SNAPYUMMY

Intelligent Cooking Assistant

User Guide

Intelligent Reasoning Systems

Practice Module

Semester One 2021/2022

PREPARED BY:

Student Name	Student ID
Chen Ziqi	A0231425R
Li Peifeng	A0231447J
Mediana	A0231458E
Xu Xuanbo	A0231436M

Contents

1 Objectives

The objective of this document is to provide an overview of SnapYummy application and the necessary information to use the application. The manual assumes that the reader has sufficient understanding on system implementation, machine learning, programming language (Python), knowledge reasoning and representation.

2 Scopes

The high-level scope of the user guide will encompass THREE (3) sections:

1. System Overview
2. Installation and Configuration
3. User Manual (Use Case)

3 System Overview

Figure 1 shows the system overview for SnapYummy. Following are the components used in SnapYummy.

1. Telegram – Chat Messaging Platform for interaction between SnapYummy and users
2. Yolov5 – Object detection to detect the ingredients uploaded by users
3. DialogFlow – Intent detection to detect users' request and intention
4. Neo4J – Knowledge graph to search for the search the recipe according to users' requirement

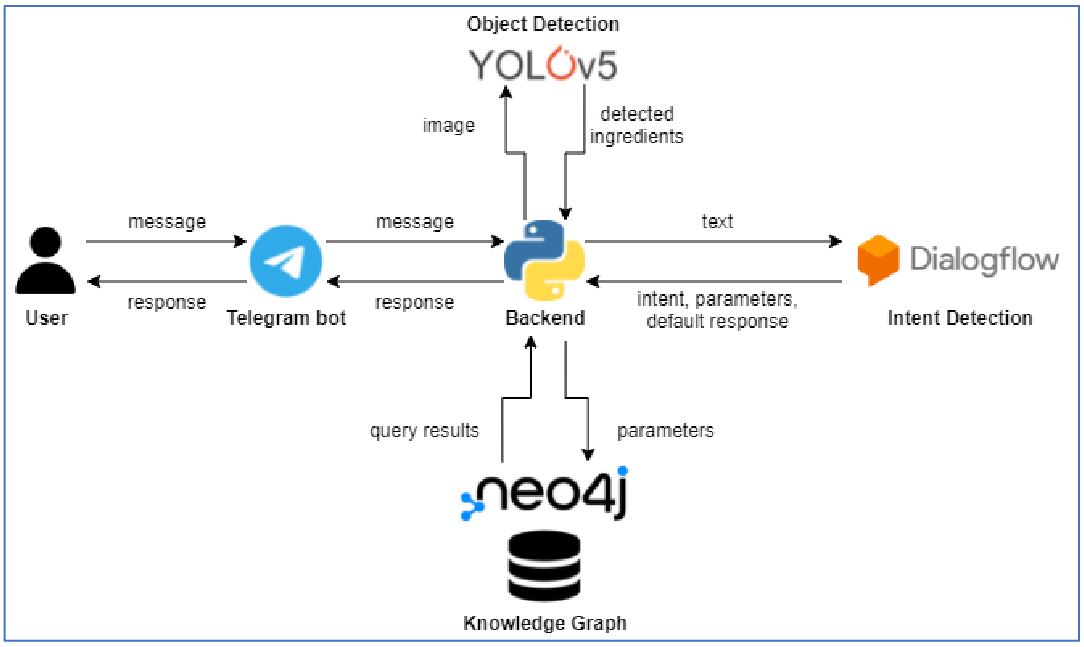


Figure 1 System Overview

4 Installation

4.1 Requirements

Description	Specification
Hardware	CPU: Intel(R) Core(TM) i5-10210U CPU @ 1.60GHz or higher GPU: *Optional RAM: 8GB or bigger Hard disk: 1GB or bigger
OS / Software	OS: Windows 10 Software: Python3.7, Docker (for Neo4j library & data)
Packages	We have included all packages needed in the requirements.txt, please use 'pip install' command to install them.

4.2 Yolo Setup and Configuration

Minor setup and configuration are required for YOLOv5. The official YOLOv5 codes and our self-trained weights are included in the project and integrated. The backend script (reqHandler.py) will automatically initialize the model when it is executed and call the model when object detection is needed.

4.3 Neo4J Setup and Configuration

We recommend users to use docker-based installation when they have not configured "neo4j" on their PCs. The docker-based method would be more convenient, simply because installing a docker is easier than installing a "neo4j" from scratch (we have configured all the things in advanced, which means you do not have to import data and create a database). If users have already installed "neo4j" on their PCs, then they can directly refer to "PC Installation Part".

4.3.1 Docker Installation Part

4.3.1.1 Docker Installation

No	Steps
1	Go to Docker[1] official website to install a Windows version of Docker. For the backend. Use WSL-2.
2	By default, the docker will be installed on your C: drive. If you want to move the data to other drive (like D:), you can refer to this Windows 10 How to change Docker's Image Directory[2]
3	Open the 'cmd' console and type 'docker' to check whether you have successfully installed it.

4.3.1.2 Download and load the Docker Image

We have shared the docker image in GDrive[3] (snapyummy.tar)

1. Use **`docker load < snapyummy.tar`** to load the local images:

```
b12172aed5b1: Loading layer[===========================>]  141.8MB/141.8MB

Loaded image: snapyummy:final
```

4.3.1.3 Check whether the load is successful by docker image ls

```
REPOSITORY           TAG               IMAGE ID       CREATED          SIZE
snapyummy            final            0ce5b4f1fc9c   2 minutes ago    862MB
```

4.3.1.4 Download and unzip the Data Volume

Download the data volume from GDrive (neo4j.zip), unzip the file into **`/home`** directory (you can create it on any directory, e.g `D:/nus/home/neo4j`)

4.3.1.5 Create Container

1. Create a docker container by

docker run

-d --name snapyummy

-p 7474:7474

-p 7687:7687

-v D:/NUS/project/home/neo4j/data:/data

-v D:/NUS/project/home/neo4j/logs:/logs

[1] Docker - https://docs.docker.com/desktop/windows/install/
[2] Windows 10 How to change Docker's Image Directory - https://www.jianshu.com/p/e79f4c938000
[3] GDrive - https://drive.google.com/drive/folders/14pni4MBB_xiyYXP45kJB1JNwOrKtDCa2?usp=sharing

-v D:/NUS/project/home/neo4j/conf:/var/lib/neo4j/conf

-v D:/NUS/project/home/neo4j/import:/var/lib/neo4j/import

--env NEO4J_AUTH=neo4j/password snapyummy:final

Some explanations:

`-v` means the docker container will create a share folder and mount it to your local file directory. (e.g local directory **`D:/NUS/project/home/neo4j/data`** is linked to docker container directory **`/data`**)

You can simply copy the following command lines:

```
docker run -d --name snapyummy -p 7474:7474 -p 7687:7687 -v
D:/NUS/project/home/neo4j/data:/data -v D:/NUS/project/home/neo4j/logs:/logs -v
D:/NUS/project/home/neo4j/conf:/var/lib/neo4j/conf -v
D:/NUS/project/home/neo4j/import:/var/lib/neo4j/import --env NEO4J_AUTH=neo4j/password
snapyummy:final
```

Use **`docker ps -a`** to check whether you have created the container.

```
CONTAINER ID   IMAGE COMMAND CREATED  STATUS    PORTS NAMES

977b09c76544   snapyummy:final      "/sbin/tini -g -- /d…"   10 seconds ago   Up 9 seconds
0.0.0.0:7474->7474/tcp, :::7474->7474/tcp, 7473/tcp, 0.0.0.0:7687->7687/tcp, :::7687->7687/tcp   snapyummy
```

By default, the docker will automatically start after the docker run command, if not, try to use **docker start (container ID)**

If everything goes smoothly, you can go to **`localhost:7474`** to check, the database name is **snapyummy**; the username is **neo4j**; and the password is **password**

4.3.2 PC Installation Part

4.3.2.1 Download Files

Please download the original csv files at GDrive neo4j_import_csv[4]

4.3.2.2 Batch Import

- The neo4j-admin script is an official neo4j script that can import data in batches. It is stored in the bin folder of the neo4j installation directory (windows and linux).

[4] GDrive neo4j_import_csv - https://drive.google.com/drive/folders/1bKofDDYTlp8p1yRtn5hioRihpPKf1Z_l

- To use this script, some csv files need to be created to meet the import conditions (such as certain mandatory fields). Please refer to this URL: https://neo4j.com/docs/operations-manual/current/tutorial/neo4j-admin-import/

- Use this script to import millions of data with one command as following (Please remember to adjust the paths)

Windows

```
neo4j-admin.bat import

--database=kg-test3

--nodes=F:\share_file\recipe_header.csv,F:\share_file\recipe.csv

--nodes=F:\share_file\ingredients.csv

--nodes=F:\share_file\meal_type.csv

--nodes=F:\share_file\main_ingredients.csv

--nodes=F:\share_file\cuisine_type.csv

--
relationships=F:\share_file\ingredients_recipe_header.csv,F:\share_file\in
gredients_recipe.csv

--relationships=F:\share_file\sub_main_ingre.csv

--
relationships=F:\share_file\recipe_meal_type_header.csv,F:\share_file\reci
pe_meal_type.csv

--
relationships=F:\share_file\recipe_cuisine_type_header.csv,F:\share_file\r
ecipe_cuisine_type.csv

--multiline-fields=true

--skip-bad-relationships
```

Linux

```
neo4j-admin import

--database=kg-test3

--nodes=F:\share_file\recipe_header.csv,F:\share_file\recipe.csv

--nodes=F:\share_file\ingredients.csv

--nodes=F:\share_file\meal_type.csv

--nodes=F:\share_file\main_ingredients.csv

--nodes=F:\share_file\cuisine_type.csv
```

```
--
relationships=F:\share_file\ingredients_recipe_header.csv,F:\share_file\in
gredients_recipe.csv

--relationships=F:\share_file\sub_main_ingre.csv

--
relationships=F:\share_file\recipe_meal_type_header.csv,F:\share_file\reci
pe_meal_type.csv

--
relationships=F:\share_file\recipe_cuisine_type_header.csv,F:\share_file\r
ecipe_cuisine_type.csv

--multiline-fields=true

--skip-bad-relationships
```

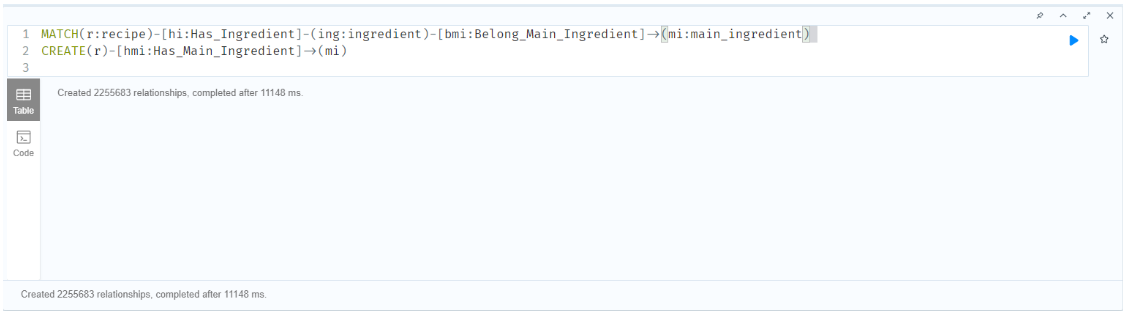
4.3.2.3 *Do the inferring*

Please refer to the session 5.2.4 Knowledge discovery using rule-based reasoning in our report

4.3.2.3.1 Main Ingredient:

① run cql below

```
MATCH(r:recipe)-[hi:Has_Ingredient]-(ing:ingredient)-
[bmi:Belong_Main_Ingredient]->(mi:main_ingredient)

CREATE(r)-[hmi:Has_Main_Ingredient]->(mi)
```



```
MATCH(r:recipe)-[hi:Has_Ingredient]-(ing:ingredient)-[bmi:Belong_Main_Ingredient]→(mi:main_ingredient)
CREATE(r)-[hmi:Has_Main_Ingredient]→(mi)
```

```
Created 2255683 relationships, completed after 11148 ms.
```

```
Created 2255683 relationships, completed after 11148 ms.
```

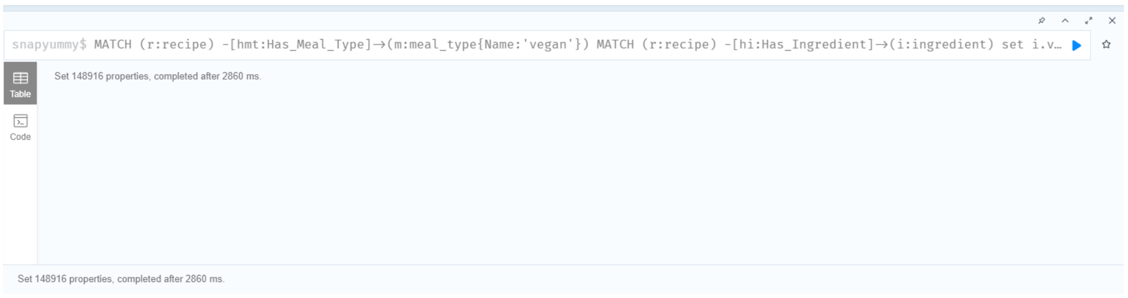
② run cql below (delete duplicate relationship)

```
MATCH (a:recipe)-[r:Has_Main_Ingredient]->(b:main_ingredient)

WITH a, type(r) as type, collect(r) as rels, b

WHERE size(rels) > 1

UNWIND tail(rels) as rel

DELETE rel
```

3.2 Vegan Part:

① run cql below

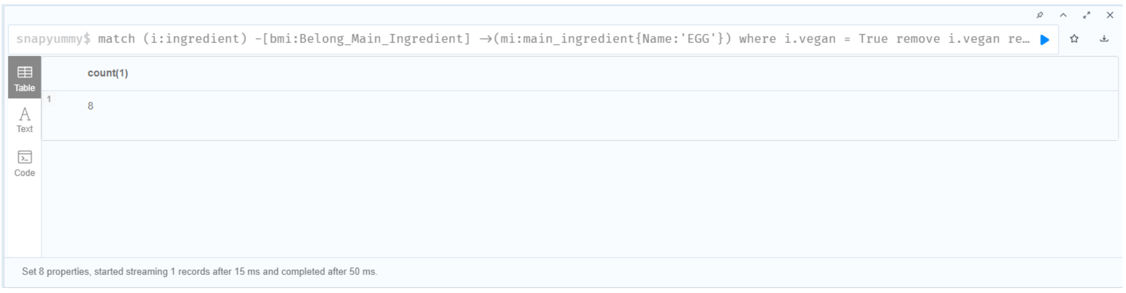
```
MATCH (r:recipe) -[hi:Has_Ingredient]->(i:ingredient)

set i.vegan = True
```



```
snapyummy$ MATCH (r:recipe) -[hmt:Has_Meal_Type]→(m:meal_type{Name:'vegan'}) MATCH (r:recipe) -[hi:Has_Ingredient]→(i:ingredient) set i.v…

Set 148916 properties, completed after 2860 ms.

Set 148916 properties, completed after 2860 ms.
```

② run cql below and replace the name 'EGG' with names in this list [MILK ,FISH, SEAFOOD, SHRIMP, CHICKEN,BEEF,LAMB,PORK,DUCK， MEATBALL， CRABMEAT， SHELLS， HAM， SAUSAGE， BUTTER] and continously run the cql

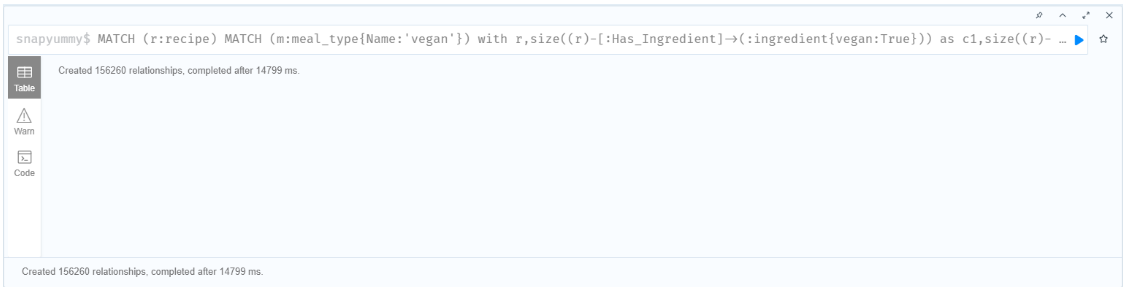
```
match (i:ingredient) -[bmi:Belong_Main_Ingredient]
->(mi:main_ingredient{Name:'EGG'})

where i.vegan = True

remove i.vegan

return count(1)
```



```
snapyummy$ match (i:ingredient) -[bmi:Belong_Main_Ingredient] →(mi:main_ingredient{Name:'EGG'}) where i.vegan = True remove i.vegan re…

    count(1)
1   8

Set 8 properties, started streaming 1 records after 15 ms and completed after 50 ms.
```

③ build new relations

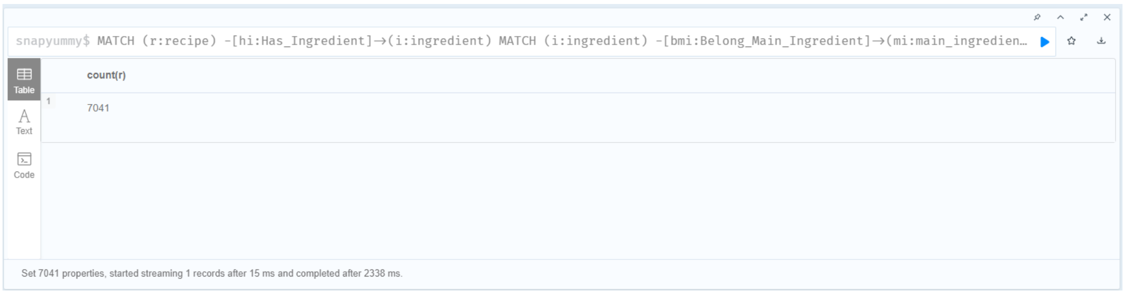
```
MATCH (m:meal_type{Name:'vegan'})

with r,size((r)-[:Has_Ingredient]->(:ingredient{vegan:True})) as
c1,size((r)-  [:Has_Ingredient]->(:ingredient)) as c2,m

where c1=c2

merge (r)-[himt:Has_Inferred_Meal_Type]->(m)
```



```
snapyummy$ MATCH (r:recipe) MATCH (m:meal_type{Name:'vegan'}) with r,size((r)-[:Has_Ingredient]→(:ingredient{vegan:True})) as c1,size((r)- …

Created 156260 relationships, completed after 14799 ms.

Created 156260 relationships, completed after 14799 ms.
```

3.3 Halal Part:

```
MATCH (r:recipe) -[hi:Has_Ingredient]->(i:ingredient)

MATCH                              (i:ingredient)                              -
[bmi:Belong_Main_Ingredient]->(mi:main_ingredient{Name:'PORK'})

set r.halal = False

return count(r)
```



```
snapyummy$ MATCH (r:recipe) -[hi:Has_Ingredient]→(i:ingredient) MATCH (i:ingredient) -[bmi:Belong_Main_Ingredient]→(mi:main_ingredien…
```

```
  count(r)
1 7041
```

Set 7041 properties, started streaming 1 records after 15 ms and completed after 2338 ms.

4.4 Telegram Setup and Configuration

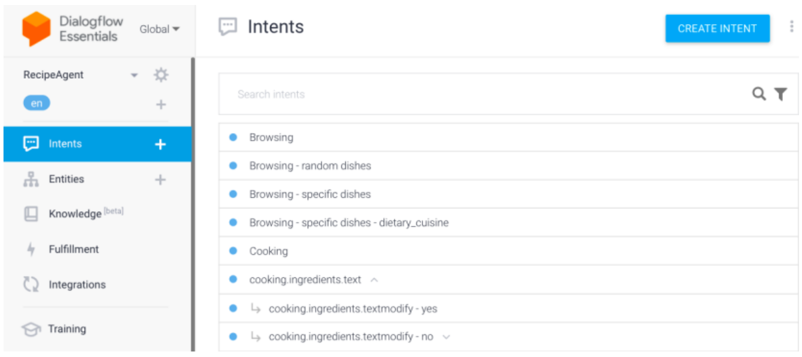
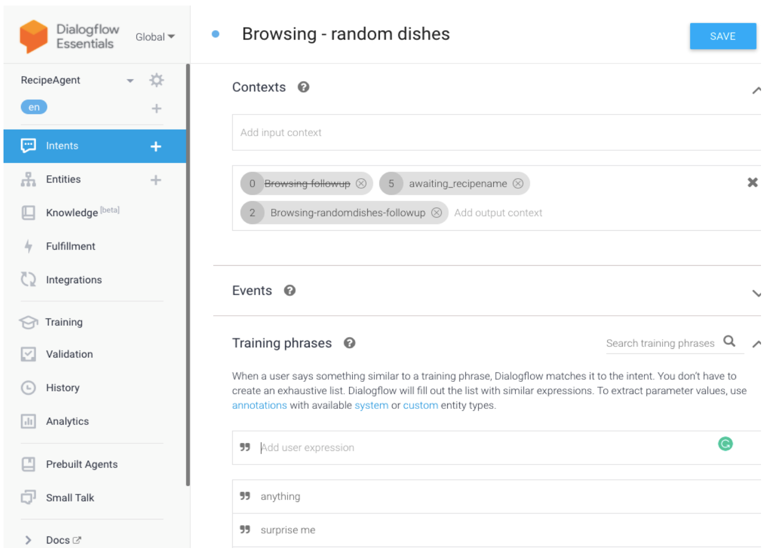
4.4.1 Generate Telegram Token

Telegram token or key is managed by *BotFather,* the token is required to connect between backend server and telegram. Below are the steps to obtain the telegram token.

No	Steps
1	Go to your Telegram account, and type BotFather in the search bar.
2	Click command /newbot to create a new bot
3	Type a name and username for your bot. If the username is already taken, BotFather will prompt you to change to another available username
4	After successfully choose a username, BotFather will provide the Telegram token along with other information.
5	Save the token and input into the backend script (refer to 4.6)
6	In the BotFather chat window, you may also click /setcommands to change the list of commands

4.5 Google Dialogflow Setup and Integration

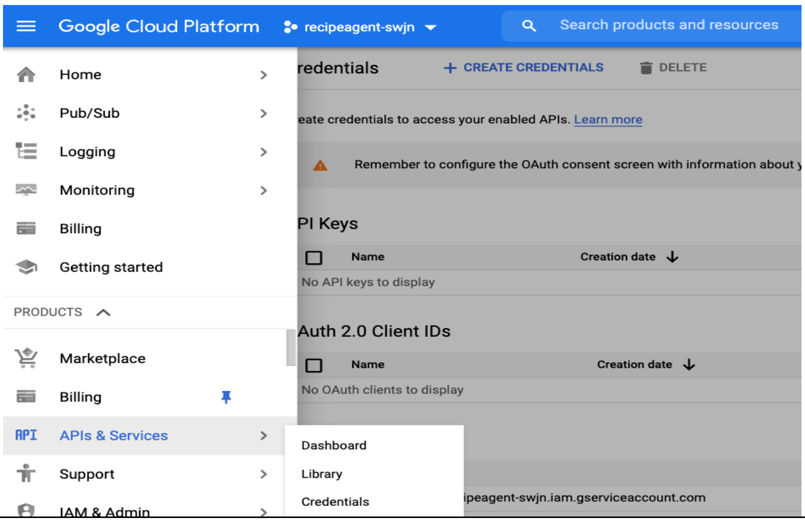
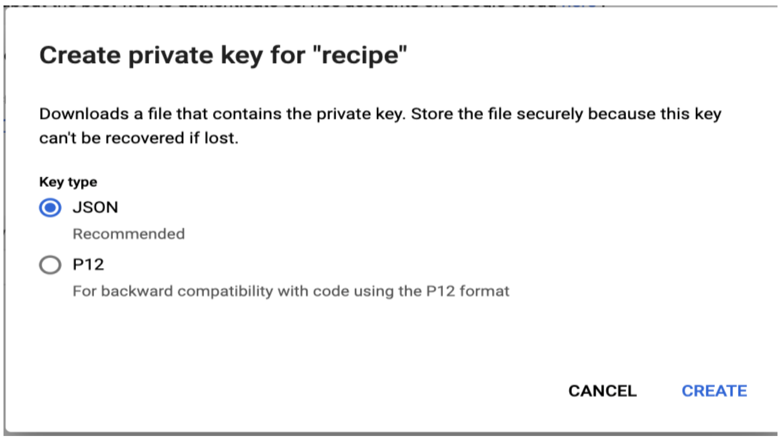
4.5.1 Build a Dialogflow Agent

No	Steps
1	Go to Dialogflow[5] and click Create new agent in the left sidebar menu
2	Choose a name for your virtual agent, set the default language, and click on Create
3	Click Intents in the left sidebar menu. Input the name of the Intents and click Create Intent  Dialogflow Essentials Global RecipeAgent en Intents Entities Knowledge [beta] Fulfillment Integrations Training Intents CREATE INTENT Search intents Browsing Browsing - random dishes Browsing - specific dishes Browsing - specific dishes - dietary_cuisine Cooking cooking.ingredients.text cooking.ingredients.textmodify - yes cooking.ingredients.textmodify - no
4	Input the name of the Intent, Contexts, Training Phrases, Action and Parameters and Responses according to the use case. You may refer to Project Report for more details and examples.  Dialogflow Essentials Global RecipeAgent en Intents Entities Knowledge [beta] Fulfillment Integrations Training Validation History Analytics Prebuilt Agents Small Talk Docs SAVE Contexts Add input context Add output context Events Training phrases Search training phrases Add user expression Browsing - random dishes 0 ~~Browsing-followup~~ 5 awaiting_recipename 2 Browsing-randomdishes-followup When a user says something similar to a training phrase, Dialogflow matches it to the intent. You don't have to create an exhaustive list. Dialogflow will fill out the list with similar expressions. To extract parameter values, use annotations with available system or custom entity types. anything surprise me
5	Alternatively, you may download the RecipeAgent (DialogFlow) from Github SystemCode/DialogFlow/RecipeAgent [6]

[5] DialogFlow - https://dialogflow.cloud.google.com/
[6] Project Github - https://github.com/SCNUJackyChen/IRS-PM-2021-11-07-IS03FT-GRP8-SnapYummy

4.5.2 Generate Google Dialogflow API Key

For backend server to reach Google Dialogflow.

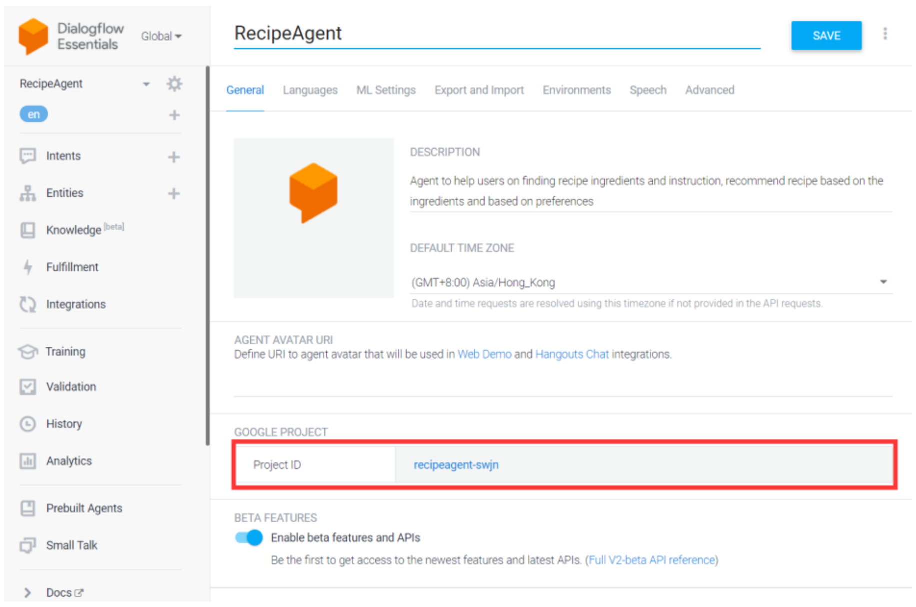
No	Steps
1	Go to Google Cloud Platform[7], click APIs & Services and then click Credentials in the left sidebar menu  Google Cloud Platform recipeagent-swjn Search products and resources Home Pub/Sub Logging Monitoring Billing Getting started PRODUCTS Marketplace Billing APIs & Services Support IAM & Admin Dashboard Library Credentials redentials CREATE CREDENTIALS DELETE PI Keys Name Creation date Auth 2.0 Client IDs Name Creation date eate credentials to access your enabled APIs. Learn more Remember to configure the OAuth consent screen with information about y No API keys to display No OAuth clients to display ipeagent-swjn.iam.gserviceaccount.com
2	In the Credentials page, click Create Credentials and choose Service Account.
3	Input the service account name and description. Service account ID will be auto populated. Then click Create and Continue.
4	Grant the service account access to project. Select a role for this service account, example Owner or Editor. Then click Done
5	Under Actions, click Manage keys to generate API key for the created service account. Choose key type JSON.  Create private key for "recipe" Downloads a file that contains the private key. Store the file securely because this key can't be recovered if lost. Key type [✓ JSON] Recommended P12 For backward compatibility with code using the P12 format
6	Save the JSON API key and input into backend server

[7] Google Cloud Platform - https://console.cloud.google.com/

4.6 Backend Server Setup and Configuration

To setup the backend server, you need:

1. Use 'git clone' command to download the project from the following URL: https://github.com/SCNUJackyChen/What2Cook
2. Open 'reqHandler.py' in the root directory of the project with any IDE.
3. At line 15, set '**telegram_botTOKEN**' to be the token you get in 4.4.1.
4. At line 14, set '**df_agentID**' to be the name of the Dialog agent you created in 4.5.1. You can find its name in the setting page:



RecipeAgent

DESCRIPTION
Agent to help users on finding recipe ingredients and instruction, recommend recipe based on the ingredients and based on preferences

DEFAULT TIME ZONE
(GMT+8:00) Asia/Hong_Kong
Date and time requests are resolved using this timezone if not provided in the API requests.

AGENT AVATAR URI
Define URI to agent avatar that will be used in Web Demo and Hangouts Chat integrations.

GOOGLE PROJECT
Project ID recipeagent-swjn

BETA FEATURES
Enable beta features and APIs
Be the first to get access to the newest features and latest APIs. (Full V2-beta API reference)

5. At line 16, set '**os.environ["GOOGLE_APPLICATION_CREDENTIALS"]**' to be the absolute path of the JSON API key you saved in 4.5.2.
6. Run 'reqHandler.py', then you can now chat with the Telegram bot (ref to 5).

5 User Guide

5.1 Use Case Overview

SnapYummy provides 2 modes, cooking and browsing. In cooking mode, SnapYummy provides the recipe to cook according to users input, ie ingredients, cuisine preference and dietary preferences. For ingredients input, users may choose to input via image or via text. When users input image, SnapYummy will use its object detection mechanism powered by Yolov5 to detect the objects. This feature enables users to just snap a photo of their refrigerator easily and inform SnapYummy on the ingredients that users have. When users are unable to snap a photo, they may also just type-in their ingredients in text form.

The second mode that users may interact with SnapYummy is browsing mode. In this mode, users may browse recipe in two ways, leisure browsing and specific browsing. Leisure browsing is for users who do not have any idea on cooking and just like to browse on available recipe for getting ideas. While specific browsing is for users who are performing leisure browsing, but with specific cuisines and dietary preferences. This is useful for users who are going to host a party for family or friends with different preferences.

Following are the specific cuisines list and dietary list provided by SnapYummy.

5.1.1 Cuisines List

Cuisine			
Nigerian	Swedish	Australian	Filipino
Lebanese	Pennsylvania Dutch	South African	Turkish
Portuguese	Ethiopian	Greek	Tex Mex
Belgian	Cajun	Southern_Us	Swiss
Czech	Cajun_Creole	Caribbean	Austrian
Finnish	European	Italian	Dutch
South American	Hawaiian	French	Egyptian
Costa Rican	Canadian	German	Danish
Moroccan	Welsh	Vietnamese	Japanese
New Zealand	Georgian	Indonesian	British
Thai	Russian	Hungarian	Chinese
Asian	African	Indian	Pakistani
Irish	Malaysian	Norwegian	Iraqi
Puerto Rican	Creole	Korean	Jamaican
Native American	Mexican	Brazilian	Cuban
Palestinian	Spanish	Scottish	Christmas

5.1.2 Dietary List

Five entries are created for dietary entity including vegetarian, halal, eggetarian, fruitarian and non-vegetarians. Each entry is defined with synonyms that will map to reference values.

Dietary	Synonyms
vegetarian	vegetarian, veggies, veg, vegan, vegetables only, veggies only, vegg, herbivorous, gluten-free, plant-eating, veggie-lovers
halal	halal, no pork, no lard
eggetarian	Eggetarian, vegetarian
fruitarian	fruitarian, fruits, fruit
non-vegetarians	non-vegetarians, non-veg, non-vegetarian, meat-lovers

5.2 Cooking

5.2.1 Select cooking mode

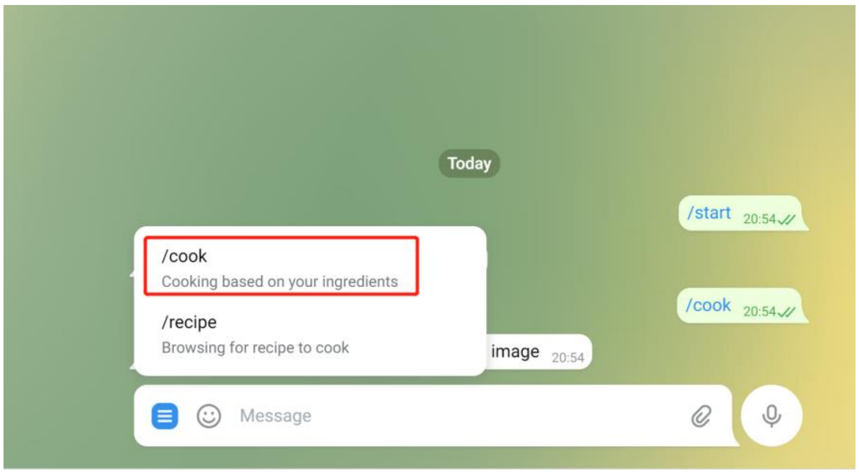
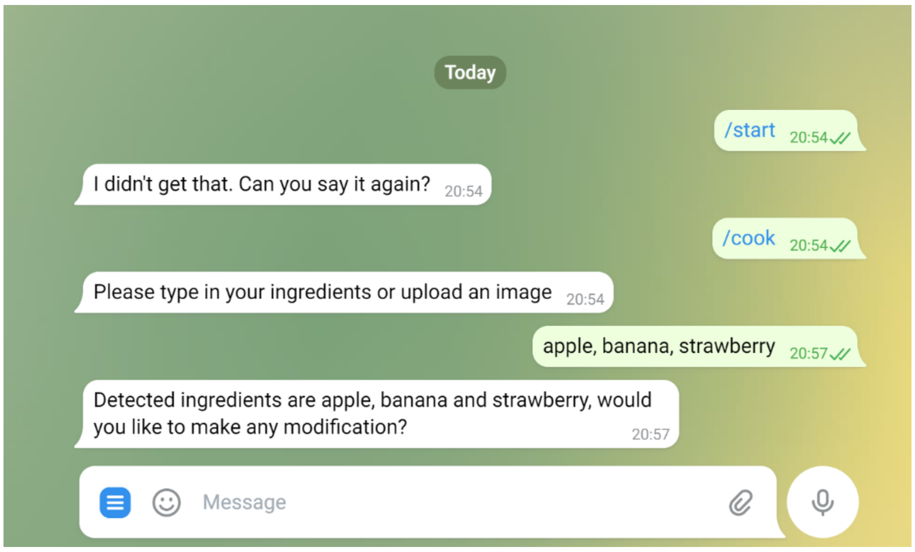

Today
/start 20:54
/cook
Cooking based on your ingredients
/recipe
Browsing for recipe to cook
/cook 20:54
image 20:54
Message

Figure 2 Cooking Mode

5.2.2 Type in the ingredients or upload an image



Today

/start 20:54

I didn't get that. Can you say it again? 20:54

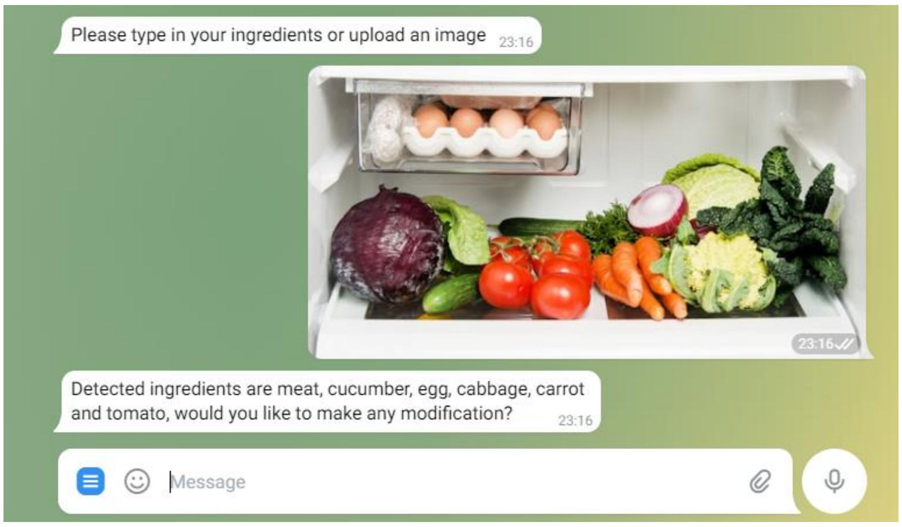
/cook 20:54

Please type in your ingredients or upload an image 20:54

apple, banana, strawberry 20:57

Detected ingredients are apple, banana and strawberry, would you like to make any modification? 20:57

Figure 3 Type in Ingredients in Text Form



Please type in your ingredients or upload an image 23:16

23:16

Detected ingredients are meat, cucumber, egg, cabbage, carrot and tomato, would you like to make any modification? 23:16

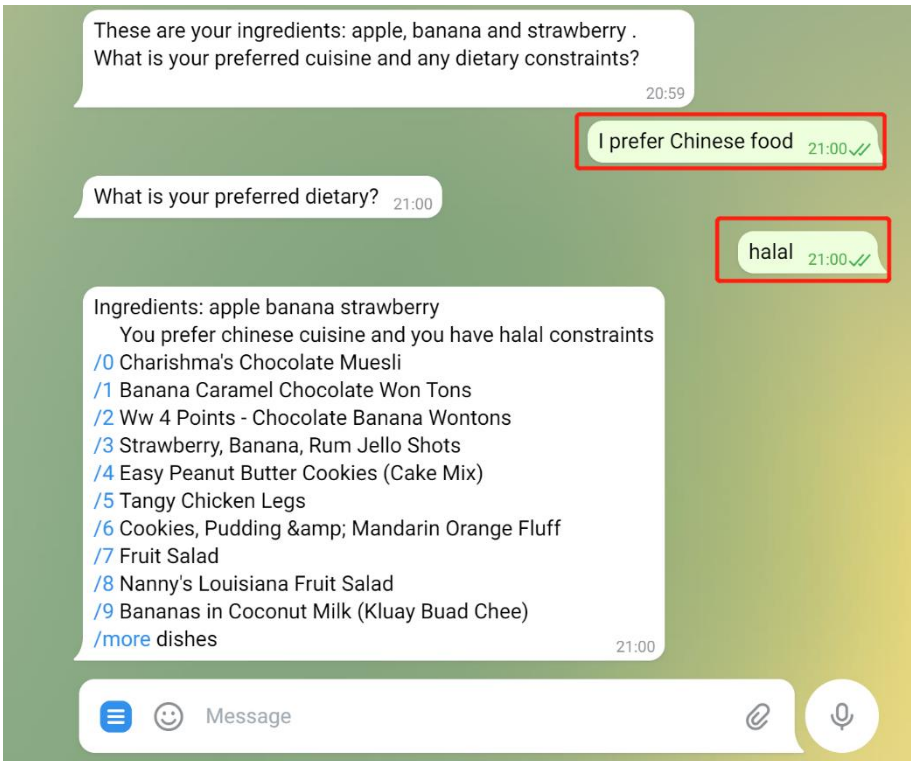
Figure 4 Upload Photo for Food Detection

5.2.3 Modify input ingredient

If you are not satisfied with the detected results, you can choose to modify them. Just enter "yes" and input them.

5.2.4 Input your preference

The robot will enquiry your dining preference for cuisine and dietary, if you do not have the preference, you can just tell it "No cuisine" or "No dietary".


These are your ingredients: apple, banana and strawberry .
What is your preferred cuisine and any dietary constraints?
20:59

I prefer Chinese food 21:00

What is your preferred dietary? 21:00

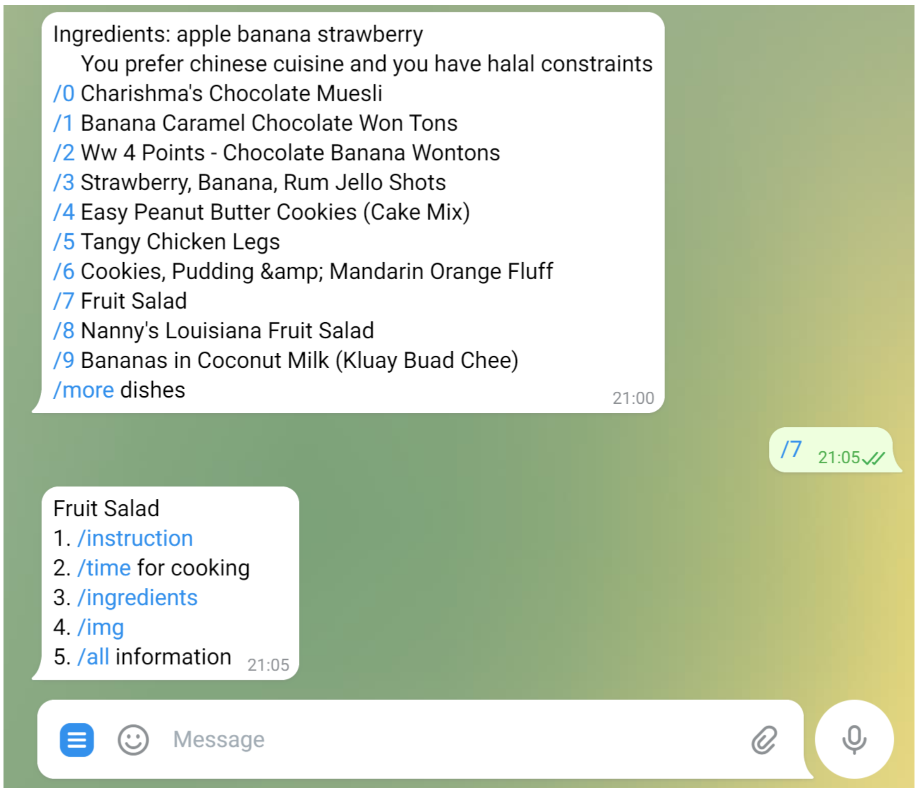
halal 21:00

Ingredients: apple banana strawberry
You prefer chinese cuisine and you have halal constraints
/0 Charishma's Chocolate Muesli
/1 Banana Caramel Chocolate Won Tons
/2 Ww 4 Points - Chocolate Banana Wontons
/3 Strawberry, Banana, Rum Jello Shots
/4 Easy Peanut Butter Cookies (Cake Mix)
/5 Tangy Chicken Legs
/6 Cookies, Pudding &amp; Mandarin Orange Fluff
/7 Fruit Salad
/8 Nanny's Louisiana Fruit Salad
/9 Bananas in Coconut Milk (Kluay Buad Chee)
/more dishes
21:00

Figure 5 Input Preferences (Dietary and Cuisine)

5.2.5 Choosing your favourite dish!

By default, the robot will recommend you 10 dishes based on your ingredients and preferences. You can click the number in blue highlight to see details or click “/more” to look at more dishes.

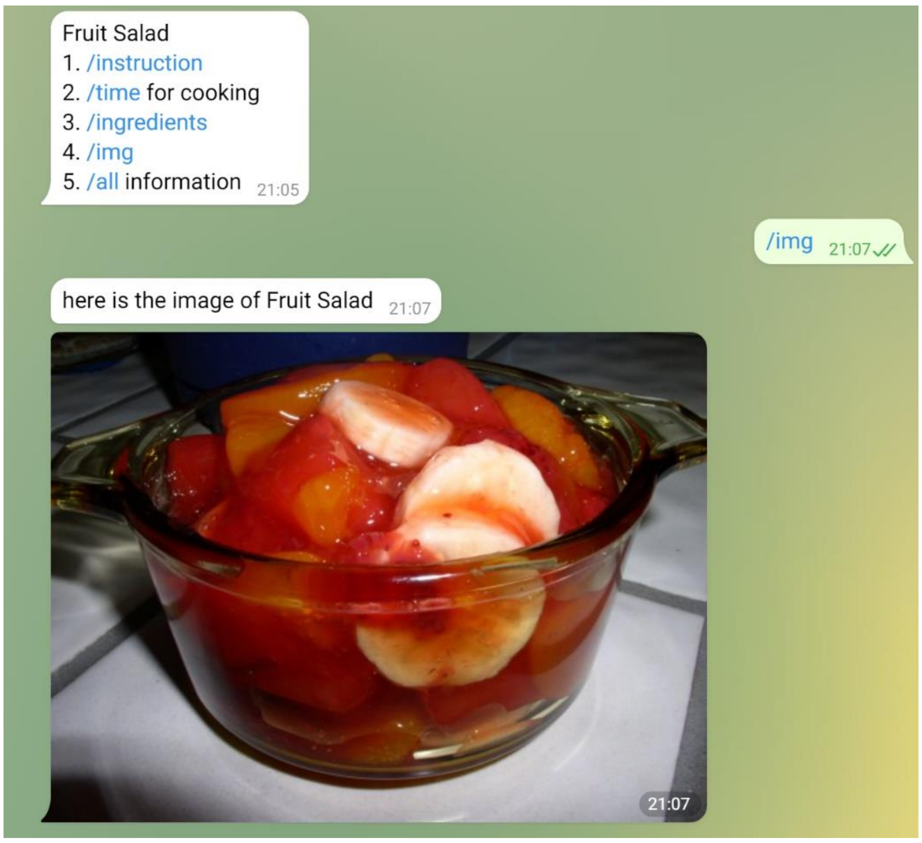

Ingredients: apple banana strawberry
You prefer chinese cuisine and you have halal constraints
/0 Charishma's Chocolate Muesli
/1 Banana Caramel Chocolate Won Tons
/2 Ww 4 Points - Chocolate Banana Wontons
/3 Strawberry, Banana, Rum Jello Shots
/4 Easy Peanut Butter Cookies (Cake Mix)
/5 Tangy Chicken Legs
/6 Cookies, Pudding &amp; Mandarin Orange Fluff
/7 Fruit Salad
/8 Nanny's Louisiana Fruit Salad
/9 Bananas in Coconut Milk (Kluay Buad Chee)
/more dishes
21:00

/7 21:05

Fruit Salad
1. /instruction
2. /time for cooking
3. /ingredients
4. /img
5. /all information 21:05

Figure 6 Selection of Recipe

5.2.6 Check details for a dish



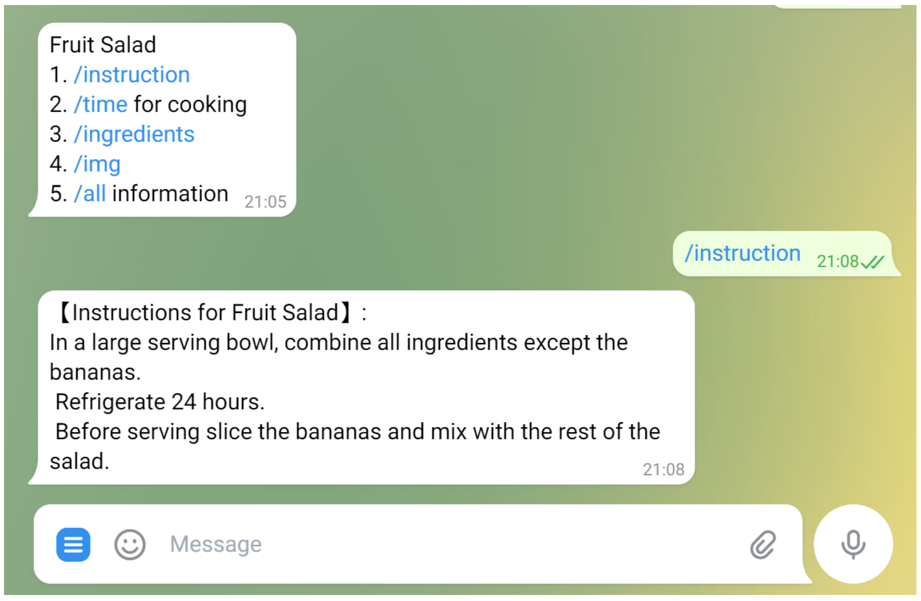
Fruit Salad
1. /instruction
2. /time for cooking
3. /ingredients
4. /img
5. /all information 21:05

/img 21:07

here is the image of Fruit Salad 21:07

21:07

Figure 7 Details of Recipe



Fruit Salad
1. /instruction
2. /time for cooking
3. /ingredients
4. /img
5. /all information 21:05

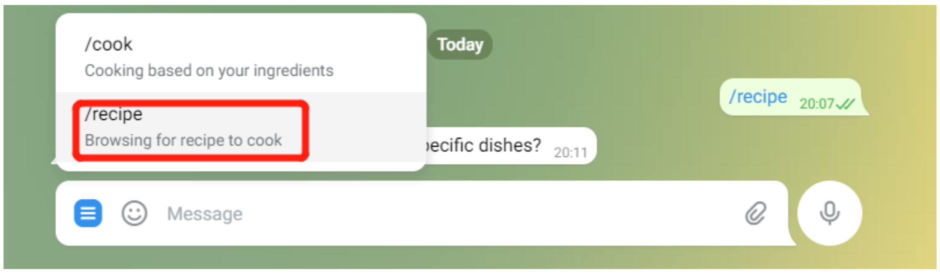
/instruction 21:08

【Instructions for Fruit Salad】:
In a large serving bowl, combine all ingredients except the bananas.
 Refrigerate 24 hours.
 Before serving slice the bananas and mix with the rest of the salad.
21:08

Figure 8 Recipe's Instructions

5.3 Browsing

If you just want to have a look at our recipes or grab some new ideas for cooking, browsing mode may satisfy your demand.



/cook

Cooking based on your ingredients

/recipe

Browsing for recipe to cook

Today

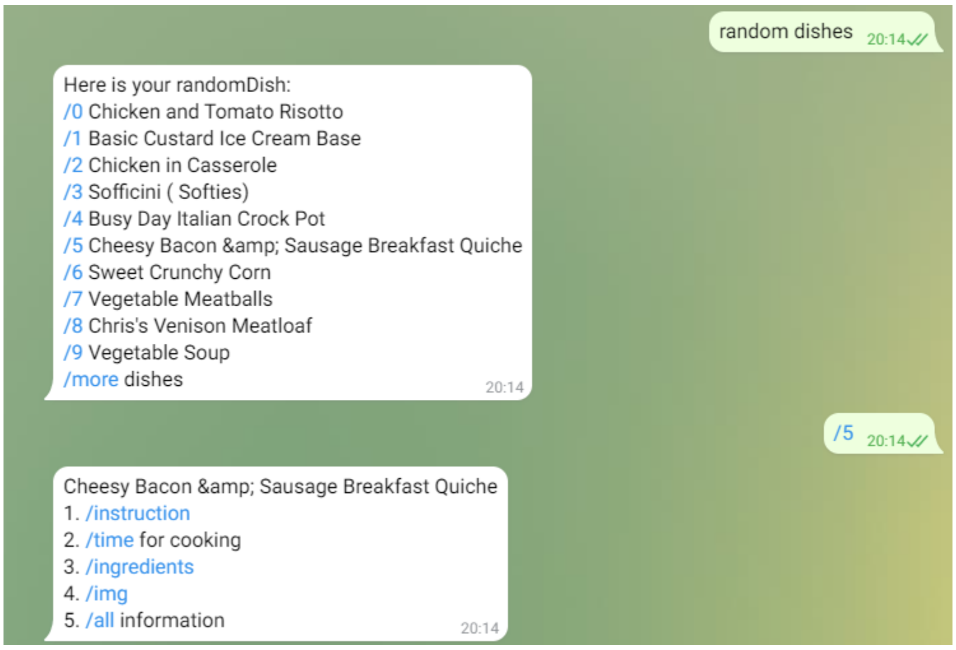
/recipe 20:07

ecific dishes? 20:11

Message

Figure 9 Browsing Mode

5.3.1 Random Dishes



random dishes 20:14

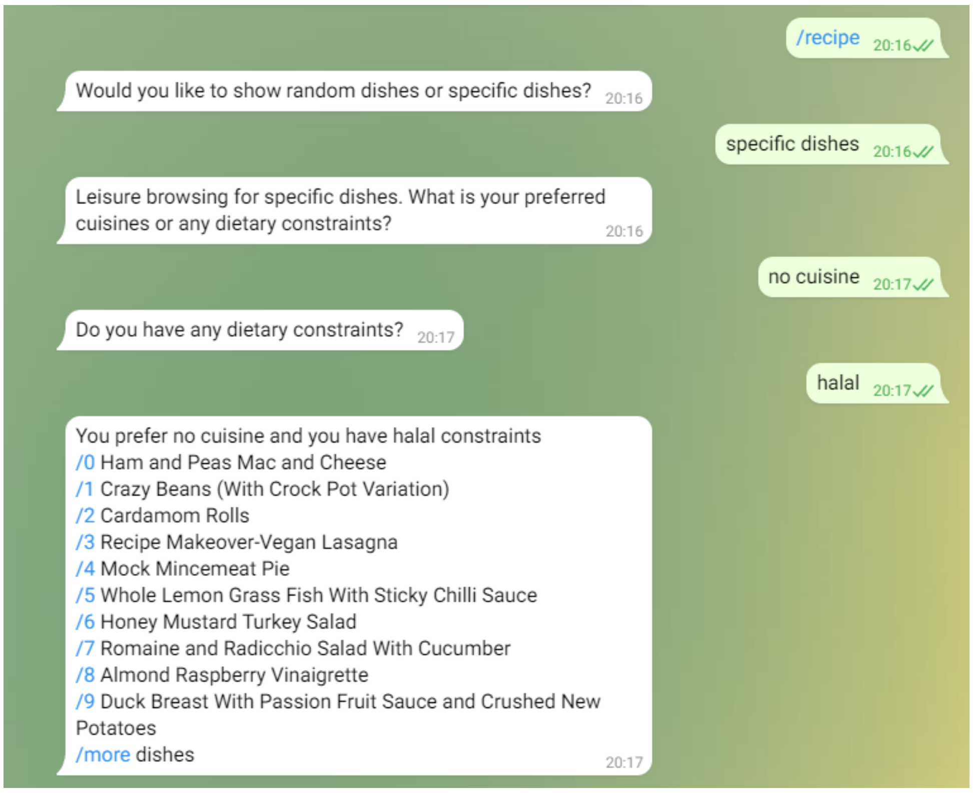
Here is your randomDish:
/0 Chicken and Tomato Risotto
/1 Basic Custard Ice Cream Base
/2 Chicken in Casserole
/3 Sofficini (Softies)
/4 Busy Day Italian Crock Pot
/5 Cheesy Bacon &amp; Sausage Breakfast Quiche
/6 Sweet Crunchy Corn
/7 Vegetable Meatballs
/8 Chris's Venison Meatloaf
/9 Vegetable Soup
/more dishes
20:14

/5 20:14

Cheesy Bacon &amp; Sausage Breakfast Quiche
1. /instruction
2. /time for cooking
3. /ingredients
4. /img
5. /all information
20:14

Figure 10 Random Recipe (Browsing Mode)

5.3.2 Specific Dishes



/recipe 20:16

Would you like to show random dishes or specific dishes? 20:16

specific dishes 20:16

Leisure browsing for specific dishes. What is your preferred cuisines or any dietary constraints? 20:16

no cuisine 20:17

Do you have any dietary constraints? 20:17

halal 20:17

You prefer no cuisine and you have halal constraints
/0 Ham and Peas Mac and Cheese
/1 Crazy Beans (With Crock Pot Variation)
/2 Cardamom Rolls
/3 Recipe Makeover-Vegan Lasagna
/4 Mock Mincemeat Pie
/5 Whole Lemon Grass Fish With Sticky Chilli Sauce
/6 Honey Mustard Turkey Salad
/7 Romaine and Radicchio Salad With Cucumber
/8 Almond Raspberry Vinaigrette
/9 Duck Breast With Passion Fruit Sauce and Crushed New Potatoes
/more dishes 20:17

Figure 11 Specific Recipe (Browsing Mode)

For ISS Use Only		
Programme Name:	**Project No:**	**Learner Batch:**
Accepted/Rejected/KIV:		
Learners Assigned:		
Advisor Assigned: Contact: Mr. GU ZHAN / Lecturer & Consultant Telephone No.: 65-6516 8021 Email: zhan.gu@nus.edu.sg		