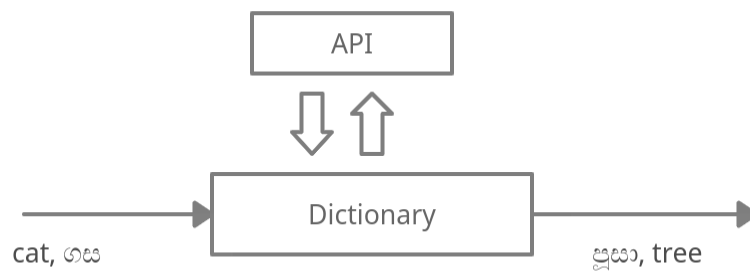


Figure 5.3.2 Sinhala English API

5.3.3 Similar Domain Names Generator

Similar domain name generator includes API configuration which enables the system to call an external URL and receive the related words to the search term. The API is developed by <http://thesaurus.altervista.org> and there are various relationships to the search term which we can configure. Such as,

1. Synonyms
2. Rhymes
3. Antonyms
4. Examples
5. In category

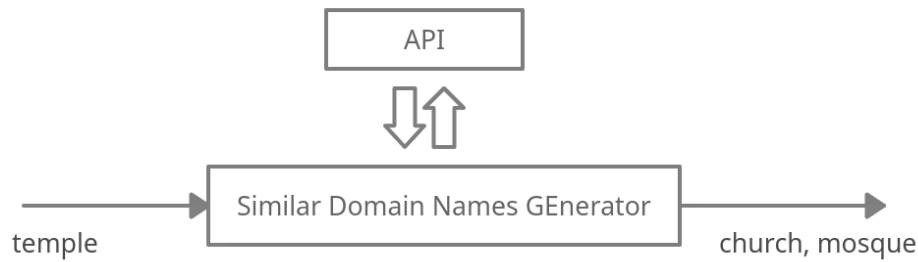


Figure 5.3.3 Similar Domain Names Generator

5.3.4 Trendy Domain Names Finder

According to the recent investigations, customers tend to buy the trending domain names as they need to increase their exposure to the internet. Therefore, suggesting trendy domains in the Domain Search display will be a very updated marketing campaign. The search terms of the customers are recorded in a database for data mining purposes. The information which will be collected are PHP Session ID, session start time and searched domain names. These information are stored in a file in the following manner.

bo8ajghlnb5dhqefvtrrnbguo6	2022-06-01 20:28:13	masterfood, yummybooking, badugodak
p1rdlh104muatk03imb6qmr1s1	2022-06-01 15:45:35	senex, ipool
06rfu7s7ndnosshlnli41112qv	2022-06-01 13:32:54	capitalmarketing, treasurespce, callofdutymobile
ioe1tp14uj56689i8uhii36nab	2022-06-01 13:22:20	codechub
a50tuae94cen50e20i2p0k2pml	2022-06-01 13:12:27	andarehouse, asquare
kd212ak5qf2nd79fcgc1tohfco	2022-06-01 12:58:51	smartconnector
d1jhuq6lhshqlstbt6pbev8lpp	2022-06-01 12:53:47	
do7b693fa44no7343stg7bgbkq	2022-06-01 12:49:43	marthatilaar
988fnoshj4edb7j1hjrt90cnt	2022-06-01 11:39:08	katakari
5tceao03l2d32akrg6q0jmmo45	2022-06-01 10:49:25	ruwal, pawra, walampuri, lantaruma
qnfn8bt4o63j0rlke3ops8cjt7	2022-06-01 09:38:16	thunhele, thunhelayan, sihanada, sirirata, sampradaya
4s6p9rni062ku8evj6dgv69gaa	2022-06-01 09:27:10	madisa
ioe1tp14uj56689i8uhii36nab	2022-06-01 09:11:47	udhayatharagai
ioe1tp14uj56689i8uhii36nab	2022-06-01 09:05:28	agrohub
9k8mrcv8rmdtrpu5ubq7pnt3g1	2022-06-01 08:28:37	valentino
voas1h70phqelpddek9f9m88d	2022-06-01 08:01:41	travelmon
r734nav2025eurjgn4hs5bts9l	2022-06-01 04:15:20	petitions, aragalaya

This information is used to identify trendy domain names.

5.3.4 Feedback Module

This module plays an important part as this information is used to increase the accuracy of the system. The customer can rate from one to five to each and every suggestion which is displayed on the user interface. The ratings from the customers are taken into an account when displaying suggestions again for the same domain search term. The user interface is as follows,

Suggested Top Level Domains (Rate the domain suggestions)			
Sinhala Words Detected -මග්නා,මග්නා			
bull.lk	★★★★★	LKR 3,000.00	Add To Cart
bullbrier.lk	★★★★★	LKR 3,000.00	Add To Cart
bullfly.lk	1★	LKR 3,000.00	Add To Cart
bullterrier.lk	★★★★★	LKR 3,000.00	Add To Cart
bulltrout.lk	★★★★★	LKR 3,000.00	Add To Cart
idiot.lk	★★★★★	LKR 3,000.00	Add To Cart
elk.lk	★★★★★	LKR 3,000.00	Add To Cart
sambur.lk	★★★★★	LKR 3,000.00	Add To Cart
stag.lk	★★★★★	LKR 3,000.00	Add To Cart
moose.lk	★★★★★	LKR 3,000.00	Add To Cart

Figure 5.3.4 Feedback Module

The information is stored in a file in the following structure.

```
pansala,pencil,5
pansala,temple,4
pansala,monastery,1
gona,bullfly,1
gona,bullterrier,1
gona,bulltrout,1
gona,idiot,5
gona,elk,5
gona,moose,5
gona,bull,5
gona,bullbrier,1
gona,sambur,1
```

The words which are rated 1 will be removed from suggestions cache file and will not be displayed for the customers for the same domain name

5.4 Summary

In this chapter novelty and the design of the system is discussed and described. The implementation is discussed in the next chapter with the algorithms used.

Chapter 6

Implementation

6.1 Introduction

The implementation is done by considering the compatibility of the existing technologies in the LK Domain Registry Domain Registration System. The framework which is used in the existing system (Silverstripe) is based on the PHP language. Therefore the new algorithms are written in PHP and the integration of the 5 modules are also done using the same programming language. This chapter describes the implementation process module by module and the integration process of the system.

6.2 Caching Process

Important outputs of each module are cached in the system in order to increase the performance of the system. The algorithms that are used in the system to generate domain name suggestions require a considerable amount of processing power. With this caching process, the amount of the time taken to generate suggestions can be decreased by a considerable amount. Example of a caching file is as follows. Meaning full suggestions are stored inside cache files.

```
_මාලා_ row,garland
_මල_ stool,template,templet,fur,faeces,bloom,rose water,dirt,dung,excreta,stain,blossom,rust,hole,ordure,feces,rose,flower
_පැන්සිල_ pencil
_පන්සිල_ temple,monastery
_ගොන_ bull,bullbrier,bullfly,bullterrier,bulltrout,idiot
_ගේන_ elk,sambur,stag,moose
_ගෙදර_ hearth,fireside,home,house
_වාහනය_ vehicle,carwheel,wagon,conveyance,wain,carriage,car,cart
_වහනය_ carriage,flow,vessel
```

```

_pansala_ පැන්සල,පන්සල pencil,temple,monastery
_mala_ මාලා,මල row,garland,stool,template,templet,fur,faeces,bloom,rose
_gona_ ගොනා,ගෝනා bull,bullbrier,bullfly,bullterrier,bulltrout,idiot,elk,sambur,stag,moose
_gedhara_ ගෙදර hearth,fireside,home,house
_wahanaya_ වාහනය,වහනය vehicle,carwheel,wagon,conveyance,wain,carriage,car,cart,flow,vessel
_dsad_ -
_dsadas_ -

```

6.2 Implementation of Singlish Translator

This module begins with generating all the possibilities which can be a singlish word interpreted. The algorithm starts with finding English vowels and consonants that can make two or more sounds.

Ex: ‘a’ - අ,ආ
 ‘e’ - එ,ඒ
 ‘la’ - ල,ළ
 ‘na’ - න,ණ
 ‘o’ - ඔ,ඕ

First of all, Singlish words are converted into sinhala letters exactly the user typed. The sinhala vowels and the constants related to the English letters are predefined in Arrays to generate the sinhala letter word. (Ex: pansala -> පන්සල)

```

/**
 * Returns sinhala letter word for singlish word
 * @param $string
 * @return string
 */
public static function SinglishToSinhalaExcatlyAsTyped($string = '') {
    $s = '';
    $r = '';
    $v = '';
    $specialConC = count(self::$specialCons);
    for ($i = 0; $i < $specialConC; $i++) {
        $string =
str_replace(self::$specialCons[$i], self::$specialConsU[$i], $string);
    }

    $specialCharsC = count(self::$specialChar);
    $consC = count(self::$cons);

    for ($i = 0; $i < $specialCharsC; $i++) {
        for ($j = 0; $j < $consC; $j++) {
            $s = self::$cons[$j].self::$specialChar[$i];

```

```

        $v = self::$consU[$j].self::$specialCharUni[$i];
        $r = $s;
        $string = str_replace($r,$v,$string);
    }
}

$vowelsC = count(self::$vowels);
for($i=0;$i<$consC;$i++){
    for($j=0;$j<$vowelsC;$j++){
        $s = self::$cons[$i].self::$vowels[$j];
        $v = self::$consU[$i].self::$pillam[$j];
        $r = $s;
        $string = str_replace($r,$v,$string);
    }

    $s = self::$cons[$i].self::$vowels[$j];
    $v = self::$consU[$i].self::$pillam[$j];
    $r = $s;
    $string = str_replace($r,$v,$string);
}

for ($i=0;$i<$consC;$i++){
    for ($j=0;$j<$vowelsC;$j++){
        $s = self::$cons[$i].self::$vowels[$j];
        $v = self::$consU[$i].self::$pillam[$j];
        $r = $s;
        $string = str_replace($r,$v,$string);
    }
}

for ($i=0; $i<$consC; $i++){
    $r = self::$cons[$i];
    $string = str_replace($r,self::$consU[$i].self::$pillam[$j],$string);
}

for ($i=0; $i<$vowelsC; $i++){
    $r = self::$vowels[$i];
    $string = str_replace($r,self::$vowelsU[$i],$string);
}
return $string;
}

```

These kinds of vowels and the constants are predefined in an array and replaced by a special character which can be replaced later and generate the possibilities.

'_'	=> ['෧', '෧෦', '']
'*'	=> ['෧෦', '෧෦']
'#'	=> ['෧෦෦', '෧෦෦']
'+'	=> ['෧෦', '෧෦']
'&'	=> ['෧', '෧']

Then the following algorithm generates all possible outcomes in which a Singlish word can be interpreted.

```
/**
 * Converts singlish words to sinhala unicode and returns all possibilities
 * @param $string
 * @return array
 */
public static function getAllPossibilitiesForSinglishWord($string) {
    $singUtils = new SinglishUtils();
    $string = self::SinglishToSinhalaExcatlyAsTyped($string);
    $consC = count(self::$consU);

    //Replace gathakuru with no pillams with an identifiers

    for ($i=0; $i<$consC; $i++){
        $r = self::$consU[$i];
        $letterPositions = StringUtils::strpos_all($string, $r);
        $count = count($letterPositions);

        for ($k=0; $k<$count; $k++){
            $letterPos = $letterPositions[$k];
            $pillama = substr($string, $letterPos+3, 3);
            $pillamaKey = array_search($pillama, self::$pillam);
            if($pillamaKey && $pillama){
                $beginning = substr($string, 0, $letterPos+3);
                $end = substr($string, $letterPos+6);
                $string = $beginning.self::$pillamIDs[$pillamaKey].$end;
            }else{
                if($letterPos == (strlen($string)-1)){
                    $string.='_';
                }else{
                    $beginning = substr($string, 0, $letterPos+3);
                    $end = substr($string, $letterPos+3);
                    $string = $beginning.'_'.$end;
                }
            }
            $letterPositions = StringUtils::strpos_all($string, $r);
        }
    }
}
```

```

$positionCombs = [];
$positions = [];
$positionsTemp=[];
$index = 0;
foreach (self::$conflicts as $item => $value){
    $positionsConflicts = StringUtils::strpos_all($string,$item);

    foreach ($positionsConflicts as $item){
        $positionCombs[] = $value;
        $positions[$item] = $index;
        $positionsTemp[] = $item;
        $index++;
    }

}

sort($positionsTemp);
$combinations = $singUtils->getCombs($positionCombs);

$splitStr = $singUtils->splitStringByPositions($string);
$words = [];
if($singUtils->is2DArray($combinations)){
    foreach ($combinations as $comb => $value){
        $str2 = '';
        $i = 0;
        foreach ($splitStr as $str){
            if(isset($positionsTemp[$i])){
                $strPos = $positionsTemp[$i];
                $valueIndex = $positions[$strPos];
                $str2.= $str.$value[$valueIndex];
            }else{
                $str2.= $str;
            }
            $i++;
        }
        $words[] = $str2;
    }
}

}else{
    foreach ($combinations as $comb){
        $words[] = implode($comb,$splitStr);
    }
}

return $words;
}

```

Then the meaning of full possibilities is extracted from all the possibilities. This includes communicating with the dictionary to get the meaningful words.

```

public static function getValidPossibilitiesForSinglishWord($word) {
    $path = 'DictionaryFiles/SinglishWords.txt';
    if($line = exec('grep _'. $word .' _ ./'. $path)){
        $engWords = explode(' ', $line);
    }
}

```

```

        return explode(',', $engWords[1]);
    }else{
        $sinWords = self::getAllPossibilitiesForSinglishWord($word);

        $meanings = [];
        $sinhala = [];
        foreach ($sinWords as $item){
            $path2 = 'DictionaryFiles/sinhalaWords.txt';
            if(exec("grep _".$item."_./".$path2)){
                if($meaning = DictionaryUtils::sinhalaToEnglish($item)){
                    $meanings = array_merge($meanings, $meaning);
                    $sinhala[] = $item;
                    sleep(2);
                }
            }
        }

        $myfile = fopen($path, "a");
        if(count($meanings)){
            $meanings = array_unique($meanings);
            $strSave = implode(',', $sinhala).' '.
                str_replace(' ', '', implode(',', $meanings));
        }else{
            $strSave = '-';
        }

        fwrite($myfile, "\n".$word.'_ '.$strSave);
        fclose($myfile);
        return $sinhala;
    }
}

```

The combinations of the words are determined by an algorithm which is newly implemented and the foundation of the word possibilities are generated from the following function.

Ex: ගොනා,ගෝනා,ගොනැ,ගෝනැ,ගොන,ගෝන

```

function getCombs($arrays, $x = 0) {
    if (!isset($arrays[$x])) {
        return array();
    }
    if ($x == count($arrays) - 1) {
        return $arrays[$x];
    }
    $tmp = $this->getCombs($arrays, $x+ 1);
    $result = array();
    foreach ($arrays[$x] as $v) {
        foreach ($tmp as $t) {
            $result[] = is_array($t) ?
                array_merge(array($v), $t) :
                array($v, $t);
        }
    }
    return $result;
}

```

```
}
```

6.3 Dictionary

Sinhala and English dictionaries are implemented in the same manner which is to connect to a dictionary file and search for the word and return the meaning of the word. Code snippet which is used to connect Sinhala English Dictionary is as follows,

```
/**
 * Returns english meaning and returns null when there is no meaning got the sinhala
 word
 * @param $string
 * @return array|void
 */
public static function getEngMeaning($string) {
    $path = 'DictionaryFiles/sinhalaWords.txt';
    if(exec("grep _".$string."_ ".$path)){
        $sch = curl_init('http://www.deranadictionary.com/?search='.$string);
        curl_setopt($sch, CURLOPT_RETURNTRANSFER, true);
        $response = curl_exec($sch);
        curl_close($sch);
        $dom = new DOMDocument('1.0', 'UTF-8');
        $internalErrors = libxml_use_internal_errors(true);
        $dom->loadHTML($response);
        libxml_use_internal_errors($internalErrors);
        $xpath = new DOMXPath($dom);
        $classname = 'col-12 col-sm-12 col-md-12 col-lg-12 col-xl-12';
        $elements = $xpath->query("//*[contains(concat(' ', normalize-space(@class), ' '), ' $classname ')]");
        $meanings = [];
        foreach ($elements as $item) {
            $ms = $item->getElementsByTagName('a');
            foreach ($ms as $m) {
                $meanings[] = $m->textContent;
            }
        }
        return $meanings;
    }
}
```

6.4 Similar Domain Names Generator

```
public static function getSimilarDomainNames($domainName) {
    $url = 'https://wordsapiv1.p.mashape.com/words/'.rawurlencode($domainName);
    $verbIsHidden = array_key_exists($domainName) ;

    $headers = [
        'X-Mashape-Key: '.$this->api_key,
```

```

        'Accept: application/json',
    ];

    $curl = curl_init($url);
    curl_setopt($curl, CURLOPT_RETURNTRANSFER, true);
    curl_setopt($curl, CURLOPT_HTTPHEADER, $headers);
    curl_setopt($curl, CURLOPT_TIMEOUT, $this->timeout);
    curl_setopt($curl, CURLOPT_SSL_VERIFYPEER, false);
    $response = curl_exec($curl);
    $statusCode = curl_getinfo($curl, CURLINFO_HTTP_CODE);
    curl_close($curl);

    if ($statusCode !== 200) {
        return false;
    }

    if (is_null($response) || empty($response)) {
        return false;
    }

    $data = json_decode($response, true);
    if ($verbIsHidden) {
        return $data;
    }
}

```

6.5 Feedback Module

```

public function ratedomainsuggestions(HTTPRequest $request){
    $rating = $request->postVar('rating');
    $domainName = $request->postVar('domainName');
    $searchedDomainName = $request->postVar('searchedDomainName');
    $myfile = fopen('DomainNameSuggestionsRatings/ratings.CSV', "a") ;
    fwrite($myfile, "\n".$searchedDomainName.','.$domainName.','.$rating);
    fclose($myfile);
}

public function updateSearchQueryFile($name){
    $path = 'SearchQueries/SearchQueries.txt';
    $strSessionID = session_id();
    if($line = exec('grep _'.$strSessionID.'_ ./'.$path)){
        exec('sed -i \'s/\'$line.'/\'$line.'-\'$name.'/g\' ./'.$path);
    }else{
        $myfile = fopen($path, "a") ;
        fwrite($myfile, "\n_".$strSessionID."_".date('Y-m-d H:i:s').",".$name);
        fclose($myfile);
    }
}

```

6.6 Summary

This chapter describes how the system to suggest meaningful domain names in the domains.lk search bar is implemented and integrated into the system. In the next chapter the evaluation process is discussed.

Chapter 7

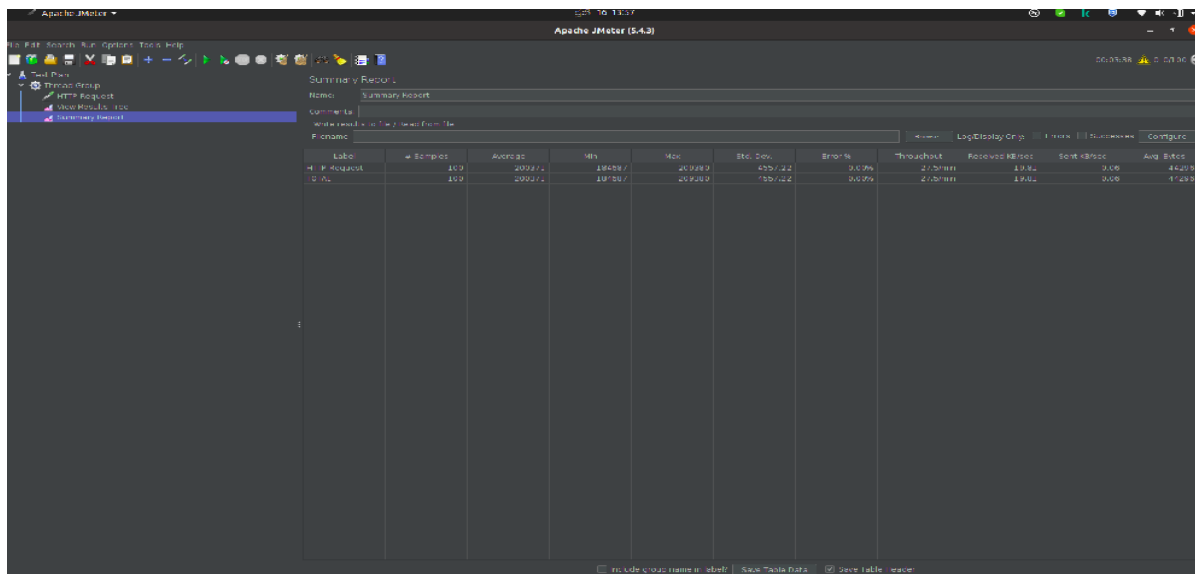
Evaluation

7.1 Introduction

The evaluation was done considering performance, accuracy and the meaningfulness of the suggested domain names. Evaluation was done in 3 stages.

1. Evaluation by QA Engineers of LK Domain Registry
2. Evaluation by Internal Staff Suggestions Ratings
3. Evolution by customers

Evolution of the system from the qa engineers are done first and foremost in order to check if there are any functional issues. Performance evaluation was done by the QA Engineers using a tool called Jmeter. The performance was at a low level prior to the implementation of the caching system.



Label	# Samples	Average	Min	Max	Std. Dev.	Error %	Throughput	Received KB/s	Sent KB/s	Avg. Bytes
HTTP Request Defaults	10	2002.1	19488.7	26928.9	4357.22	0.0%	2.75req/s	19.6s	0.0s	4420.9
	10	2002.1	19750.7	26928.9	737.22	0.0%	2.75req/s	19.6s	0.0s	1.433

The URL which is used to get domain name suggestions was tested five times with the same conditions.

Test	Response Time before Caching System Implementation (seconds)	After (seconds)
1	2.312	1.298
2	2.301	1.003
3	2.001	1.176
4	1.900	1.213
5	1.652	1.231

Table 7.1 Jmeter Responses

Meaningfulness of the suggestions were evaluated using the internal Software Team. Each of the 7 members was given 5 chances to search a domain and they had to rate the suggestions from 1 to 5. Average rating was taken as a measurement of accuracy of the final output of the system.

--	--	--

Chapter 8

Conclusion and Further Work

8.1 Introduction

In this research we have worked upon a solution to give meaningful domain suggestions in domains.lk search bar for its users. And our intension was to let the customer select the most suitable domain for themselves in their native language.

Our initial solution catered for Sinhala translations and transliterations due to the fact that most of the domain searches are in that language in Sri Lanka. But there is room and requirement

for further improvements since we have Tamil as another main language with its own characters and character rules.

In the current level of this solution it can give domain suggestions for Sinhala transliterations in English. For an example if a customer search a domain as ‘amma’, the system is capable enough to understand that the word is අම්මා and give suggestions as mother, mom etc.

8.2 Limitations

We have come up with our own method to find Sinhala meanings and Sinhala transliterations using set of rules defined only for this project. Since there are no common standards for such rules our set of rules may not be able to use in a different system which has the same requirement. But any other system can certainly use the methodology we have used with their own set of translation and transliteration rules.

Also the logics and algorithms we have used in this system are quite tricky and requires much more processing power compared to a simple web application. So a fully sophisticated domain search bar would require high-end server performance to operate concurrently on several domain searches inserted by different client.

8.3 Future Developments

This particular system developed targeting Sinhala translations and transliterations, but as a further work anybody can develop the same system targeting the Tamil translations and Transliterations.

Also in our character rules we have considered only the common scenarios to define translation and transliteration rules. But in future this system can be improved to cover all the possible scenarios.

Ex: We covered rules for common words such as සමනලයා, but we did didn’t set rules for possibilities such as සමනලයා, සමනලයා etc. Meaningless character combination of words can be eliminated by verifying with the help of dictionary module.

The advantage of this expansion would be that we will not miss out any word in the vocabulary. Apart from that we can improve this system to provide similar domains with plurals, synonyms etc in both Sinhala and English languages.

Another important improvement of this system could be that it will be able to cater new words adding to each languages vocabulary. Since the system is using a dictionary module as the base for words repository a dynamic dictionary module will make this a reality.

8.4 Summary

This chapter concludes the thesis by providing a basic explanation about the solutions we have provided in this research and limitations involved in further development of the solutions. And here it has giving some further improvement which can be added to expand this solution to a reasonably complete domain search bar which support all native languages in Sri Lanka.

References

- [1] LK Domain Registry, "Overview," LK Domain Registry, 2020. [Online]. Available: <https://www.domains.lk/index.php/about-us/overview>. [Accessed 17 July 2021].
- [2] M. Meyler, "A-Z of Sri Lankan English: S is for Singlish," GROUNDVIEWS, 14 06 2012. [Online]. Available: <https://groundviews.org/2012/06/14/a-z-of-sri-lankan-english-s-is-for-singlish/>. [Accessed 17 07 2021].
- [3] Cambridge University Press, Cambridge University Press, 2021. [Online]. Available: <https://dictionary.cambridge.org/dictionary/english/transliteration>. [Accessed 17 July 2021].
- [4] N. Halai, "Making use of bilingual interview data: Some experiences from the field," in *eCommons@AKU*, Karachi, 2007.
- [5] K. Regmi, J. Naidoo and P. Pilkington, "Understanding the Processes of Translation and Transliteration in Qualitative Research," *International Journal of Qualitative Methods*, vol. 9, no. 1, pp. 16-26, 2010.
- [6] L. Bilge, E. Kirda, C. Kruegel and M. Balduzzi, "EXPOSURE: Finding Malicious Domains Using Passive DNS Analysis," *NDSS*, pp. 1-17, February 2011.
- [7] T. Fonseka, R. Naranpanawa, P. Ravinga and U. Thayasivam, "English to Sinhala Neural Machine Translation," in *IALP 2020*, Kuala Lumpur, 2020.
- [8] N. Donges, "A Guide to RNN: Understanding Recurrent Neural Networks and LSTM," 16 June 2019. [Online]. Available: <https://builtin.com/data-science/recurrent-neural-networks-and-lstm>. [Accessed 17 July 2021].

- [9] Google Cloud; How to Guides, "Translating text (Basic)," Google Cloud, 8 July 2021.
[Online]. Available: <https://cloud.google.com/translate/docs/basic/translating-text>.
[Accessed 17 July 2021].
- [10 B. Hettige and K. Asoka, "Transliteration System for English to Sinhala," in *ICIIS 2007*,
] Peradeniya, 2007.
- [11 J. R. Thomas, S. K. Bharti and K. S. Babu, "Automatic Keyword Extraction for Text
] Summarization in e-Newspapers," 2010. [Online]. Available:
https://www.researchgate.net/publication/306891181_Automatic_Keyword_Detection_for_Text_Summarization_in_e-Newspapers. [Accessed 17 July 2021].
- [12 O. Khriyenko, "Customer Feedback System: Evolution towards semantically-enhanced
] Systems," in *Proceedings of the 11th International Conference on Web Information Systems and Technologies*, Finland, 2015.
- [13 M. F. Akan, "Transliteration and Translation from Bangla into English: A Problem
] Solving Approach," *British Journal of english Linguistics*, vol. 6, no. 6, pp. 1-21, 2018.
- [14 N. Durrani, H. Sajjad, A. Fraser and H. Schmid, "Hindi-to-Urdu Machine Translation
] Through Transliteration," Institute for Natural Language Processing, University of Stuttgart, Stuttgart.