AUTOMATED DOCTOR APPOINTMENT MANAGEMENT SYSTEM FOR MEDICAL DOMAIN IN SINHALA USER(ADAMS).

2020-175

Project Proposal Report

D.D.S. Rajapakshe

(U.L.N.P. Uswatte, K.N.B. Kudawithana, N.A.B.D. Nishshanka)

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology

Sri Lanka

March 2020

AUTOMATED DOCTOR APPOINTMENT MANAGEMENT SYSTEM FOR MEDICAL DOMAIN IN SINHALA USER(ADAMS).

2020-175

Project Proposal Report

B.Sc. (Hons) Degree in Information Technology

Department of Information Technology

Sri Lanka Institute of Information Technology
Sri Lanka

March 202

DECLARATION

We declare that this is our own work and this proposal does not incorporate without acknowledgement of any material previously submitted for a degree or diploma in any other university or Institute of higher learning and to the best of our knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Name	**IT Number**	**Signature**	Date
D.D.S Rajapakshe	IT16234062		25/02/2020
U.L.N.P. Uswatte	IT17029278		25/02/2020
Kudawithana K.N.B	IT17255820		25/02/2020
Nishshanka N.A.B.D	IT17043656		25/02/2020

Supervisor

---------- ----------

Prof. Koliya Pulasinghe Date

Co-Supervisor

---------- ----------

Ms. Vijani Piyawardana Date

ABSTRACT

In Sri Lankan medical systems are in the English language (Doc990, e-Channelling PLC) and majority of the Sri Lankan cannot understand the English terms (According to 2012 Department of Census and Statistics server (DCS)) and they do not possess user experience about the web Information system. the majority medical information systems are now looking forward to the automated Sinhala conversation system to help the customers (patients) to make a decision on their health issues.

Human Computer Interaction (HCI) is the majority suitable method to interact with humans and machines. Developing with Natural Language Processing (NLP) system, it can satisfy users. According to DCS survey 70.5 percent of those using the internet had connected through their smartphone or tablet in the first half of 2018, up from 59 percent a year earlier, while 27.1 percent used a desktop or laptop, down from 38.1 percent [1].

Thus, the majority suitable Human interaction method in Sri Lanka is a mobile application. The main objective of this system is developing an **A**utomated **D**octor **A**ppointment **M**anagement **S**ystem for medical domain in Sinhala user (**ADAMS**). and as a sub-objective, Determine the optimum identification to sort out the Human Diseases (patient is given), Keyword analyser and database query generation, RASA framework Implementation and Voice and Text Dialogue generation and blockchain creation. This document presents a survey, starting from the literature review, Mobile application architecture, evaluation methods/criteria, comparison of evaluation methods, Project requirements and Budget Justification.

Keywords -: RASA, NLU, NLP, AI, Ontology, Blockchain

TABLE OF CONTENT

LIST OF FIGURES

LIST OF TABLES

1 INTRODUCTION

ADAMS mobile conversation application provides HCI solution to support the Doctor channelling service of hospitals and deliver convenience to patients. The system has been designed to safeguard both patient information as well as doctor information.

Through ADAMS, the patient has the facility to channel a doctor without visiting the hospital using this Mobile application and further facilitates the option to get, Disease identification and Doctors, Hospitals and Channelling pieces of information. ADAMS is compatible only voice command and also can understand the Sinhala language.

It is a comprehensive and sophisticated system with various modules. However, it is designed in a user- friendly manner to provide convenience to its customers. The system is updated in real time, so that patients can see the most updated schedule of a consultant, make appointment with a consultant and conduct the payment at their own convenience.

1.1 Background & Literature Survey

1.1.1 Background

The Internet is a huge platform for people to use and communicate with each other. It is the global lattice of governments, companies, organizations, individual units of networks and so on. Became a few years, the internet has become a universal and common source of information for people globally. It has delineated business communication in new ways and, in the past few years, social media platforms and mobile phones have changed the way people use the Internet. Modern busy lifestyles, people are trying to use their time is effective. Therefore, most of the people are looking for an online-based service. As a DSC survey, 70.5 percent of those using the internet have connected through their smartphone or tablet.

One of the medical industry-based online IT solutions is e-channel. What is an e-channel? It is a service provider and providing a platform for patients to get an appointment with a doctor over the online and it improved the overall effectiveness and efficiency of patients and Hospitals. Sri Lanka only has two e-channels (Figure 1.1 : Doc990 & e-Channeling) Those systems are in the English language and it will be compatible with keyboard input.

Figure 1.1 : Doc990 & e-Channelling

1.1.2 Literature Survey

1. Ontology-driven semantic profiling and retrieval in medical information systems [1].

This paper presents a OntoMOVE Framework –'Ontologies on the MOVE' – that combines the use of ontology driven annotations and the application of a novel sub-ontology extraction methodology (called MOVE) for achieving interoperability and improving the effectiveness of information retrieval for the specific MIS domain [1]. It is extracting a sub-ontology that meets the user requirements [1] and is based on the Unified Medical Language System(UMLS) knowledge sources(User Recode Ex: patient records, administrative health data, bibliographic), namely the UMLS Semantic Network (UMLS-SN) and UMLS Metthesaurus [1]. The methodology of sub-ontology based semantic requirement profiling and information retrieval is generally applicable in any domain where it is possible to annotate resource-sets [1]. Designed OntoMOVE to be specifically applied in the medical information systems domain [1].

2. Owlready: Ontology-oriented programming in Python with automatic and high-level constructs for biomedical ontologies [2].

Research output is to design a tool for accessing easily the entities of an OWL ontology, with high-level constructs helping with biomedical ontologies [2]. The ontologies rely on the open-world assumption, whereas the medical reasoning must consider only evidence-based medical knowledge as true [2]. They designed a Python module for ontology-oriented programming [2]. It allows access to the entities of an OWL ontology as if they were objects in the programming language. They propose a simple high-level syntax for managing classes and the associated "role-filler" constraints [2]. They also propose an algorithm for performing local closed world reasoning in simple situations [2]. They developed Owlready, a Python module for high-level access to OWL ontologies [2]. The paper describes the architecture and the syntax of the module version 2 [2]. It details how they integrated the OWL ontology model with the Python object model [2]. The paper provides examples based on Gene Ontology (GO) [2]. We also demonstrate the interest of Owlready in a use case focused on the automatic comparison of the contraindications of several drugs [2]. This use case illustrates the use of the specific syntax proposed for manipulating classes and for performing local closed-world reasoning [2]. Owlready is an Open-Source software.

3. Applying Ontology to WBAN for mobile applications in the context of sport exercises [3].

This paper details a new mobile application for runners that calculates the burned calories based on a dedicated ontology [3]. Some of the main goals for using this ontology are interoperability management, the ease of information management and

the minimization of the battery consumption of the mobile device by requiring less network resources and data processing due to the use of ontologies [3]. They used WSN generic ontology [3].

Application, they are used the Protégé 4.3 for defining the ontology model and the Jena API for implementing the ontology on the mobile application and the server [3]. Jena API is the Semantic Web framework for Java. Rules are added to the ontology using Semantic web Rule Language SWRL [3]. The Pellet reasoner is used in order to extract inferred information from the ontology [3]. SPARQL is dedicated to querying knowledge from the ontology [3].

4. The Unified Medical Language System (UMLS): integrating biomedical terminology [4].

Research projects such as TAMBIS [5] have addressed the specific issue of integrating disparate resources for bioinformatics through a model of domain knowledge [4]. While TAMBIS [5] provides a framework for integrating resources, its coverage is currently limited to five sources [4]. In this paper, they present a different approach to information integration through terminology integration [4]. The Unified Medical Language System (UMLS), developed over more than 15 years, which covers the entire biomedical domain [4]. The UMLS was developed by the National Library of Medicine (NLM) as 'an effort to overcome two significant barriers to effective retrieval of machine-readable information' the variety of names used to express the same concept and the absence of a standard format for distributing terminologies [4]. By integrating more than 60 families of biomedical vocabularies, the UMLS Metathesaurus currently provides not only an extensive list of names (2.5 million) for its 900 551 concepts but also over 12 million relations among these concepts [4]. Its scope is broader and its granularity finer than that of any of its source vocabularies [4].

5. SWRL rule-selection methodology for ontology interoperability [6].

In this paper, they focus on improving the performance of queries addressed over ontology alignments expressed through SWRL rules [5]. Indeed, when considering the context of executing queries over complex and numerous alignments, the number of SWRL rules highly impacts the query execution time [5]. Moreover, when hybrid or backward chaining the reasoning is applied, the query execution time may grow exponentially [5]. Still, the reasoners involved deliver performant results (in terms of execution time) when applied over reduced and simpler rule sets [5]. Based on this statement, and to address the issue of improving the query execution time, they describe a novel approach that allows, for a given query, to ignore unnecessary rules [5]. The proposed Rule Selector (RS) is a middleware between the considered systems

and the reasoner present on the triple store side [5]. This paper is summarized as follows [5].

1. They describe their approach for improving the performances of queries addressed over ontology alignments described in terms of SWRL rules [5].
2. They define a novel architecture for our rule selector (RS), architecture that contains two main modules [5].
    a. The Rule Pre-processing module is responsible for transforming existing SWRL rules describing the ontology alignment into canonical rules [5]. In other words, this module performs a normalization of the existing ruleset [5].
    b. The Query execution module aims at 2 goals [5].
        i. Identify the necessary and sufficient rule subset that allows answering a given query [5].
        ii. Eventually, remove rules from the previously identified rule set (step 2.b.i) that produce rule cycles (e.g. A → B → A) [5].
3. They specify the algorithms used by the two previously described modules(Their own Algorithm) [5].
    a. Algorithm for the Rule Pre-processing module [5].
    b. Algorithm for the Query execution module [5].
4. They provide an evaluation of the performance of their approach, by applying 4 different queries over an alignment of 2 ontologies [5]. Moreover, they compare their approach with existing ones [5].

6. Ontology-Oriented Programming for Biomedical Informatics [7].

The research project presenting information about new drugs to physicians [7]. To guarantee independent information, this information is produced automatically by comparing the new drug to the older ones, using criteria such as efficacy in clinical trials, contraindications, and known adverse effects [7]. The automatic comparison of drug properties like contraindications is not easy because contraindications are often expressed at different levels of granularity in drug databases, therefore they designed an ontology of contraindications [7]. They were experienced difficulties in connecting the ontology to the database and the website, and more generally to manipulate the ontology in the computer program [7]. They are a reflection about methods for accessing ontologies, and to the development of OwlReady [7], a Python 3 module for ontology-oriented programming with full class-support, including dynamic class creation and classification of classes at run time using the HermiT OWL reasoner [7]. OwlReady supports OWL 2.0 [7]. An experimental feature allows them to automatically generate dialog boxes for editing individuals and classes in ontology [7]. Populating the ontology from the results of SQL requests, calling the reasoner, and generating HTML pages from the ontology was easy, from a computer-science point of view [7]. The automatic generation of dialog boxes was very convenient, it allowed them to modify the ontology without having to manually update the editing interface [7]. The computation time of ontology-oriented programming with OwlReady and

Python was higher than a traditional API in Java [7]. However, the difference is insignificant compared to the time consumed by the HermiT OWL reasoner, or the time they gained during software development (shorter source code implies faster development) [7].

7. Continuous Sinhala Speech Recognizer. [8]

The objective of this research was to apply existing continuous speech recognition mechanisms to develop a continuous Sinhala speech recognizer, which is not bound to any specific domain [8]. The building of an ASR system mainly consists of designing two models, namely the Acoustic Model and the Language Model [8]. The Acoustic model is responsible for detecting phonemes which were spoken, and the Language Model is responsible for detecting connections between words in a sentence [8]. The implementation of the ASR using Hidden Markov Model Toolkit (HTK) which was developed by Cambridge University, UK (Young et al, 2006) [8]. HTK is primarily designed for speech recognition using Hidden Markov Models (HMMs) [8].

8. Diabot: A Predictive Medical Chatbot using Ensemble Learning [9].

The paper presents a state-of-the-art Diabot design with an undemanding front-end interface for a common man using React UI, RASA NLU based text preprocessing, quantitative performance comparison of machine learning algorithms as standalone classifiers and combining them all in a majority voting ensemble [9]. It is observed that the chatbot is able to interact seamlessly with all patients based on the symptoms sought [9]. The accuracy of the Ensemble model is balanced for general health prediction and highest for diabetes prediction among all weak learners considered which provides motivation for further exploring ensemble techniques in this domain [9]. At a high level, the system consists of a front-end User Interface (UI) for the patient to chat with the bot [9]. It is a developed using react UI platform which uses HTML and JavaScript [9]. The chatbot communicates with the NLU engine at the backend via API calls [9]. The NLU engine is developed using RASA NLU platform [9]. Two models- one for generic health prediction and another for advanced diabetes prediction are trained at the backend using the general health dataset and the Pima Indian diabetes dataset which provides the necessary diagnosis decision to the NLU engine which is trained to provide an output based on user queries [9].

9. Chatbot, a chatbot for the Spanish “La Liga” [10].

This work describes the development of a social chatbot for the football domain [10]. The chatbot, named chatbot, aims at answering a wide variety of questions related to the Spanish football league “La Liga” [10]. A chatbot is deployed as a Slack client for text-based input interaction with users [10]. One of the main Chatbol’s components, an NLU block, is trained to extract the intents and associated entities related to user’s questions about football players, teams, trainers and fixtures [10]. The information for

the entities is obtained by making SPARQL queries to the Wikidata site in real-time [10]. Then, the retrieved data is used to update the specific chatbot responses [10]. As a fallback strategy, a retrieval-based conversational engine is incorporated to the chatbot system [10]. It allows for a wider variety and freedom of responses, still football oriented, for the case when the NLU module was unable to reply with high confidence to the user [10]. The retrieval-based response database is composed of real conversations collected both from an IRC football channel and from football-related excerpts picked up across movie captions, extracted from the Open Subtitles database [10].

10. Quinn: Medical Assistant for Mental Counseling using Rasa Stack [11].

A chatbot called Quinn is developed to serve the users as a personal assistant to solve the problem [11]. It is a chat robot that interacts with the user to analyse their thoughts or state of mind and suggests appropriate solutions [11]. Thus, the conversational service can provide individuals with personalized counselling [11]. One-to-one conversation can effectively resolve the isolation [11]. Quinn notifies the users' dangerous status when there is an accidental mental disorder, such as panic and suicidal impulse [11]. Besides, the system observes the mood swings continuously for users who are manic-depressive [11]. This conversational service for psychiatric counseling adapts methodologies to understand counselling contents based on high-level natural language understanding (NLU) using Rasa Stack [11].

11. Text-based Healthcare Chatbots Supporting Patient and Health Professional Teams: Preliminary Results of a Randomized Controlled Trial on Childhood Obesity [12].

Health professionals have limited resources and are not able to personally monitor and support patients in their everyday life [12]. Against this background and due to the increasing number of self-service channels and digital health interventions, they investigate how text-based healthcare chatbots (THCB) can be designed to effectively support patients and health professionals in therapeutic settings beyond on-site consultations [12]. Technology-based self-service channels and digital health interventions have the potential to support patients in their everyday life and health professionals likewise [12]. Although there are scalable self-service channels in the form of digital voice assistants [12]. They used the Mobile Coach chatbot framework to develop a project and it text base chatbot [12]. It has already been evaluated in the public health context and provides a modular architecture and rule engine for the design of fully automated digital health interventions [12]. It also supports the implementation of RCTs (Randomized controlled trial) and micro-randomized trials. Also, it can hadal a patient and health professional live communication [12].

This paper presents a OntoMOVE Framework –'Ontologies on the MOVE' – that combines the use of ontology driven annotations and the application of a novel sub-ontology extraction methodology (called MOVE) for achieving interoperability and improving the effectiveness of information retrieval for the specific MIS domain [1]. It is extracting a sub-ontology that meets the user requirements [1] and is based on the Unified Medical Language System(UMLS) knowledge sources(User Recode Ex: patient records, administrative health data, bibliographic), namely the UMLS Semantic Network (UMLS-SN) and UMLS Metthesaurus [1]. The methodology of sub-ontology based semantic requirement profiling and information retrieval is generally applicable in any domain where it is possible to annotate resource-sets [1]. Designed OntoMOVE to be specifically applied in the medical information systems domain [1].

Research output is to design a tool for accessing easily the entities of an OWL ontology, with high-level constructs helping with biomedical ontologies [2]. The ontologies rely on the open-world assumption, whereas the medical reasoning must consider only evidence-based medical knowledge as true [2]. They designed a Python module for ontology-oriented programming [2]. It allows access to the entities of an OWL ontology as if they were objects in the programming language. They propose a simple high-level syntax for managing classes and the associated "role-filler" constraints [2]. They also propose an algorithm for performing local closed world reasoning in simple situations [2]. They developed Owlready, a Python module for high-level access to OWL ontologies [2]. The paper describes the architecture and the syntax of the module version 2 [2]. It details how they integrated the OWL ontology model with the Python object model [2]. The paper provides examples based on Gene Ontology (GO) [2]. We also demonstrate the interest of Owlready in a use case focused on the automatic comparison of the contraindications of several drugs [2]. This use case illustrates the use of the specific syntax proposed for manipulating classes and for performing local closed-world reasoning [2]. Owlready is an Open-Source software.

12. Festival-si: A Sinhala Text-to-Speech System.

The conversion of text to speech involves many important processes. These processes can be divided mainly in to three stages; text analysis, linguistic analysis and wave-form generation. The text analysis stage is responsible for converting the non-textual content into text. This stage also involves tokenization and normalization of the text; identification of words or chunks of text [13].

Text normalization establishes the correct interpretation of the input text by expanding the abbreviations and acronyms. This is done by replacing the nonalphabetic characters, numbers, and punctuation with appropriate text-strings depending on the context. The linguistic analysis stage involves finding the correct pronunciation of words, and assigning prosodic features (e.g. phrasing, intonation, stress) to the phonemic string to be spoken [13].

The final stage of a TTS system is waveform generation which involves the production of an acoustic digital signal using a particular synthesis approach such as formant synthesis, articulatory synthesis or waveform concatenation. The text analysis and linguistic analysis stages together are known as the Natural Language Processing (NLP) component, while the waveform generation stage is known as the Digital Signal Processing (DSP) component of a TTS System [14].

According to this implementation and evaluation of a Sinhala text-to-speech system based on the diphone concatenation approach. The Festival framework was chosen for implementing the Sinhala TTS system [13].

TTS systems have been developed using the Festival framework for different languages including English, Japanese, Welsh, Turkish and Hindi, Telugu, among others. However, no serious Sinhala speech synthesizer has been developed this far [13].

Sinhala Phonemic Inventory and Writing System :

The Sinhala phonemic inventory :

Sinhala is one of the official languages of Sri Lanka and the mother tongue of the majority - 74% of its population. Spoken Sinhala contains 40 segmental Sinhala [13].

phonemes; 14 vowels and 26 consonants as classified below in Table 1 and Table 2.

	Front		Central		Back	
	Short	Long	Short	Long	Short	Long
High	i	i:			u	u:
Mid	e	e:	ə	ə:	o	o:
Low	æ	æ:			a	a:

Figure 1.2: Spoken Sinhala vowed classification

		Lab.	Den.	Alv.	Ret.	Pal.	Vel.	Glo.
Stops	Unvoiced	p	t̪		ʈ		k	
	Voiced	b	d̪		ɖ		g	
Affricates	Unvoiced					ʧ		
	Voiced					ʤ		
Pre-nasalized voiced stops		ᵐb	ⁿd̪		ᶯɖ		ᵑg	
Nasals		m		n		ɲ	ŋ	
Trill				r				
Lateral				l				
Fricatives	Unvoiced	f	s			ʃ		h
	Voiced	v						
Approximants						j		

Figure 1.3 : Spoken Sinhala constant classification

There are four nasalized vowels occurring in two or three words in Sinhala. They are /a/, /a:/, /æ̃/ and /æ̃:/ [4]. Spoken Sinhala also has following Diphthongs; /ai/, /iu/, /eu/, /au/, /ou/, /au/, /ui/, /ei/, /oi/ and /ai/ [13].

The Sihala writing System :

The Sinhala character set has 18 vowels, and 42 consonants as shown in Table 3 [13].

Vowels and corresponding vowel modifiers (within brackets):

අ ආ(ා) ඇ(ැ) ඈ(ෑ) ඉ(ි) ඊ(ී) උ(ු) ඌ(ූ) ඍ(ෘ) ඎ(ෲ) ඏ(ෟ) ඐ(ෳ) එ(ෙ) ඒ(ේ) ඓ(ෛ) ඔ(ො) ඕ(ෝ) ඖ(ෞ)

Consonants:

ක ඛ ග ඝ ඞ ඟ ච ඡ ජ ඣ ඤ ඥ ඦ ට ඨ ඩ ඪ ණ ඬ ත ථ ද ධ න ඳ
ප ඵ බ භ ම ඹ ය ර ල ව ශ ෂ ස හ ළ ෆ ං ඃ

Special symbols: ෟ ෲ ෳ ඥ
Inherent vowel remover (Hal marker): ්

Figure 1.4 : Sinhala Character set

Sinhala characters are written left to right in horizontal lines. Words are delimited by a space in general[18]. Vowels have corresponding full-character forms when they appear in an absolute initial position of a word. In other positions, they appear as 'strokes' and, are used with consonants to denote vowel modifiers [14].

Diphone Database Construction :

Designing, recording, and labeling a complete diphone database is a laborious and a time-consuming task. The overall quality of the synthesized speech is entirely dependent on the quality of the diphone database. This section describes the methodology adopted in the construction of Sinhala diphone database [13].

Natural Language Processing Modules :

When building a new voice using Festvox, templates of the natural language processing modules required by Festival are automatically generated as Scheme files. The NLP modules should be customized according to the language requirements [13].

The NLP modules should be customized according to the language requirements. Hence, the language specific scripts (phone, lexicon, tokenization) and speaker specific scripts (duration and intonation) can be externally configured and implemented without recompiling the system [13]. The NLP related tasks involved when building a new voice are:

- Defining the phone-set of the language
- Tokenization and text normalization
- Incorporation of letter-to-sound rules
- Incorporation of syllabification rules Assignment of stress patterns to the syllables in the word
- Phrase breaking
- Assignment of duration to phones
- Generation of f0 contour

13. Health Record Management through Blockchain Technology.

A blockchain is a decentralized, distributed, immutable, shared & tamperproof data structure to store a continuously growing list of the transaction. When consider Blockchain as a register containing transaction records into timestamp blocks. Each block has its own identity called cryptographic hash [15].

Each block is provided with the hash value of the block that came before it. Because of which a link is established between the blocks, thus creating a chain of blocks. We get a clear picture of how a Blockchain works only when we explore how a Blockchain network runs. It is a peer-peer network where each node holds the record of each transaction that's been carried out in the network [15].

To carry out transactions each node has its own wallet. The interaction between the user and the network is via a pair of private & public keys (Cryptographic keys). A private key is used to sign their own transaction whereas the public key is visible to all the nodes in the network [15].

Someone who wants to carry out transaction should send a message by signing the transaction with their private key, when this is combined with the public key it forms a digital signature. This transaction is broadcasted onto the Blockchain network where it is verified by the miners [15].

Miners are the nodes in the Blockchain with high processing power. Miners make the transaction unaltered & irreversible using a consensus algorithm called Proof of work. There is a competition among miners to generate a valid block and the one who generates a valid block is rewarded [15].

A block of the transaction is approved only when it is verified by all the miners in the network and if more than 50% of the miners validate the transaction then this block is considered as a valid block and is added to the longest Blockchain [15].

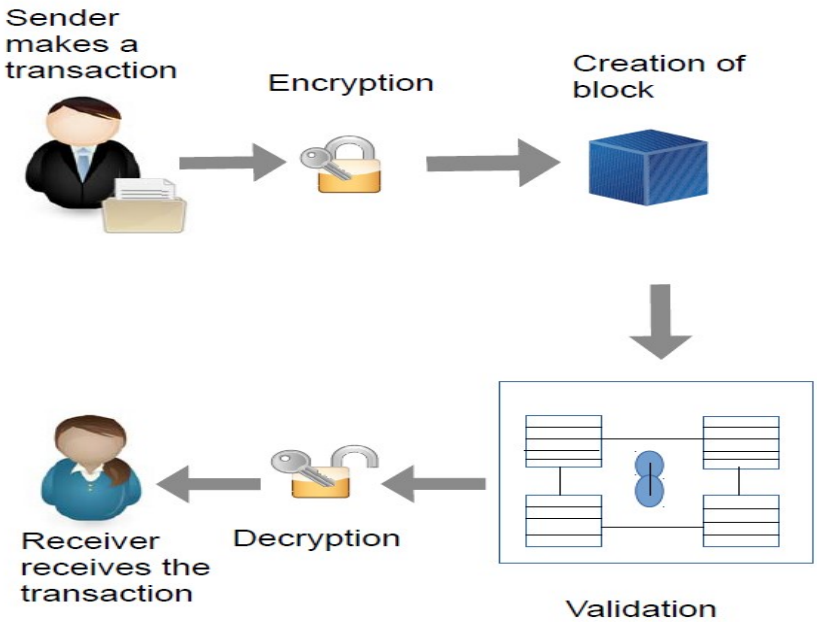

Figure 1 : Working of Block-Chain Technology [19]

1.1.3 Research Gap

Feature	***ADAMS***	***WBAN*** [3].	***Diabot*** [9].	***La Liga*** [15]	***Quinn*** [11].	***THCB*** [12].	***Festive-si:***	***MaryTTS***
Sinhala voice to Text	Yes	No	No	No	No	No	No	No
User Authentication and session	Yes	Yes	Yes	Yes	Yes	Yes	No	No
RASA connectivity with Flutter	Yes	No	No	No	No	No	No	No
Ontology	Yes	Yes	No	Yes	No	No	No	No
Ontology medical domain	Yes	Yes	No	No	No	No	No	No
Mobile Chatbot	Yes	No	Yes	Yes	Yes	Yes	No	No
RASA Framework	Yes	No	Yes	Yes	Yes	No	No	No
Create Sinhala voice module	Yes	No	No	No	No	No	Yes	Yes
Train Sinhala voice modules	Yes	No	No	No	No	No	Yes	Yes
Sinhala medical facts dictionary	Yes	No	No	No	No	No	No	No
Implement Sinhala	Yes	No	No	No	No	No	No	No

text using medical terms knowledge								
Phoneme mapping	Yes	No	No	No	No	No	Yes	Yes
Dialogue generation & display the conversation in mobile interface	Yes	No	No	No	No	No	No	No
Sinhala Text to Speech conversion	Yes	No	No	No	No	No	Yes	Yes
Sinhala voice output	Yes	No	No	No	No	No	Yes	Yes
Used of Block-Chain mechanism to manage historical and updating data	Yes	No	No	No	No	No	No	No

Table 1-1 : Research Gap

2 RESEARCH PROBLEM

Nowadays in Sri Lanka, all of the e-channelling systems are based on the English language. Mostly e-channelling systems are used by people in western province comparing to other provinces. The reason is the erudition that they have is very impecunious in the English language.

Sometimes, people having diseases such as Short-Term and Long-Term Incapacitation and People with dyslexia, are incapable of using communicate an E-channel system.

Most of the Sri Lankan e-channelling systems are web-predicated and utilizing the web-predicated system in mobile is Minimize Celerity, sometimes browser support is very impuissant and not facile to utilize on mobile. It is not user-friendly for mobile.

Most of the patients don't know what the specialization of the doctors is, and who are the best doctors of their diseases. They only know the diseases that they have at that moment.

Some of the patients know doctor's name but they will not know the details about the doctor, such as the hospital where, the doctor is available, time schedules of the doctors.

Some patients know all the details and they want to get an appointment with doctors, but they couldn't make it, because they will face interaction problems with a system, the system will perform the English language they can't understand it.

Patient have to pace on above difficulties in this kind of a challenging situation.

3 OBJECTIVES

3.1 Main Objective

Human-computer interaction (HCI) is a major role in interfaces between humans and machines. Dialogue system, conversational system, Chatbot, voice controller interface, and personal assistants' system are the example of HCI system that has been developed to Interaction with a human using Natural language. That type of system helps users (In our system user is a patient) to find useful information for their needs. Thus, a number of hospital organizations prefer to use an automated system such as a chatbot to automate their customer service. Because that type of system is easily delivered to the user. As the first solution, we identified to develop a Mobile chat form for interaction with a patient. For some disabled people who might struggle, or find it impossible, to work with a mouse or keyboard, speech recognition enables them to communicate with a system effectively. It lends a hand to those with physical impairments and dyslexia and also reduced time and it less stressful than conventional handwriting or typing. Therefore, the second solution is which allows a user to perform voice commands such as text commands. Today mobile usage is increased in the

world, and most people are using smartphones. Also, Sri Lankan people are the same and most of them used the internet on their smartphones. As a solution, we developed a mobile app, which is easy to deliver a better service in patients. In Sri Lankan population is 22,576,592 (July 2018 est.). in 87% percentage can speak the Sinhala Language. 23.8% lower percentage of people can speak the English language (2012 est.) and 91.9% percentage can read and write. (Copyright @ Department of Census and Statistics server details) According to the survey Most of the Sri Lankan people can communicate and understand the Sinhala language. Thus, as the language problem solution is, we are using the Sinhala language. Therefore, we used Natural Language Processing (NLP). NLP is a way of computers to analyse, understand and derive meaning from a human language such as English, Sinhala and etc. Thus, the need for using artificial intelligence has been increasing due to the needs of automated services. However, devolving smart Mobile chat form that can respond at the human level is challenging. And it can give to the user a better service and solve the user problem. In our research, we identified user problems, is getting doctor visiting appointments for patients and normal person doesn't have a knowledge of human diagnosing. therefore, our main objective is developing an Automated Doctor appointment management system for medical domain in Sinhala user (ADAMS).

3.2 Specific Objectives

3.2.1 DOI-HD:

Determine the optimum identification to sort out the human diseases (patient is given). -:

Ontologies are widely used in the biomedical domain [2]. While many tools exist for the edition, alignment or evaluation of ontologies, few solutions have been proposed for ontology programming interface [2], i.e. for accessing and modifying an ontology within a programming language [2]. Existing query languages (such as SPARQL) and APIs (such as OWLAPI) are not as easy-to-use as object programming languages are [2]. Moreover, they provide few solutions to difficulties encountered with biomedical ontologies [2]. My objective was to design a tool for accessing easily the entities of an OWL ontology, with high-level constructs helping with biomedical ontologies [2]. As a description I want a Knowlagent base, therefore I used ICD API. ICD API allows programmatic access to the International Classification of Diseases (ICD) [13]. It is an HTTP based REST API [13]. The information for the entities is obtained by making Owlready Ontology framework and OWL rules to the ICD REST API site in real-time [13].

1. DOI-HD ontology for medical domain (DOI-HD-Ontology):
   A sub-objective, **D**etermine the **O**ptimum **I**dentification to sort out **H**uman **D**iseases (The Patient is a given) is one of the ones (**DOI-HD**). Most of the people doesn’t have clear idea about their unwellness. As core objective is to use a huge Owlready Knowledge Base using a sort out a human unwellness by DOI-HD-Ontology. There for design OWL rules for the ontology will help determine the

optimum identification to sort out Human Diseases. Owlready is an opensource and it is a python library. It used ontology in medical domain.

2. DOI-HD STT(UVP):
   Developing a **U**ser **V**oice **P**rocessing (**UVP**) system will Analyse a Sinhala voice command of the patient. The system is converted user voice to Sinhala text using Mozilla deep speech or google cloud API using. Using API can minimize an app loading time and can increased an efficient of app. Running so many services, system will be a lower efficient and can be increase app loading time.

3. DOI-HD User Authentication:
   protecting user sensitive data user validation is most important and those methods are mostly used in a mobile app Ex: Email verification, Face recognition and Image Encryption. Implementing firebases Authentication cloud function service can secure patient recode.

4. Plugging development:
   Mobile app interfaced is develop using flutter and it is independent of platform. So, have to develop a plugging for get connectivity with backend. The plugging of developing in a system, DOI-HD and RASA framework plugging.

3.2.2 Appointment Management System

1. Create Firebase Database:

Firebase is a trending Cloud based (BaaS) backend as a service which provides Real Time syncing with all the clients subscribed to the server at any given instance. Firebase supports multiple platforms/frameworks like AngularJS, Backbone.js, iOS7, OSX, Android and programming languages like Ruby, Node.js, Python, Java, PHP, Perl & JavaScript. Firebase relies on web sockets to update all the clients about any data changes instantly.so ADAMS can use Firebase as Database.

2. Translate Given Sinhala and English Sentence:

The Translator is used to translate an English base word into a Sinhala base word with the help of the bilingual dictionary. The translator translates subject, object and verb in the English sentence separately. This is a method we have used to reduce the complexity of the translation process. This translator is a simple one and it does not automatically handle the semantics of sentences. Note that, this stage can be supported by human intervention to generate the most appropriate translation for some words in a sentence. As such, handling semantic, pragmatic and Multiword expressions must be addressed with the support of humans, for which we introduce an intermediate editor.

3. Divide medical facts from given sentences:

In this process users converted sentences going to be separate because in this process need to separate medical facts for identify diseases. Identified verbs, objects and nouns are comparing with database storing in firebase and collect only important medical facts from those sentences. Gradient Algorithm is choosing for the medical facts' dividend process.

4. Generate Query from Identified Facts:

From those selected medical facts data should need to generate queries from these all things.

5. Flutter application to collect data about medical centers:

create simple flutter application for collect data about medical centers and all of these data will be saved in firebase database table,

3.2.3 Dialogue management

1. Implementation of the Sinhala text:

By gathering all the important medical facts of the user's input, query will be generated. Using that query have to implement the medical answer for the user's medical question. These import facts of the medical question use to identify the what is the problem that having to the user. According to these medical facts, response medical answer will be generated through the system. End of this process system Sinhala text will be created.

2. Dialogue management:

User's medical question and the response medical answer will be set as a conversation. Through this whole conversation between patient and the bot, have to manage as a dialogue.

3. Development of the Sinhala Voice Modules:

Sinhala voice modules will be developed using Sinhala voice commands. Corresponding sentences were recorded by a professional speaker. By using those Sinhala medical terms Sinhala voice module will be developed.
Mapping between Sinhala Voice Modules and the Sinhala Phoneme [1] (Text to Speech conversion)The conversion of text to speech involves many important processes. These processes can be divided mainly in to three stages; text analysis, linguistic analysis and wave-form generation.

The text analysis stage is responsible for converting the non-textual content into text. This stage also involves tokenization and normalization of the text; identification of words or chunks of text. Text normalization establishes the correct interpretation of the input text by expanding the abbreviations and acronyms. This is done by replacing the nonalphabetic characters, numbers, and punctuation with appropriate text-strings depending on the context. The linguistic analysis stage involves finding the correct

pronunciation of words, and assigning prosodic features (e.g. phrasing, intonation, stress) to the phonemic string to be spoken. The final stage of a TTS system is waveform generation which involves the production of an acoustic digital signal using a particular synthesis approach such as formant synthesis, articulatory synthesis or waveform concatenation.

4. Sinhala Voice output:

At the end of the above three processes Sinhala voice output will be here as a voice command. Sound waves will be here in smooth, quality and better speed that healthy level to the humans. Implementation of the Block-Chain related to the user registration functionality

The users who are involve with the system have to register to the system. As a patient or as a medical agent. All the records related to the users are maintained through a block-chain mechanism. Through this mechanism user registration details and the historical data will be managed.

When consider Blockchain as a register containing transaction records into timestamp blocks. Each block has its own identity called cryptographic hash. Each block is provided with the hash value of the block that came before it. Because of which a link is established between the blocks, thus creating a chain of blocks [15].

3.2.4 RASA Framework

1. Convert dialog into the json format:

The dialogue which includes patients' medical questions and response medical answers are move to the RASA NLU. In that case want some proper method to move this dialogue. There are two methods available. First one is Json and the other one is Markdown method. In here most suitable method is Json format. Because Jason format is compatible for the mobile application development rather than Markdown format. By using Json format method system will enrich training data. Then convert that Sinhala dialogue into the Json format.

2. Automatically json format generation:

Using Json format that data will move to the RASA NLU. After the retrieve of the dialogue into this stage, that Sinhala dialogue automatically convert in to the json format. This json query generation is a temporary process. It will not be storing any database. Intent and Entities available inside this json query.

3. Json query move to the RASA NLU:

Then dialogue automatically convert into the Json format and it will move to the RASA NLU. Inside this RASA NLU, extract that Json query. Intent and Entities available

inside this Json query. If the Json query extract inside this RASA NLU both entity and intent parts automatically move.

4. RASA NLU move to the RASA core:

Rasa NLU will be moved inside to the RASA core and it will extract inside this RASA core. Purpose of the RASA core is continuing the conversation. Rasa core will be managed patient's medical questions and response answers. It will identify the correct questions and generate reply for the related questions that patient asked.

4 METHODOLOGY

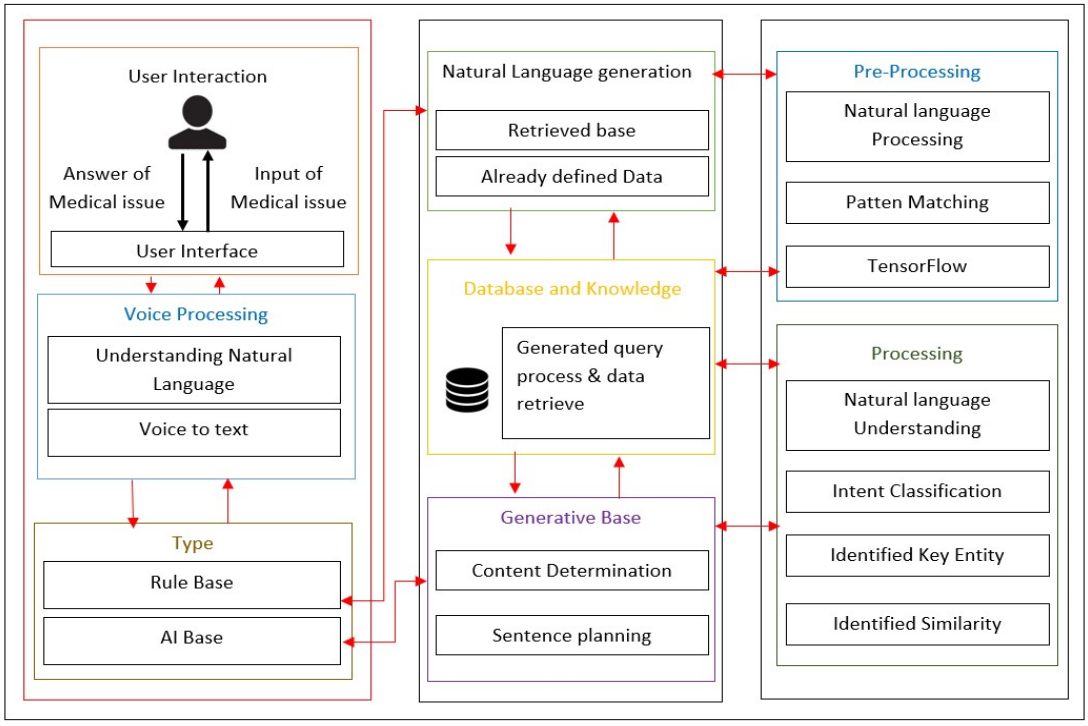

Figure 4.1 : High level architecture diagram for ADAMS

This research's outcome is implemented Two Mobile applications for MIS domain. Fist app is used to register the Medical centres and other app is for patients. Both apps are developed using Flutter SDK for interface, chat framework is RASA Framework, Authentication done using Firebase cloud function, Firebase used to Database connectivity, Ontology Framework is python Owlready and OWL rules, getting speech to text use Google cloud speech to text API, Ontology data recourse is ICD API, Text to speech done by using Neural network and Cryptographic Hashing for Blockchain mechanism. Below mentions the sub objective of the research:

4.1 DOI-HD:

Determine the optimum identification to sort out the human diseases (patient is given) (DOI-HD):

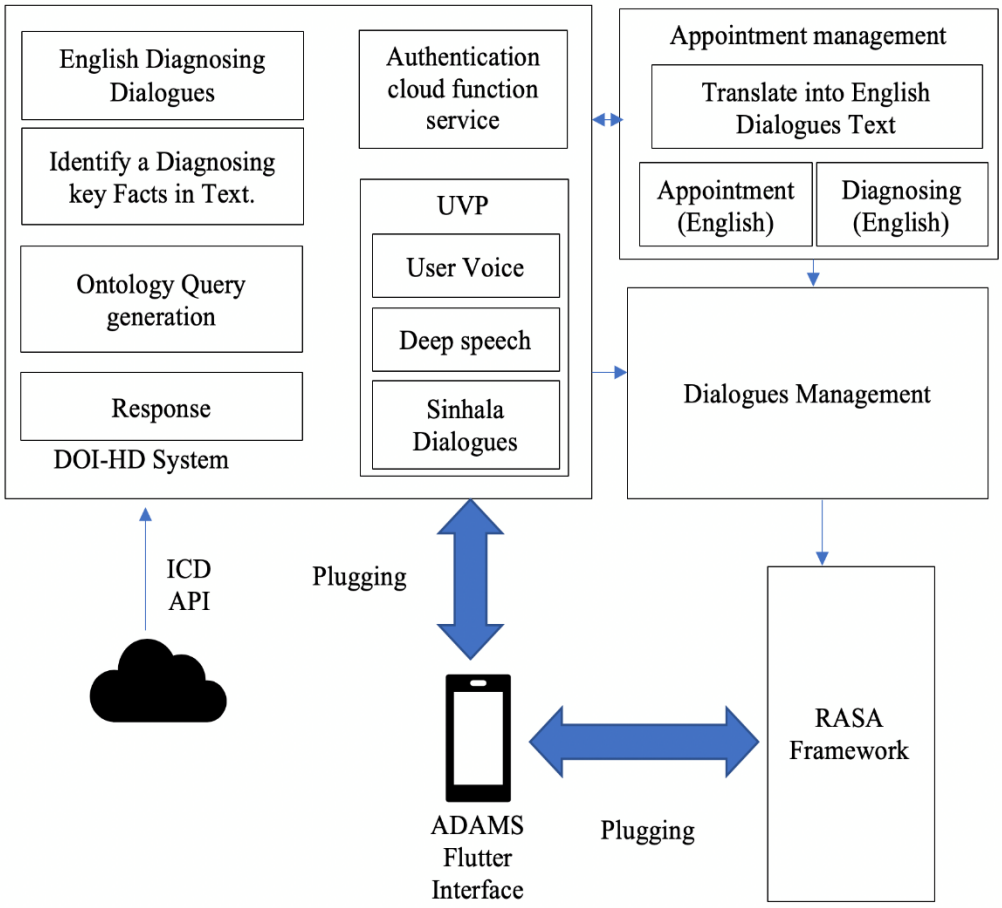

Figure 4.2: High level architecture diagram for DOI-HD

When a user open app user authentication and user session are handled by the Firebase cloud service. therefore, I implement a cloud function. After validation user, he or she can use an ADAMS service. ADAMS only compatible with the Sinhala user's voice and That user's voice converted to Sinhala Text. A specific set of the identified keyword (Medical domain keyword) extracted from a user speech (Sinhala speech Text translate to the English text; it is done by Appointment Management) is compared with a set of identified keywords that are fetched by ICD API. As a fetching data used ontology, using ontology easy to retrieval a domain specified data. Used Owlready Ontology framework and OWL rules. It is a python framework. In our Mobile app interface base on a Flutter framework, it is a platform-independent framework. For that reason, I developed RASA DOI-HD service connection plugging, RASA framework plugging, and Firebase Connection plugging to connecting a Flutter interface. That plugging using can integrate our service into the Flutter interface.

4.2 Appointments Management

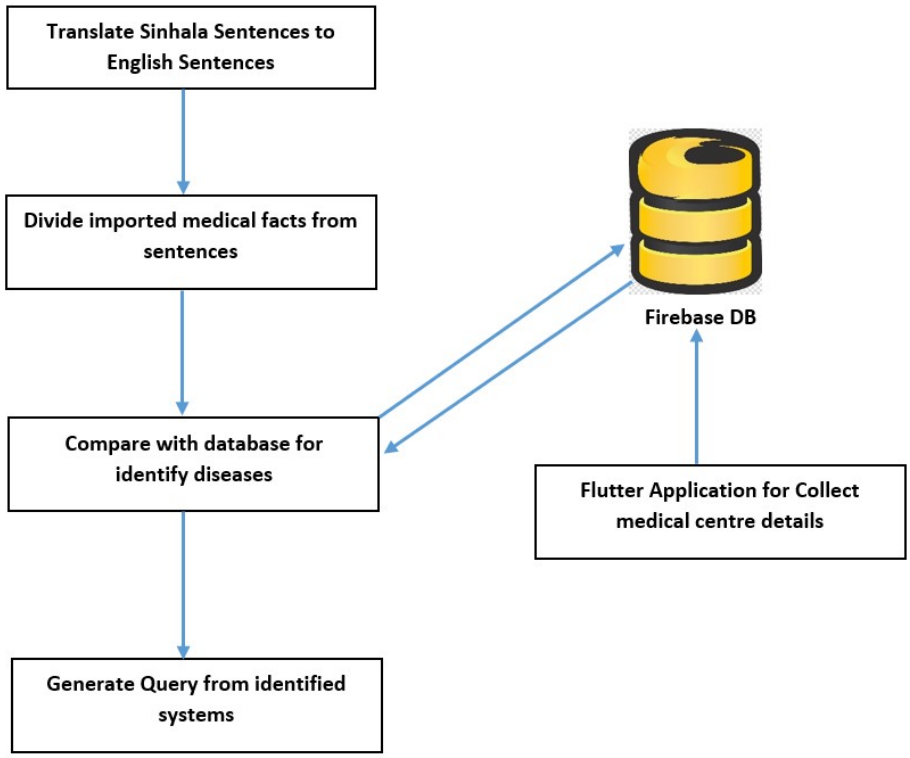

Figure 4.3: High level architecture diagram for Appointments Management

Create Database is an organized collection of structured information, or data, typically stored electronically in a computer system. A database is usually controlled by database management system. Database management system stores, organizes and manages a large amount of information within a single software application. In our application we have to make database with huge capacity of data. These data should retrieve fast from database. So, we decide to create our database on firebase. The main thing is firebase database easy to Connect with flutter application and it will be NoSQL database. Natural Language Processing usually shortened as NLP is part of artificial intelligence that works as interaction between computers and human using the natural Language. The final objective of NLP is to Read, Decipher, Understand and make sense of the human language in a manner that is valuable. Most NLP techniques rely on machine learning to derive meaning from human language. In this research NLP use for understand given English sentence.

- Structure extraction – identifying fields and blocks of content based on tagging
- Identify and mark sentence, phrase, and paragraph boundaries – these markers are important when doing entity extraction and NLP since they serve as useful breaks within which analysis occurs.
- Language identification – will detect the human language for the entire document and for each paragraph or sentence. Language detectors are critical to determine what linguistic algorithms and dictionaries to apply to the text.

  - Open source possibilities include Google Language Detector or the Optimize Language Detector or the Chromium Compact Language Detector

Fact Identification is important to separate important words from given sentences for make answers. This application has to identify filtered English medical domains from given English sentences. Those medical domains should compare with database.

- Tokenization – to divide up character streams into tokens which can be used for further processing and understanding. Tokens can be words, numbers, identifiers or punctuation (depending on the use case)

  - Open source tokenizers include the Lucene analyzers and the Open NLP Tokenizer.

  - Basis Technology offers a fully featured language identification and text analytics package (called Rosette Base Linguistics) which is often a good first step to any language processing software. It contains language identification, tokenization, sentence detection, lemmatization, decompounding, and noun phrase extraction.

- Search Technologies has many of these tools available, for English and some other languages, as part of our Natural Language Processing toolkit. Our NLP tools include tokenization, acronym normalization, lemmatization (English), sentence and phrase boundaries, entity extraction (all types but not statistical), and statistical phrase extraction. These tools can be used in conjunction with the Basis Technology' solutions.

- Acronym normalization and tagging – acronyms can be specified as "I.B.M." or "IBM" so these should be tagged and normalized.

  - Search Technologies' token processing has this feature.

4.3 Dialogue Management

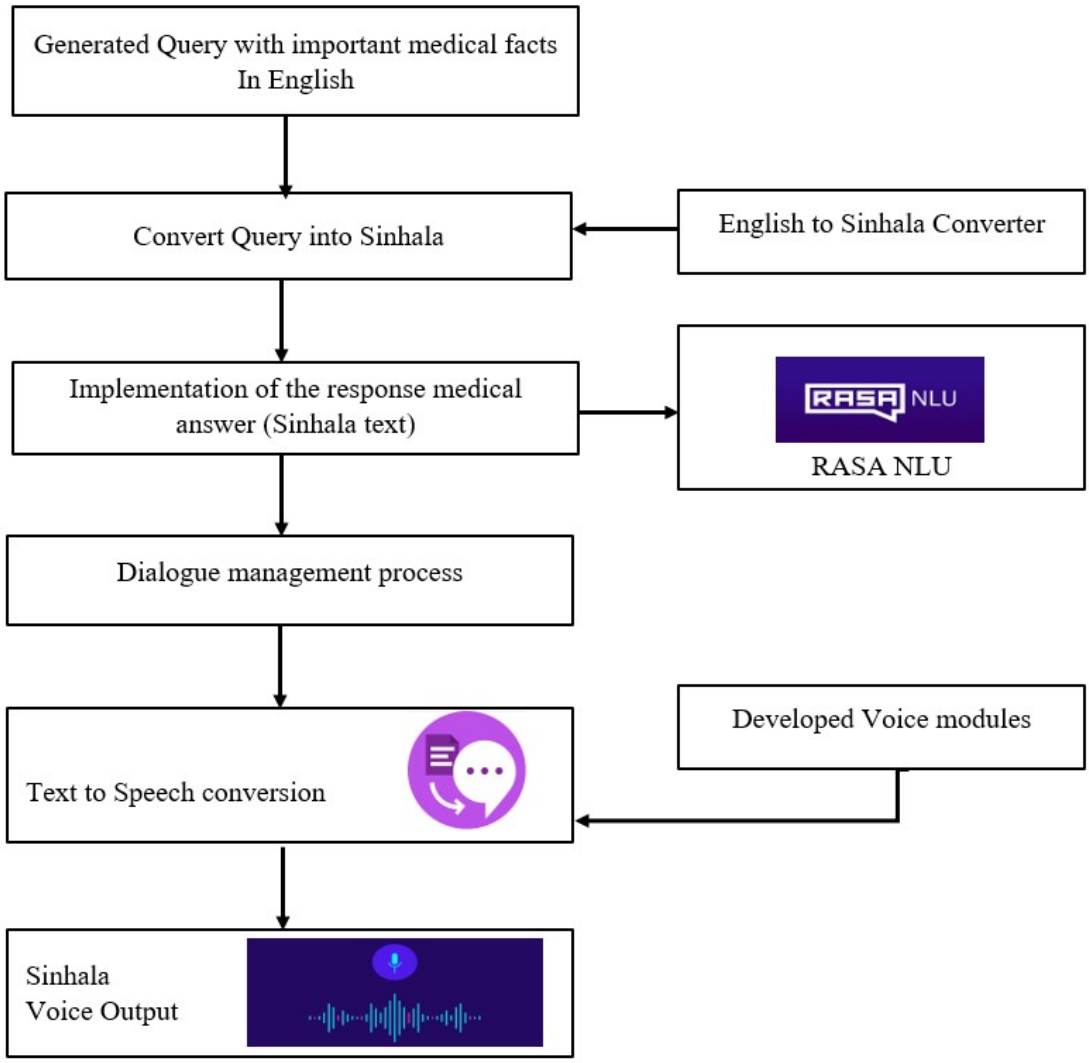

Figure 4.4 : High level architecture diagram for Dialogue Management

The knowledge of the trained medical terms is used to generate the query. This query is in English, this has to be converted into Sinhala language using English Sinhala converter. Using that query have to implement the medical answer for the user's medical question. These import facts of the medical question use to identify the what is the problem that having to the user. According to these medical facts, response medical answer will be generated through the system. End of this process system Sinhala text will be created.

User's medical question and the response medical answer will be set as a conversation. Through this whole conversation between patient and the bot, have to manage as a dialogue. To output Sinhala voice command there should be a Sinhala voice module which is developed already. There is mapping mechanism between trained voice modules and the generated Sinhala response text. Sinhala voice modules will be developed using Sinhala voice commands. Corresponding sentences were recorded by

a professional speaker. Recorded Sinhala words according to the medical domain is used to mapping mechanism. By using those Sinhala medical terms Sinhala voice module will be developed. There is a voice module train mechanism to use this whole text to speech process. The conversion of text to speech involves many important processes. These processes can be divided mainly in to three stages; text analysis, linguistic analysis and wave-form generation [13].

The text analysis stage is responsible for converting the non-textual content into text. This stage also involves tokenization and normalization of the text; identification of words or chunks of text. Text normalization establishes the correct interpretation of the input text by expanding the abbreviations and acronyms. This is done by replacing the nonalphabetic characters, numbers, and punctuation with appropriate text-strings depending on the context.

The linguistic analysis stage involves finding the correct pronunciation of words, and assigning prosodic features (e.g. phrasing, intonation, stress) to the phonemic string to be spoken. The final stage of a TTS system is waveform generation which involves the production of an acoustic digital signal using a particular synthesis approach such as formant synthesis, articulatory synthesis or waveform concatenation [13].

At the end of the above three processes Sinhala voice output will be here as a voice command. Sound waves will be here in smooth, quality and better speed that healthy level to the humans.

4.4 RASA Framework

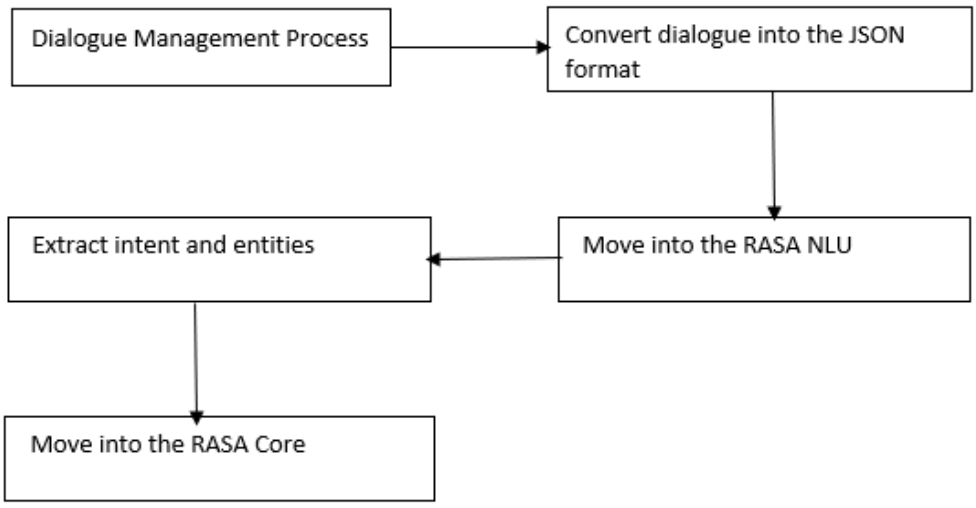

Figure 4.5 : High level architecture diagram for RASA Framework

In here Sinhala text message should move to the RASA NLU (Natural Language Understanding). For that case rasa provides two data training types. First one is Markdown format and second one is Json format.

Markdown format is the easiest data training format available inside the Rasa NLU. Listed using the unordered list syntax, e.g. minus -, asterisk *, or plus + examples for this markdown format. These examples can group by intent and entities. e.g. [entity] (entity name) [9].

Second data training type is Json format. In this Json format includes a top-level object called Rasa NLU data. “common examples”, “entity synonyms” and “regex features” are the keys of RASA NLU data. The important part is “common examples”.

```
{
  "rasa_nlu_data": {
    "common_examples": [ ],
    "regex_features" : [ ],
    "lookup_tables"  : [ ],
    "entity_synonyms": [ ]
  } }
```

Trained the model using “common examples”. After that include all of training examples inside to the “common examples” array. “Regex features” tool help the classifier to detect entities or intents in the RASA NLU and improve the performance of RASA NLU [9].

For those reasons JSON format is the most appropriate solution. And also, Jason format is compatible for the mobile application development rather than Markdown format. By using Json format method system will enrich training data so this project prefers using a User Interaction to do so. However, some power users interested in Markdown as it is really easy to understand, some points to consider.

The text message wants to move into the RASA NLU. That text message automatically convert into the JSON format. In that Json format has included intent (describes what type of messages) and entities (what specifically a user is asking about).

Below example sentence is for RASA NLU which is an open source natural language processing tool intent classification and entity extraction.

“ශ්‍රී ලංකාවේ අද කාලගුණය කෙසේද?”

This example’s intent is “ask_weather_location”

After the moving inside a RASA NLU it will extract entities and intents which are inside in a JSON format. Nlu.md file use to write the training data. Training data is usually stored in a markdown file. There are number of different components available in Rasa NLU, which together make a pipeline. This process can include it into the NLU model pipeline, once the training data is ready [10].

RASA NLU move to the RASA Core. Can build AI assistance using dialogue engine is the purpose of RASA CORE. It's part of the open source RASA framework. RASA CORE use as a machine learning model trained to decide what to do next.

5 DESCRIPTION OF PERSONAL AND FACILITIES

- Determine the optimum identification to sort out the human diseases (patient is given). -DOI-HD
- Appointments Management
- Dialogue Management
- RASA Framework

This project would be done by a group of four members and the research and development workload of the project is being distributed among all the members. Detailed explanation of the assigned components has been discussed in the previous sections of the document.

Member	Component
D.D.S Rajapakshe	DOI-HD
Kudawithana K.N.B	Appointment Management
U.L.N.P. Uswatte	Dialogue Management
Nishshanka N.A.B.D	Rasa framework

Table 5-1 : Description of personal

6 PROJECT REQUIREMENTS

Require engineering is a process of establishing

- The services that a customer requires from a system and

- The constraints under which it operates and is developed.

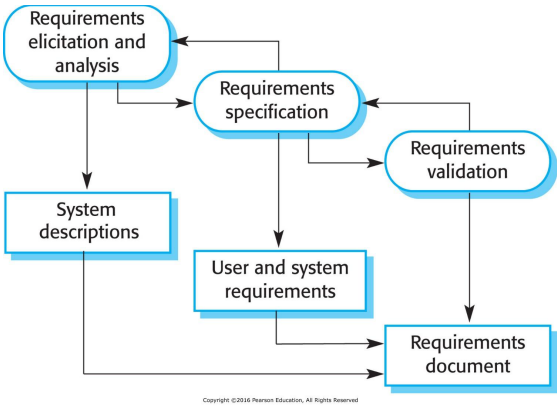

Table 6-1 : Requirements Engineering Process

Two types of Requirements Levels

1. User requirements
   - Platform to patient to e-channeling
   - Platform to dispensary/ hospital to register their system
   - Platform to doctor to register their system

2. System requirements

   A structured document setting out detailed descriptions of the system's functions, services and operational constraints. System requirements are the configuration that a system must have in order for a hardware or software application to run smoothly and efficiently. Failure to meet these requirements can result in installation problems or performance problems. The former may prevent a device or application from getting installed, whereas the latter may cause a product to malfunction or perform below expectation or even to hang or crash.
   There are two types of System Requirements

   1. Functional requirements
      Statements of services the system should provide, how the system should react to inputs and how the system should behave in particular situations.Describe functionality or system services, Depend on the type of software, expected users and the type of system where the software is used, Functional user requirements may be high-level statements of what the system should do, Functional system requirements should describe the system services in detail

      - Patient registration to the system
      - Patient assistant through the system
      - User interact with system using Sinhala voice command
      - Doctor appointment for the patients
      - Patients can get the information about doctors, hospitals and channeling

- Doctor channeling through the system
- Medical service centers registration to their mobile application

2. Non-functional requirements

   Constraints on the services or functions offered by the system. Often apply to the system rather than individual features or services. These define system properties and constraints. Non-functional requirements may more critical that functional requirements. If these are not met, the system may be useless.
   - Automated conversational mobile application
   - The quality of Sinhala Voice commands and the performance like System response time, throughput, utilization, static, volumetric
   - Reliability of the system data is the most important fact for both Patients and Medical Service Centres
   - System and its whole functionality (doctor channelling, information accessibility) should be available anytime for the patients.
   - How much easier to users when interact with this system's functionalities
   - Total cost for system implementation, developments and use of the services
   - How the system localization for Sinhala users with interaction of Sinhala voice commands

7 WORK BREAKDOWN STRUCTURE

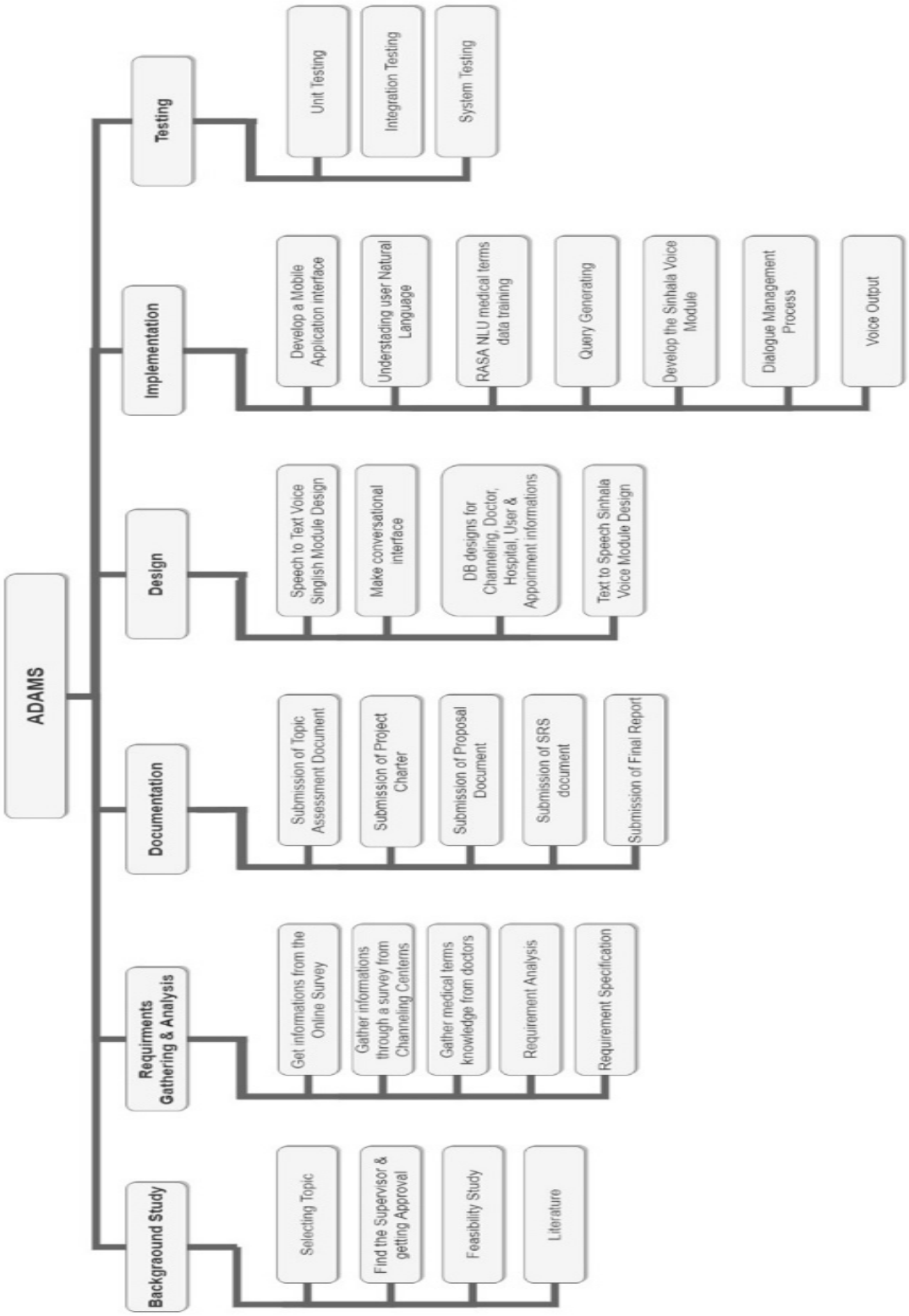

Figure 7.1 : Work breakdown structure

8 BUDGET AND BUDGET JUSTIFICATION

The Budget justification about the **ADAMS**. These will be the budget that going to estimate in this project. In Research Project is planned to use Firebase Realtime database. Because it's a mobile compatible database. Firebase has two types of database planning. First one is Spark Plan and the second one is Blaze Plan. Spark Plan is a free plan, but it can't customize. Then the best plan is Blaze Plan. Because it can customize as the developer wish.

Realtime-Database:1 GB free storage has already been given. But it's not enough for this project requirements. Then hope to add 4 GB as an extra storage. Then the total storage is 5GB and the price of that storage is $20. 10 GB transferred data has already been given. But another 10 GB data must be added. For that extra 10GB $10 must be paid.

Firebase Cloud Storage:5GB free data has already been provided for the cloud storage. Mainly in this research wants This cloud storage also provides a 5GB free data. This cloud storage also provides us a 5GB free data. Mainly store two voice modules then we want some high capacity storage. Mainly we want to store two voice modules then high capacity storage must be needed. Then 95 GB extra storage must be added. Then total storage capacity is 100 GB. Then $2.47must be paid additionally. 30 GB free storage for data transfer and 2,100,000 operations are provided additionally.

Google Play (Play Store): if this application is used for the Android, a Google Play (Play Store) account must be created. $25 must be paid for this but it is one-time registration payment. MasterCard, Visa, American Express, discover (U.S. only) and Visa Electron (outside of the U.S. only) payment methods will be used in this payment proses.

App Store: If the application is used for the IOS, App Store account must be created. $ 99 must be paid for this but it is annual fee. But If this is an Apple Developer Enterprise Program account, $299 must be paid annually. Some Stationary, Communication and Printing cost are related to this research. These prices can be changed according to the quantity and the value of dollar.

Budget		**Cost**
1. Firebase		

1.1. Realtime-Database		
1.1.1. Storage(5GB)	$20	
1.1.2. Transferred data(20GB)	$10	$30
1.2. Cloud Storage		
1.2.1. Storage(100GB)	$2.47	$2.47
2. Total Cost for the Firebase		$32.47
3. Google Play (Play Store)		$25
4. App Store		$99
5. Stationaries (12 months)		
5.1.A4 sheets	$1.38	
5.2.Rough sheets	$0.55	
5.3.Binding	$1.38	
5.4.Pens & Pencils	$0.83	
5.5.Other Stuff	$1.10	$5.24
6. Communication (12 months)		
6.1.Internet service	$13.78	
6.2.Phone Cost	$16.54	$30.32
7. Printing Cost (12 months)		
7.1.Reports	$2.76	
7.2.Photocopy cost	$5.51	$8.27
Total Cost		**$200.30**

Table 8-1 : Budget

9 TIME PLAN

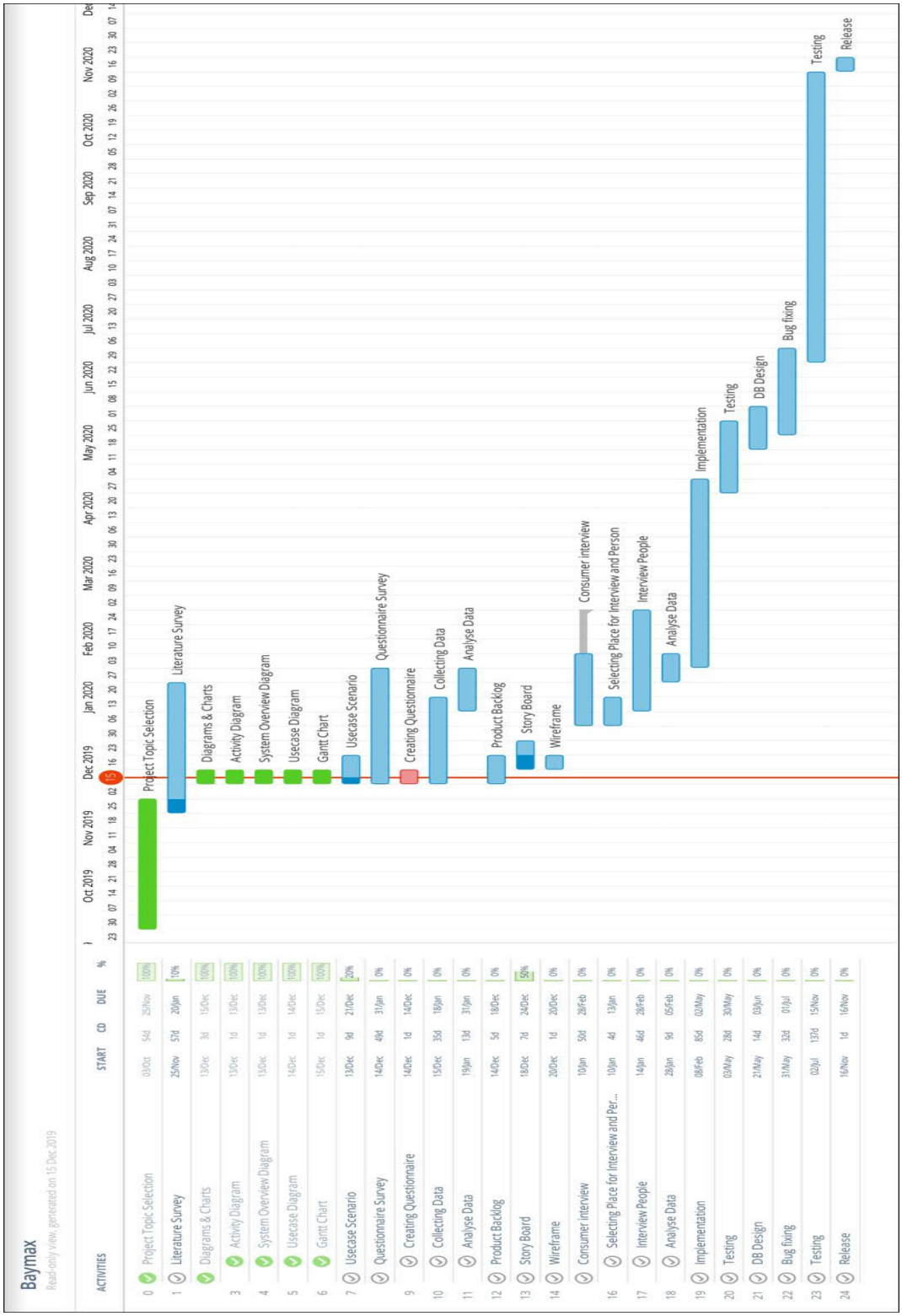

Figure 9.1 : Gantt Chart

10 REFERENCE

[1] W. R. S. P. S. C. W. Mehul Bhatta, "Ontology driven semantic profiling and retrieval in medical information systems," 8 March 2009. [Online]. Available: https://www.sciencedirect.com/science/article/abs/pii/S1570826809000201. [Accessed 27 May 2009].

[2] J.-B. Lamya, "Owlready: Ontology-oriented programming in Python with automatic classification and highlevel constructs for biomedical ontologies," August 2017. [Online]. Available: https://www.researchgate.net/publication/319126534_Owlready_Ontology-oriented_programming_in_Python_with_automatic_classification_and_high_level_constructs_for_biomedical_ontologies.

[3] L. Nachabe, "Applying Ontology to WBAN for mobile application in the context of sport excersises," September 2014. [Online]. Available: https://www.researchgate.net/publication/267152523_Applying_Ontology_to_WBAN_for_mobile_application_in_the_context_of_sport_excersises.

[4] O. Bodenreider, "The Unified Medical Language System (UMLS): integrating biomedical terminology," 1 january 2004. [Online]. Available: https://www.ncbi.nlm.nih.gov/pmc/articles/PMC308795/.

[5] P. B. S. B. G. N. A. J. N. W. P. C. A. G. A. B. Robert Stevens, "TAMBIS: Transparent Access to Multiple Bioinformatics Information Sources," ISCB, 1 February 2000. [Online]. Available: https://academic.oup.com/bioinformatics/article/16/2/184/232155.

[6] T. M. d. Farias, "SWRL rule-selection methodology for ontology interoperability," 2015 September 2015. [Online]. Available: https://www.researchgate.net/publication/282248747_SWRL_Rule-selection_Methodology_for_Ontology_Interoperability.

[7] J.-B. LAMY, "Ontology-Oriented Programming for Biomedical Informatics," IOS Press, 2016. [Online]. Available: https://cmapspublic2.ihmc.us/rid=1Q75R3F5C-TX84TD-XM/2016%20-%20Transforming%20healthcare%20with%20the%20internet%20of%20thing.pdf#page=76.

[8] T. Nadungodage, "Continuous Sinhala Speech Recognizer," May 2011. [Online]. Available:

https://www.researchgate.net/publication/265992312_Continuous_Sinhala_Speech_Recognizer.

[9] M. Bali, "Diabot: A Predictive Medical Chatbot using Ensemble Learning," 2019. [Online]. Available: https://www.academia.edu/39258471/Diabot_A_Predictive_Medical_Chatbot_using_Ensemble_Learning.

[10] C. Segura, "Chatbol, a chatbot for the Spanish “La Liga”," May 2018. [Online]. Available: https://www.researchgate.net/publication/327426302_Chatbol_a_chatbot_for_the_Spanish_La_Liga.

[11] V. V. Sandra V A, "Quinn: Medical Assistant for Mental Counseling using Rasa Stack," 06 June 2019. [Online]. Available: https://www.irjet.net/archives/V6/i6/IRJET-V6I6297.pdf.

[12] T. Kowatsch, "Text-based Healthcare Chatbots Supporting Patient and Health Professional Teams: Preliminary Results of a Randomized Controlled Trial on Childhood Obesity," 2017. [Online]. Available: https://www.alexandria.unisg.ch/252944/2/2368_Kowatsch%20et%20al%202017%20-%20THCB%20PEACH%20Workshop.pdf.

[13] I. API, "ICD API," [Online]. Available: https://icd.who.int/icdapi.

[14] E. M. Company, "Sri Lanka computer literacy falls amid shift to smartphones, tablets," 29 01 2019. [Online]. Available: https://economynext.com/sri-lanka-computer-literacy-falls-amid-shift-to-smartphones-tablets-12907/.

[15] D. Khurana, "Natural Language Processing: State of The Art, Current Trends and Challenges," August 2017. [Online]. Available: https://www.researchgate.net/publication/319164243_Natural_Language_Processing_State_of_The_Art_Current_Trends_and_Challenges.

[16] R. Hall´en, "A Study of Gradient-Based Algorithms," 02 March 2017. [Online]. Available: http://lup.lub.lu.se/luur/download?func=downloadFile&recordOId=8904399&fileOId=8904400.

[17] A. Nanayakkara, "A Human Quality Text to Speech System for Sinhala," August 2018. [Online]. Available: https://www.researchgate.net/publication/328068273_A_Human_Quality_Text_to_Speech_System_for_Sinhala.

[18] R. Weerasinghe, "Festival-si: A Sinhala Text-to-Speech System," September 2007. [Online]. Available:

https://www.researchgate.net/publication/221151931_Festival-si_A_Sinhala_Text-to-Speech_System.

[19] V. M. Harshini, "Health Record Management through Blockchain Technology," April 2019. [Online]. Available: https://www.researchgate.net/publication/336439409_Health_Record_Management_through_Blockchain_Technology.