Sheffield Hallam University

Faculty of Arts, Computing, Engineering & Sciences

Department of Computing

Final Year Individual Project (SEGM)

[55-604708]

2017/18

Author:	**Perera P.A.H.E (27045240)**
Date Submitted:	**20-08-2018**
Supervisor:	**Mr. Lakmal Rupasinghe**
Degree Course:	**BEng (Hons) in Software Engineering**
Title of Project:	**Smart Dietician Bot**

Confidentiality Required?
NO ☐
YES ☐

TABLE OF CONTENTS

Chapter 1: Introduction

Abstract

The diet we eat will have a profound effect on our healthiness. Everybody has a strong opinion, diverse assertions about What is Healthy Nutrition. Changes in diet help various health problems including obesity, diabetes and certain risk factors for heart disease and cancer. Diet planning is the discipline of how food and nutrition effect on public 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 combines meal plan management and built in Assistant (chat bot) to respond in a timely manner and be overall 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 communication as easy and fast as possible to confirm that the time of the is not wasted and that they get what they want without any struggle 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 realize 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.

In Sri Lanka, we have converted to an overweight society. Our busy lifestyles and the plenty of ease foods have nurtured our increasing 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 cafeterias deliver a fast meal for people always 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.

When look at the background of other meal plan apps or services, they are written programs but not intelligent. Therefore, to build a meal plan app that has intelligent which can understand human language and give proper meal plans, as a solution for this problem, integration of a chat bot to meal plan application is the best choice to overcome the current situation.

Therefore, to achieve the goal of implementing proposed intelligent dietitian, exploring technologies and backgrounds of AI bots are vital. AI bots (also known as Artificial Conversational Entity, chatbot) is a computer program that communicate through textual messages or audio. In 1950, Alan M.Turing point out "Can machines think?" [2], 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. To find further explanations and solutions for the proposed system, dig deep down into existing projects is compulsory.

1.3: Existing Projects and Research

This section covers the background research that conducted into different kind of chatbots, existing meal plan apps and some of the advance technologies explored.

Existing bots

The historic chatbots are ELIZA in 1966 (Weizenbaum,1966) [3] which was mimicked human conversations by pattern matching and substitution methodology however passed the turning artificial intelligence test. ELIZA's target was to act as a Rogerian psychologist. Since ELIZA, there has been growth in the development of more intelligent chatbots and computer programmers and business owners understood the usefulness Bots can deliver to end users, specially when the data can be categorized into concrete and predictable subjects.

In 1972, Kenneth Colby at Stanford created PARRY (Colby, 1999) [4], PARRY was designed to imitate as a paranoid schizophrenic [5]. PARRY was more advance than ELIZA, also called "ELIZA with and attitude".

In 1988, Rollo Carpenter who is a British programmer created Jabberwacky (Carpenter,2005) [8], Jabberwacky was design to Keep in touch with users and imitate human relationships. Jabberwacky's learning technology is mainly focused on entertainment, rather than other AI applications. In 2005 Jabberwacky won the Loebner prize [6].

In 1995, Richard Wallace created A.L.I.C.E(Wallace,2009) [7], a meaningfully more intelligent bot that process answers thru pattern matching inputs in contradiction of <pattern> (input) <template> (output) pairs kept as forms in a knowledge base. These writings are written in the artificial intelligence of Markup Language (AIML), an extension of XML, currently we are using. ALICE won Loebner prize [6] [7] three times, an annual contest held every year which challenges to pass the Turing Test, and offers the prize to smartest intelligent chatbot.

Modern chatbots include: Amazon's Echo and Alexa, Apple's Siri, and Microsoft's Cortana [13](Weinberger,2017) are more advance and featured NLP that can learn from user utterance. They can access APIs to get information users such as weather ,news, time etc. They can even prepare and place orders through the chat bot interface. The architectures and retrieval processes of these bots take lead of advances in machine learning to offer intelligent "information retrieval" processes.

Existing meal plan apps

MyFitnessPal is a trending mobile app which is created by Albert Lee and Mike Lee. “MyFitnessPal allows users to track calories, monitor progress toward weight-management goals, and gain support from an online community” (Rebedew,2015) [12]. Currently this app available in android, windows and iOS platforms. Also, this app has a big data set of nutrition and more than 5 million foods. MyFitnessPal notes on its blog that “75 million people using the app have lost a total of more than 180 million pounds”.

SapoFitness (Silva, Lopes, Rodrigues, Ray,2011) [9] is an application personalized to its user keeping a daily record of user’s food intake and everyday workouts. The key objective of this app is to provide weight loss and a stimulating tool for enhancing physical activity. SapoFitness is a challenged mobile application that delivers the action to the user, motivating for a healthy life pattern.

Keas is a meal plan app which is created by Damir Zekhtser (Zekhtser,2010) [10]. Keas produces a meal based on user health data, targets and other details. The data regarding user health goals, eating habits, current health condition, and eating preferences is accessed. Then diet plan is generated based on at least part of the information obtained. The meal plan may be modified based on user selected replacements, automatically generated substitutions, or in some other manner. Information on the food plan is updated when changing the meal plan.

According to the survey of “Diet app use by sports dietitians” (Jospe, Fairbairn, Green, Perry,2015) [11], target to study the distribution of using smartphone diet planning apps for dietary valuation in five countries to tracking among sports dietitians. Following figure explains the preference of diet application among sports dietitians.

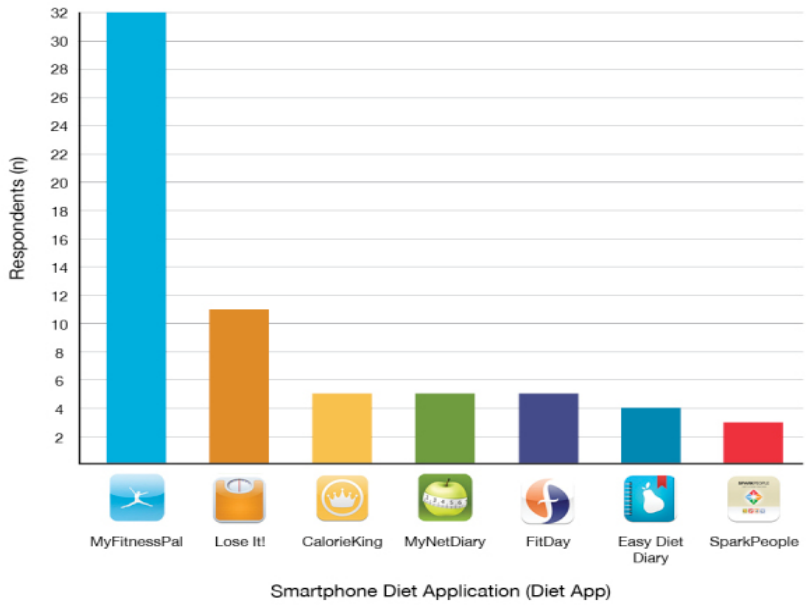

Figure 3.1 - a (Diet app use by sports dietitians)

"MyFitnessPal was overwhelmingly the most popular diet app, used by 56% (32/57) of diet app users" [11] and in conclusion most of the sports dietitians used mobile meal plan apps in sports food science and rated them as provide a great support to measure the dietary intake of athletes.

1.4: Findings

After deeply go through the existing projects above, this section outlines the technical approaches, language and algorithms and tricks used in existing projects.

Year	Chatbot	Technology	Language Tricks
1991	PC Therapist III (Weintraub, 1986)*	parsing, pattern matching, word vocabulary, remembers sentences	non sequitur, canned responses
1992	PC Professor (Weintraub, 1986)*		
1993	PC Politician (Weintraub, 1986)*		
1994	TIPS (Whalen, 1994; Hutchens, 1997)	Pattern matching, database like system	Model of personal history
1995	PC Therapist (Weintraub, 1986)*	Same as in 1991	Same as in 1991
1996	HeX (Hutchens, 1997)	Pattern matching, Markov chain models to construct some replies	database of trick sentences, Model of personal history, not repeating itself
1997	CONVERSE (Levy, 1997)	Statistical parser, pattern matching, modular with weighted modules, WordNet synonyms, list of proper names, ontology, database for storing facts	Proactivity
1998	Albert One (Garner, 1995)	Pattern matching, hierarchical composition of other chat bots (Eliza, Fred, Sextalk)	Proactive monologues
1999	Albert One (Garner, 1995)		
2000	A.L.I.C.E (Wallace, 2003)	AIML (Artificial Intelligence Mark-up Language – advanced pattern matching)	
2001	A.L.I.C.E (Wallace, 2003)		
2002	Ella (Copple, 2009)	Pattern matching, phrase normalization, abbreviation expansion, WordNet	Monologues, not repeating itself, identify gibberish, play knock-knock jokes
2003	Jabberwock (Pirner, 2003)	Parser, pattern matching – simpler than AIML, Markov Chains, Context free grammar (CFG)	
2004	A.L.I.C.E (Wallace, 2003)	Same as in 2000	
2005	George (Carpenter, 2006)	Based on Jabberwacky chatbot. No pattern matching or scripts. Huge database of people's responses.	
2006	Joan (Carpenter, 2006)		
2007	UlrtraHAL by Robert Medeksza*	Combination of VB code and pattern matching scripts	
2008	Elbot (Roberts, 2007)*	Commercial NLI system	
2009	Do-Much-More (Levy, 2009)*	Commercial property of Intelligent Toys Ltd.	
2010	Suzette (Wilcox, 2011)	ChatScript (AIML successor. Concepts, triples, variables)	
2011	Rosette (Wilcox, 2011)		
2012	Chip Vivant (Embar, 2011)	Not publicly disclosed. Common ontology and AI, responses taken from ChatScript (but not in ChatScript format or engine).	

Figure 1.4 - a (winners of Loebner winners)

Algorithms AND Technical approaches

Pattern Matching
This is the most common methodology used in chatbots. Distinctions of some pattern matching algorithm exist in every present chatbots. The pattern matching approaches can vary in their complexity, but the basic idea is the same. ELIZA and PC Therapist used this technology to give responses as a human.

Parsing
Textual Parsing is a method which takes the original text and converts it into a set of words (lexical parsing) with features, mostly to determine its grammatical structure On top of that, the lexical structure can be then checked if it forms allowable expression (syntactical parsing).
The earlier parsers were very simple, looking for recognizable keywords in allowed order. Example of such parsing would be that sentences "please take the gold" and "can you get the gold" would be both parsed into "take gold". With this approach the chatbot with a limited set of patterns can cover multiple input sentences. The more complicated parsers used in latter chatbots do the complete grammatical parsing of the natural language sentences.

Markov Chain Models
The Clue behind Markov Chain Models is that each incidence of a note or a word in some written dataset happens with a fixed possibility. The order of a model means how many successive incidences the model takes into the account. For example, if an input text is "agggcagcgggcg", then the Markov model of order 0 predicts that letter 'b' occurs with a possibility 2/13. The model with order 1 would state that each letter still occurs with a fixed probability, but that probability depends on the letter before.
Markov Chain Models process correct responses more viable. In some scenarios these models were even used to produce a garbage sentence that sounds right, as a failback method.

AIML
AIML's composition is XML based (A.L.I.C.E bots' methodology) and contains frequently of input rules with suitable output. The pattern must cover the entire input and is case insensitive. It is possible to use a wildcard which binds to one or more words. Due to simple and effective explanation, this and as well the Other examples are extract from papers.
AIML lets chatbots to have topics which give it a way to rank the patterns. AIML has <that> pattern as well, If it matches the transcription of the previous sentence then it has importance over the other rules.

ChatScript
ChatScript is intended to be a successor to the AIML language.
It focuses on a better signal that makes maintenance easier. It introduces additional functionality such as concepts, continuous existence, logical and / or variables, real triangles, and functions that solve problems in the zero language.

Language approaches and tricks

Non Sequitur
Non sequitur (Latin) is a row that has decisions which does not suggest from its properties. For a Example conversation would be: “Life is life and fun is fun, but it’s all so quiet when the goldfish die.”

Simulating keystrokes and typing errors
The judges interpret loop protocols as Loebner competes transparently. This push chatbots to “pretend” they are tying word by word. Some bots even lie spelling and spelling mistakes errors. This protocol is most controversial.

Canned responses
Canned responses are hard coded responses to all questions. To some range all of the chatbots designs could be calculated as canned responses. This would massively growth the number of patterns and would make them even more uncontrollable, so these replies are typically used only for things which cannot be protected with the primary chatbot technology.

Natural Language processing
Generally, modern bots use Natural Language processing techniques to Input versus analysis and output. Natural Language Processing (NLP) is the field of letting computers realize human languages. 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.

In conclusion of the findings as modern bots are using NLP, therefore best solution for building smart dietitian app is selecting NLP approach powered with machine learning. Therefore, this project uses the API provided by Microsoft called Language Understanding Intelligent Service (LUIS). It’s a well-developed REST API for Language Understanding.

Chapter 2: PROJECT MANAGEMENT AND TOOLS

2.1: Project Management

A project management strategy and tool selection tool were very important for the project because of the research nature of the project. The dietitian bot is result of a strategic review. In the market there is a good reputation to meal app. However, after do a strategic review that could find many of them are proceed with manual way whereas with the competition of the world people are spend with a busy life style there for there is a need of automate dietitian app rather than having a meeting a real dietitian. With that in order to successfully achieve goals of project like this there should be a good project management.

2.1: Project Planning

Project planning is very important for the success of the project. For the successful of the project need to manage following,

- Schedule - To dietitian app there was a well-defined schedule. But in some instance schedule get change due to external problems. However, by managing schedule in a proper way it helps to successful development of the app.
- Scope - As for the app initially scope was unclear how ever due to proper project management could map a well define scope. There for at the middle of the development process there were no any requirement changes as there is a good understand about the scope.
- Risk - in order to avoid from obstacles and meet the objectives of the application need to manage risk properly.
- Resources - By managing the resources could avoid from lack of resources problem. There for there was no any break point in the process of development due to any lack of resources.
- Quality of the work - by managing overall quality of the product could give a satisfactory output.

In order to manage projects following project planning tools were helped.

- Favro – This tool supported to divide the entire project workload into several tasks and set deadlines and reminders for each divided task.

To version control and to manage file following tools were used

- GitHub
  From the beginning of development all program resource files were maintained in a private GIT repository on GitHub. Some key benefits of using GitHub for this project were reverting to a previous commit which is error free and create different branches for different application areas and unsure application areas.

- GitKraken
- Git kraken is a client app with pro features like built in merge conflict output editor, and with a perfect GUI for version control rather than bash. Git kraken provides Git hosting service and connect to preferred remote repositories easier. Gitkraken benefits were same as before.

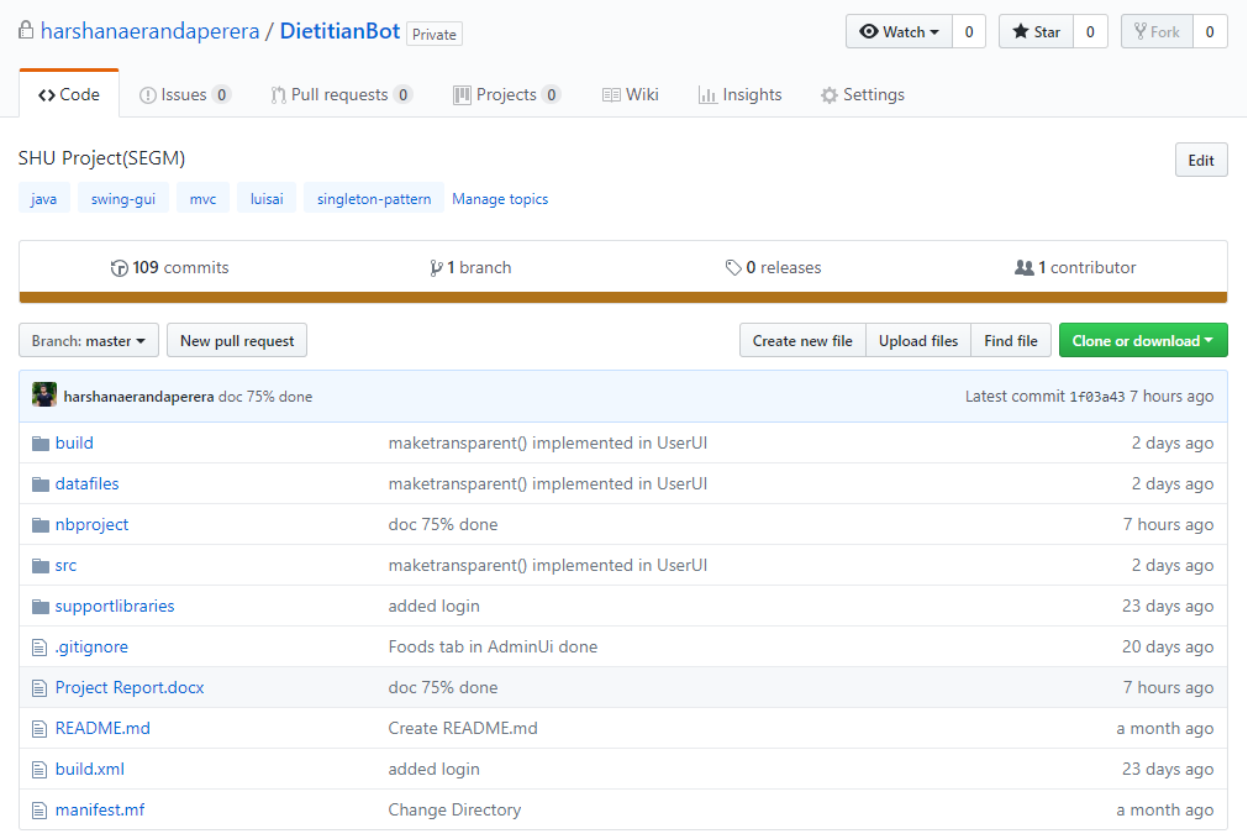

Figure 2.1 - a (GitHub Repository)

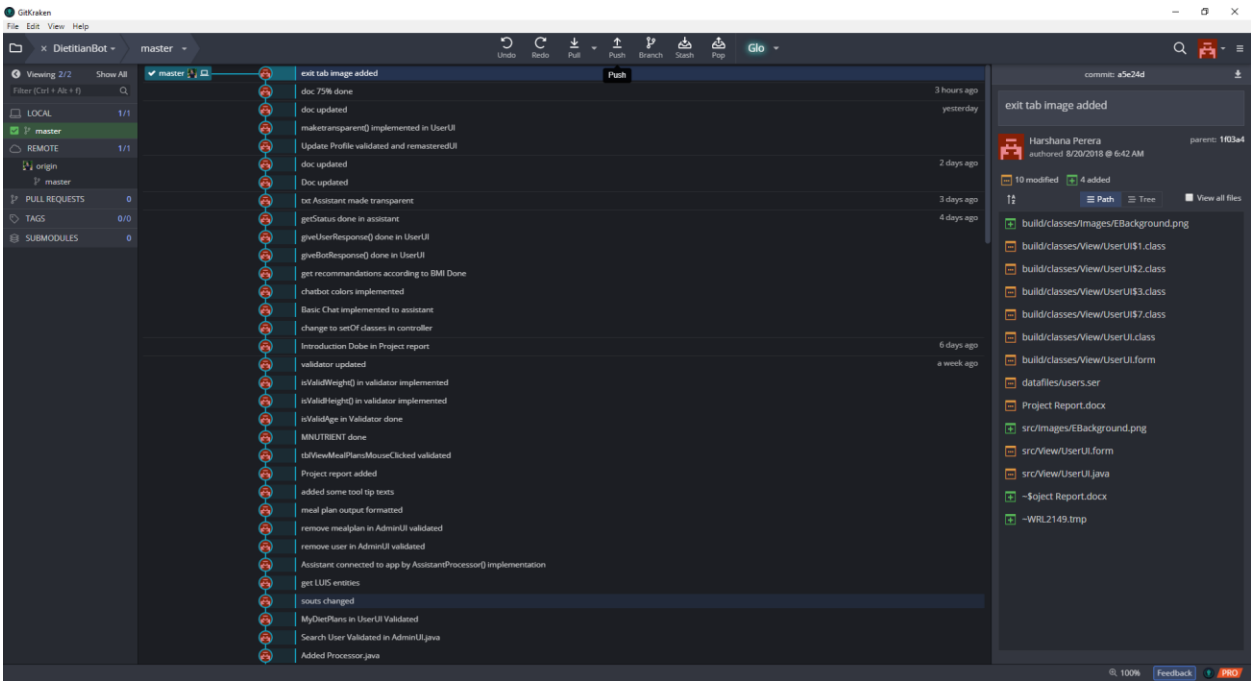

Figure 2.1 - b (GitKraken Client)

Chapter 3: DESIGN AND DEVELOPMENT

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

After deeply go through the requirements above the entire process was outlined in the architecture overview.

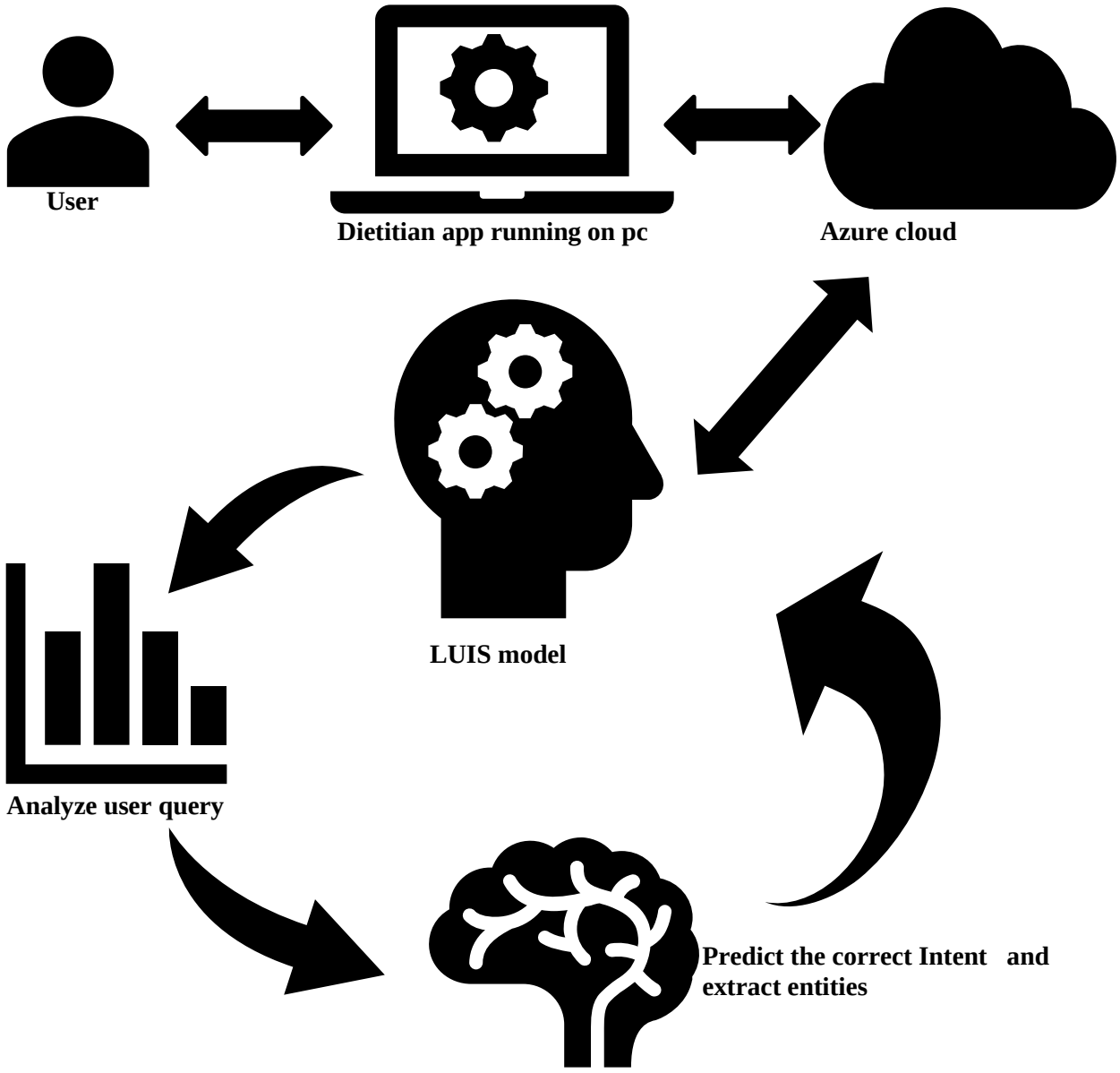

Figure 3.2 - a (architecture overview)

Above diagram represents the process of entire application in short. The process of dietitian app can be break down into following Steps.

- User start the dietitian app.
- Application validates the user's identity. If user didn't verify, user must follow the sign-up procedure.
- After successful user identification, user can use the meal plan app in offline or get the help from dietitian assistant.
- If user prefer to get all information (suggested meal plans, recommendations etc.) from assistant, user can ask questions related to application domain. If user ask questions out of application domain app, then assistant gives a suitable response according to the situation.
- If user ask query from assistant, the application pass the user's query to Azure cloud.
- After that Azure cloud connect with LUIS model.
- LUIS model analyzes the query and predict overall meaning, and extract attributes which is important to process in the application.
- Then LUIS model returns the attributes back to the application.
- Finally, application give good responses to the user by processing received attributes (intents, entities etc.) from LUIS model.

3.3: PROGRAMMING LANGUAGE

Smart dietitian bot is fully developed using Java, as there are built-in functions which are different API of JDK that supporting for app. Therefore, java provides rich solutions for this project criteria.

- ArrayList
  Model the real-world objects in ArrayLists are easy, ArrayLists are more than just arrays. In this app SetOfUsers and SetOfMealPlans are the ArrayLists that used to store users and meal plans.

- String class
  Operation of String concatenation support for populating responses of the assistant in a proper way. Also, tostring () function used to conversion process of strings.

- StringBuilder
  StringBuilder plays a major role in assistant by appending user query along with bot response. Append function in StringBuilder supported to make better live chat interface.

- parseInteger () and parseDouble ()
  These in-built functions in Integer and Double classes helped to conversion process of double and Integers and values.

- Vector class

Vector class also like ArrayList and implement dynamic array but has some changes. In this application vector class help to add rows to some tables in admin interface.

- Regex functions
  In Java.util package provide regex built in functions like Pattern.compile(),matcher() Supported to validate some components in application like user registration, profile creation and guard from wrong entity values coming from the assistant etc.

```
public boolean isValidEmail(String email) {
    return email.matches("^([\\w\\.\\-]+)@([\\w\\-]+)((\\.(\\w){2,3})+)$");
}

public boolean isValidName(String name) {
    return name.matches("[a-zA-Z]+");
}

public boolean isValidNumber(String num) {
    return num.matches("[0-9]{1,13}(\\.[0-9]*)?");
}
```

Figure 3.3 - a (regex pattern)

- Support for file-oriented storage
  In java.io package provide Serializable interface for Java serialization. Also, FileInputStream and FileOutputStream do the file read and write process. Following is a screen shot of how the application store data in a file.

```
public void SerializeUser() {
    try {
        FileOutputStream ufos = new FileOutputStream(new File("datafiles/users.ser"));
        ObjectOutputStream uboos = new ObjectOutputStream(ufos);
        uboos.writeObject(users);
        uboos.flush();
        uboos.close();
    } catch (Exception e) {
        System.out.println(e);
    }
}
```

Figure 3.3 - b (Serialize users to a file)

- Support of handling http request/response
  Apache HttpComponents Client external library is used to make http requests and responses. Purpose of using this external library is to connect with Microsoft LUIS API.

- Support of handling json objects

After hit the LUIS endpoint then it returns the result as json response. Then from the methods of the java-json.jar external library, json response converted to a java object and pull the relevant data from that java object.

- Support to build attractive user interface for desktop application.
  For the front end of application swing API provides a bunch of lightweight components that, to the Highest level possible, work parallel on all platforms.

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.”(LUIS Documentation)

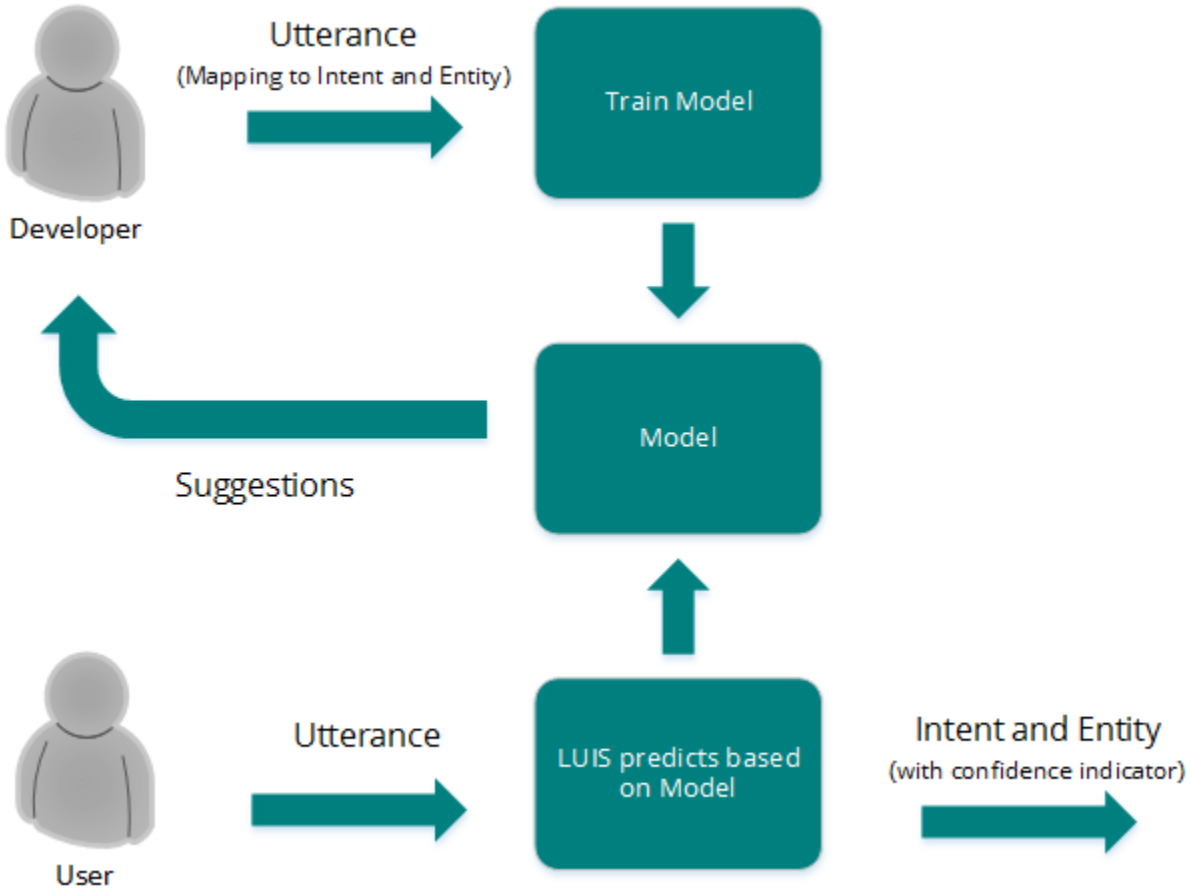

Figure 3.4 - b (LUIS overview (Pedley, 2017))

A LUIS model includes:

- **intents**: An intent represents a task or action the user wants to perform
- **entities**: exact kinds of information in utterances such as email, number or name
- **example utterances**: instance script a user enters in the application

Intents

An intent's purpose or goal expressed in a user's utterance. Each utterance is going to be classified into one intent, that is like the concept of "class" in supervised learning. 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" 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

The entity represents complete information found within the utterance that is relevant to the user's request. An utterance can include many entities or none. Entities are optional but highly recommended. In this project Assistant use the list of entities as parameters to start an action. For example, in the utterance "Set my weight to 50 kilograms", a user asking from assistant to update his/her current weight to 50 kilograms, two entities are found "50" indicating user’s current weight and " kilograms" indicating the unit of mass. Then assistant call the Setters of related java classes to set the new value which is pull from user utterance and update current values. After that Assistant populate successful response to the user interface.

Example utterances

The example word is text input from the user that the consumer application must interpret. To train LUIS to extract intents and entities from them, it's important to add variety of different example utterances for each intent. Active learning, or the procedure of continuing to train on new utterances, is absolutely necessary to machine-learned intelligence that LUIS provides. It can be a sentence, like “Let me know how many kilograms to loss to be in the ideal weight?”, or a part of a sentence, like “my BMI.” Utterances are not consistently well-formed, and there can be abounding utterance variation (that mean identical however made another way in word length and word placement) for a specific intent. Due to that reason, this project includes at least twelve example utterances to each intent.

Design plan of LUIS app for Smart Dietitian Bot

It is necessary to plan app before start making it in LUIS. Therefore, the design plan breakdown into several stages.

- Prepare a schema of the possible intents and entities that are pertinent to the domain of real dietitian and food nutrition science.

- Create an app in LUIS.
- Build model by adding intents, example utterances and label with entities according to the schema which is planned before.
- Train the model and test prediction results.
- Improve prediction accuracy by reviewing endpoint utterances, adding phrase lists and patterns.
- Finally retrain the model and publish.

Example user utterance	**Intent**	**Entities**
"Show my diet plans? "	getDietPlans	
"My name is **John**"	setName	John
"Set my name to **Peter**"	setName	Peter
"Hello, Set my age to **30**"	setAge	30
"I am **40** years old"	setAge	40
"Could you please tell my **email** "	getEmail	
"What is my **email** "	getEmail	
"Show my **email** "	getEmail	
"Set my height to **180** centimeters"	setHeight	180, centimeters

Example JSON endpoint responses


```
{
 "query": "show my diet plans",
 "topScoringIntent": {
  "intent": "getDietPlans",
  "score": 0.9131645
 },
 "entities": [],
 "sentimentAnalysis": {
  "label": "positive",
  "score": 0.813097
 }
}
```



```
{
 "query": "set my name to John",
 "topScoringIntent": {
  "intent": "setName",
  "score": 0.89418155
 },
 "entities": [
  {
   "entity": "john",
   "type": "username",
   "startIndex": 15,
   "endIndex": 18,
   "score": 0.8410931
  }
 ],
 "sentimentAnalysis": {
  "label": "positive",
  "score": 0.8668431
 }
}
```

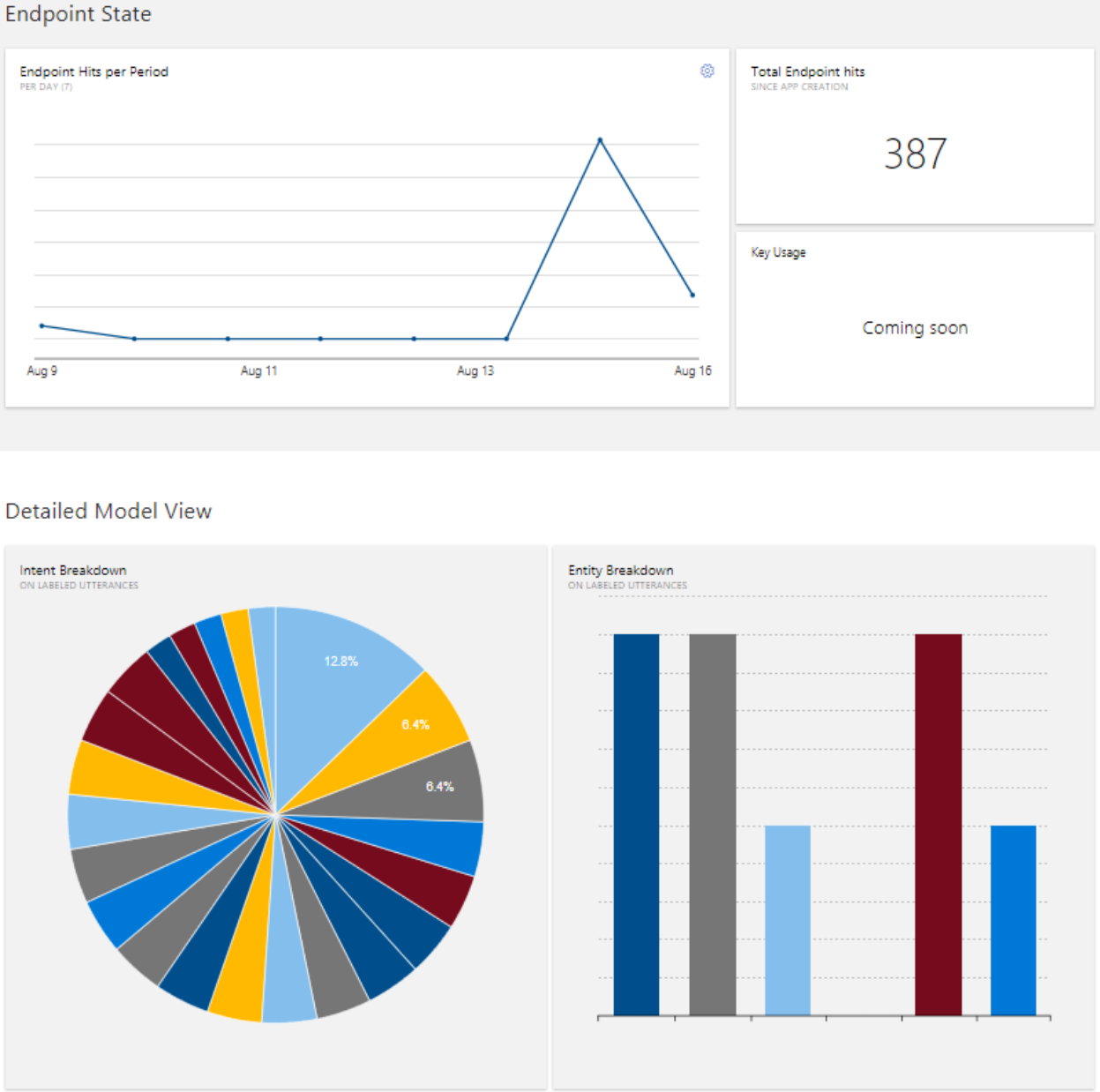

Figure 3.4 - c (LUIS dashboard)

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.

3.6: DATABASE

As this dietitian project is a portable application, picking the most appropriate database or a file-oriented system was one of the major tasks. Following were the critical points to take a decision about final storage approach.

- Support for data encryption and authentication to ensure security of the application.
- Support for data compression.
- Support for object-oriented programming to ensure flexibility of data models.
- Should be maintain less.
- Support for the schema-less approach.
- Support for portability.
- Support from the community.

After deeply analyzing above critical points and need of developing portable desktop application, Java serialization mechanism selected as the best choice. Object serialization is a method in which the object's state is transformed into a byte stream while deserialization is the reverse process of serialization. The most exciting fact is that the whole process is JVM independent, that means support for multiple platforms. Ultimately, this dietitian bot can run on different platforms without any data loss and without maintain. Especially the community like Stack overflow, Stack exchange were used to finish the job properly.

3.7: TOOLS

NetBeans by Oracle Corporation was used as the IDE for this project. Here are top reasons to use the NetBeans IDE to develop the application.

- Provide a great Support for Java Standards and Platforms
- Powerful GUI Builder for built attractive user interface for application
- Free and Open Source
- Profiling and Debugging Tools
- Integrated Terminals
- Inbuilt Git Integrations make it easy to push and pull.

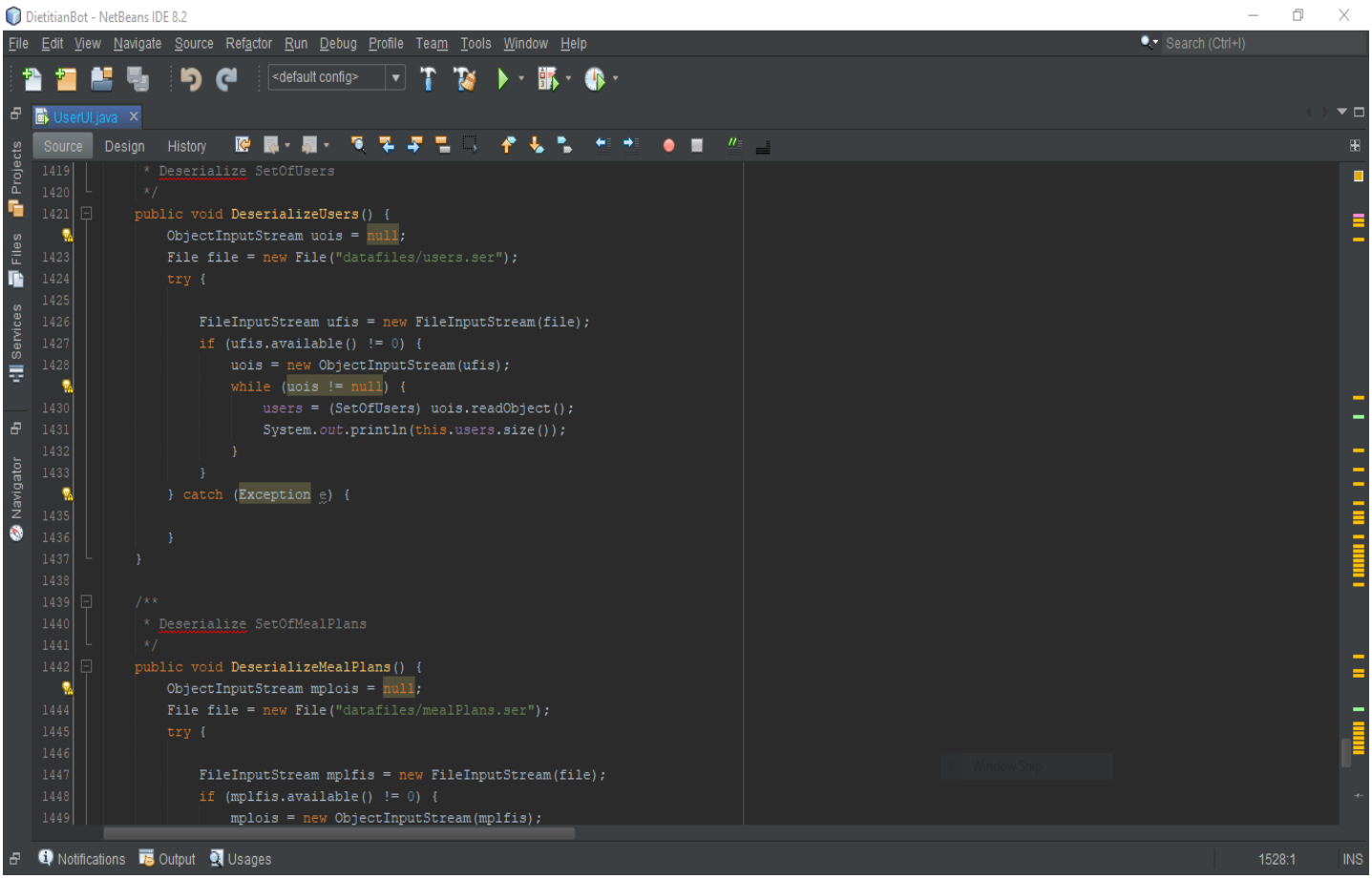

Figure 3.7 - a (Netbeans IDE)

3.8: APPLICATION UI

Netbeans support for Swing and JavaFX UI frameworks. Swing was selected to design the entire application as it's easy to build. Following are the example figures of application which is created by Swing.

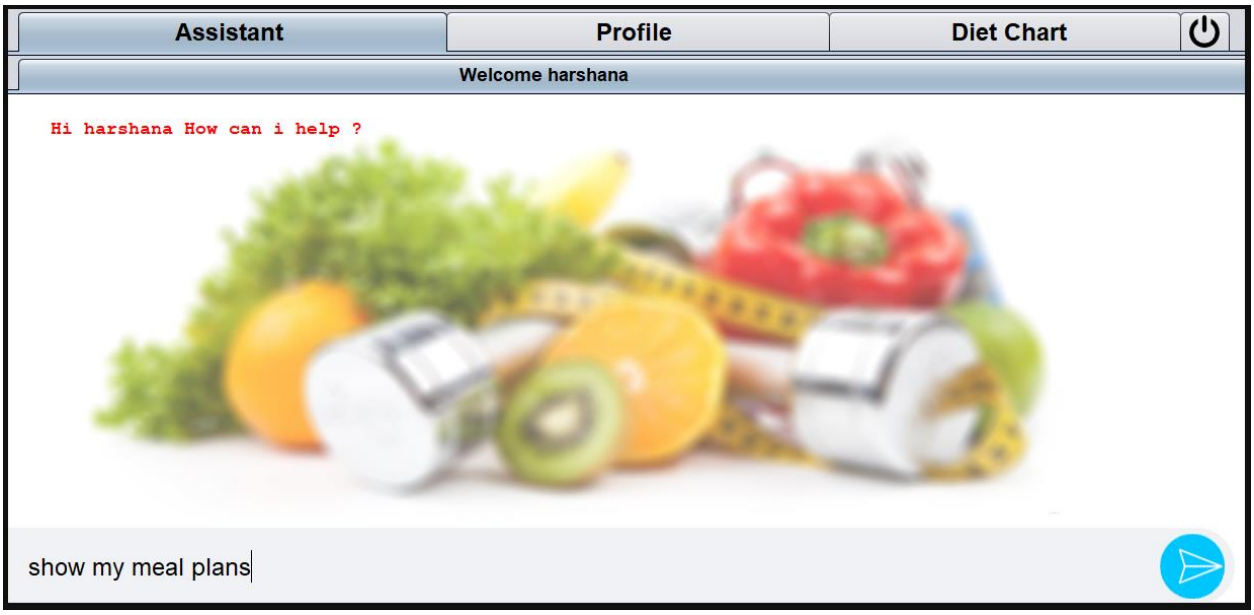

Figure 3.8 - a (diet assistant)

Figure 3.8 - b (user profile)

Chapter 4: Critical Reflection and Evaluation

Evaluation perspectives

There are a number of different perspectives on how to evaluate chatbot performance. From an information retrieval (IR) perspective, chatbots have specific functions: there are virtual assistants, question-answer and domain-specific bots. Evaluators should ask questions and make requests of the chatbot, evaluating effectiveness by measuring accuracy, precision, recall, and Fscore relative to the correct chatbot response. [14]

From a user experience perspective, the goal of the bot is, arguably, to maximize user satisfaction. Evaluators should survey users (typically, measured through questionnaires on platforms such as Amazon Mechanical Turk), who will rank bots based on usability and satisfaction. From a linguistic perspective, bots should approximate speech, and be evaluated by linguistic experts on their ability to generate full, grammatical, and meaningful sentences. [14]

Finally, from an artificial intelligence perspective, the bot that appears most convincingly human (e.g. passes the Turing Test best) is the most effective. [14]

4.1: Test Plan

Testing of a software product is very vital because of the following reasons.

- help to identify the bugs that have occurred in development.
- It ensures the final product is a software with minimum bugs.
- It ensures customer satisfaction for the built product.
- It ensures maximum reliability expected from the customer.

Following are the main dimensions of testing criterion needed for this project.

- Performance
- Acceptability

Performance and acceptability are the key factors that must be ensured. The system anticipated that the response from the diet assistant should end in 4 seconds to ensure that experience is guaranteed. For the quick response, the performance and accuracy of LUIS model is important. To ensure the acceptability, conduct a survey is necessary.

Performance of LUIS Model

After proper training in LUIS model, testing is necessary because sometimes LUIS predicts the wrong intent as top score intent. To avoid this testing with sample utterances are important to ensure if the recognized intents and entities are correct and error free. For an example If the expected intent is incorrect make it by selecting the correct intent from the drop-down list is necessary. Figure 4.1 represent the testing panel of LUIS.

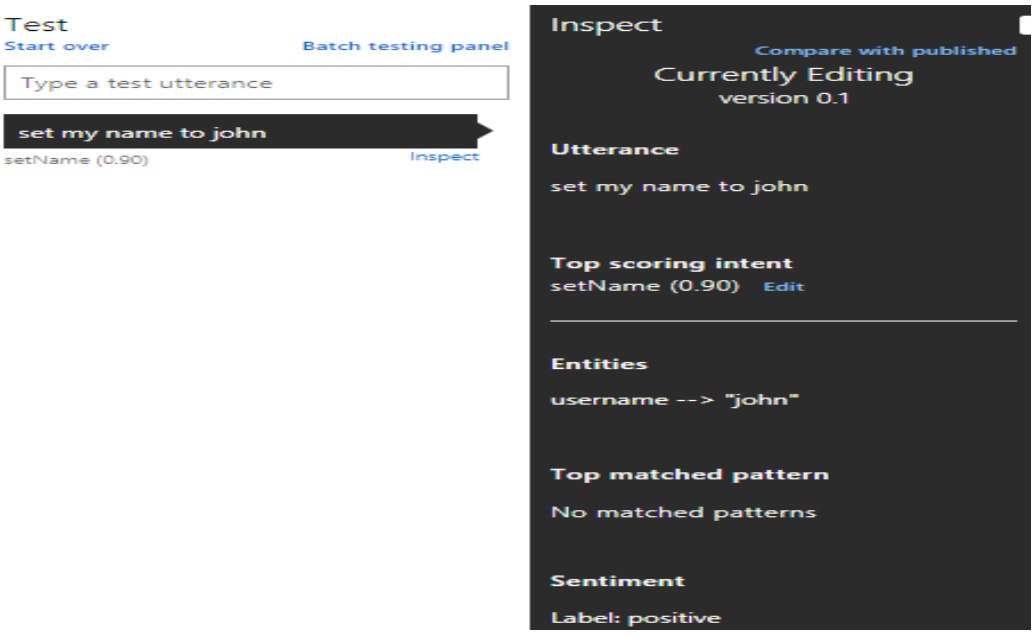

Figure 4.1 - a (Improving prediction accuracy of LUIS.)

Project acceptability

The acceptability is measured by conducting a survey. For the sample 30 students were selected as users of dietitian app and let users to give rates for the critical points of the application.

As system is a desktop application, the most suitable method to conduct the survey was the paper-based method. The survey form and distributed among the selected evaluators(users) after they completed the inspection on application. Then they gave their individual responses and suggestions through the survey form.

Survey Form / questionnaire

Please put a ✓ n the selected box and if you have any comments or suggestions regarding our system please write them down under the comments and suggestions columns.

1- Excellent/Very Satisfied
2- Very good/Satisfied
3- Good/ Somewhat Satisfied
4- Fair/ Somewhat Unsatisfied
5- Poor/ Very Unsatisfied

		1	2	3	4	5	COMMENTS
i.	How well the conversation logic of assistant?						
ii.	How well the Accuracy of language understanding?						
iii.	How well the application's ease of use?						
iv.	How likely is that you would recommend this app to other party?						
v.	How well the reliability of this application?						
vi.	How satisfied are you with generated meal plan?						
vii.	How satisfied are you with the dietitian app recommendation?						
viii.	How well the applications security?						
ix.	How well the consistency with Interface?						
x.	How well the overall performance?						

4.2: Test results

Following responses have been obtained from the results of the survey.

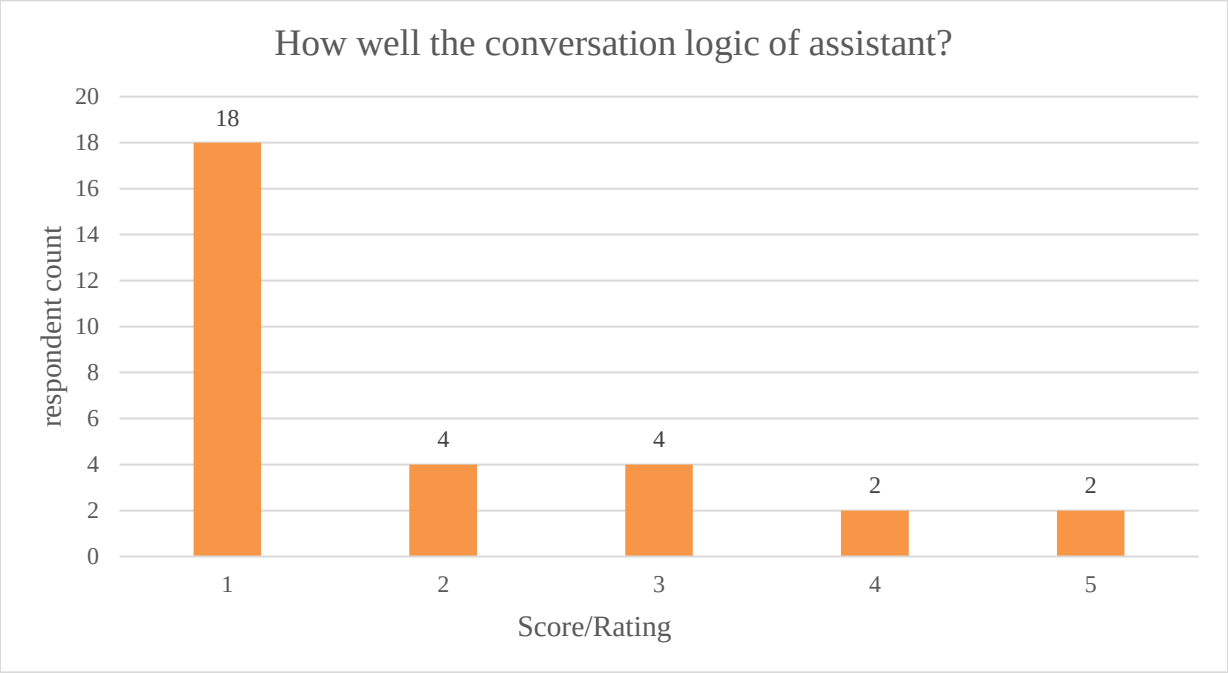

Figure 4.2 - a (Survey Result 1)

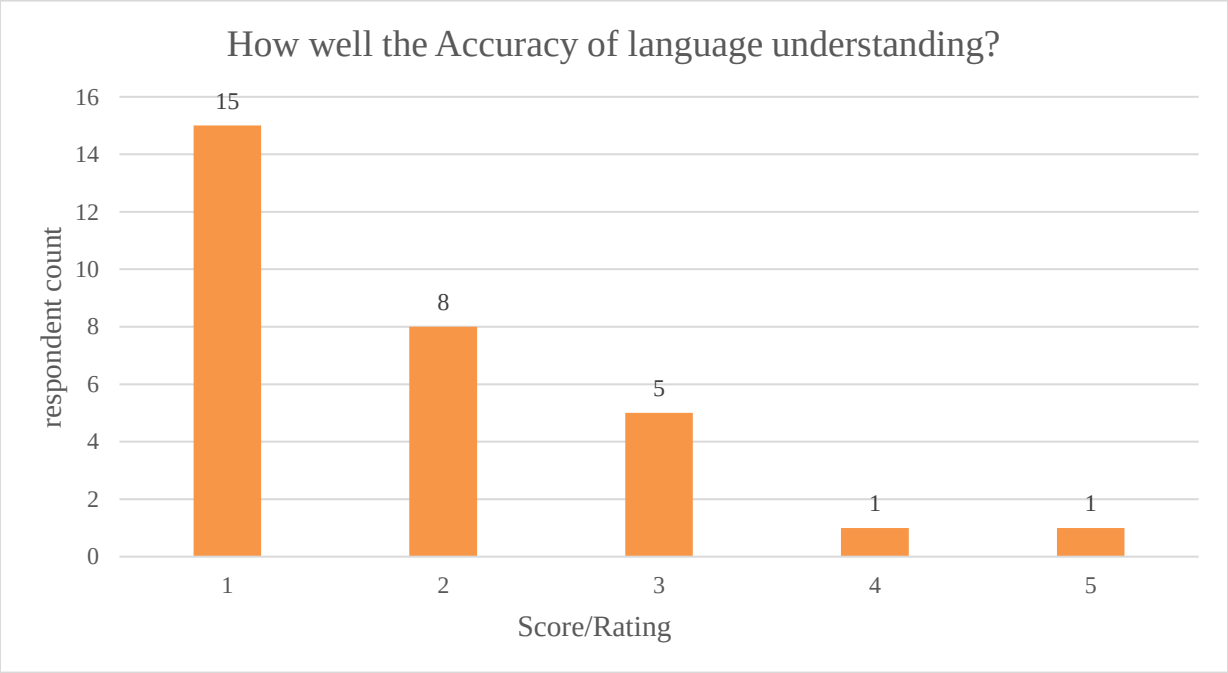

Figure 4.2 - b (Survey Result 2)

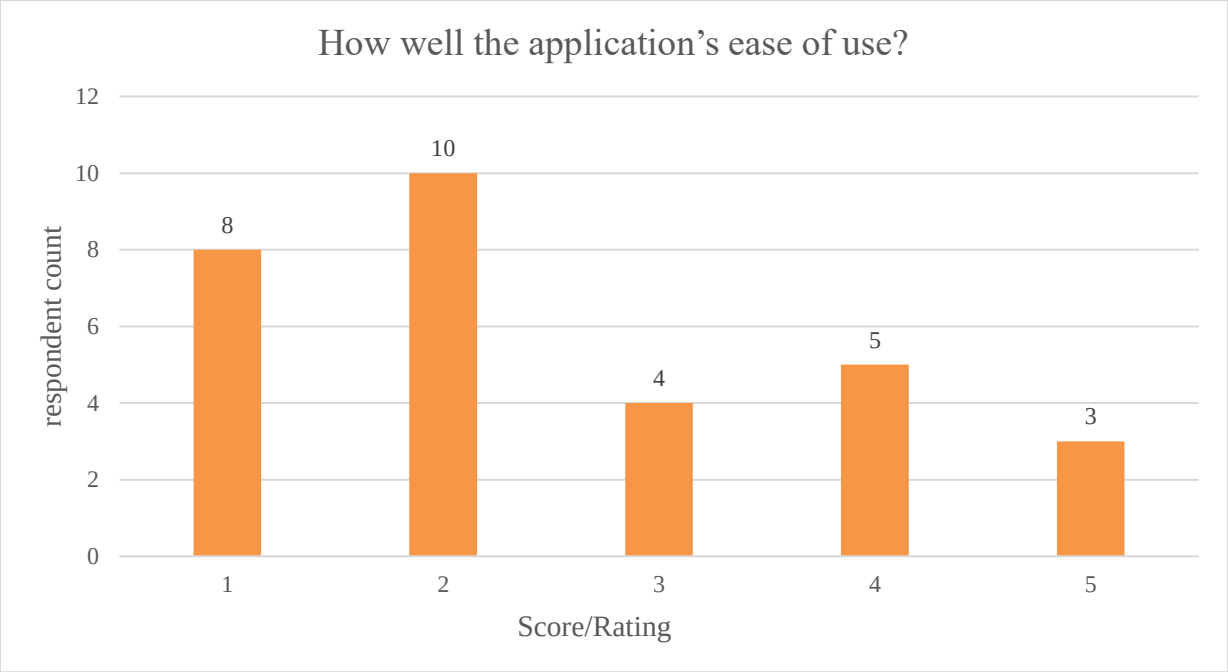

Figure 4.2 - c (Survey Result 3)

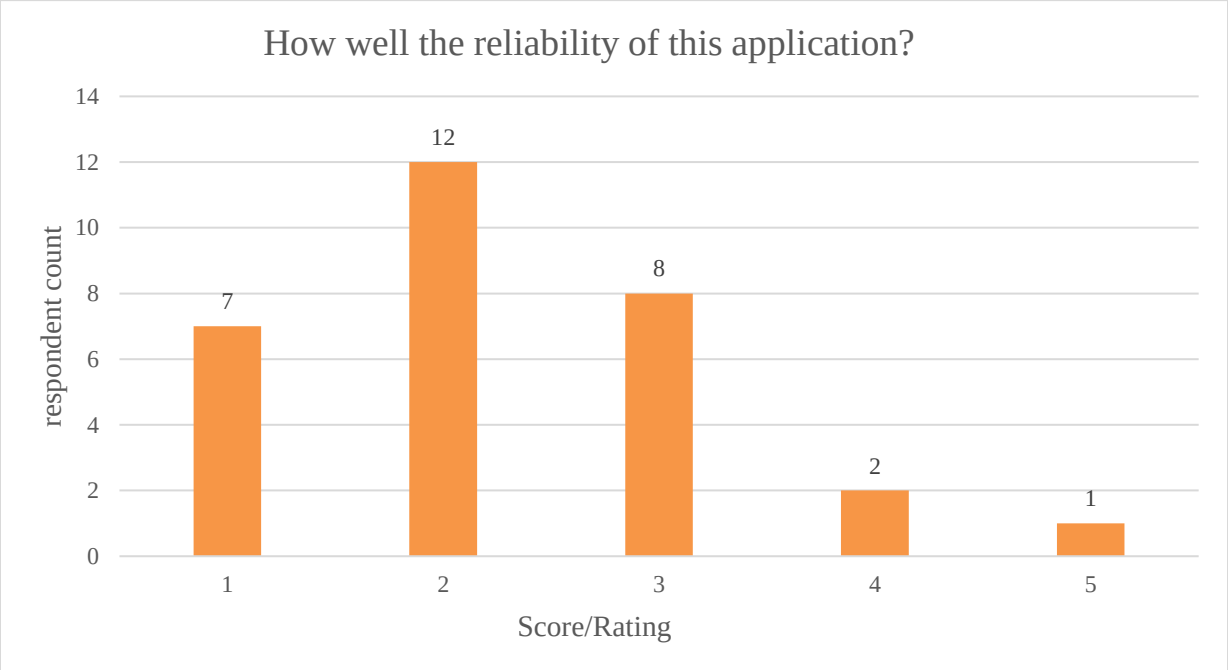

Figure 4.2 - d (Survey Result 4)

Figure 4.2 - e (Survey Result 5)

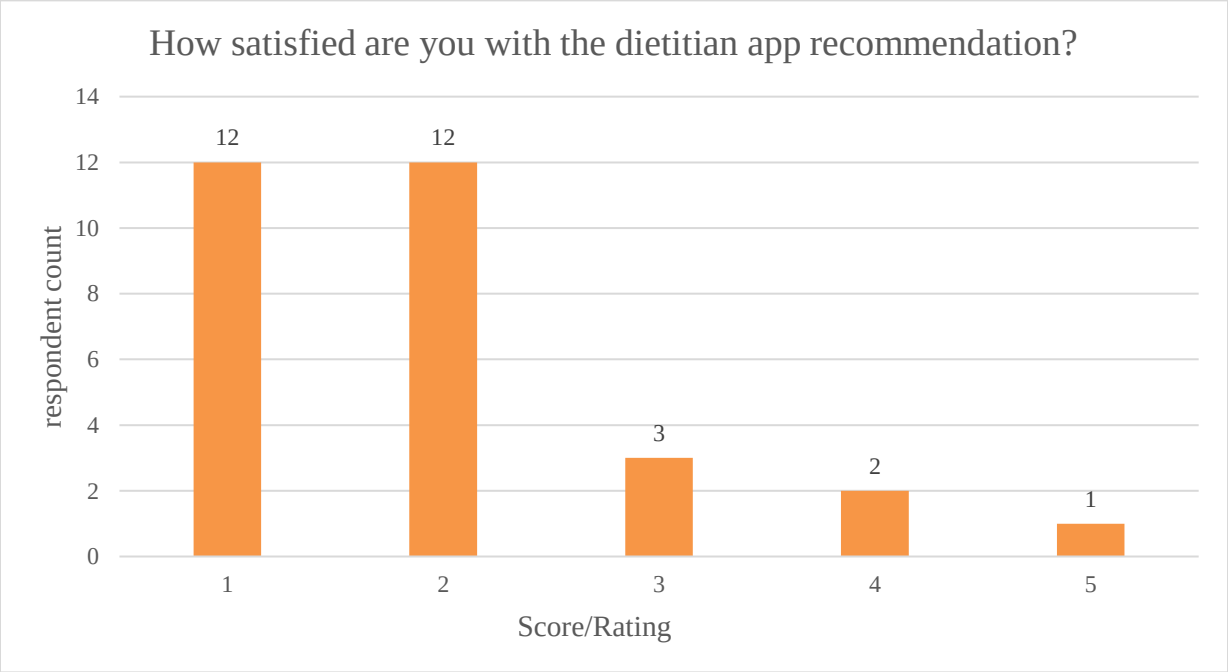

Figure 4.2 - f (Survey Result 6)

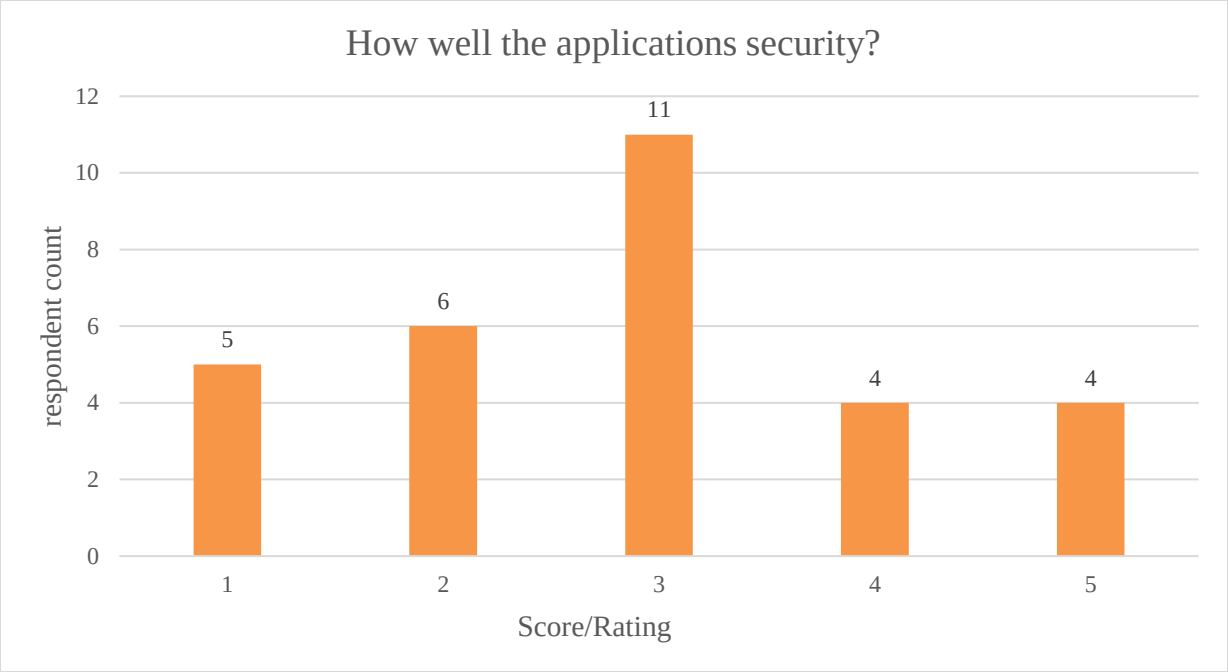

Figure 4.2 - g (Survey Result 7)

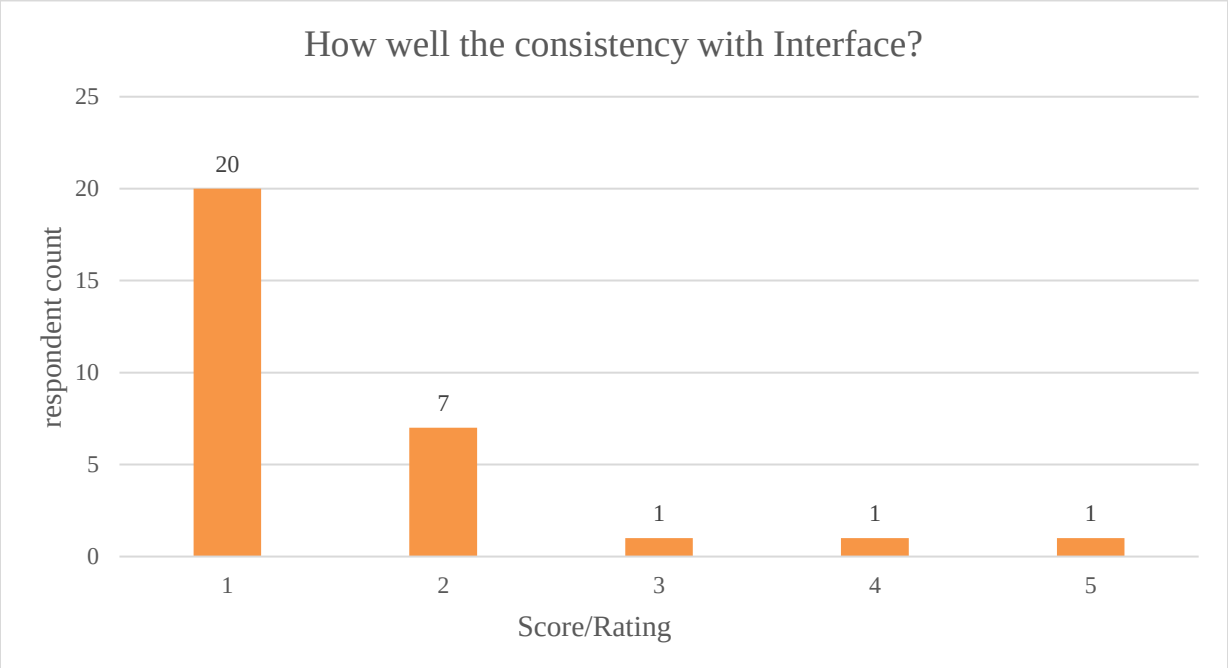

Figure 4.2 - h (Survey Result 8)

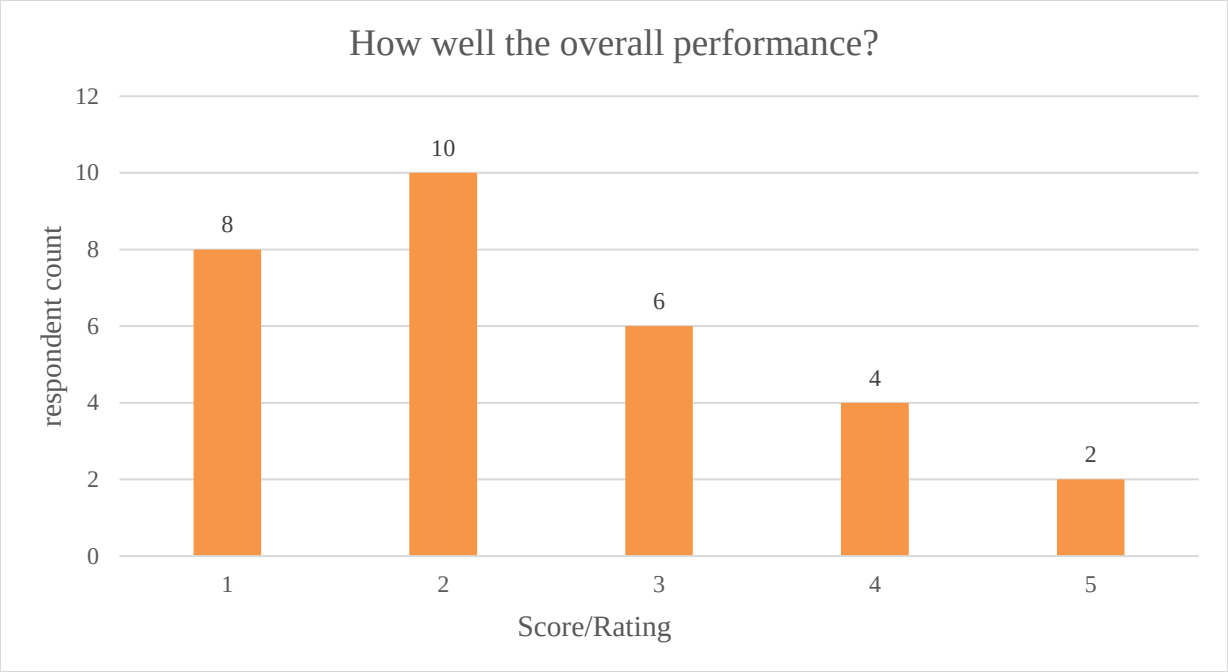

Figure 4.2 – i (Survey Result 9)

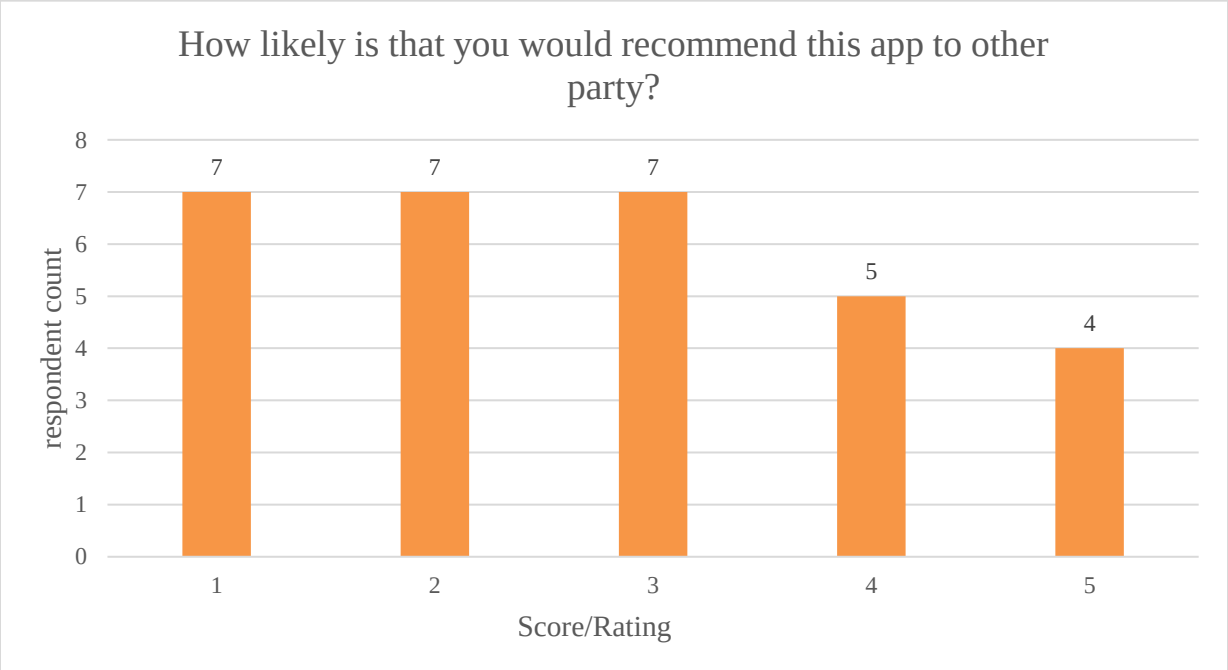

Figure 4.2 – j (Survey Result 10)

4.3: Findings

Following are the key findings from the survey results.

According to test results above section,

- In figure 4.2 – a, highest number of respondents rated as excellent on assistant as it has a good conversion logic. Only four respondents rated as fair and poor similarly.

- In figure 4.2 – b, highest number of respondents rated as excellent on assistant and eight respondents rated as very good as it has a good accuracy on language understanding. Only one respondent rated as poor.

- In figure 4.2 – d, highest number of respondents rated as very good on assistant and eight respondents rated as good as application has a good reliability. Only one respondent rated as poor.

- In figure 4.2 – g, highest number of respondents rated as good, but eight respondents given negative comment on application. Therefore, security of this application cannot be guaranteed.

- In figure 4.2 – h, overall rating for consistency with interface is great because 20 respondents rated excellent.

- In figure 4.2 – i, overall performance is good because 24 respondents given a positive rating while eight responses are negative.

4.4: DISCUSSION

According to the above founded results in 4.3: Findings, this proves that Smart Dietitian Bot is working properly. Because overall comment is positive, but there is some sort of few negative ratings found which is considerably low. First key finding ensures that “need of conversation between user and assistant should flow and always try to keep the user in control of the conversation which is mention as a design requirement” and “If user ask questions beyond the domain of application, then application should understand the domain and give a proper reply.” in subsection 3.1 in design and development phase. Second key finding ensures that “The application should predict the correct intent (highest score intent) by analyzing user given utterance” and “The application should extract the correct entities by analyzing user given utterance.” which is mention as a design requirement.
Also, fifth key finding ensures that “User interface of the client-side should be user-friendly to provide better user experience”. After analyzing all key finding in above subsection 4.3, This dietitian project successfully fulfilled the all requirements in design phase. The survey did a great job to ensure the project is successful.

4.5: FUTURE DEVELOPMENT

This project opens a gateway to many additional features and functions in the future if proper planning, development and integration will be carefully carried out. Most of the ideas were ignored because It does not make sense to implement them in the initial release. Some of the few innovative concepts that will add for the stable releases as given as follows.

- Integration of speech to application
  As currently the dietitian bot is working with text conversations. Integration of voice will make the bot more human. Then users can ask questions from the bot in a more intelligent manner. Voice assistant will be a big step in the application.
- Implement a mobile app with vision enabled and object identification feature.
  With the idea of enabling vision, then bot can identify the foods and give quick recommendations for user to eat or not.
- Improve security of the application
  The results of the survey show that the application has some backdoors. Therefore, in the next release will close these backdoors.

With the above mission, there are many features targeted to build a stable version and make it cross platform for increase the audience rapidly.

4.6: GAINED LESSONS

This project was the most challenging experience faced because selecting a best approach to build a bot within the limited time and the area of artificial intelligence is a totally new theory.
During the testing phase of the project, had some problems with balancing the project work and other subjects. Time was limited as a student, and if better time management was planned out perhaps at least one idea in the future development could be achieved.
When fell in to a precise approach of developing dietitian bot was a very fascinating learning experience. As a student, Interest in the creation of new apps increased.
Efficient planning is important to obtain proper results for a project. A project plan must be repeated over and over again, and possible changes must be made before starting the project work. When planning this project by creating a timeline and carefully identifying each milestone that helped to develop management skills. Also, the objective of the project is very challenging because during the project, I expanded my knowledge in different areas which help to achieve targets in future career path.

4.7: CONCLUSION

The successful outlook on the Smart dietitian bot depends on the market acceptance.
When comparing with other meal plan apps which is in the market is not intelligent. From the survey results I think I did a really good job within the limited time period. The deliverable meets most of the aims of the project which confirmed in section 4.4: DISCUSSION.
Whenever there is a need of choosing projects, Typically I'll try to select the projects that will give me good learning exposure. Therefore, by considering the learning curve I selected AI dietitian app without any doubt.
It was exciting to visit people and specially my supervisor and explain this project to them and listen to their tips and advice on how this could be improved further and how this project could be brought to a success.
Finally, this project has supported me with my time management, planning and developer skills.
Even though I have finished the deliverable for SHU Final Year project module, as this project was positive and promising, I will continue the development until stable releases.

Chapter 5: References

1. Katulanda, P., Jayawardena, M. A. R., Sheriff, M. H. R., Constantine, G. R., & Matthews, D. R. (2010). Prevalence of overweight and obesity in Sri Lankan adults. *Obesity reviews*, *11*(11), 751-756.
2. Turing, A. M. (2009). Computing machinery and intelligence. In *Parsing the Turing Test* (pp. 23-65). Springer, Dordrecht.
3. Weizenbaum, J. (1966). ELIZA—a computer program for the study of natural language communication between man and machine. *Communications of the ACM*, *9*(1), 36-45.
4. Colby, K. M. (1999). Human-computer conversation in a cognitive therapy program. In *Machine conversations* (pp. 9-19). Springer, Boston, MA.
5. Colby, K. M. (1981). Modeling a paranoid mind. *Behavioral and Brain Sciences*, *4*(4), 515-534.
6. Bradeško, L., & Mladenić, D. (2012). A survey of chatbot systems through a loebner prize competition. In *Proceedings of Slovenian Language Technologies Society Eighth Conference of Language Technologies* (pp. 34-37).
7. Wallace, R. S. (2009). The anatomy of ALICE. In *Parsing the Turing Test* (pp. 181-210). Springer, Dordrecht.
8. Carpenter, R., & Freeman, J. (2005). Computing machinery and the individual: the personal Turing test. *http://www. jabberwacky. com/personaltt*.
9. Silva, B. M., Lopes, I. M., Rodrigues, J. J., & Ray, P. (2011, June). SapoFitness: A mobile health application for dietary evaluation. In *e-Health Networking Applications and Services (Healthcom), 2011 13th IEEE International Conference on* (pp. 375-380). IEEE.
10. Zekhtser, D. (2010). *U.S. Patent Application No. 12/604,872*.
11. Jospe, M. R., Fairbairn, K. A., Green, P., & Perry, T. L. (2015). Diet app use by sports dietitians: a survey in five countries. *JMIR mHealth and uHealth*, *3*(1).
12. Rebedew, D. (2015). MyFitnessPal. *Family practice management*, *22*(2), 31-31.
13. Weinberger, M. (2017). Why Amazon's Echo is totally dominating–and what Google, Microsoft, and Apple have to do to catch up. *Business Insider Inc*.
14. A framework for chatbot evaluation. (2017, January 24). Retrieved from https://isquared.wordpress.com/2017/01/24/a-framework-for-chatbot-evaluation/