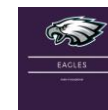


hello birdie

Final Presentation - PRI Project Voice Caddie

Team EagleS

Paris, Jule 17, 2020





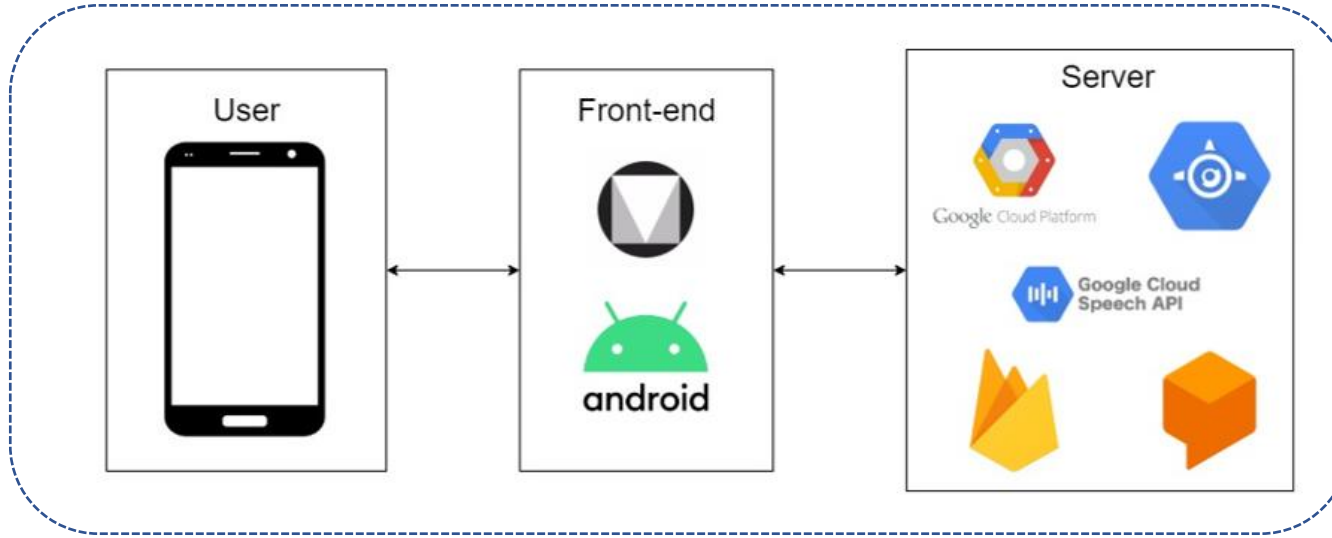
Project Background



As the one of the most popular mobile applications for golfers, Hello Birdie aims to strengthen the performance of the app through powering an AI - Voice Recognition function. With potential 90 Million golfers in the world, Hello Birdie wants to enhance the user experience to the next level. Therefore, Hello Birdie key founders have initiated the Voice Caddie project to achieve this trendy product innovation with two main objectives:

- Develop a conversational tool to provide pre-calculated strategic advice and voice-controlled input of score and shot sequences
- With a dedicated vocabulary set - the speech to text shall be trained to work with >90% of the time for 5 input languages

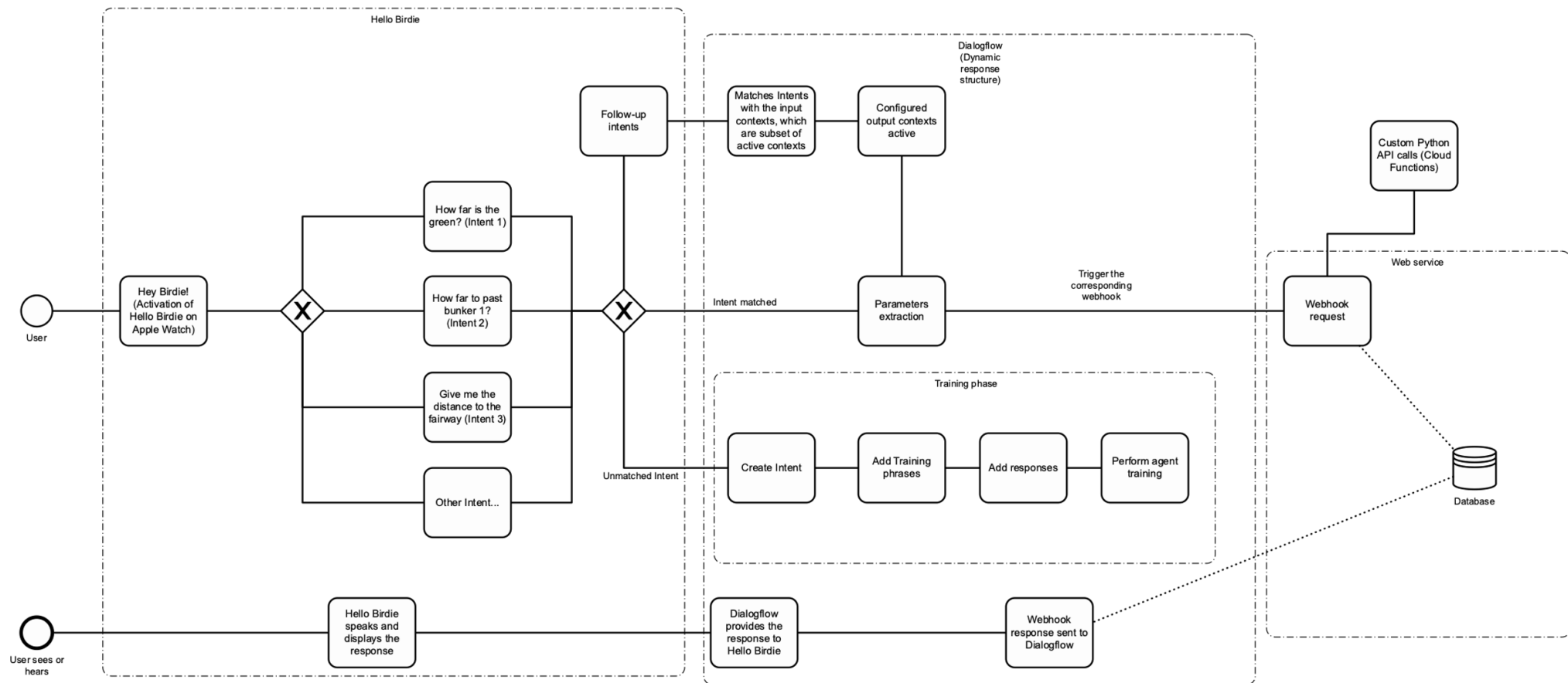
Overview and Tools



- ✓ Users can integrate with Hello Birdie through Voice Caddie. The Principle of Application is: Voice input and Voice output.
- ✓ Our project uses online communication platform SLACK and Git where members can communicate with sponsors, share the ideas, input and checking status of our project.
- ✓ Despite the impact of Covid 19, our communication still went smoothly and ensured the tracking frequently between project team and sponsor.



Application Structure



Firestore

Here are the configuration details of the Firestore service we have used.

```
In [1]: import pyrebase
config = {
    "apiKey": "AIzaSyDSwviBwGLJEA09j8xZ6sPnhmx8U9Wy0B8",
    "authDomain": "hb-epita-voice-recognition.firebaseio.com",
    "databaseURL": "https://hb-epita-golf-conversational-agent.firebaseio.com",
    "projectId": "hb-epita-voice-recognition",
    "storageBucket": "hb-epita-voice-recognition.appspot.com",
    "messagingSenderId": "845561190896",
    "appId": "1:845561190896:android:3e2adc1210b2e9982ae535",
    #"measurementId": "G-LEJ0DZJXWF"
}
```

These are available in the Firestore console of the project in a json

```
In [2]: import json_reader

config = json_reader.json_loader("hb-google-services.json")

print(config.values())

6
dict_values([{'project_number': '845561190896', 'firebase_url': 'https://hb-epita-voice-recognition.firebaseio.com',
'project_id': 'hb-epita-voice-recognition', 'storage_bucket': 'hb-epita-voice-recognition.appspot.com'}, {'client_info': {'mobilesdk_app_id': '1:845561190896:android:3e2adc1210b2e9982ae535', 'android_client_info': {'package_name': 'fr.epita.hellobirdie'}}, 'oauth_client': [{'client_id': '845561190896-8fo1a3kf12mldagkmrlblcitbk3otmm.apps.googleusercontent.com', 'client_type': 3}], 'api_key': [{'current_key': 'AIzaSyDSwviBwGLJEA09j8xZ6sPnhmx8U9Wy0B8'}],
'services': {'appinvite_service': {'other_platform_oauth_client': [{'client_id': '845561190896-i2vr4su3nt4gpkc5q1s97kg9bfffmod.apps.googleusercontent.com', 'client_type': 3}]}}], '1']])
```

The service is initialized with the configuration file using the Python wrapper for Firestore called “Pyrebase.” The authentication services are enabled using the auth() method. A user is then created using an email and a password. The user is then signed in to avail the service.



Firestore



Dialog Flow

Main components of DialogFlow:

1. Agents
2. Intents
3. Entities
4. Contexts
5. Fulfillment for integrations



Dialogflow

Step 1: Create an agent on Dialog flow

A virtual agent that handles conversations with the end-users.

Hello-Birdie-Caddie

CREATE

DEFAULT LANGUAGE

English — en

Primary language for your agent. Other languages can be added later.

DEFAULT TIME ZONE

(GMT+2:00) Europe/Kaliningrad

Date and time requests are resolved using this timezone.

GOOGLE PROJECT

Create a new Google project

Enables Cloud functions, Actions on Google and permissions management.

AGENT TYPE

Set as Mega Agent

Combine multiple Dialogflow agents (i.e. sub agents) into a single agent (i.e. mega agent).

Step 2: Create an intent for the agent

It is used to categorize an end-user’s conversation. A Dialogflow agent can have multiple intents.

Intents

CREATE INTENT

Search intents

compositeDistance

Default Fallback Intent

Default Welcome Intent

getGameplanInfo

UserAsksForDistanceInfo

UserAsksForShotAcquisition

UserAsksForShotAcquisition - getDetails

Step 3: Define and map entities

golf_course

SAVE

Define synonyms

Regep entity

Allow automated expansion

Fuzzy matching

GreenCenter	green center, pin, flag, hole, green centre
Teebox	tee box, teeing ground, teefox
water hazard	water hazard
rough	rough
cart path	cart path
Bunker 1	Bunker 1, bunker one, 1st bunker, first bunker
Fairway	fairway, fairway 1
Tee 1	Tee 1, tee
Bunker 2	Bunker 2, bunker 2, bunker two, second bunker, 2nd bunker
Bunker 3	Bunker 3, bunker three, Bunker three
Bunker 4	Bunker 4, bunker four, bunker 4, Bunker four
Bunker 5	Bunker 5, bunker five, bunker 5, Bunker five
Bunker 6	Bunker 6, bunker 6, Bunker six
Bunker 7	Bunker 7, bunker seven, Bunker seven

Step 4: Fulfilment responses:

By default, the agent responds to an intent with a static response.

Responses

DEFAULT

GOOGLE ASSISTANT

+

Text or SSML Response

1

The distance to the \$destination.original is (distance). You can use a (club) here.

2

You have to play (distance) to the \$destination.original. My suggestion is to use a (club) here.

3

You have to play (distance) to \$distanceintentAction the \$destination.original. You should use a (club).

4

In order to \$distanceintentAction the \$destination.original, you have to play (distance). I suggest, that you use a (club) here.

5

You are (distance) far to the \$destination.original. I recommend to use a (club) here.

6

Enter a text or SSML response variant

Step 5: Machine Learning settings.

Choosing the ML classification threshold: If the return value is less than the threshold, it will trigger the fallback intent.

HB-Voice-Caddie

SAVE

General

Languages

ML Settings

Export and Import

Environments

Speech

Advanced

Select the match mode that suits your agent best.

Use the Hybrid (Rule-based and ML) mode for agents with a small number of examples/templates in intents, especially the ones using composite entities.

Use ML only mode for agents with a large number of examples in intents, especially the ones using @sys.any

Hybrid (Rule-based and ML)

ML CLASSIFICATION THRESHOLD

Define the threshold value for the confidence score. If the returned value is less than the threshold value, then a fallback intent will be triggered, or if there is no fallback intents defined, no intent will be triggered.

0.72

AUTOMATIC SPELL CORRECTION

Allow ML to correct spelling of query during request processing.

AUTOMATIC TRAINING

Disable automatic re-training the agent after every agent modification.

Automatic training may allow UI responsiveness and is not recommended for large agents. You can kick off agent training manually through API or by clicking the "TRAIN" button below.

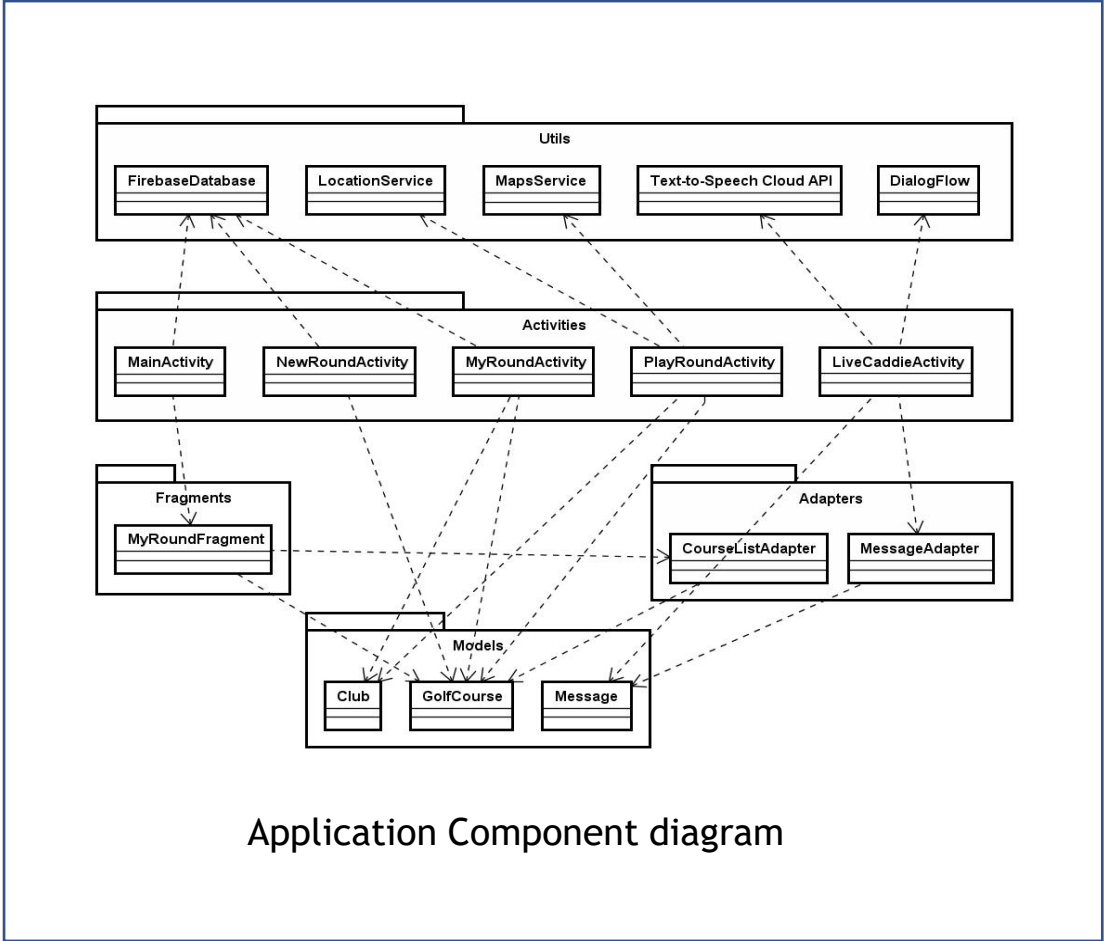
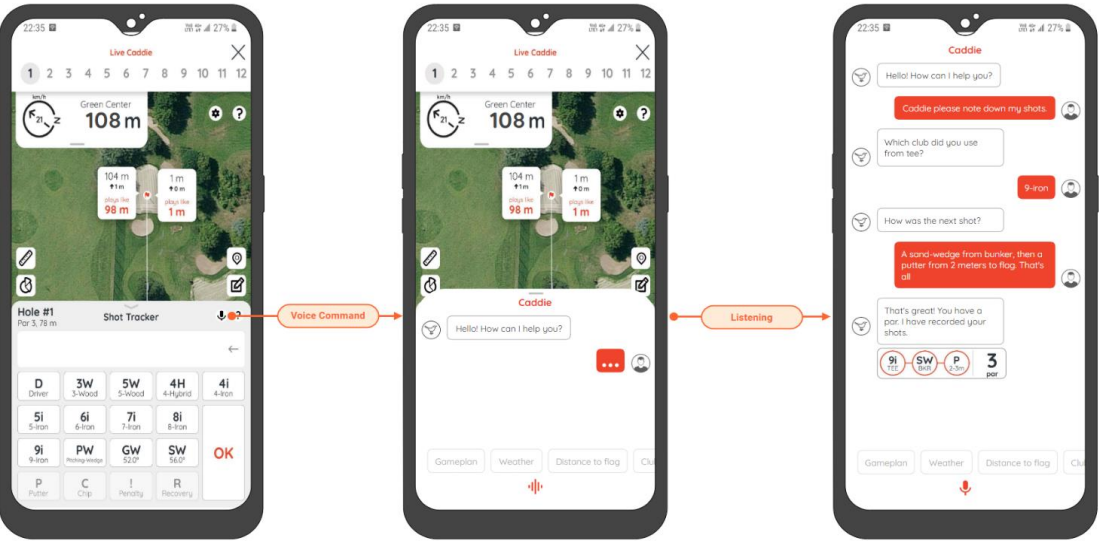
AGENT VALIDATION

Automatically validate the agent when agent training is performed.

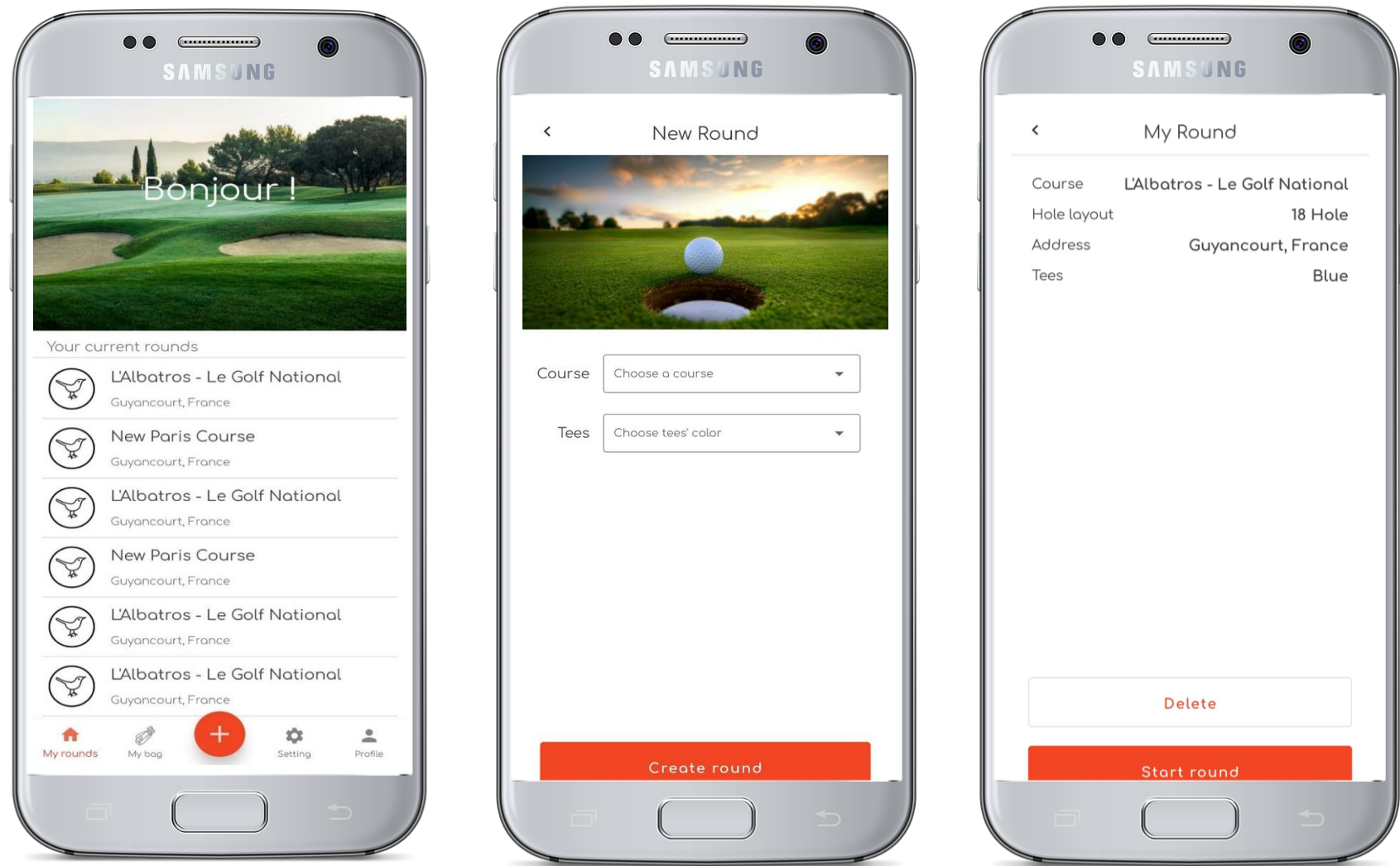


Front End

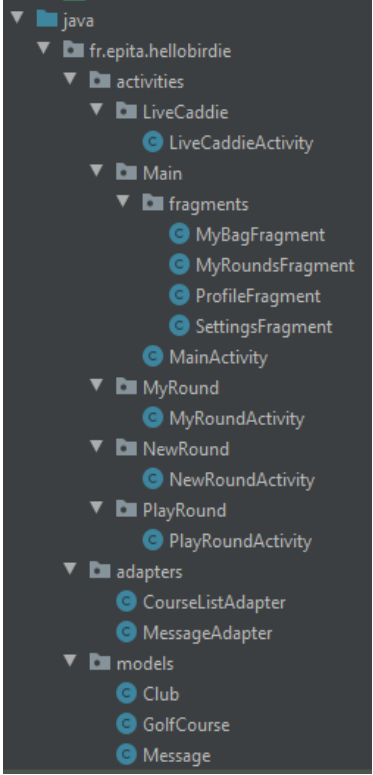
Front-end here is an Android Mobile application, including the interface for users to provide functions related to create new round, playing a round and talk with the Caddie.



Main screens and features



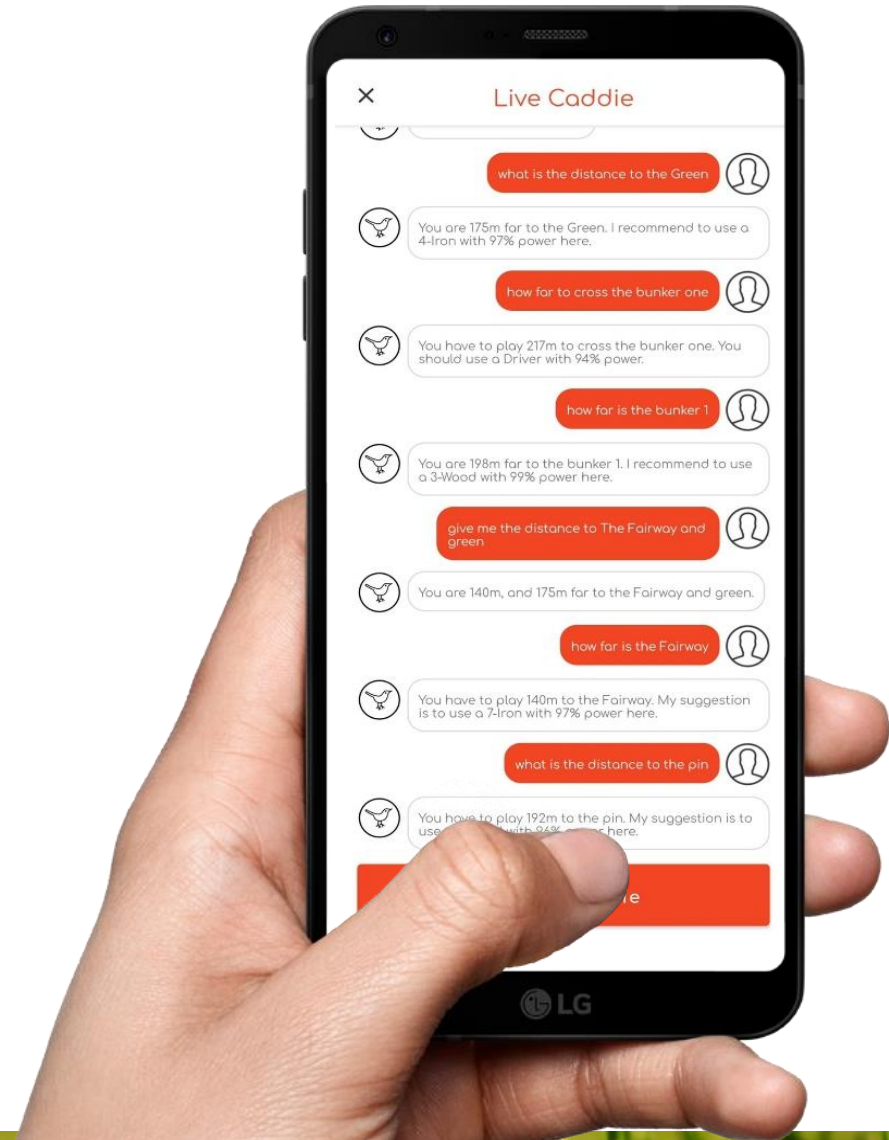
Front-end structure



Final Users'stories

Distance: Voice Caddie can detect multiple targets and calculate the multiple distances as the need from user. This function has the advantage of reducing the numbers of requests, consequently, it helps to optimize the cost of using Dialog Flow.

Club Recommendation: Voice Caddie can classify different target and recommend the club accordingly. User can ask multiple targets, in the responding, Voice Caddie not only show the distance, but recommend to player which type of club they can use accordingly. We ensure the high speed respond and high accuracy rate above 90% as the requirement of this project. Besides, the google mapping has added in to improve the visualization. All these tools are to improve the performance of user once using the Voice Caddie function from Hello Birdie.



Final Deliverables

Original agreements	Updated agreements	Final Delivery	Evaluation
3 Users' stories	1 Compete User' story	2 Compete User 'stories	Over Delivery
IOs Platform	Android Platform	Android Platform	Meet expectation
No UI/UX, Apple Watch Interfacings	Front-end with UI	Front-end + UI with Google Map Display	Over Delivery
Voice to text out put	Text to Speech output	Text to Speech output	Meet expectation

The project – Voice Caddie has 4 key items with 2 key items meet expectation and 2 key items over delivery. As the final agreements, Voice Caddie can detect the voice and respond to the user with Voice output as well. This is the best way to interact with users and improve the performance of users while using Hello Birdies.

Besides, we have ensured the voice detection accuracy at above 90% by using DialogFlow for our application.

Further Improvement: The 3rd user' story can be finished with the right update of the database. The final source code and application have been delivered to Sponsor. This will ensure Voice Caddie can further develop the Gameplan.





hello
birdie

Thank you!

Team EagleS

Thanh Tung TRINH
Sanjeet MAISNAM
Anh Tu NGUYEN
Shuhbam RANA