TABLE OF CONTENTS

Chapter 1: Introduction #

1.1: PROJECT AIM #

1.2: PROJECT BACKGROUND #

1.3: Existing PROJECTS #

CHAPTER 2: PROJECT MANAGEMENT AND TOOLS #

2.1: PROJECT MANAGEMENT

2.1.a: Project Management Tools #

2.2: PROJECT PLANNING #

2.2.a: App Development Methodology #

2.2.b: Scope #

2.2.c: Challenges #

2.2.d: Risks #

2.2.d: Identified Tasks #

Chapter 3: dESIGN #

3.1: REQUIREMENTS #

3.2: ARCHITECTURE #

3.2.a: Class Diagram #

3.2.b: Sequence Diagram #

3.3: PROGRAMMING LANGUAGE

3.4: METHODOLOGY

3.5: TECHNICAL CHALLENGES #

3.6: DATABASE #

3.5.a: Challenges #

3.5.b: Approach #

CHApter 4: Implementation #

4.1: TOOLS #

4.1.a: Ide #

4.1.b: Debugging #

4.1.c: Other Tools #

4.2: APPLICATION #

4.2.a: UI #

4.2.a: Content #

CHApter 5: testing #

5.1: TEST PLAN #

5.2: TEST REASULTS #

CHApter 6: Evaluations #

6.1: PROJECT PLANNING #

6.2: DESIGN #

6.3: IMPLIMENTATION AND TESTING #

6.4: FUTURE DEVELOPMENT #

6.5: GAINED LESSONS #

6.6: CONCLUSION #

CHApter 7: REFERENCES #

**Chapter 1: Introduction**

**Abstract**

The food we eat will have a profound effect on our health. Everybody has a strong opinion, diverse assertions about What is Healthy Nutrition. Changes in diet help many health problems including obesity, diabetes and certain risk factors for cancer and heart disease. Diet planning is the science of how food and nutrition effect on human health. Therefore, People around the world seeking to maintain their weight by limiting junk foods and eating more nutrition foods. For this an automated Dietitian is required to help people improve their health.

Smart dietician bot is an AI system that can gather daily intake of calories, weight, height, age, working hours (Activity Level) and analyze the given data and consult as a real dietician. Most importantly this dietician can take health condition (like diabetes or cardiac patients) into account and suggest their meal plans and suitable workout routines. Furthermore, this provides full details of the nutritional formula required for the body and necessary number of calories to burn fat/maintain BMI, increase with the plan by answering some queries.

This is appropriate for users who need to improve their health. Also appropriate for users who need to prevent from certain risk factors and to have care and consultation. Also, people can be in touch with the nutritional formula required for their body.

Artificial intelligent bot become visible as an important research area in recent past. Study about existing work on dietician Artificial intelligent bot useful for construct, a new solution.

1.1: Project AIM

We all know the adage, "You are what you eat." Maintaining your health is one of the first steps to managing weight, and a big step to maintaining your health is eating well every day.

Therefore, the project Dietitian-Bot goal is to process diet plans and give some better recommendations according to user conditions and needs. This application includes meal plan management plus Assistant to respond in a timely fashion and be all round user friendly. Assistant is the chatbot of the meal plan app. User don’t have to go through the application to get their generated diet plans and suggested recommendations by changing tabs, button clicks etc. Assistant makes the user interaction as easy and fast as possible to ensure that the users time is not wasted and that they get what they want without any difficulty or misunderstanding

from the application in one place by asking some queries. Ultimate aim of the project is to help people around the world to build a healthy society by consult as a real dietitian free of charge.

1.2: Project Background

To understand any complex system, at first, an abstracted high-level introduction of the subject and explanation of the architecture which perfectly considering the complex system and allows for a better understanding of details later is not only compulsory but also important. Therefore, in this section will cover the background research that I have conducted into different kind of chatbots and some of the advance technologies I’ve explored.

In Sri Lanka, we have become an overweight society. Our busy lifestyles and the abundance of convenience foods have fostered our expanding waistlines. Our society supports working long hours followed by responsibilities to our families, children and other things that take up time. Convenience food items and fast food restaurants provide a quick meal for people constantly on the go. An April 2010 Prevalence of overweight and obesity in Sri Lankan adults [1] report noted that “relatively high prevalence of overweight and obesity, particularly, abdominal obesity among adults in Sri Lanka which is a middle‐income country. Urgent public health interventions are needed to control the problem at an early stage.” The idea for this project was born when understanding the future risk of obesity and other related risk factors.

AI Bots (also known as Artificial Conversational Entity, chatbot) is a computer program or an artificial intelligence that conduct conversation via audio or

textual messages [2].

In 1950, Alan Turing’s well-known article “Computing and Machinery and Intelligence” [3], which projected the Turing test as a criterion of intelligence that depend on the fact that a real written discussion with a computer program to imitate a human in a real-time written conversation with a human judge. The historic chatbots are ELIZA (1966) which was mimicked human conversations by pattern matching and substitution methodology however passed the turning artificial intelligence test and PARRY (1972) was more advance than ELIZA also called “ELIZA with and attitude”. From 1966 onwards, computer programmers and business owners understood the usefulness Bots can provide to end users, specially when the information can be categorized into concrete and predictable subjects. Modern chatbots are more complex and feature natural language processing that can learn from user inputs. They can access APIs to get information users such as news, weather, time etc. They can even process orders and make bookings entirely through a chatbot interface. Chatbots are well suited for mobile devices as messaging is at the heart of a mobile phone.

Generally, bots use Natural Language processing techniques to Input versus analysis and output. Natural Language Processing (NLP) is the study of letting computers understand human languages [4]. Without NLP, human language sentences are just a series of meaningless symbols to computers. Computers don’t recognize the words and don’t understand the grammars. NLP can be regard as a “translator”, who will translate

human languages to computer understandable information. Traditionally, users need to follow well-defined procedures accurately, in order to interact with computers. For example, in Linux systems, all commands must be precise. A single replaces of one character or even a space can have significant difference. However, the emergence of NLP is changing the way of interacting. Apple Siri [5], Microsoft Cortana [6] and Google Assistant [7] have made it possible to give command in everyday languages and is changing the way of interacting. Assistant Photos

In this project uses the API provided by Microsoft called Language Understanding Intelligent Service (LUIS). It’s a well-developed REST API for Language Understanding.

1.3: Existing Projects

**Chapter 3: DESIGN**

Before step in to the design stage of SDLC, project need to ensure if it is practical or not. Therefore, feasibility study must be done. It was challenging to accept final decision of software design for this project. After some several software designs, a practically possible software design was chosen which is technically feasible and has Schedule feasibility.

3.1: REQUIREMENTS

The following are the main system requirements which is to be catered by the software design.

* The application should predict the correct intent (highest score intent) by analyzing user given utterance.
* The application should extract the correct entities by analyzing user given utterance.
* An utterance can have only one top scoring intent, but it can have many entities.
* If user ask questions beyond the domain of application, then application should understand the domain and give a proper reply.
* The conversation should flow and always try to keep the user in control of the conversation.
* The application Should process accurate meal plans for users by analyzing user profile.
* The application should have an admin role for meal plan management and user management.
* User interface of the client-side should be user-friendly to provide better user experience.

3.2: ARCHITECTURE

The entire application is programmed using Java following the MVC architecture.

This uses the following external libraries.

* Apache HttpComponents Client – handling http request/response
* java-json.jar – handling json objects

After deeply go through the requirements above the entire process was outlined in the Architecture Overview.

Architecture Overview (Figure Process of the input to output)

3.2.a: Class Diagram

After getting clear idea about the application architecture by careful analysis of the requirements above, the class diagram and sequence diagram were drawn.

Place Figure cd

In Figure cd illustrates the class diagram according to the MVC architecture. For this, three types of objects have been used in the application.

**Model** - The model represents data and the rules that govern access to and updates of this data [8].

In Figure cd LUIS, User and MealPlan are the model classes.

**View** - The view renders the contents of a model. It specifies exactly how the model data should be presented. If the model data changes, the view must update its presentation as needed [8].

In Figure cd AdminUI, SignInUI, SignUpUI, and UserUI are the View classes.

**Controller** - The controller translates the user's interactions with the view into actions that the model will perform [8].

In Figure cd SetOfUsers, SetOfMealPlans, DietMaths and Validator are the Controller classes.

3.2.b: Sequence Diagram

Figure sequence diagram

3.3: PROGRAMMING LANGUAGE

3.4: METHODOLOGY

The design of the LUIS module is key to the correctness of the Natural Language

Processing and thus critical to the performance of diet assistant. Because the learning algorithm is close-sourced, most important task is to define the ​intent

​and ​entity clearly and labeling the sentence based on design.

Language Understanding (LUIS) is a cloud-based service that applies custom machine-learning to a user's conversational, natural language text to predict overall meaning, and pull out relevant, detailed information.

A screenshot of a cell phone

Description generated with high confidence

A LUIS model includes:

* [**intents**](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#intents): An intent represents a task or action the user wants to perform
* [**entities**](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#entities): specific types of data in utterances such as number, email, or name
* [**example utterances**](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#example-utterances): example text a user enters in the application

**Intents**

An [intent](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-add-intents), short for intention, is a purpose or goal expressed in a user's utterance. For this project need to create an intent for each action like Greeting, getDietPlans, getStatus, getBMI, setAge, setHeight, setName and None intent etc. After Define a set of intents that corresponds to actions in assistant use the top scoring intent to trigger an action. For example, when "getDietPlans" intent is returned from LUIS, assistant call the getters of related java classes and populate to User Interface. LUIS model come with the predefined intent, "[None](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-intent#none-intent-is-fallback-for-app)" which is the fallback intent and used to teach LUIS utterances that are not significant to the app domain. For example, when user enter some utterance like “can you send a message to Alan” which is not important for the dietitian application then assistant prompt a proper message for user “sorry I didn’t get it, I’m trying to do my best for you”.

**Entities**

An [entity](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-add-entities) represents detailed information found within the utterance that is relevant to the user's request. For example, in the utterance "Book a ticket to Paris", a single ticket is requested, and "Paris" is a location. Two entities are found "a ticket" indicating a single ticket and "Paris" indicating the destination.

After LUIS returns the entities found in the user’s utterance, the client application can use the list of entities as parameters to trigger an action. For example, booking a flight requires entities like the travel destination, date, and airline.

### Example utterances

An example [utterance](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-add-example-utterances) is text input from the user that the client application needs to understand. It may be a sentence, like "Book a ticket to Paris", or a fragment of a sentence, like "Booking" or "Paris flight." Utterances aren't always well-formed, and there can be many utterance variations for a particular intent. Add 10 to 20 example utterances to each intent and mark entities in every utterance.

## **Improve prediction accuracy**

After your LUIS app is published and receives real user utterances, LUIS provides several methods to improve prediction accuracy: [active learning](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#active-learning) of endpoint utterances, [phrase lists](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#phrase-lists) for domain word inclusion, and [patterns](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/home#patterns) to reduce the number of utterances needed.

### Active learning

In the [active learning](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-how-to-review-endoint-utt) process, LUIS allows you to adapt your LUIS app to real-world utterances by selecting utterances it received at the endpoint for your review. You can accept or correct the endpoint prediction, retrain, and republish. LUIS learns quickly with this iterative process, taking the minimum amount of your time and effort.

### Phrase lists

LUIS provides [phrases lists](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-feature) so you can indicate important words or phrases to your model domain. LUIS uses these lists to add additional significance to those words and phrases that would otherwise not be found in the model.

### Patterns

Patterns allow you to simplify an intent's utterance collection into common [templates](https://docs.microsoft.com/en-us/azure/cognitive-services/luis/luis-concept-patterns) of word choice and word order. This allows LUIS to learn quicker by needing fewer example utterances for the intents. Patterns are a hybrid system of regular expressions and machine-learned expressions.

3.5: TECHNICAL CHALLENGES

This section delineates the technical challenges faced all over the project.

**HTTP request and respond delay –** Handling http request/response via the java application has some delay than expected high speed. For this project Apache HttpComponents Client library is used to hit the Microsoft LUIS API endpoint. Apart from that there is another popular lightweight HTTP library called Unirest which is built and maintained by mashape. After checking the Unirest, there is no significant difference.

**API is under development – ​**Because API is still under development, therefore cannot fix to a version for the API, the API may change overtime. Besides, there are conflicts between the APIs and their documents or sample codes.

**Azure account –** Azure education account is limited to 10,000 transactions per month / 5 transactions per second to querying the LUIS endpoint.

**Need Proper training for LUIS model –** Training is the process of teaching Language Understanding (LUIS) app to improve its natural language understanding. Training LUIS app after updates to the model such as editing, adding, labeling, or deleting intents, entities, or utterances. Training and testing an app is an iterative process. Next trained LUIS app needs to test it with sample utterances to see if the intents and entities are predicted correctly. If they're not, make updates to the LUIS app, train, and test again.

**Chapter 7: References**

1. https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-789X.2010.00746.x
2. [*"What is a chatbot?"*](http://searchdomino.techtarget.com/sDefinition/0,,sid4_gci935566,00.html). techtarget.com*. Retrieved 30 January 2017*.
3. [*Turing, Alan*](https://en.wikipedia.org/wiki/Alan_Turing)*(1950), "Computing Machinery and Intelligence", Mind,****59****: 433–60,*[*doi*](https://en.wikipedia.org/wiki/Digital_object_identifier)*:*[*10.1093/mind/lix.236.433*](https://doi.org/10.1093/mind/lix.236.433)
4. Lexalytics [Internet]. USA: NLP in 5 Minutes. [updated 2016 April 27, cited 2016 Oct 21]. Available from: ​<https://www.lexalytics.com/lexablog/2016/nlp>​.
5. Apple [Internet]. USA: Siri. [cited 2016 Oct 21]. Available from: <http://www.apple.com/ios/siri>​.
6. Microsoft [Internet]. USA: What is Cortana. [updated 2016 Sep 10, cited 2016 Oct 21]. Available from: ​<https://support.microsoft.com/en-hk/help/17214/windows-10-what-is>​.
7. <https://assistant.google.com/explore?hl=en_us>
8. https://www.oracle.com/technetwork/articles/javase/mvc-136693.html