Providing personalized information with Voicelog

Chang Hee Kim
Undergraduate School of Information System
Hanyang Univ.
Seoul, Korea
flyc4@hanyang.ac.kr

Taek Joon Kim
Undergraduate School of Information System
Hanyang Univ.
Seoul, Korea
xorwns2765@hanyang.ac.kr

Hyun Kook Jeon
Undergraduate School of Information System
Hanyang Univ.
Seoul, Korea
ksp707@hanyang.ac.kr

Heon Nam Chu
Undergraduate School of Information System
Hanyang Univ.
Seoul, Korea
cngjsskaisme@hanyang.ac.kr

***Abstract*—Giving personalized information by collecting and analyzing daily diaries using NUGU AI speaker is the main goal of the project. Firstly, User asks daily lives to the speaker. Then the machine learning model which receives data from the speaker analyzes the user's emotion and reports the overall emotion every week. With the reports, the speaker recommends items based on the emotion that is analyzed. The project will be done with NUGU dashboard and python.**

***Index Terms*—NUGU AI speaker, voice, voicelog, emotion, recommendation**

I. Introduction

According to research from SKT, people regard AI speakers as an interlocutor as the number of single households increases [1]. Also, people feel familiar with AI speakers when they share personal experiences. This experience builds trust in the information given from an AI speaker [2]. Our team assumed that when people talk about their daily lives and the emotion to AI speakers, they feel familiar with AI speakers which can lead to building trust in recommendation of services and commercial items. We will induce conversation on daily life with AI speakers by recording a voicelog with AI speakers. For example, first the AI speaker asks the user 'How was your day?' at the certain time in daily basis. Then the user will answer to AI speaker and stores the given sentences. The machine learning model will analyze emotion from stored text and sends the result to the AI speaker. If the AI speaker found that the user feels much 'depressed', it will recommend heartwarming songs or movies, or introduce famous spa to get relaxed.

II. Requirements

A. Application) Main Page

1) Loading screen: After executing application, show loading screen until the app calls up the required data.
2) Lock screen: If a user sets a password, the app asks for a 4-digit password to the user. If the user typed the correct password, then show Main screen. If the password was wrong, erase the input and show Lock screen again. If the user did not set a password, skip the lock screen and show Before log-in screen.
3) Log-in screen: If the value of string type variable in application "AccessToken" is NULL (AccessToken == NULL), show "Log-in with XX" button. (ex. Login with Google)
4) Main menu screen: If value of variable "AccessToken" is not empty (AccessToken != NULL), show Monthly diary screen.
5) Log-out: If the user selects "Log out" button, set the value of AccessToken to NULL (AccessToken == NULL) and show Before Log-in screen.

B. Application) Log-in with OAuth

1) First Log-in screen: If a user selects "Log in with XX" button from Before log-in screen when the user is not logged in, show policy about using information in the user's device, a checkbox for agreement to policy, "Accept" button, and "Cancel" button.
2) Agreement to policy and Authorization: If a user selects "Accept" button without check for agreement to policy, show pop-up window which show a message "Please agree with all of the policies". If a user selects "Accept" button with a check for agreement to policy, show pop-up windows which ask for agreement to accessing required functions (ex. Recording, picture gallery, file, etc.) with "Accept" and "Cancel" buttons.
3) Request for OAuth certification: After Agreement to policy and Authorization is done, request OAuth certification to Backend proxy by sending the user's ID and password at Resource Server. If certification is failed, repeat this process.
4) Save Access Key: Save access key that received from "Transmission of Access Token" process.
5) Show main menu: After Save Access Key process, show Main menu screen.
6) Cancel first log-in: If the user selects "Cancel" button in First Log-in screen, go back to Before log-in screen

C. Application) Monthly Diary screen

1) Monthly diary screen (Main screen): Show Menu button for selection of "Log out" and "Configuration". Show calendar of the chosen month and a Button for choosing a year/month to see a calendar (Default value is the current month in the device's system). Dates that the user wrote his/her diary are checked with circle mark so that the user can notice that he/she wrote a diary on

those days (Diary from text input has a different color of circle mark with diary from voice record).

2) Preview of daily diary screen: If the user selects a date that checked with circle mark from Main screen, show score of the day (1 5). Also, if the diary's format is text, show part of the diary. If the diary's format is voice record file, show file name. When the user selects a date that not checked with circle mark, print message "There is no diary". Default value is 1st day of the month.
3) Button for choosing a year: If the user selects this button, show years (before 4-year after 4 years from the current year in device's system) and "Go back" button. The user can select a year, and the screen shows a list of months in the year. The user can see the previous nine years by selecting a left arrow button, and the next nine years by selecting a right arrow button. These arrow buttons are displayed at the bottom of the screen. Selecting "Go back" button will show the previous menu.
4) Retrieve data of monthly diary: Include selected year and month value in object and request diary in corresponding period. If there is no input for year and month, include year and month of currently accessed date in object and requests to backend proxy.
5) Button for choosing a month: After selecting a year, show buttons with numbers from 1 to 12, and "Go back" button. If the user selects a month, show Monthly diary screen of selected month/year.
6) Button for previous/next month: There are two buttons attached at a calendar that can switch the calendar to the previous/next one: button for the previous month (<) and next month (>). When the current calendar is January, (<) button will switch the calendar to December last year. If the current calendar is December, (>) button will switch the calendar to January next year.
7) Retrieve data of monthly diary: Include selected year and month value in object and request diary in the corresponding period. If there is no input for year and month, include year and month of currently accessed data in object and request backend proxy.
8) Receive data of monthly diary: Optimize retrieved diary data from Monthly diary transmission for the application's UI and show it.

D. Application) Daily diary screen

1) Writing a diary: When user select Preview of daily diary screen and if there is no diary, show "Write in text", "Write in voice", "Go back" button.
2) Look up a diary: When user select Preview of daily diary screen and if the diary is created as text, show Text diary screen. If the diary is created in voice, show Voice diary screen.
3) Text diary screen: Show score of the date, contents of the diary and "Delete", "Convert to voice diary" buttons. If a user touches the text diary screen, show Writing a text diary screen containing its score and contents.
4) Voice diary screen: Show score of the date, a button with the name of the recorded file and "X", "Delete", "Convert to text diary" buttons.
5) Button for playing recorded file: If the user selects a button with the name of the recorded file, load file by executing Recorded file transmission and play the file. If the button is selected during play, pause it. After playing the entire file, set playtime to 00:00 and stop it.
6) Delete recorded file of voice diary: If the user selects "X" button next to filename button, show pop-up window containing message "Existing file will not be recovered. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete recorded file and show Writing a voice diary screen. If "Cancel" button is selected, maintain the original state.
7) Convert to voice diary: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Writing a voice diary screen. If "Cancel" button is selected, maintain the original state.
8) Convert to text diary: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Writing a text diary screen. If "Cancel" button is selected, maintain the original state.
9) "Delete" button: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Monthly diary screen. If "Cancel" button is selected, maintain the original state.
10) "Go back" button: If the button is selected, show the recently shown screen.
11) Request for authorization: If the user selected "Cancel" button at Agreement to policy and Authorization and select "Look up the diary", show pop-up window containing message "You can't write a diary without authority" and "Confirm" button.
12) Pop-up window for authorization: If the user selects "Confirm" button at Request for authorization, show pop-up window for requesting missing authorization for the application.
13) Referring daily diary data: Include selected date in the request object and request diary of the date to backend proxy
14) Receiving daily dairy date: Optimize the diary from Daily diary data transmission for the application's UI and show it.

E. Application) Writing a diary screen

1) "Write in text" button: If button is selected, show Writing a text diary screen.

2) “Write in voice” button: If button is selected, show Writing a voice diary screen.
3) “Go back” button: If button is selected, go back to Monthly diary screen.

F. Application) Writing a text diary screen

1) Date: Show the current year/month/day.
2) Score contents of diary: Show radio box for selection of today’s score and box for writing a diary.
3) “Confirm” button: When the user selects “Confirm” button, check whether a daily score is selected or not. If the score is not selected, show pop-up message “Please select daily score.”. If the score is selected, including score and contents of the diary in requesting object, send the request to the server, and execute Create daily diary. After that, show Text diary screen of the diary.
4) “Cancel” button: When the user selects “Cancel” button, show pop-up message “Changes will not be saved. Will you cancel it?” and “Yes”, “No” button. If “Yes” button is selected, go back to Monthly diary screen. If “No” is selected, maintain the original state.

G. Application) Writing a voice diary screen

1) Date: Show current year/month/day
2) Record/Stop: When the application is not recording a diary, show “Record” button and “Push this button to record your diary” message. Start recording if the button is pressed. When the application is recording a diary, then the button changes into "Stop" button. When the user press "Stop" button, the application finishes recording and show a pop-up window for deciding the name of the recorded file and “Confirm”, “Cancel” button.
3) Screen after saving recorded file: After record at Record/Stop, the user can change the file’s name to the user’s input. If “Confirm” button is selected, change the file’s name, create a file to .wav format, and save file through Save recorded file execution. After these processes, show Writing a voice diary screen. Also, show a button with the file name and “X” button instead of “Record/Stop” button. If “Cancel” button is selected, the file will not be saved and show Writing a voice diary screen.
4) Delete recorded file/activate Record/Stop: If “X” button in Screen after saving recorded file is selected, delete recorded file name from the screen and execute Delete recorded file. After that, show Record/Stop button.
5) Saving diary and transmission: If “Confirm” button in Writing a voice diary screen is selected but the score is not selected, show a pop-up message “Please select daily score”. If the score is selected, include recorded file and score in request object, send it to backend proxy and execute Create daily diary. After that, show Voice diary screen. (Saving in server)
6) “Cancel” button: If “Cancel” button is selected, show pop-up window containing the message “If there is a recorded file, it will be deleted. Will you cancel?” and “Yes”, “No” buttons. If “Yes” button is selected, execute Delete recorded file and show Monthly diary screen. If “No” button is selected, maintain the current window.

H. Application) Configuration

1) Menu screen: If a user selects “Configuration” button in main screen, show Password setting menu option which can decide whether to activate Lock screen function or not.
2) Password setting menu: If the user selects Password setting menu option in Menu screen and password is already set, deactivate Lock screen function. If a password is not set, show Password setting screen.
3) Password setting screen: Get 4-digit input from the user. After the first input, ask the user to input one more time to check. If the second input is different from the first one, show a message “It is different from the first input password” and delete the user’s input. If the second input is corresponding with the first one, set a password to the 4-digit input.
4) “Go back” button: If the button is selected, show the previous screen.

I. Application) Request to backend proxy

1) Header setting: If a request to backend proxy occurs, transmit variable to the header that can identify the subject of the request is “Application”.

J. Application) Application error handling

1) Network error: If the network is unavailable, show a pop-up window containing the message “Sorry. The network connection is unavailable now. Please check the network connection setting and try again.” and “Confirm” button. If “Confirm” button is selected, show the previous screen.
2) Backend proxy connection error: If Backend proxy connection is unavailable, show a pop-up window containing message “Sorry. The server is unavailable now. Please try again later.” and “Confirm” button. If “Confirm” button is selected, show the previous screen.

K. Speaker) Account

1) Account registration/change/delete: A speaker can register/change/delete an account that is currently linked or will be linked.
2) Account information transmission: Whenever a speaker requests to backend proxy, it transmits the account’s information stored on the speaker to backend proxy.

L. Speaker) Voice recognition

1) Wake-up: Execute Wake-up function on a specific word.
2) EPD (End Point Detector): Make a proper judgment of start/end point of the user’s speech.
3) Noise cancellation: Remove any noise without the user’s speech so that it can be recognized correctly.

4) ASR (Automatic Speech Recognition): Transfer speech into text without any error
5) Data forwarding – voice recognition: Transmit text from ASR as an input value of NLP.

M. Speaker) NLP (Natural Language Processing)

1) Text processing: Perform domain/intent classification of text from Data forwarding – voice recognition.
2) Data forwarding – NLP: Forward data result of Text processing to DM.

N. Speaker) DM (Dialog Manager)

1) Conversation – Look up daily diary: User can look up his/her diary within a year through NUGU speaker. If the speaker recognizes the speech, it includes user account information which is in the speaker in request object and requests "Daily diary data transmission" to backend proxy. If diary data exists, forward score and contents of the diary to NLG. If diary data does not exist, requests NLG to make a proper response such as "There is no diary in (date)". ex) User: "Please show me a diary of December 1st." Speaker: "Let me read a diary of December 1st. (Read the diary)" or "A diary of December 1st does not exist."
2) Conversation – Weekly score: User can look up the most frequently appeared score during a week (Weekly score). If the speaker recognizes the speech, forward the result of "Calculating the most frequently appeared daily score" to NLG. If data does not exist, requests NLG to make a proper response such as "Calculation is not finished, or there is no diary in last week. Please try again on Sunday." Ex) User: "Please tell me weekly score of last week." Speaker: "Weekly score of last week is (1 5)." Or "Calculation is not finished, or there is no diary in last week. Please try again on Sunday."
3) Conversation - Product recommendation: User can look up products which are recommended based on diary of last week. If the speaker recognizes the speech, forward the result of "Product recommendation" to NLG. If data does not exist, requests NLG to make a proper response such as "Calculation is not finished, or there is no diary in last week. Please try again on Sunday.". Ex) User: "Please recommend me product based on last week's diary" Speaker: "Recommended product of this week is (product) of (company)." or "Calculation is not finished, or there is no diary in last week. Please try again on Sunday."
4) Conversation – Weekly score Product recommendation: User can use function "Conversation – Weekly score" and "Conversation – Product recommendation" at the same time.

O. Error handling response

1) Speech recognition failure: Make a proper response such as "Sorry, it is too difficult contents to me.".
2) Backend proxy connection failure: Make a proper response such as "Sorry. Network connection is unavailable now. Please try later.".

P. Data forwarding

DM Forward all responses from DM to NLG.

Q. Speaker) NLG

1) Response conversion: Convert all responses that NLG received to text-format sentences which has preset tone manner.
2) Data forwarding – NLG: Forward sentence from Response conversion.

R. Speaker) TTS

1) TTS execution: Convert all test-format sentences in TTS to voice data.
2) Response: Output the result of TTS execution.

S. Emotion analysis model for sentence

1) Learning: Multiple times of learning is possible using learning data sets from developer.
2) Emotion analysis: Using input sentence, return "Positive" or "Negative" value which indicates emotion.

T. "Product" entity extraction algorithm

1) Learning: Multiple times of learning is possible using learning data sets from developer.
2) "Product" entity extraction: Extract the most frequently appeared entity of product from input sentence. Ex) "I drank beer today" -> Both "beer" and "today" appeared, but "today" is not a product. Only "beer" entity must be extracted.

U. Product recommendation model

1) Learning: Multiple times of learning is possible using learning data sets from developer.
2) Product recommendation: Get information such as product for recommendation, price and where to buy based on input word. Ex) Input word: Beer -> Get name of the beer that recently popular or new, its price and where to buy.

V. Backend proxy

1) OAuth certification: Perform OAuth certification using account information from "Request for OAuth certification"
2) Access Token transmission: After OAuth certification, transmit issued access token to application/speaker.
3) Monthly diary data transmission: Transmit diaries of year/month in request object to application/speaker.
4) Daily diary data transmission: Transmit diaries of year/-month in request object to application/speaker.
5) Create daily diary: Create diary using account information, date and diary contents in request object.
6) Revise daily diary: Revise requested diary's contents from application to contents in request object.

7) Delete daily diary: Delete diary of requested date.
8) Save recorded file: Save requested record file in Backend proxy.
9) Recorded file transmission: Transmit requested recorded file to application.
10) Delete recorded file: Delete requested recorded file from backend proxy.
11) Load embedded model: Load model embedded in back-end proxy (Emotion analysis model, entity extraction model, product recommendation model)
12) Scheduling: Perform the function every time and interval specified by developer.
13) Recorded file -> text conversion: Convert speech in .wav recorded file to text.
14) Distinguishing requisition principal: Distinguish requisition principal which called backend proxy (Application, speaker, etc.)
15) Backend proxy error handling: If error is occurred while backend proxy is running, terminate the request after recording type of error and occurrence time in log.

W. Weekly analysis in backend proxy (Functions that executed every Saturday 2:00 AM. Functions are executed sequentially.)

1) Analysis product entity by score: Forward contents of diaries in a week of each user to product entity extraction model and execute "Product" entity extraction. Result of analysis is saved by user's account and score.
2) Daily emotion analysis: Forward contents of diaries in a week of each user to emotion analysis model and execute Emotion analysis. Result of analysis of each diary is stored by user's account and score.
3) Calculation of the weekly score: Calculate modal value of daily score in a week of each user. However, daily score which is contradictory to the result of Daily emotion analysis will be excluded. For example, if score of a daily diary is higher than 3, but the result of Daily emotion analysis is "Negative", the data is excluded. Also, if score of a daily diary is lower than 3, but the result of Daily emotion analysis is "Positive", the data is also excluded. The result of modal value is stored by each user.
4) Saving recommended product information: Load each user's result of Calculation of the weekly score. Load product entity from Analysis product entity by score corresponding to each user's weekly score. Forward each product entities to Product recommendation model and execute Product recommendation. The result is stored by each user.

III. Specification

A. Application) Main Screen

1) Loading screen: After executing application, show loading screen until the app calls up the required data.
2) Lock screen: If a user sets a password, the app asks for a 4-digit password to the user. If the user typed the correct password, then show Main screen. If the password was wrong, erase the input and show Lock screen again. If the user did not set a password, skip the lock screen and show Before log-in screen.
3) Log-in screen: If the value of string type variable in application "AccessToken" is NULL (AccessToken == NULL), show "Log-in with SKT" button.
4) Main menu screen: If value of variable "AccessToken" is not empty (AccessToken != NULL), show Monthly diary screen.
5) Log-out: If the user selects "Log out" button, set the value of AccessToken to NULL (AccessToken == NULL) and show Before Log-in screen.

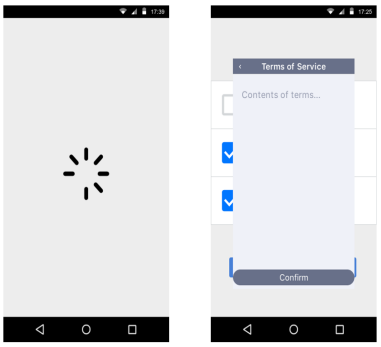

Figure III-A.1. Loading/Locked Screen

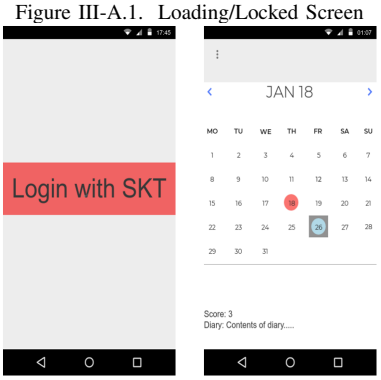

Figure III-A.2. Before/After Login

B. Application) Log-in with OAuth

1) First Log-in screen: If a user selects "Log in with SKT" button from Before log-in screen when the user is not logged in, show policy about using information in the user's device, a checkbox for agreement to policy, "Accept" button, and "Cancel" button.
2) Agreement to policy and Authorization: If a user selects "Accept" button without check for agreement to policy, show pop-up window which show a message "Please agree with all of the policies". If a user selects "Accept" button with a check for agreement to policy, show pop-up windows which ask for agreement to accessing required functions (ex. Recording, picture gallery, file, etc.) with "Accept" and "Cancel" buttons.
3) Request for OAuth certification: After Agreement to policy and Authorization is done, request OAuth certi-

fication to Backend proxy by sending the user's ID and password at Resource Server. If certification is failed, repeat this process.

4) Save Access Key: Save access key that received from "Transmission of Access Token" process.
5) Show main menu: After Save Access Key process, show Main menu screen.
6) Cancel first log-in: If the user selects "Cancel" button in First Log-in screen, go back to Before log-in screen

C. Application) Monthly Diary screen

1) Monthly diary screen (Main screen): Show Menu button for selection of "Log out" and "Configuration". Show calendar of the chosen month and a Button for choosing a year/month to see a calendar (Default value is the current month in the device's system). Dates that the user wrote his/her diary are checked with circle mark so that the user can notice that he/she wrote a diary on those days (Diary from text input has a different color of circle mark with diary from voice record).
2) Preview of daily diary screen: If the user selects a date that checked with circle mark from Main screen, show score of the day (1 5). Also, if the diary's format is text, show part of the diary. If the diary's format is voice record file, show file name. When the user selects a date that not checked with circle mark, print message "There is no diary". Default value is 1st day of the month.
3) Button for choosing a year: If the user selects this button, show years (before 4-year after 4 years from the current year in device's system) and "Go back" button. The user can select a year, and the screen shows a list of months in the year. The user can see the previous nine years by selecting a left arrow button, and the next nine years by selecting a right arrow button. These arrow buttons are displayed at the bottom of the screen. Selecting "Go back" button will show the previous menu.
4) Button for choosing a month: After selecting a year, show buttons with numbers from 1 to 12, and "Go back" button. If the user selects a month, show Monthly diary screen of selected month/year.
5) Button for previous/next month: There are two buttons attached at a calendar that can switch the calendar to the previous/next one: button for the previous month (<) and next month (>). When the current calendar is January, (<) button will switch the calendar to December last year. If the current calendar is December, (>) button will switch the calendar to January next year.
6) Retrieve data of monthly diary: Include selected year and month value in object and request diary in the corresponding period. If there is no input for year and month, include year and month of currently accessed data in object and request backend proxy.
7) Receive data of monthly diary: Optimize retrieved diary data from Monthly diary transmission for the application's UI and show it.

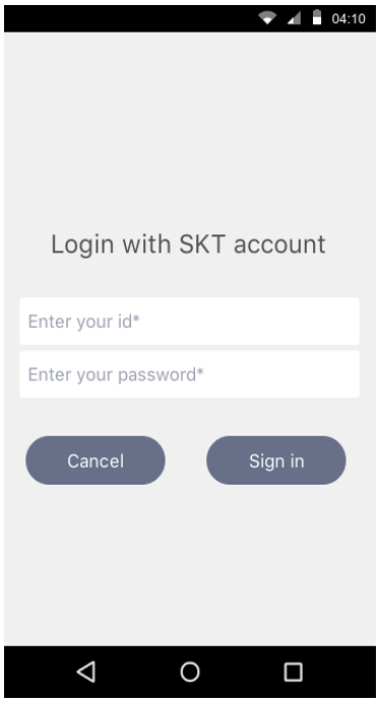

Figure III-B.1. Signing in SKT account

Figure III-B.2. Terms, pressing 'more detail'

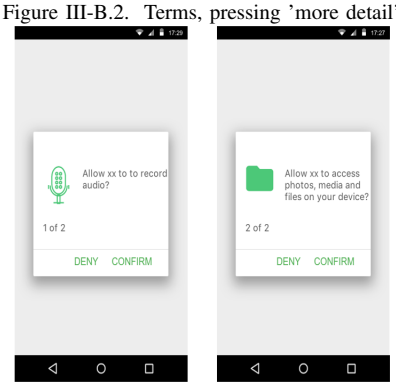

Figure III-B.3. Permission

Figure III-C.1. Monthly Log

Figure III-C.2. Pressed 'JAN 18' in Fig III-C.1

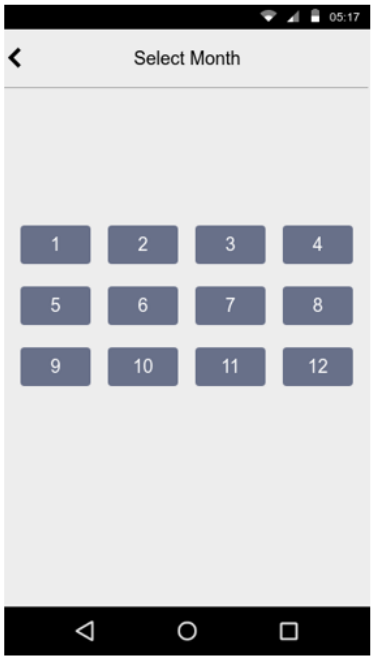

Figure III-C.3. Pressed a year in Fig III-C.2

D. Application) Daily diary screen

1) Writing a diary: When user select Preview of daily diary screen and if there is no diary, show "Write in text", "Write in voice", "Go back" button.
2) Look up a diary: When user select Preview of daily diary screen and if the diary is created as text, show Text diary screen. If the diary is created in voice, show Voice diary screen.
3) Text diary screen: Show score of the date, contents of the diary and "Delete", "Convert to voice diary" buttons. If a user touches the text diary screen, show Writing a text diary screen containing its score and contents.
4) Voice diary screen: Show score of the date, a button with the name of the recorded file and "X", "Delete", "Convert to text diary" buttons.
5) Button for playing recorded file: If the user selects a button with the name of the recorded file, load file by executing Recorded file transmission and play the file. If the button is selected during play, pause it. After playing the entire file, set playtime to 00:00 and stop it.
6) Delete recorded file of voice diary: If the user selects "X" button next to filename button, show pop-up window containing message "Existing file will not be recovered. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete recorded file and show Writing a voice diary screen. If "Cancel" button is selected, maintain the original state.
7) Convert to voice diary: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Writing a voice diary screen. If "Cancel" button is selected, maintain the original state.
8) Convert to text diary: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Writing a text diary screen. If "Cancel" button is selected, maintain the original state.
9) "Delete" button: If the button is selected, show a pop-up window containing a message "Existing diary's contents will be deleted. Will you delete it?" and "Confirm", "Cancel" buttons. If "Confirm" button is selected, execute Delete daily diary and show Monthly diary screen. If "Cancel" button is selected, maintain the original state.
10) "Go back" button: If the button is selected, show the recently shown screen.
11) Request for authorization: If the user selected "Cancel" button at Agreement to policy and Authorization and select "Look up the diary", show pop-up window containing message "You can't write a diary without authority" and "Confirm" button.
12) Pop-up window for authorization: If the user selects

"Confirm" button at Request for authorization, show pop-up window for requesting missing authorization for the application.

13) Referring daily diary data: Include selected date in the request object and request diary of the date to backend proxy
14) Receiving daily dairy date: Optimize the diary from Referring daily diary data for the application's UI and show it.

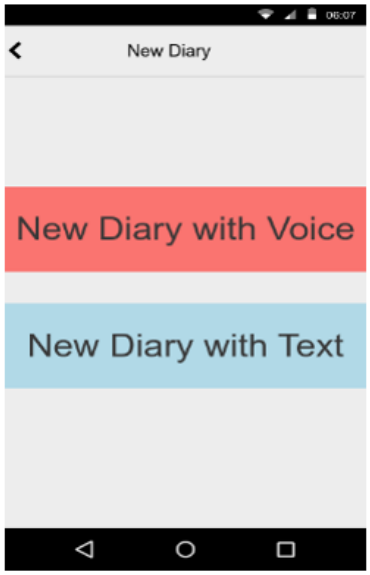

Figure III-D.1. Write a new log

Figure III-D.2. Log written in wave file

E. Application) Writing a diary screen

1) "Write in text" button: If button is selected, show Writing a text diary screen.
2) "Write in voice" button: If button is selected, show Writing a voice diary screen.
3) "Go back" button: If button is selected, go back to Monthly diary screen.

Figure III-E.1. Writing a text log

F. Application) Writing a text diary screen

1) Date: Show the current year/month/day.
2) Score contents of diary: Show radio box for selection of today's score and box for writing a diary.
3) "Confirm" button: When the user selects "Confirm" button, check whether a daily score is selected or not. If the score is not selected, show pop-up message "Please select daily score.". If the score is selected, including score and contents of the diary in requesting object, send the request to the server, and execute Create daily diary. After that, show Text diary screen of the diary.
4) "Cancel" button: When the user selects "Cancel" button, show pop-up message "Changes will not be saved. Will you cancel it?" and "Yes", "No" button. If "Yes" button is selected, go back to Monthly diary screen. If "No" is selected, maintain the original state.

G. Application) Writing a voice diary screen

1) Date: Show current year/month/day
2) Record/Stop: When the application is not recording a diary, show "Record" button and "Push this button to record your diary" message. Start recording if the button is pressed. When the application is recording a

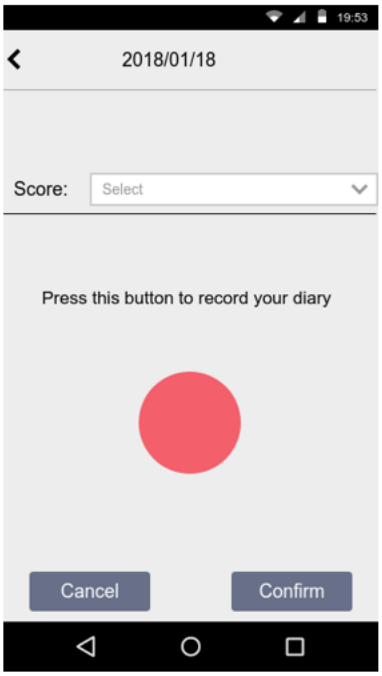

Figure III-F.1. Writing a voice log

diary, then the button changes into "Stop" button. When the user press "Stop" button, the application finishes recording and show a pop-up window for deciding the name of the recorded file and “Confirm”, “Cancel” button.

3) Screen after saving recorded file: After record at Record/Stop, the user can change the file’s name to the user’s input. If “Confirm” button is selected, change the file’s name, create a file to .wav format, and save file through Save recorded file execution. After these processes, show Writing a voice diary screen. Also, show a button with the file name and “X” button instead of “Record/Stop” button. If “Cancel” button is selected, the file will not be saved and show Writing a voice diary screen.
4) Delete recorded file/activate Record/Stop: If “X” button in Screen after saving recorded file is selected, delete recorded file name from the screen and execute Delete recorded file. After that, show Record/Stop button.
5) Saving diary and transmission: If “Confirm” button in Writing a voice diary screen is selected but the score is not selected, show a pop-up message “Please select daily score”. If the score is selected, include recorded file and score in request object, send it to backend proxy and execute Create daily diary. After that, show Voice diary screen. (Saving in server)
6) “Cancel” button: If “Cancel” button is selected, show pop-up window containing the message “If there is a recorded file, it will be deleted. Will you cancel?” and “Yes”, “No” buttons. If “Yes” button is selected, execute Delete recorded file and show Monthly diary screen. If “No” button is selected, maintain the current window.

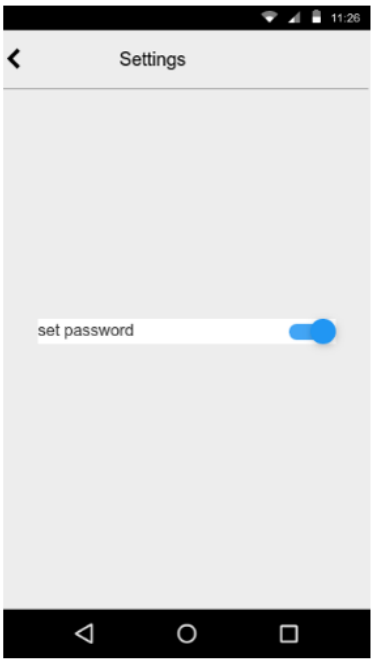

Figure III-G.1. Activating/Deactivating password

H. Application) Configuration

1) Menu screen: If a user selects “Configuration” button in main screen, show Password setting menu option which can decide whether to activate Lock screen function or not.
2) Password setting menu: If the user selects Password setting menu option in Menu screen and password is already set, deactivate Lock screen function. If a password is not set, show Password setting screen.
3) Password setting screen: Get 4-digit input from the user. After the first input, ask the user to input one more time to check. If the second input is different from the first one, show a message “It is different from the first input password” and delete the user’s input. If the second input is corresponding with the first one, set a password to the 4-digit input.
4) “Go back” button: If the button is selected, show the previous screen.

I. Application) Request to backend proxy

1) Header setting: If a request to backend proxy occurs, transmit variable to the header that can identify the subject of the request is “Application”.

J. Application) Application error handling

1) Network error: If the network is unavailable, show a pop-up window containing the message “Sorry. The network connection is unavailable now. Please check the network connection setting and try again.” and

"Confirm" button. If "Confirm" button is selected, show the previous screen.

2) Backend proxy connection error: If Backend proxy connection is unavailable, show a pop-up window containing message "Sorry. The server is unavailable now. Please try again later." and "Confirm" button. If "Confirm" button is selected, show the previous screen

K. NUGU Speaker) Account

1) Account registration/change/delete: A speaker can register/change/delete an account that is currently linked or will be linked to it through NUGU application.
2) Account information transmission: Whenever NUGU speaker requests to backend proxy, it transmits Access Token which is assigned to 'context.session.accessToken' of HTTP Header.

L. NUGU Speaker) NLP (Natural Language Processing)

1) Text processing: Perform domain/intent classification of text from Data forwarding – voice recognition. The developer can predefine the system of domain/intent classification using NUGU play-kit
2) Data forwarding – NLP: Forward data result of Text processing to DM.

M. NUGU Speaker) DM (Dialog Manager)

1) Conversation – Look up a diary: User can look up his/her diary within a year through the speaker. When the speaker recognizes that the user tries this conversation, it includes user account information which the speaker transfer in request object and requests Referring daily diary to backend proxy. If a diary exists, the speaker forwards the score and contents of the diary to NLG. If a diary does not exist, the speaker requests NLG to make a proper response such as "There is no diary in (date)". ex) User: "Please show me a diary on December 1st." Speaker: "Let me read a diary on December 1st. (Read the diary)" or "A diary of December 1st does not exist."
2) Conversation – Weekly score: A user can look up the modal value of the score during a week (Weekly score). When the speaker recognizes that the user tries this conversation, it forwards the result of "the modal value of daily score" to NLG. If data does not exist, the speaker requests NLG to make a proper response such as "Calculation is not finished, or there is no diary in last week. Please try again on Sunday." ex) User: "Please tell me the weekly score of last week." Speaker: "Weekly score of last week is (1 5)." Or "Calculation is not finished, or there is no diary last week. Please try again on Sunday."
3) Conversation - Product recommendation: User can look up products which are recommended based on the diaries of last week. When the speaker recognizes that the user tries this conversation, it forwards the result of "Product recommendation" to NLG. If data does not exist, the speaker requests NLG to make a proper response such as "Calculation is not finished, or there is no diary in last week. Please try again on Sunday.". ex) User: "Please recommend a product based on diaries of last week" Speaker: "Recommended product of this week is (product) of (company)." or "Calculation is not finished, or there is no diary last week. Please try again on Sunday."
4) Conversation – Weekly score Product recommendation: User can use both functions Conversation – Weekly score and Conversation – Product recommendation at the same time. When the speaker recognizes that the user tries this conversation, it forwards the result of Calculating of weekly score and Product recommendation to NLG. If the data does not exist, the speaker requests NLG to make a proper response such as "Calculation is not finished, or there is no diary in last week. Please try again on Sunday.". If the result of Calculating of weekly score only exists, the speaker requests NLG to make a proper response such as "Weekly score is (1 5), but product recommendation is not available now.". If the result of Product recommendation only exists, the speaker requests NLG to make a proper response such as "Recommended product of this week is , but weekly score is not available now.".
   ex) User: "Please tell me weekly score and recommended products." Speaker: "Weekly score of last week is 4, and the recommended product is (product) of (company)." or "Calculation is not finished, or there is no diary in last week. Please try again on Sunday."
5) Sentence aliasing: If a sentence from the user's speech is contextually the same with a sentence which developer embedded on the speaker, the speaker can make proper sentence-corresponding response for the speech.
6) Request for essential entity: Speaker re-ask essential entity to user for performing user's request.

N. Error handling response

1) Speech recognition failure: Make a proper response such as "Sorry, it is too difficult content to me.".
2) Backend proxy connection failure: Make a proper response such as "Sorry. Network connection is unavailable now. Please try later.".

O. Data forwarding – DM

Speaker forwards all responses from DM to NLG.

P. Speaker) NLG

1) Response conversion: Speaker converts all responses that NLG received to text-format sentences which has preset tone manner. This can be set from the 'Response Filters' menu in the NUGU play-kit.
2) Data forwarding – NLG: Speaker forwards sentence from Response conversion.

Q. Speaker) TTS

1) TTS execution: Speaker divides the requested sentences by syllables through the G2P process supported by the NUGU speaker, and defines the sections to rest for a while, how to pronounce them, etc. It also defines nonverbal elements such as pitch, tone, energy, and duration into syllables in sentences through the Proxy Model after the G2P process. The information defined in these two courses is combined to complete a voice that will respond to the user through the Wave form Generation process.
2) Response: Speaker outputs the result of TTS execution.

R. Emotion analysis model for sentence

1) Learning: We implement a sentence-emotional analysis model through TextBlob, the Python NLP library. However, since TextBlob cannot perform analysis of Hangul morpheme, we use the library of Python Hangul Morphology Analyzer to perform the analysis of Hangul shapeshifters
2) Emotion analysis: Using input sentence, it returns “Positive” or “Negative” value which indicates emotion.

```
from textblob.classifiers import
   NaiveBayesClassifier
from konlpy.tag import Mecab
pos_tagger = Mecab()
train = [
        ("I felt good since I met
           Hyun-sik." , 'Positive'),
        ("I had a hard time today
           since I ate the food gone
           bad", 'Negative')
        ]
train_data = pos_tagger( Every
   sentence in the 'train' )
model = NaiveBayesClassifier(
   train_data)   // end of training
   data (requirement 1 of this
   section)
       test = [
             ("I felt good since I
                 met Hyun-sik.", '
                Positive')
             ("I had a hard time
                today since I ate
                the food gone bad
                ", 'Negative')
             ]
         model.classify(test) // end
            of analysis (requirement 1
             of this section)
```

S. Product Entity Extraction

We perform the analysis of Hangul morpheme in Korean through the Python Korean Morphology Analyzer Library(), and record the number of Entities belonging to the "list of products" among the elements divided through this process. Ex) “I drank beer today” -> Both “beer” and “today” appeared, but “today” is not a product. Only “beer” entity must be extracted.

```
from konlpy.tag import Mecab
pos_tagger = Mecab()
sentence = [
        ( Because  of the glass of beer,
            I could make today worthy.
          )
        ( Today 's beer tastes great.
          )
        ]
words = pos_tagger( every elements in the
       sentence   )
product_list = consists of every product
investigate every elements of    words
   :
        if theres a    word    which
           is an element of
              product_list
record that    word    appeared once
```

T. Product recommendation model

Based on the word entered, the model obtains the product name, price, and location information that is recommended to users through the crawling of 11th Street site. ex. The word entered: Beer / Product: Model gets information of the product name, price, and place of purchase of the beer with the highest volume of sales or new.

U. Inside of Backend Proxy

1) OAuth certification: Using Authorization Server from SKT, perform OAuth certification which the way of Authorization Code Grant. If certification is success, perform Access Token Transmission. If certification is failed, return message “Authentication failed. Please re-enter your ID and password please”.
2) Access Token transmission: It transmits Access token to the application.

```
@app.route(   /get_access_key,
   method=[   GET   ]):
Do Auth certification()     //like
   picture above
if Auth certification completed:
Access_Key = get Access key from
   Authorization server
return Access_Key  // Access Token

          else:
             return  Authentication
                 failed. Please re-
```

```
                enter your ID and
                password please
```

3) Monthly diary data transmission: It looks up diaries of year/month in request object using "find" function in MongoDB. Transmit the data to application/speaker.

```
@app.route( /monthly_diaries,
   method=[ GET ]):
client = identify request endpoint(
   app or speaker)
year = year that client requested
month = month that client requested
userId = ID of a user that requested
   through client
collection( diaries ).find(year:
   year, month: month, userId: userId
   }, callback(data){
        res.send(data)
})
```

4) Daily diary data transmission: It looks up diaries of year/month in request object to application/speaker using "findOne" function in MongoDB. Transmit the diary's daily score and contents to the application/speaker. If the requested diary does not exist, backend proxy transmits message text "Cannot find the diary" to the application/speaker.

```
@app.route( /daily_diary, method
   =[ GET ]):
client = identify request endpoint(
   app or speaker)
date = date that client requested
userId = ID of a user that requested
   through client
collection( diaries ).find({date:
    date, userId: userId}, callback(
   data){
if data !=        then
res.send({score: data.score, contents
   : data.conetents})
           else
res.send( Cannot find the diary )
})
```

5) Create daily diary: It creates diary using account information, date and diary contents in request object using "insertOne" function in MongoDB.

```
@app.route( /daily_diary, method
   =[ POST ]):
date = date that the user requested
userId = ID of the user
score = score that the user requested
    to be saved into the new diary
contents = contents that the user
   requested to be inserted into the
   new diary (text or wav)
collection( diaries ).insertOne(
   date, userId, score, contents)
return
```

6) Revise daily diary: Revise requested diary's contents from application to contents in request object using "findOneAndUpdate" function in MongoDB.

```
@app.route( /daily_diary, method
   =[ PUT ]):
date = date that the user requested
userId = ID of the user
score = score that the user requested
    to be saved into the new diary
contents = contents that the user
   requested to change from previous
   contents
collection( diaries ).
   findOneAndUpdate({ $and: {date:
   date, userId: userId}}, {score:
   score, contents: contents})
return
```

7) Delete daily diary: Delete diary of requested date using "deleteOne" function in MongoDB.

```
@app.route( /daily_diary, method
   =[ DELETE ]):
client = identify request endpoint(
   app or speaker)
date = date that client requested
userId = ID of a user that requested
   through client
collection( diaries ).deleteOne({
    $and: {date: date, userId: userId
   })
return
```

8) Save recorded file: Save requested record file with date information in Backend proxy using "boto3" module in flask, "s3" in AWS and "insertOne" function in MongoDB.

```
@app.route( /record_diary,
   method=[ POST ]):
s3.upload_fileobj( file, bucket_name)
date = date that client requested
userId = ID of a user that requested
   through client
collection( diaries ).insertOne({
   date: date, userId: userId, url:
    https ://bucketname.s3.amazonaws
   .com/filename.extension })
return
```

9) Recorded file transmission: Transmit requested recorded file to application/speaker using "findOne" function in MongoDB and "s3" in AWS. If requested recorded file does not exist, transmit message text "Cannot find the record" to application/speaker.

```
@app.route( /record_diary,
   method=[ GET ]):
client = identify request endpoint(
   app or speaker)
date = date that client requested
userId = ID of the user
record = collection( diaries ).
   findOne( {date: date, userId:
   userId}, function(data){

if data != then
res.send(data[0].url)
else
res.send( Cannot find the record
   )
```

10) Delete recorded file: Delete a requested recorded file from backend proxy using "deleteOne" function in MongoDB and "boto" library in flask.

```
@app.route( /record_diary,
   method=[ DELETE ]):
date = date that the user requested
userId = ID of the user
collection( diaries ).deleteOne({
   date: date, userId: userId},
   function(data){
b = bucket which contains the record
   file
k.key = data.url
b.delete_key(k)
})
return
```

11) Load embedded model: Load model embedded in backend proxy (Emotion analysis model, entity extraction model, product recommendation model) using "import" in flask. : Perform the function every time and interval specified by the developer using Lambda service in AWS.
12) Recorded file -> text conversion: Convert speech in .wav recorded file to text using transcribe service in AWS.
13) Distinguishing requisition principal: Distinguish requisition principal which called backend proxy (Application, speaker, etc.) using header from Header setting.
14) Backend proxy error handling: If error occurs while backend proxy is running, record the type of the error and occurrence time using print(file=sys.stderr) and Cloud Watch in AWS.

V. Weekly analysis in backend proxy (Functions that executed every Saturday 2:00 AM. Functions are executed sequentially.)

```
userInfo = {userId: Id value of of
   each user}
// this object stores information on
   a user whose ID is corresponds to
    userInfo .userId and is saved
    in MongoDB
diary = {date, userId, contents,
   score}
// this object stores information on
   a diary whose ID is corresponds to
      userId and date is
   corresponding to date .
   contents: contents of the diary,
   score: score of the diary. This
   object is saved in MongoDB
```

1) Analysis product entity by score: Forward contents of diaries in a week of each user to product entity extraction model and execute "Product" entity extraction. The result of analysis is saved by the user's account and score using functions in MongoDB.

```
model = Product-Entity-Extract model
date = Sunday of current week
userInfoList = consists of
      userInfo table for each
   users.
userInfo = First element of
   userInfoList
for userInfo reaches end of
      userInfoList
while date becomes Saturday
diary = MongoDB.findDiary(date,
   userInfo.userId)
result = model.analyze(diary.contents
   )
upsert ScoreEntitySet which
   saves {score: x , entity:
    entity name } pair into
   userinfo.
date++
           date = date of Sunday of
              current week
move on to next userinfo
```

2) Daily emotion analysis: Forward contents of diaries in a week of each user to the emotion analysis model and execute Emotion analysis. The result of the analysis of each diary is stored by the user's account and score

```
model = Emotion-Detect-Model
date = Sunday of current week
userInfoList = consists of
      userInfo table for each
   users.
userInfo = First element of
   userInfoList
for userInfo reaches end of
      userInfoList
while(date becomes Saturday)
```

```
        diary = diary object that
           corresponds to  userInfo
           .userId and   date
Append result of  model .analyze(
   diary.contents) to the
      diary
date++
move on to next   userId
```

3) Calculation of the weekly score: Calculate modal value of daily score in a week of each user. However, daily score which is contradictory to the result of Daily emotion analysis will be excluded. For example, if score of a daily diary is higher than 3, but the result of Daily emotion analysis is "Negative", the data is excluded. Also, if score of a daily diary is lower than 3, but the result of Daily emotion analysis is "Positive", the data is also excluded. The result of modal value is stored by each use

```
 rmodel = Emotion-Detect-Model
date = Sunday of current week
userInfoList = consists of
      userInfo   table for each
   users.
userInfo = First element of
   userInfoList
mostFrequentScore = {  1  : 0,
      2  : 0,   3  : 0,   4  :
   0,   5  : 0}
for   userInfo   reaches end of
      userInfoList
while(date becomes Saturday)
diary = diary object that corresponds
    to  userInfo .userId and
      date
if (diary.score >= 3 && diary.emotion
    =   bad  ) || (diary.score < 3
   && diary.emotion =   good  )
   then
        move on to next date
else
mostFrequentScore.toText(diary.score)
   ++
        date++
frequentScore = save the most
   frequent score that appears in
   diaries of a user
upsert { frequentScore:
      frequentScore  } into
      userInfo
reset   mostFrequentScore
move on to next   userInfo
```

4) Saving recommended product information: Load each user's result of Calculation of the weekly score. Load product entity from Analysis product entity by a score corresponding to each user's weekly score. Forward each product entity to the Product recommendation model and execute Product recommendation. The result is stored by each user.

```
model = Product-Recommendation-Model
userInfoList = consists of
      userInfo   table for each
   users.
userInfo = First element of
   userInfoList
for   userInfo   reaches end of
      userInfoList
entities = load list of   entity
   in   ScoreEntitySet   that
   matches   frequentScore   of a
   user
           upsert  model .analyze(
              entities)   into
                 userInfo
    move on to next   userInfo
```

IV. Development Environment

A. Development environment

1) Flask framework of Python in Linux(Ubuntu 18.04): To implement backend proxy of Linux and integrate models. Many machine learning models are implemented in Python. And Flask is one of the most popular web framework for Python. Ubuntu 18.04 is easy to use in Windows through 'WSL (Windows Subsystem for Linux)' and the latest version.
2) Serverless framework: To easily deploy flask application to AWS lambda. For beginners, it is hard to use AWS lambda directly. This framework eases use of AWS lambda. and to
3) AWS lambda: To save Flask application. AWS free tier does not provide 'VPC peering' which means anyone who knows IP address of the server can access our server. So we should choose serverless. AWS Lambda is one of the most popular service for serverless system.
4) AWS S3: To save audio file. The only way AWS save audio files.
5) AWS Cloud Formation: To check errors of flask web framework
6) MongoDB: To save user data and url for each audio files. MongoDB handles data in json format which is easy to manipulate for beginners.
7) MongoAtlas: Cloud database service for MongoDB
8) React-Native: To implement application for both Android and iOS
9) NUGU play-kit: To handle NUGU speaker (connect to Flask framework, setting NLP for NUGU speaker and so on)

B. Software

1) KoNLPy: Most popular Korean NLP that can be implemented using Python (https://konlpy-

ko.readthedocs.io/ko/v0.4.3/)

2) Surprise Library: Product Recommendation model available in Python (http://surpriselib.com/)
3) Naive Bayesian Classification model of TextBlob(https://textblob.readthedocs.io/en/dev/) library: To analyze emotion of the sentence. It is also available in Python.

REFERENCES

[1] "'주인님이 제일 멋져요!"... 미래 AI, 정답보다 공감에 초점," 한국일보, 07-May-2018. [Online]. Available:https://www.hankookilbo.com/News/Read/201805071591384777. [Accessed: 01-Oct-2019].

[2] Bonseung Koo, Sookyung Baek, Hyoungnyoun Kim and Kwangsu Cho "Personalization of Conversational Agent Influences Users' Intimacy and Information Reliability," 한국HCI학회 학술대회, Jan-2016. 한국HCI학회 학술대회, vol. 15, no. 144, pp. 456-458,Jan.2016.[Online]. Available:http://www.dbpia.co.kr/journal/articleDetail?nodeId=NODE06645321/ [Accessed: 30-Sep-2019].