Connect to parent – FAFA

(communicate tool for kids via voice recognition)

Park Hyeong Jin
2016026271
Dept. of Information System
College of Engineering,
Hanyang University
Seoul, Rep. of Korea
Email : jin5378@hanyang.ac.kr

Lee Jeong Seon
2016026435
Dept. of Information System
College of Engineering,
Hanyang University
Seoul, Rep. of Korea
Email : com2769@gmail.com

Lee Jun Seok
2016026444
Dept. of Information System
College of Engineering,
Hanyang University
Seoul, Rep. of Korea
Email : junslee0912@gmail.com

Yoon Seung Gwon
2016026371
Dept. of Information System
College of Engineering,
Hanyang University
Seoul, Rep. of Korea
Email : csyoon1472@gmail.com

***Abstract*— FAFA is a parent-child connection service that is powered by artificial intelligence-based voice recognition technology. The penetration rate of mobile phones among South Korea's senior elementary school students surpassed 90% in 2018. However, the gap between the lower grade of elementary school(58.8%) is too big, and it is expected to be bigger if infants are included. In addition, the penetration rate of fixed-line telephones in households was 51.9% in 2019, the lowest ever. Given the trend of these figures, the penetration rate of communication ways(fixed-line telephone, mobile phone) for kids is low, which leads to one-way remote communication between parents and kids(caregiving application for parents).**

To solve this problem, we suggest using NUGU speaker based on voice recognition technology. The service will enable two-way communication with parents by providing tool of communication to kids.

Role Assignments

Roles	Name	Task description and etc.
User	Lee Jun Seok	The child who is in the home alone without mobile phone is main user of our service. He will gather information about child's needs. In order to select the main function of our services he will investigate what child want to ask to parent when he/she is in the home alone.
Customer	Lee Jeong Seon	Our main customer are parents with young children who don't have a mobile phone and landline phone. He will gather information about Parent's needs. In order to select the main function of our services he will investigate what parent want to communicate with their child.
Software Developer	Park Hyeong Jin	Software Developer is main developer of our services . he is responsible for designing our services and writing source code. He will investigate the API we can use for our services and choose development tool.
Development manger	Yoon Seung Gwon	A development manager plans, directs, and coordinates all activities related to software programs. He will schedule development and decide which functions will be included in the service.

1. Introduction

Motivation

Child Location Based Services (LBS), which is widely used by parent, is a child management service for parents. Parents who are concerned about their child's safety check their child's current location through the LBS service.However, parents are not the only ones who are worried about their child, and their child wants to know where their parents are now and when they will come back home. Also, child who is left alone at home wants to communicate with his or her parents. We understood these demands and felt the need for a way to communicate child at home with parents.

Problem statement (client's needs)

According to a report by the Korea Information Society Development Institute, the penetration rate of mobile phones in the lower grades children of elementary school in 2018 was 58.8%. Including preschoolers, more than half of children under the age of 10 don't have cell phones. In addition, the penetration rate of landline phones in 2018 was 51.9%.Parents are reluctant to buy their young children's cell phones for various reasons, including their children's addiction to smartphones and the burden of costs. When young children are alone at home, they often want to communicate with parents when they come to the house and where they are. However, Sometimes he can't contact his parents due to the absence of communication means. In order to solve the demand for communication with young children and parents, we proposes a "parent-child connection" service that use NUGU speakers.

Solution

FAFA is a parent-child communication service by artificial Intelligence speech recognition technology of SKTelecom's NUGU speaker. By using FAFA, Young child can communicate with their parents about where they are and when they will come back home. When the child asks the NUGU speaker where the parents are, it searches parent's location and informs it. Also, When the child arrives at home and talks to the NUGU speaker, the alarm goes off on parent's cell phone. The FAFA service uses NUGU speakers installed in the home to provide communication between parents and children at no additional cost.

Research on any related software

1) iSharing Lifestyle

iSharing by iSharingSoft is an app that provides a real-time locator service allowing family members and close friends to privately share their location information and communicate with each other. iSharing help parents and caregivers reduce anxiety around the whereabouts of their loved ones with easy tracking and alerting messages. There are four main functions.

1) Place alert : receive real-time alerts when family arrive at or leave destination
2) Panic alert : Just shake phone to send notification messages to your family member
3) Walkie-Talkie : Turn your phone into a Walkie-Talkie.
4) Location History : See the location history of family member

2) NUGU call

'NUGU-to-NUGU Call' is service about talking to your NUGU device or NUGU call subscriber.

This call is linked to data. You can use 'normal mode' to non NUGU call subscriber. A phone call is linked through the mobile phone of the account connected to the NUGU device.

3) NUGU SOS

SOS service is that sends pre-set text messages to designated recipients. You can set the sender's and recipient's information and the emergency SOS message to be sent. If you request an emergency SOS to the NUGU speaker, we will send an emergency SOS message to your registered number.

4) KAKAO mini

You can use Kakao Mini to send and read messages you received from users you want. Kakao Mini reads new messages from Kakao Talk's 1:1 chat room and group's Kakao Talk chat room. It reads messages from friends and reply. If it is not a text message, such as an emoticon or video, it tells the format of the message.

2. Requirements

2.1 User Side (Kids)

2.1.1 Ask parent's location via voice

1) SKT NUGU speaker handles this

2) Deliver request to server

2.1.2 View parent's location info

1) Provide map info in GUI(smartphone, TV) through application

2.1.3 Send alarm to parent

1) SKT NUGU speaker handles this

2.2 Client side (Parents)

2.2.1 Log-in

1) To identify clients, require client to log in

2.2.2 View kid's alaram

1) Client could see alarm that kids sent

2) Client could see alarm history already sent

2.2.3 Set location info & permitting option

1) Set a specific location to the desired word

2) Set permission option of client's location to NUGU

2.3 Server side

2.3.1 Send alarm to database

1) Server get alarm from NUGU speaker

2) Server keep these alarm in database

2.3.3 Request and Response the client’s location

1) Receive request of finding location with STT function in NUGU speaker

2) Deliver the request to client

3) Get response(client’s location)

4) Deliver the response to NUGU speaker with TTS function

2.3.4 Make json for NUGU

1) Server should make json file for learning of NUGU speaker

3. Development Enviroment

3.1 Software Development Platforms

We chose web enviroment to develop our project. Native app or other enviroment could be an answer. But, we don’t need too heavy enviroment, web is enough for our project. In addition, we will use AWS commercial cloud service such as Elastic Beanstalk for deploy. Lastly, SKT’s NUGU API will be used to analyze kids’ utterance and to recognize their intent.

3.1.1 React Native

React Native is a JavaScript framework for writing real, natively rendering mobile applications for iOS and Android. It’s based on React, Facebook’s JavaScript library for building user interfaces, but instead of targeting the browser, it targets mobile platforms. Similar to React for the Web, React Native applications are written using a mixture of JavaScript and XML-esque markup, known as JSX.

3.1.2 Django (web framework)

Django is a Python-based free and open source web framework that follows the model-template-views(MTV) architectural pattern. It is maintained by the Django Software Foundation. Django's primary goal is to ease the creation of complex, database-driven websites. The framework emphasizes reusability and "pluggability" of components, less code, low coupling, rapid development, and the principle of don't repeat yourself. Python is used throughout, even for settings files and data models.

3.1.3 SQLite

SQLite is a relational database management system (RDBMS) contained in a C library. In contrast to many other database management systems, SQLite is not a client–server database engine. Rather, it is embedded into the end program. SQLite is a popular choice as embedded database software for local/client storage in application software such as web browsers. It is arguably the most widely deployed database engine, as it is used today by several widespread browsers, operating systems, and embedded systems (such as mobile phones), among others. SQLite has bindings to many programming languages.

3.1.4 Amazon Web Service Elastic Beanstalk(EB)

AWS Elastic Beanstalk is an orchestration service offered by Amazon Web Services for deploying applications which orchestrates various AWS services, including EC2, S3, Simple Notification Service (SNS), CloudWatch, autoscaling, and Elastic Load Balancers. Elastic Beanstalk provides an additional layer of abstraction over the bare server and OS; users instead see a pre-built combination of OS and platform.

3.1.5 SKTelecom NUGU API

Based on SK Telecom’s technical skills such as voice recognition, voice synthesis and understanding of natural language through NUGU developers, the company can develop new functions through voice command in devices or applications owned by its affiliates. We will recognize and categorize the user’s voice commands through the NUGU API and send output results to the user via voice.

3.1.6 Kakao maps API

The Kakao Map API provides a variety of functions to produce map-based services on Web (Javascript) and mobile applications (Android, iOS). Local API provides contents and data of Kakao Map through REST API method.

3.2 Programming Languages

3.2.1 Javascript

Javascript is a high-level, interpreted scripting language that conforms to the ECMAScript specification. Javascript has flexible grammars: freedom from indentation, loose type checks. Also, it adopts modern progamming padigms and has convenient and great features: function programming, reactive programming. By using this language we can learn various modern progamming paradigms. Javascript is used in web browsers, which means it does not require any special working environment to run program written by Javascript.

3.2.2 Python

Python is an interpreted, high-level and general-purpose programming language. Created by Guido van Rossum and first released in 1991, Python's design philosophy emphasizes code readability with its notable use of significant whitespace. Its language constructs and object-oriented approach aim to help programmers write clear, logical code for small and large-scale projects.

3.3 Cost Estimation

This project rely on Amazon Web service. The cost estimation is in Table 1. This is calculated by Amazon Web Service Cost Calculator.

Table 1. cost estimation

Service	Region	Cost(monthly)
Amazon EC2	Asia Pacific (Seoul)	USD($) 11.65

3.4 Development Enviroment Description

Used development enviroment tools inforatmion is described in Table 2.

Table 2. development environment tools

Name	Version	Description
Windows	10 Home	Operating System made by Microsoft
macOS	Catalina(10.15)	Operating System made by Apple, used in Macbook
Visual Studio code	1.50.1	Text editor and integrated development editor made by Microsoft

3.5 Market Research & Software in Use

3.5.1. Market research

Location-based service(LBS) market size stood at USD 16.14 billion in 2018 and is expected to reach USD 66.61 billion by 2026. There are lots of services based on LBS, and one of the most popular services is the tracking services which provide the location of somebody for safety or other reasons. Most of these applications are for the parents who concern about their child's safety. For example, 'iSharing' and 'Google family link' are one of the popular apps that provides a real-time locator service allowing family members and close friends to privately share their location. However, kids who don't have mobile phones cannot use these apps, and these apps are mainly for parents who want to know their kids' locations. There are not many services for children. Also, 'Zenly' is the most popular social networking apps among teenager and it surpassed 10 million users in 2019. It provides a map that lets users see their friends and what they are up to. In the case of 'Zenly', we find out teenagers wonder about what their parents are doing and where they are.

3.5.2. Voice Recognition AI

Voice or speaker recognition is the ability of a machine or program to receive and interpret dictation or to understand and carry out spoken commands. Voice recognition has gained prominence and use with the rise of AI and intelligent assistants, such as Nugu, Siri, and Bixby. Voice recognition systems enable consumers to interact with technology simply by speaking to it, enabling hands-free requests, reminders and other simple tasks.

3.5.3. Located-based services (LBS)

Location-based services offer a wide range of opportunities because of their ability to provide information associated with a particular location or place. LBS use real-time data through mobile devices and keep track of the geographical location of the phone. Increased used of smart devices and location-aware technologies drive the growth of LBS market. The market of LBS has grown gradually because of its convenient functions such as navigation services, tracking services and information services.

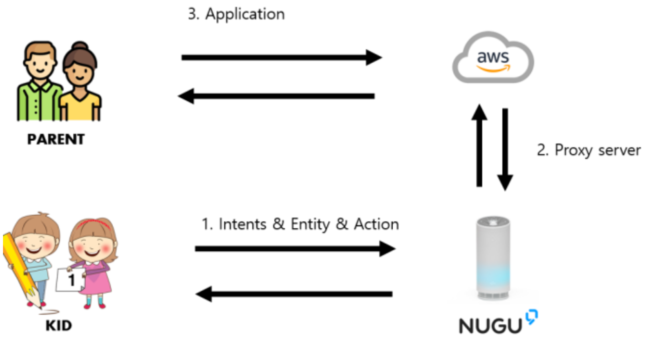
3.6 Task distribution

Task distribution is shown in table 3. Note that each of us periodically switched off our roles for sharing our ideas and improving the accuracy and efficiency of our project.

Table 3. Task distribution

Task	Name	Description
User	Lee Jun Seok	Gather and investigate users' requirements in this project
Customer	Lee Jeong Sun	Gather and investigate customers' information in this project.
Software Devloper	Park Hyeong Jin	Design our service and write source code. Also, investigate and choose the appropriate API.
Development Manager	Yoon Seung Gwon	Manage the overall project. Also, schedule the development of project.

4. Specification



This image shows our service's process. Kid is user of NUGU speaker. Parent send their location data through web application. And web application act as proxy server for NUGU speaker.

4.1 User Side (Kids)

4.1.1 Ask parent's location by voice (NUGU play)

1) Custom Intent

a. Intent name : find.FAFA

Name of the function that the child uses to find parents.Only English, numbers, ., -, _ are available, Duplicate is not allowed. Naming rules such as 'ask.weather' are recommended.

b. Sentence : "Where is father"

Enter the sentences that you expect the child who wants to find the parent will say. The more diverse, the better. At least 30 registration is recommended.

c. Type : FAMILY(mother, father)

Classify words that play the same rol within a sentence into types. If there is not build-in type, you should add custom type and classify.

d. Synonym : Mother, Mama, Mom…

When there are multiple words that refer to same entity, it is normalized by adding synonyms.

2) Custom Action

a. utterance parameter : FAMILY

Specify the 'FAMILY' parameter in the use's utterance. Use this to request from backend-proxy-server and configure the NUGU speaker's response statement.

b. backend parameter : LOCATION, STATUS

Get 'LOCATION', 'STATUS' parameter from backend-proxy-server. Location may be Office or Home. And Status may be 'coming', 'working' and others.

c. branch action

The backend parameter changes depending on the parent's location. Write branch action considering each situation.

Ex1) Father is coming to home from office.

Ex2) Mommy is working at office now.

4.1.2 Send notification to parent

1) When kids find parent, make log (save it up to 1 month)

2) Send this log to parent through backend server

4.2 Client side (Parents)

4.2.1 Log-in

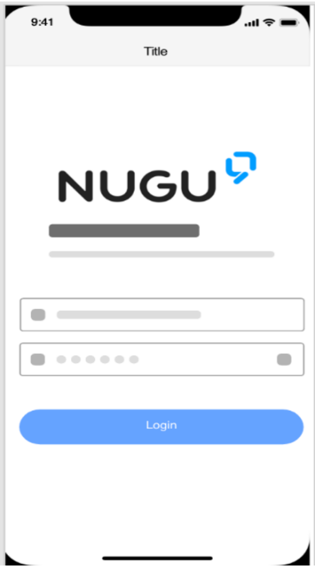


Figure 1. Login

1) Client could login with SKT ID

a. OAuth certification

Using Authorization server form SKT, perfrom OAuth certification with the way of authorization code grant. If certification is success, perfrom access token transmission. If certification is failed, return message “Authentication failed. Please reenter your ID and password please”

2) After log-in, client could use service

4.2.2 View kid’s notification

1) Parent could see log list of the request by kids

2) “Kid is finding you” fixed message and timestamp will appear.

4.2.3 Set location info & permitting option

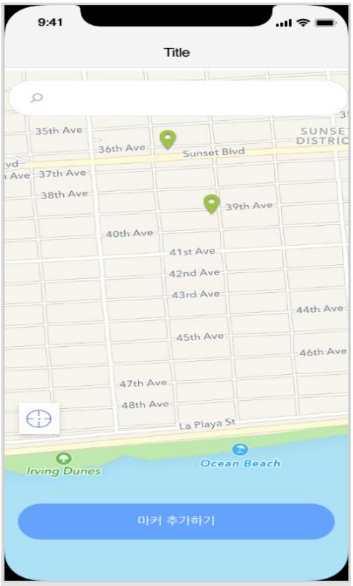


Figure 2. Set the marker

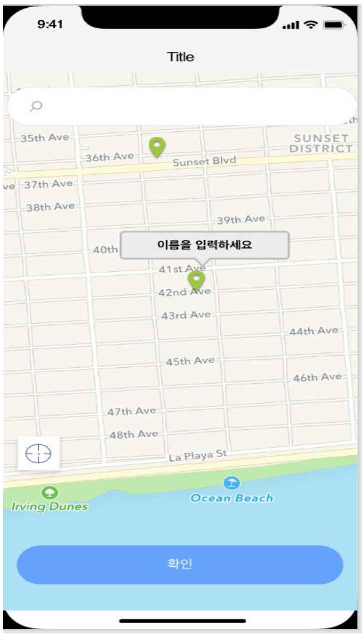


Figure 3. Name the marker

1) Specify a specific location

Use Kakao Map API’s function to search a specific location or return the current location. Set name the location and save it with latitude and longitude.

- location : varchar2(10)

- latitude, longitude : float

Ex) {location :’회사’, latitude:36.232, longitude:35.231}

2) Add marker at the stored location

3) Send user’s current location to server

Use the background feature of the native app to send current location to server. Option for using location information must be checked by user before send this data.

4.3 Server side

4.3.1 Request and Response the client’s location

Server return different backend-parameter depending on parent’s location.

```
if CurrentLocation== KnownLocation
   return CurrentLocation, status
//{location : ‘회사’, status:’일하는 중’}
elif CurrentLocation between KnownLocation
  return KnownLocation, status
```

```
//{location:['회사', '집'], status:'퇴근하는 중'}
else
  return Status
// {status : '외출 중'}
```

4.3.2 Make json for NUGU

1) When parent update location, server should make json file. Django Rest Framework cover this automatically.

5. Architecture Design

5.1 Overall Architecture

Our service is web-based application and heavily depends on Amazon Web Service for deployment. Our application's UI is implemented with React Native. Server is implemented by Python with Django framework web server on it to serve data to frontend UI as REST API, and interact with SKTelecom NUGU's API and speaker. Entire API server is running on Amazon Web Service EC2 instance. Figure 1 shows overall architecture of the service.

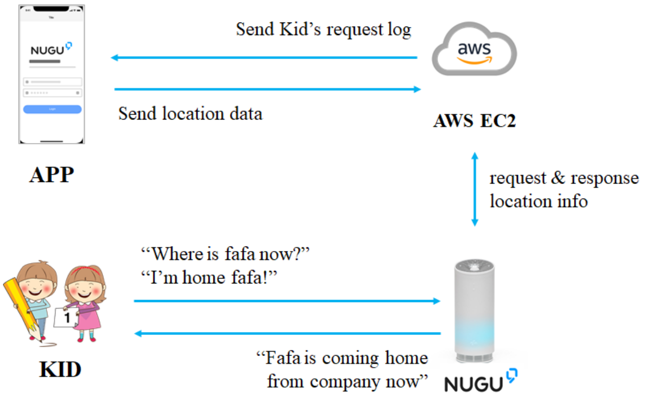


Figure 7. Overall System Architecture Diagram

5.2 NUGU Architecture Design

Below are in-voice commands required to make interaction with SKTelecom NUGU Speaker.

5.2.1 General Setting

1) Backend Proxy Server

2) Web URL : TBA

3) Exception message : "Connect Error"

5.2.2 Intent

1) inform.home

Inform to parent that kids get home now.

Command : (FAMILY_NAME), (ending of word) ex) "Mom, I'm home"

Example mention	**Mom**	**I'm home**
Category	FAMILY_NAME	ending of word
Entity	FAMILY_NAE	STATEMENT_HOME

Action for inform home Intent : alert_NUGU

Example mention	엄마	에게	알려드렸어요.
Category	FAMILY_NAME	Postposition	Ending of word
Utterance Parameter	FAMILY_NAME	Fixed postposition	Fixed statement

2) ask.location

Ask for family's location

Command : (FAMILY_NAME), (ending of word) ex) 엄마 어디야?

Example mention	엄마	어디야?
Category	FAMILY_NAME	ending of word
Entity	FAMILY_NAE	STATEMENT_LOCATION

- Action for ask.location Intent

a) now_location

When family member is in designated place, now_location tells the location of family member.

Example mention	엄마	는	회사	에 있어요.
Category	FAMILY NAME	Postposition	Location	Ending of word
Utterance Parameter	FAMILY_ NAME	Fixed post position	LOCATION	Fixed Statement

b) between_location

When family member is between home and company, between_location tells the starting point, destination, and status.

Example mention	엄마	는	회사	에서	집	으로	퇴근하는	중이에요
Category	FAMILY_ NAME	Postposition	Location	Postposition	Location	Postposition	Status	Ending of word
Utterance Parameter	FAMILY _NAME	Fixed postposition	START_ LOCATION	Fixed Postposition	DESTI_LOCATION	Fixed postposition	STATUS	Fixed Statement

c) except_location

When family member is not in designated place and not between two designated places, except_location tells family member is gone.

Example mention	엄마	는	지금 외출중이에요.
Category	FAMILY_NAME	Postposition	Statement
Utterance Parameter	FAMILY_ _NAME	Fixed postposition	Fixed statement

5.2.3 Entity

1) FAMILY_NAME

Family members

Parameter	Synonym
엄마	어머니
아빠	아버지
형	오빠, 형님
누나	언니

2) STATEMENT_HOME

Fixed Statement for ending of word when kid come back home

Parameter	Synonym
나 집이야	나 지금 집이야, 나 도착했어, 나 집왔어…

3) STATEMENT_LOCATION

Fixed Statement for ending of word when kid ask for family member's location.

Parameter	Synonym
어디야	어디야, 지금 어디야, 어디에요, 지금 어디에요…

5.2.4 Actions

1) alert_NUGU

- Custom Action

- Using it when kid come back home.

- Trigger

  - inform.home

  Ex) "엄마, 나 도착했어"

- Prompt

  Ex) "{{FAMILY_NAME_}}에게 집에 왔다고 알려 드렸어요."

2) location

- Custom Action

- Root Action

- Trigger: ask.location

- Output

  6.1) now_location

  - Branch Action

  - Using it when family member is in designated place.

- Trigger: When LOCATION(Backend Parameter) exist

- Prompt

Ex)”{{FAMILY_NAME}}은 {{LOCATION}}에 있어요”

6.2) between_location

- Branch Action

- Using it when family member is going to designated place or home.

- Trigger: When START_LOCATION and DESTI_LOCATION(Backend Parameter) exist.

- Prompt

Ex)“{{FAMILY_NAME}}는 {{START_LOCATION}}에서 {{DESTI_LOCATION}}으로 {{STATUS}} 중이에요”

6.3) except_location

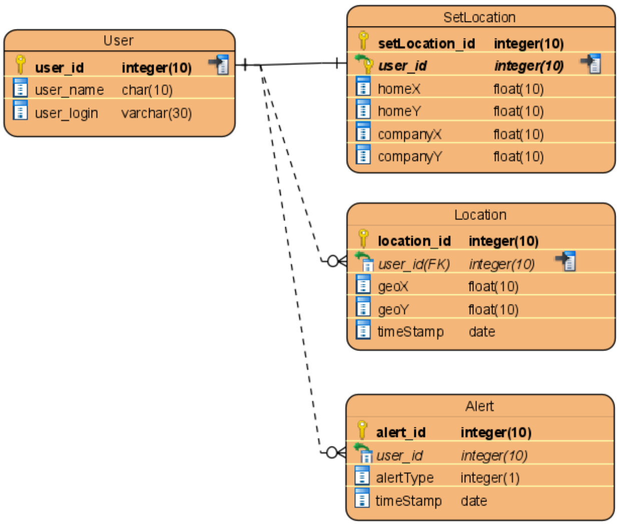
- Branch Action

- Using it when family member is not in designated place or not between two designated places.

- Prompt

Ex) “{{FAMILY_NAME}}은 지금 외출 중이에요.”

5.3 Database Design



5.3.1 User

- user_id(PK) : identification of user

- user_name : user’s name

- user_login : information of user login(like token)

5.3.2 SetLocation

- setLocation_id(PK) : identification of SetLocation

- user_id(FK) : reference ‘User’ table

one user could set only one home and company

- homeX : longtitude of user’s home

- homeY : latitude of user’s home

- companyX : lontitude of user’s company

- companyY : latitude of user’s company

5.3.3 Location

- location_id(PK) : identification of Location

- user_id(FK) : reference ‘User’ table

one user could make multiple location logs

- geoX : lontitude of user

- geoY : latitude of user

- timeStamp : time of this data made

5.3.4 Alert

- alert_id(PK) : identification of Alert

- user_id(FK) : reference ‘User’ table

one user could make multiple alert logs

- alertType : define what request is this

- timeStamp : time of this data made

5.4 Directory Organization – Front End

Table 4 shows the directory organization of React-Native frontend application’s project.

TABLE 4. DIRECTORY ORGANIZATION FOR FRONTEND APPLICATION PROJECT

Directory	File names	Module name in use
/src/components	Index.tsx	styled-components/native
/src/Assets/Images	.png	-
/src/Screens/Navigator	Index.tsx	react-navigation
/src/Screens/Login	Index.tsx	react-native-community/async-storage
/src/Screens/CheckLogin	Index.tsx	react-native-community/async-storage
/src/Screens/Map	Index.tsx	react-native-maps, react-native-geolocation-service

5.4.1 /src/components

React-native uses functional-component-based development method. This folder contains components that are frequently used like button, Input components. We will use these components by export and import function.

5.4.2 /src/Assets/Images

Inside the Asset folder, there are resources such as image files and fonts needed for the application. There are app icons, button images, and marker images in the image folder images.

5.4.3 /src/Assets/Screens

The Screens folder contains functions that make up the screens in the application. Navigator is a function that controls the movement of the screen. The screen consists of a Login screen for logins, a CheckLogin screen for tokens checking, and a Map window for positioning.

5.4.4 styled-components/native

Styled Components is an open-source library that helps with the application of styling of react and react native. It can create a style on a single JavaScript file. In other words, you can do CSS work in JavaScript files without CSS files.

5.4.5 react-navigation

A React Navigation is a chain of navigators that define the *screen* flow of your app. React Navigation's stack navigator provides a way for your app to transition between screens and manage navigation history.

5.4.6 react-native-community/async-storage

React Native Async Storage is an asynchronous, unencrypted, persistent, key-value storage system for React Native. Async Storage send and receive data like token, user information, location of home and company with Backend Server.

5.4.7 react-native-geolocation-service

React Native Geolocation Service tell user’s current location and allows to track user’s location. Using this module, we took user’s current location and whenever the user’s location changed, we used fetch module to send the location information to backend server.

5.4.8 react-native-maps

React Native Maps is a component system for maps. Using this module, we displayed Google Maps on the screen and marked house, company, user’s current location with marker

5.5 Directory Organization – Back End

TABLE 5. DIRECTORY ORGANIZATION FOR BACKEND APPLICATION PROJECT

Directory	File names	Module name in use
/.ebextensions	django.config	Django, AWS ElasticBeanstalk
/.elasticbeanstalk	config.yml	AWS ElasticBeanstalk
/NUGU	settings.py, urls.py, wsgi.py	Django, Django RestFramework
/FAFA	models.py, serializers.py, urls.py, views.py	Django, Django RestFramework
/static	.css, .js, .img,	-
.	.gitignore, db.sqlite3, manage.py, requirements.txt	Django

5.5.1 /.ebextensions

In order to implement Django module on AWS ElasticBeanstalk environment, path configuration needed. ‘django.config’ file contains option for setting and covers deploy problem.

5.5.2 /.elasticbeanstalk

To upload directory on AWS instance, application name or region and the other configuration needed. ‘config.yml’ would cover upload problem.

5.5.3 /NUGU

In order to use Django framework which is run by Python, allowed hosts, url, apps, packages and other things should be set on this directory. On ‘settings.py’ and ‘urls.py’ files, there are some customized setting for this project.

5.5.3 /FAFA

Django RestFramework module is used in this directory. ‘models.py’ make models (DB table). ‘urls.py’ handles routing. ‘views.py’ gets request and sends response by function that we made. ‘serializers.py’ make response in json form from queryset.

5.5.3 SQLite

SQLite is relatively light embedded database for applications. To store data, only one file ‘db.sqlite3’ is needed. Small and concise DB would run in local, so you don’t have to worry about the cost of network configuration, firewalls, network address translation, and so on. We can only focus on code level.

5.5.4 AWS ElasticBeanstalk

A fully managed service of AWS that deploy, expand and manage web application. It handles capacity provisioning, load balancing, auto scaling, monitoring, and hosting environment automatically. So user just deploy application. In addition, there is no unnecessary expenditure to pay. Through easy deployment, this module makes us focus on code level during development.

5.5.6 Django

Django is Web framework used in our project. Using this module, we don’t ‘reinvent the wheel’ about HTTP service. Django provide simple and convenient middle-ware like routing, error-handler and parser. Many of above middle-ware can be implemented in custom way that we want to make.

5.5.5 Django Rest Framework(DRF)

DRF is a powerful and flexible module for creating WEB APIs. To develop backend proxy server of NUGU-PLAY in REST API form, we used DRF. This module has powerful ‘viewset’ that enables us to develop CRUD functions easily and quickly. Also we can customize a general view.