

Taxi Sharing – YA!TA!

Saroho

Kim Munui 2015005323
College of engineering,
Hanyang University
Dept. of Information system
Seoul, Korea
ansml1024@naver.com

Kim Sohee 2015005350
College of engineering,
Hanyang University
Dept. of Information system
Seoul, Korea
sh960726@naver.com

Park Saro 2015005469
College of engineering,
Hanyang University
Dept. of Information system
Seoul, Korea
salo3678@naver.com

Abstract— Our team is trying to develop an application that will allow students in the vicinity of Hanyang University to take a taxi together in order to accommodate students. In fact, we realized that there are a lot of commute students who live near Hanyang University, and that they pay a certain amount of transport traffic cost to attend school. The application not only provides matching, but also provides users with timetables, chat rooms, alarms, and easy payments. Our team matches the customers who have chosen the same course at the same time, and makes an environment where they can take taxis together. It provides a platform to match customers who use our application with one another and to communicate with each other, so it is cheaper and easier to get a taxi together than other public transportation when student go to school and this application will provide the cost-effective and time-saving convenience of the students.

Role Assignments

Roles	Name	Task description and etc.
User/Customer	Kim Munui	The students from Hanyang University who live in school near school are chosen as the main application customers. Through the questionnaire, we gather basic information about users for the initial application development. Then, she investigates the major public transportations that users ride to go school. In order to select a course to be provided by the application, it is necessary to investigate the places where the users are located and determine the starting point.

Software developer	Park Saro	Software developer thinks about the software system in general. She analyzes, categorizes the related software and examines the software that can be used to implement the application. She will investigate about what development tool we will use to implement.
Development manager	Kim Sohee	Development manager should confirm if the functions of software fulfill users' need. When they do not satisfy users, she gathers some information from users and customers. Then, she advises software developer to update it and mediate both of them.

I. INTRODUCTION

Motivation

We focused on taxis sharing as the means of transportation to school, and created this application. Students use various means of transportation, including taxis, buses and subways to go to university. Especially Hanyang Universities have a huge campus, so students mention the difficulties and inconveniences to attend university. Furthermore, there is a certain distance from the bus platform to university. Also, the distance between a lot of stations (except for Hanyang University station) and the school are too far to walk, Students transfer to buses again or run to school not late to class.

Although Taxis are the most convenient means of transportation for students, taxi is a very expensive means of transportation for students who go to university every day. So that many students cannot use taxi, they even also know taxi is most convenient transportation. By the way, we have surprisingly found that students who live around the school often use taxis to attend school, not go to walk. Students living around the school pay a basic fare of around won and often take a taxi to school. There are lots of students who use taxis to go to school around school if they are late because students think taxi base fare are relatively cheap to spend per day. student who live near a taxi's base fare have little financial burden of occasional use of a taxi, so they prefer to take a taxi in unavoidable circumstances. For that reason, we thought that it would be good for the students to provide a taxi sharing ride service for students who want to go to school by more cheaper transportation cost.

If someone who are a student living in the nearby Hanyang University, or if someone get off at a nearby station, they may be able to use a taxi to go to school as easily method. In this situation, if people take a taxi with other students who have the same destination, student can save money. If students take a taxi around the university, they will have to pay a taxi fare of around 3000~5000 won. But if student take taxi together, they can divide the cost. So one student have to pay the taxi cost at a low cost of 1000-1500 won, which is similar cost to other public transportation buses and subway fare. This application will match students who want to use a taxi around the school using low cost. This application help students to get a taxi ride together more safe and easy. We surveyed the major residential areas of students who lived around Hanyang University through questionnaires that were created for the main system of this application. Since Hanyang University's major occupied area are large in scope, we have selected lots of specific residential areas. In addition, we analyzed the main starting place for these students to take taxis. Because of the large circulation of vehicles around the major roads and major locations of the student-resident areas, students realized that the actual residence starts from a variety of similar locations, which was used to set the starting point for this application. All of the destinations of this application have been designated as Hanyang University. This application can help user quickly get a taxi ride with a quick matching system. Even if there is a burden on the cost side, there are many reasons why student used to go to school by taking taxi. When student late for class, user use this application, so we make this application to user can get on the taxi quickly in a short time. Also, when students enroll their information, they have to receive the university certification process so user can safely board taxi with others. This application is available through a simple enroll phase and login. User also can use the chat features in this application to view the matching party's information and send and receive required messages. We have created this application with a focus on providing a system that allows users to easily and quickly take a cab ride system.

Problem statement (client's needs)

- The rate of students staying vicinity of school is about 52.6% because the dorm of university cannot accommodate a lot.
- Although the students live near school, when they go to school by public transportation or on foot in the morning, it takes 40 minutes in average. This is so ambiguous.
- In Hanyang University, there is no shuttle bus on campus and the bus stops around the school are also inaccessible.
- Taking a taxi is very convenient compare to using other transportations and it takes people to the destination where they want very fast.
- Taxis are the most effective transportations in terms of time and convenience, but they are so expensive when using it alone.

- The taxi applications like 'Kakao Taxi' which is on the market are hard for local visitors. In addition, the students have difficulty in using the carpool application because they should specify a certain time and place to use. However, the students' time tables are different every day.

Research on any related software

A. Kakao Taxi

It is an application that connects taxi drivers and customers. Also, it prompts the user to enter the departure and destination locations, and simply input the taxi type and payment method. "Kakao Taxi" provides a platform for automatically paying taxi fares automatically when you register your card information in the application. In this application, I thought that the matching system that connects the taxi driver and the customer is similar to the application that our team envisions.

B. Poolus

The representative carpool application "Poolus" is linked to the SNS, so you can enter the phone number and the authentication number and subscribe through the email address. Within this application, the user can specify the origin and departure time destination. "Poolus" utilizes Google Maps, allowing users to click on a location on the screen to select it. This application can be paid through a domestic credit card or a check card when a user registers card information or uses a carpool when registering. If you register a place as a favorite, you can use a faster matching service, and you have a booking function that allows you to specify a time and save time for matching. When the matching is confirmed, a notification is sounded and a system is provided to send a relieved message to the acquaintance. In addition, drivers can be assessed and secured through the report / help menu for the driver to form a

carpool culture. Finally, it reconfirms the itinerary and helps to make payment by providing payment and receipt. It is an application that helps the driver to contact the driver again within 24 hours after getting off the bus, helps the user when there is a lost property, and provides convenience to the user who carries the car with the system in which the information is not recorded in 24 hours.

C. Hopsee

Hopsee which is a taxi ride made overseas is similar to the application that our team wants to implement. In this application, a simple subscription procedure is available. If you enter only the starting point, time, and destination after logging in, a chat window will be opened and a waiting person can help you get a taxi together.

D. LINE

The users can sign up for the app through mobile phone authentication or Facebook linkage. Line is a messenger that provides free messages, free calls. The line is available in 52 countries, including Japan, Thailand, Taiwan, and Spain. Users can enjoy international calls free of charge, and payment systems that are easy to use overseas are also available. Domestic line account users can register cards that can be used abroad. The card registration is configured so that the user can register the card one time in consideration of the convenience of the user. The line app also supports free chat, free voice & video calls, and mobile chat with PC.

E. Kakao Talk

It is one of the most popular applications to support both mobile and PC. Users can easily sign up for this app via phone verification. Users can send and receive messages with subscribed users, and send and receive messages to a variety of people, including common chat and group chats. Users can send and receive photos and videos from the chat screen, and can add a variety of services such as group calls, contact connections, and wire transfers.

F. Kakao Pay

The users set their frequent card and their own password. After filling the card information, they can easily use the system only by inputting the password from the next pay. The most useful function from this application is that it connects with 'Kakao Talk'. 'Kakao Talk' users do not have to sign up additionally and they can use gifticon function without downloading 'Kakao Pay'.

G. KT mobile pay

The users input their personal data like name, phone number, phone authentication through message. After all identifications, they can go to charge screen.

H. Toss

It helps users to transfer money to others without inputting others' account number or security card number. It only needs the user's own password and receiver's phone number.

I. KB App Card

The users set their KB card information and password like other applications. The most different part of this is it provides spot payment system. The users can pay in the store by showing the bar code of this app card. They don't need to keep their wallet.

II. REQUIREMENT ANALYSIS

A. Main Page

If the user downloads the application, the screen that explains the simple tutorial should be displayed first. It briefly explains the function of the application such as linkage with the time table, setting location, chatting room matching, payment system, etc. It concisely described in accordance with the order and this page can be skipped. Main page is the second screen when downloading this application. The main screen consists of the origin and destination, the time user want to match, a map showing the route briefly, and a 'create' button.

B. Sign Up

Membership requires basic information and a number field for phone number authentication and a nickname request. It consists of input screen. Log-in consists of field for inputting ID and password, searching ID or password function. Terms of use and features are available by clicking View Terms and Conditions on the Membership page. The subscription page allows user to easily sign up using only their nickname, mobile number, and password. This is to make it easier for the user to use by guiding the user to join more easily. Kakao Talk, Facebook, and Naver Log-in, Hanyang-in mail to download the member information to each server.

- a. Mobile phone number subscription Receive name (nickname) and cell phone number input. All items must be listed as required information. Upon successful membership, a message or information screen informs user that he/she has been successfully registered.
- b. Request an authorization number

1. Enter cell phone number and click the button for requesting an authentication number.
2. If the user clicks "Enter Authentication Number" and the key value is authenticated, the "Membership Success" screen will appear.
3. When the user clicks the Request Authentication Number button and send the authentication number, the button is changed into the confirmation of the authentication number and the authentication number retransmission.
4. If the user clicks the authentication number retransmission button, the authentication number will be resent.

c. Registration of profile image

1. Clicking the register button pops up the profile image registration pop-up.
2. Click 'Select Album' to select a photo from the user's mobile phone gallery. Click the OK button in the gallery and the image will be registered as a profile image.
3. If the user chooses 'take a picture', he/she can take pictures with their own phone.

d. Error message after clicking the member button

1. It is not activated when the item 1 is not written.
2. If the authentication number is duplicated, 3 error messages will be displayed.
3. Membership is completed only when all items are entered and the cell phone number is authenticated.

C. Log-in

Log-in page has a message that is invoked when a user clicks on a page that only members can access.

- a. Mobile phone number login - Membership registration must be number authentication. (Self-certification required)
- b. Automatic Log-in Default value: checked (selectable)
- c. Find password: Go to the password search page.
- d. Sign up: Go to the member registration page.

D. My Page

There is 'My Page' screen which shows members' information. It is shown only when users log-in. Users can change their own pictures which they set in the sign-up page. It will be useful on matching for recognizing each other. They can also change their password in this page, too. For an option, the users can connect this application with their own time table. If they connect, they can choose their preferred taxi time by day and we can provide a pre-alarm like 'The class is 30 minutes left! We recommend taking taxi for arriving on time!'. The users can fluidly decide everyday if they will take taxi or not according to their circumstances.

E. Set and start the matching

The users can decide where they will take taxi and destination. Also, they should set when they will depart. We are also planning about reservation function. If users set their time into later time, it has the same meaning with reservation. This will make users reserve before they leave, before they go to sleep at night, and so on. We will give some options of locations and let them select it. We will decide the options based on our questionnaire later. People who have used our application before do not need to set their settings again because they are set into previous record. If users finish the settings and click the 'Start the matching' button, the server will match each other by locations and time.

F. Make the chat room

If the server finds people who meets with settings, it invites people in the chat room. The maximum number of people in the chat room is 4. When more people are found, it creates the new chat room. It finishes search. People can change their state into ready or exit. When only 2 or 3 people are found, but their states are all ready, then they can go. If not, the search is on-going. When the server terminates the search and there are 4 people in the chat room but if there is a person who does not change the state will get some penalties and automatically goes out.

G. In the chat room (chat, go out, penalty)

We provide a chat room service where users who take a taxi can communicate with each other. Restricted to a minimum of 2 people and a maximum of 4 people. Until all the people in the chat room are gathered, "Waiting for joiner" is displayed on the bottom screen, and if they are gathered, "Matching complete" status will be clicked. We plan to create a platform similar to the normal messenger service (Kakao Talk, Naver, Line). When the customer enters the chat screen and presses the "send" button, the dialog entered on the screen will be displayed chronologically through the server and will be made into a system that customers can share. If customer click on the other person's icon in the chat room, user will be able to see the brief information that he/she entered when signing up for their profile. If user do not like the match, create a "EXIT" button so that user can leave the chat room at any time. When they click this button, they double-check with "Are you sure?" And select either "Yes" or "No". And when they leave the chat room, they will return to their previous state and select "Join ride" to re-match and enter the chat room. When one of the chat room customers presses the "Ready" button on the screen, they are converted to ready to complete the match and other people in the chat room decide to complete the matching by pressing the "Ready" button within 5 minutes to do. When the status of the customers in the chat room is all "ready", the message "I hope you will accompany me happily" will be displayed. Otherwise, customers who have not pressed the "Ready" button in 5 minutes will be forced to leave the chat room.

H. Finish the matching / Get in the taxi

In this phase, our application receives the situation of user, whether they are ready to ride or not. When the users meet through the chat room, they will press the button to complete the matching. We check all users are "ready" situation, we show "complete" to check they take taxi together. In this phase, this button is used to check whether our application compete its task. If users click complete the matching system is done. We want to make sure that the time from the beginning to the end of the match is as fast as possible. We think of ways to make all users press the "complete" button in up to 10 minutes. Next phase is additional step. It processes regardless of car sharing system. Our team want to get user's feedback, so we apply some questionnaire page. For the sake of situation likes, users don't want to answer this questionnaire, we make "ignore" button. In the questionnaire, we asked simple question. First, the score of passengers. Second, the feeling of satisfaction with using this application. Last, we asked writing, if users have any additional request. These three questions will help to maintain and develop our application.

I. Additional page

We are going to receive an evaluation of the application users' use of the application through an additional page. This step is an additional page that will go through after the match is completed, and unwanted customers can skip this step. We will ask three simple questions in this step.

Q1) How much score would you give to the passenger who boarded the taxi?

Q2) Are you satisfied with this application?

Q3) Please write, if users have any additional request.

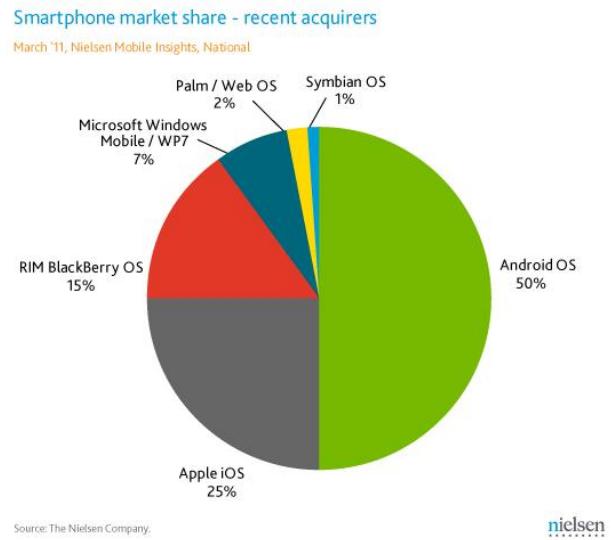
The above three questions will lead us to additional steps for system maintenance and performance enhancement. We plan to create an application so that you can receive answers to these questions in an interoperable email. Our application only matches the users to get the cabs together, so we will check that the users use the taxi and see if there are any additional inconveniences or problems. Users who do not want these extra steps will be able to quit the app or return to the first step.

III. DEVELOPMENT ENVIRONMENT

A. Choice of software development platform

We are going to develop applications on Windows using Linux based platform through a virtual machine and Mac OS. Android is the most famous mobile operating system, which has large users. Android is a mobile operating system developed by Google, based on the Linux Kernel and designed primarily for

touchscreen mobile devices such as smartphones and tablets. According to the chart below, we can know that market share of Android OS is about 50% and it is the most part. Also, there are lots of tutorials for guidelines of mobile applications. This is because our Android studio and open fire server are available for both Linux and Mac OS. Our team will use Android studio (Android studio 2.3.1 version) to develop these applications. The language used is Java (Java SE 8u131), MySQL and HTML will be used if the function to be linked with web.



Tool and language	Reason
JAVA	JAVA is a general language that is concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. JAVA is a base language used in Android studio.
MySQL	We need to create a database where we can save information about users when they sign-up. When users try to log-in, the server compares the information which they input and the information which is saved in the database. To look and manage the information easily, we choose MySQL as a database management tool.
PHP	PHP is a programming language which is used in servers. We will use PHP to develop between the server and web environment because we have learned it in another course, 'Web system programming 2'. Therefore, we think that it will be useful for developing our server part.
Google push	Google push is a technology that automatically and periodically pushes

	the latest information directly to the computer screen by simply selecting the list of information that the user is interested in using the Internet environment. The push technology is a way that the server automatically provides the information that the user wants. It is an automated technology that allows users to receive specific information without having to request it every time.
--	---

Cost estimation

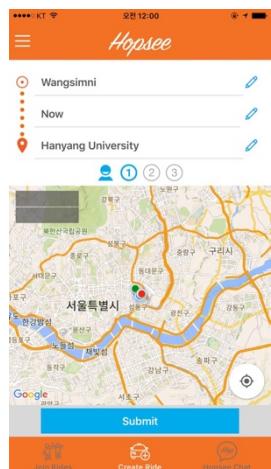
Device	Price (won)
Server (Filezilla)	10400

We will use open source primarily to develop application but we need server, so we pay some money for the server only.

B. Software in use

a. Hopsee

There is an application 'Hopsee' which is very similar with our application. Its purpose is sharing taxi. However, our application is more safe and we will receive feedbacks from users. Therefore, our application can be developed better in some ways.



b. Wifi technology

This application is available in both Wifi and LTE environments. In addition, since it is an application used in mobile phones, it is essential when Wifi devices are connected. The main advantage of a wireless network over a wired one is that users can move around freely within the area of the network with their laptops, handheld devices etc. and get an internet connection.

c. Database management system

Database management system is software that handles the storage, retrieval, and updating of data in a computer system. Database Management System or DBMS in short refers to the technology of storing and retrieving users' data with utmost efficiency along with appropriate security measures. This tutorial explains the basics of DBMS such as its architecture, data models, data schemas, data independence, E-R model, relation model, relational database design, and storage and file structure and much more. DBMS is a collection of inter-related data and set of programs to store & access those data in an easy and effective manner. Therefore, we are going to save information of users in database.

d. Android studio

The IDE itself is based off the very popular IntelliJ IDEA from JetBrains and is being offered by Google for free. On the other hand, Eclipse is more mature than ever and Google's ADT plugin which transforms the popular IDE into a fully featured Android developing environment has become very stable. Instead, lots of developers want to get associated with Android application because of incredible growth. Besides, Android Studio platform developers also use Eclipse to develop applications, but always thought of Eclipse like a "Student-Project IDE (Integrated Development Environment)" and learned about it.



e. Git hub

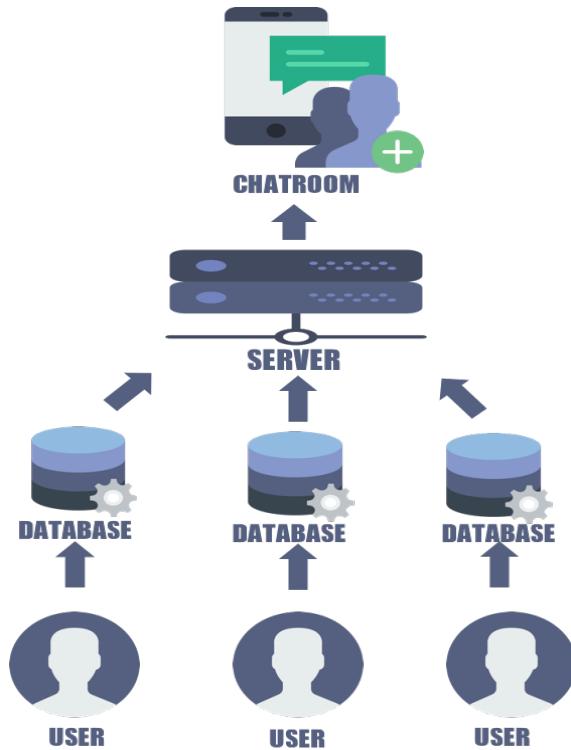
GitHub is a Git repository hosting service, but it adds many of its own features. Git is a command line tool and GitHub provides a Web-based graphical interface. It also provides access control and several collaboration features, such as a wikis and basic task management tools for every project. This will help us to co-work each other and see our own code more easily.

C. Task distribution

Name	Task
Kim Munui	Chat room page (Android Studio, Server)
Kim Sohee	Main page, My page (Android Studio)
Park Saro	Sign up, Log-in page (Android Studio, MySQL, Server)

IV. SPECIFICATIONS

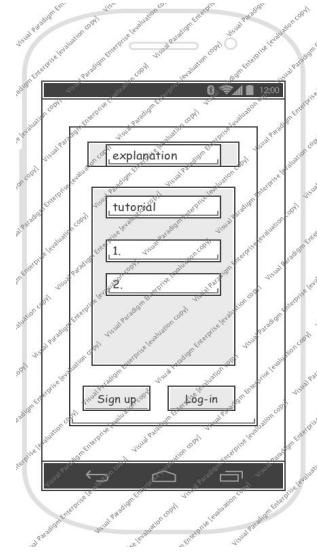
This picture shows our whole application. The information of users are saved in our database. They are saved in server and these data are based on our chat room settings.



A. Main Page

Main page is the first screen when downloading this application. If user run the application, the description of this application will be displayed on the main page first. After users read and understood this, they will see the main screen of the application. The main screen consists of the origin and destination, the time user want to match, a map showing the route briefly, and a 'create' button.

a. Tutorial of the application (Description)



If the application is executed, the description of this application will be displayed on the main page first, and if user read it thoroughly, click the OK button and the main screen of the application will appear. The description of the application includes the following contents.

1. Usage and purpose of the application

This page is the first page that comes up when user run the app. It helps user to get this app to accompany user, so user can lower the taxi fee and explain the purpose of the app to help user get to school.

2. Explain how to use this application

Describe how to use the application according to the order.

1) Set origin and destination

Select the desired departure and destination from the list.

2) Time when to match

Users can explain that they can match at any time

3) Map

After setting the departure and destination, the desired time for matching, the route appears on the map based on the route selected by the user on the map.

4) 'CREATE' button

Pressing this button completes the matching.

5) 'Sign Up' button and 'Log-in' button

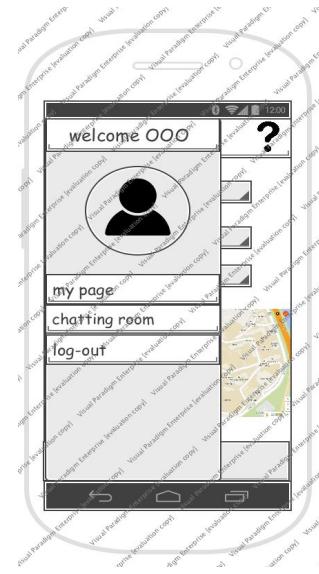
If the user has read this tutorial, there are a 'Sign up' button and a 'Log-in' button. The 'Sign up' button will go to the Sign Up page when a user who has not yet subscribed to the page clicks the 'Sign Up' button. If users press the 'Log-in' button, the existing user will go to the Log-in page to use the application.

b. Main page



The main page consists of menus and application names at the top as shown in the following figure. In the middle, it is composed of a starting point for matching, a destination setting, a desired time, and a map. At the bottom is a button called "create" that starts matching.

1. Menu



It contains several components of the application and user can navigate to the selected page by selecting the list. The menu consists of the following items and is linked to each item.

- (1) Chat room
- (2) My page
- (3) Log-out

2. Name and logo of the application



The name of the application is 'YATA!', and the logo is located at the top of the application.

3. Set departure and destination for matching

The origin and destination are based on the questionnaire written by the surrounding residents. The questionnaire consists of questions such as the place where the respondent resides, how to commute to school, the cost of commuting to school, difficulties of using tax. Analyze the written questionnaire and make it a route that students often use.

<Questionnaire>

The questionnaire consists of the following questions.

1. Where are you currently residing?

2. How long does it take to go to school?
 1. Within 10 minutes
 2. Within 20 minutes
 3. Within 30 minutes
 4. Within 40 minutes
 5. More than 50 minutes

3. What type of transportation do you use most often when you go to school?
 1. Bus
 2. Subway
 3. Taxi
 4. Walk

4. How much does it cost to attend school? (Monthly average)

5. What is the most convenient way to use a taxi?

6. What is the most uncomfortable thing about taxis?

<Conclusion>

The students who were near the school were also using public transportation to go to school and paid a certain portion of the transportation fee each month. Students also preferred taxis on several public transportations because they were the fastest and most comfortable, but costly. Therefore, it is expected that students will be able to solve the problem of time and money to attend school by selecting the area where the students reside according to the questionnaire results and matching the taxi companions.

<Result>

Q1)

Students residing in the vicinity of Hanyang University mainly resided in the vicinity of Shindap-dong, Dapsimni (25%), Wangsimni station (20%), Majang station (15%) and Sageundong (20%).

Q2)

within 30 minutes: 30 %, within 20 minutes and 40 minutes: 20%, within 10 minutes and 50 minutes: 15% respectively

Q3)

The most common means of transportation for students was no different and occupied a similar percentage.

Q4)

The average cost for students to go to school from 50000 won a month to 100 million won.

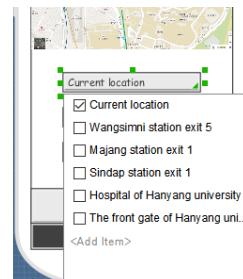
Q5)

The most convenient thing for students to use taxis was to be able to go to school quickly and to be comfortable.

Q6)

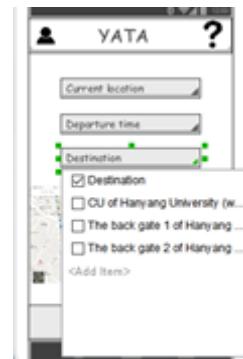
The most uncomfortable thing was that taxi fare was expensive for students.

(1) Departure List



- Wangsimini station exit 5
- Majang station exit 1
- Sindap station exit 1
- Hospital of Hanyang university
- The front gate of Hanyang university

(2) Destination List



- CU of Hanyang university
- The back gate 1 of Hanyang university
- The back gate 2 of Hanyang university

4. Time



The time is basically set to 'Select your starting time'.

5. Map



The map briefly shows the route user have selected and this route is selected via the questionnaire.

6. 'CREATE' button

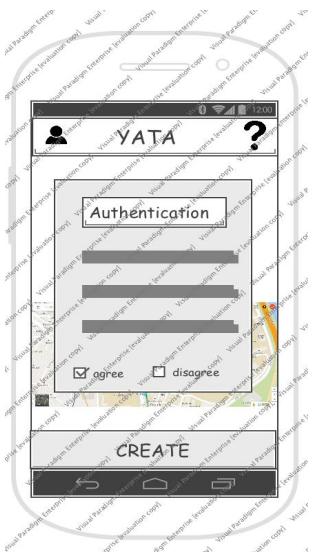


If user want to set the starting, destination, and matching time, click the 'CREATE' button. Pressing this button starts matching.

B. Sign Up

This page appears when an existing subscribed user has moved to use this app after reading the tutorial.

a. User Authentication



The user must authenticate his or her identity in order to use the application. Users can authenticate their identity through their mobile phone number, Naver, Facebook, and Hanyang-in mail. Users must read the Terms of Use and select the Agree or Disagree button before authenticating their identity.

1. Authentication

Users can authenticate their identity through a mobile phone number, Naver, Facebook login, and Hanyang in-mail. KakaoTalk, Facebook, and Naver login to download the member information to each server.

- (1) The user enters his cell phone number and clicks the Request Authentication Number button.
- (2) Click "Enter Authentication Number" and if the key value is authenticated, "Authentication Verification Success" screen appears.
- (3) When the user click the Request Verification Number button and send the verification number, the button changes to Verification and Confirm Verification.
- (4) If the user click the authentication number redirection button, the authentication number will be resent. When the authentication is completed, it goes to the member registration page.

b. Sign Up

information		Registration date 연도. 월. 일.
ID	<input type="text"/> check	
PASSWORD	at least 10 characters long.	
PASSWORD identification	Check your password.	
nickname		
profile image	<input type="button" value="파일 선택"/>	선택된 파일 없음
<input type="button" value="Registration"/> <input type="button" value="cancel"/>		

1. ID

<input type="text" value="ID"/>	<input type="button" value="check"/>
---------------------------------	--------------------------------------

If the user succeeds in authenticating his / her identity, the user moves to a page where he / she can register. The ID can be a combination of English and numbers, and no special characters are allowed. The maximum number of characters is 10 characters. After writing the desired ID, the user must check that his or her ID overlaps with the ID of another user. If it is duplicated, pop up screen "ID is already in use" to inform them that they need to select another ID.

2. Password

<input type="text" value="PASSWORD"/>	at least 10 characters long.
<input type="text" value="PASSWORD identification"/>	Check your password.

The password is a combination of English and numbers and can be up to 10 characters long. The password must be marked with asterisks on the screen when typing and the password must be entered to complete the password verification.

3. Nickname

nickname	<input type="text"/>
----------	----------------------

Users can set a nickname for this application. Passwords can be up to seven characters in length, and can be in English, numbers or special characters.

4. Profile image

profile image	<input type="button" value="파일 선택"/> 선택된 파일 없음
---------------	--

The user can register the profile image.

- (1) Clicking the registration button pops up a profile image registration popup screen.
- (2) Click ‘Select Album’ to select a photo from user’s phone gallery.
- (3) Asking if the user want to allow access to user’s gallery before selecting photos.
- (4) Allow users to access the gallery and click the OK button in the gallery to register the image as a profile image.
- (5) If the user chooses to take a picture, he/she can take a picture with user’s own mobile phone.
- (6) Photo sizes from 320px up to 900px are possible.

5. ‘Registration’ and ‘Cancel’ button

<input type="button" value="Registration"/>	<input type="button" value="cancel"/>
---	---------------------------------------

When the user completes the subscription items and presses the registration button, the following situation occurs.

- (1) If the user correctly filled out the membership items: Users are subscribed.
- (2) If the user does not enter some subscription items, if the user ID or password is incorrect, or if the nickname is not entered: A popup screen will be displayed.: ‘When you are finished, go to login page.’
- (3) Press the Cancel button to go to the main page.

c. How to implement

Kakao Talk, Facebook, and Naver login to download the member information to each server. The app page consists of a view and the developer completes the development of the data connection operation and transferring the subscription data to the database.

Main menu	Sub menu and functions	Explanation
Sign Up	Terms of Use	Terms and Conditions
	Sign Up	Phone number (ID) and password input, User Authentication mandatory, nickname
Log-in	Log-in screen	Log-in screen
	Email or ID Log-in	Phone Log-in
	Confirm the password	Phone number and Changing the password

1. App server back-end development

Based on the ERD created in the design stage, we work on the member DB. Membership Configure a web server that communicates data to be input at the front desk and delivers it to the database server. Of course, we also work on middle servers that will deliver useful values to the database via queries.

2. App Front Development

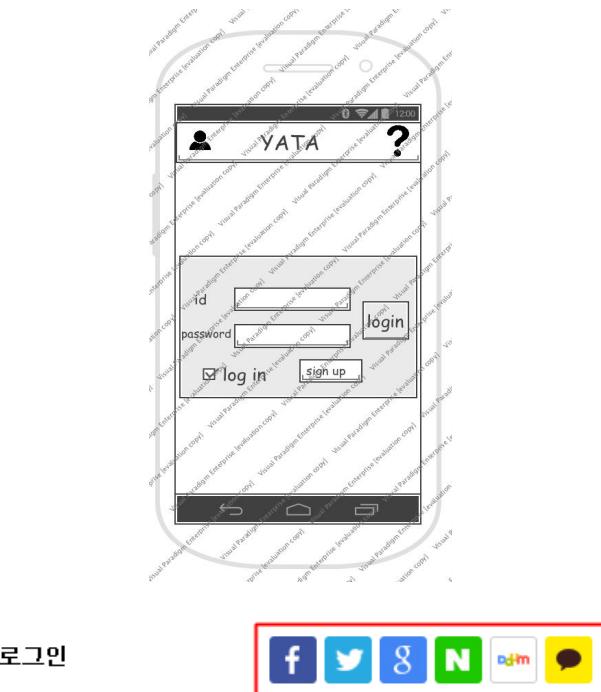
The app page has a view that is configured and the developer completes the development of data connection work to transfer membership data to the database.

3. App development testing and Q&A (to maximize performance)

Expand all function reference definitions, ERDs, class specifications, and storyboards and perform function-specific tests to ensure that sign-up & sign-in functionality is implemented correctly. Confirm precisely whether the subscription input value is properly delivered to the database and the manager page, and whether the subscription status and subscription data appear on the server and front without missing. If there is no problem with the function-specific test, we perform the integration test whether there is a bug from the sign-up to the login. At the same time, designers and planners check carefully to see if they have a first design problem. If a bug is recorded during the inspection, or if there is a problem with the design, immediately address the problem, make a correction, or change the function.

C. Log-in

a. Log-in



1. Enter ID and password

The user must enter his / her ID and password in the ID and password fields for Log-in, respectively. Automatic Log-in is selectable and is selected by default.

2. Log-in button

- (1) If the user's ID and password are established: Approved
- (2) If the user's ID and password are not established: Pop up screen indicating that login failed
- (3) If the login is successful: the main page appears

3. Find username and password

If the user forgets his / her ID and password, the user can confirm or reset his / her ID and password through authentication of the mobile phone or Hanyang-in mail.

4. Membership

If the user is not registered, he / she can register. When the user clicks the subscription tab, the user goes to the subscription page.

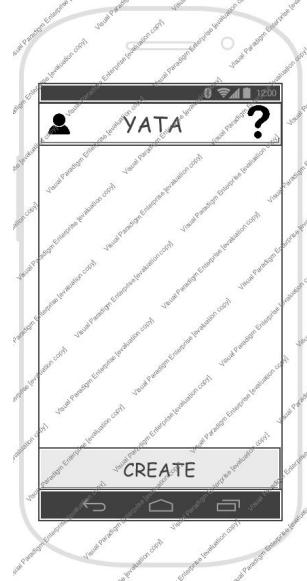
b. How to implement

With Amazon Cognito, we can add user sign-in and sign-in capabilities to mobile and web apps.

D. My Page

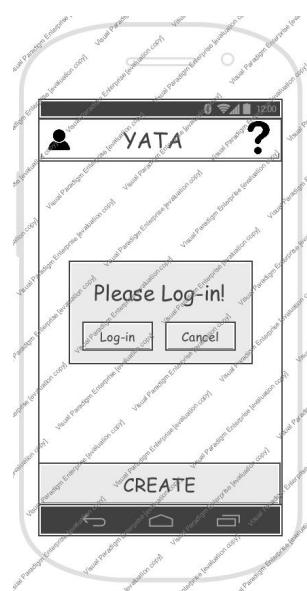
a. Go to My page screen

When the users click the small icon on the upper left one in the main screen, the main screen will change into my page screen. My page screen will be shown only when they log-in.



1. If the users don't log-in

If they don't log-in and click the icon, there comes pop-up message like "Please log-in!" and at the below, there will be 2 buttons: 'Log-in' and 'Cancel'.



(1) When 'Log-in' button is clicked

When the first button is clicked, users will move to Log-in

page.

(2) When ‘Cancel’ button is clicked

When the second button is clicked, pop-up message go over and users will move to previous page, main page.

2. If the users log-in

After log-in, users can go to my page screen which is for settings of users.

b. My page option (required)

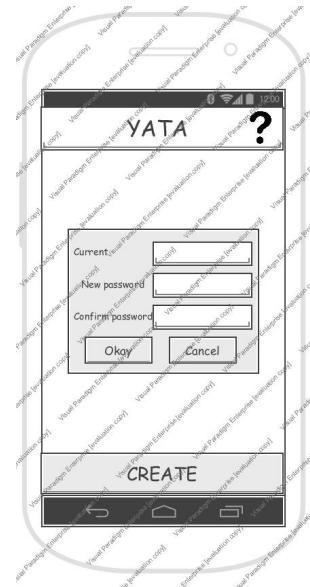
1. Changing the image

If users click their image which they registered in sign-up page, they can modify their pictures. It is for identifying each other when they are matched because people can find each other more easily. Whenever they want, they can change their pictures. If they do not add it, they cannot use this application.



2. Changing the password

People can also change their password, too. If users click the ‘Changing the password’ button under their names, pop-up message which can change the password but they should keep the form of application which was provided in the sign-up page.

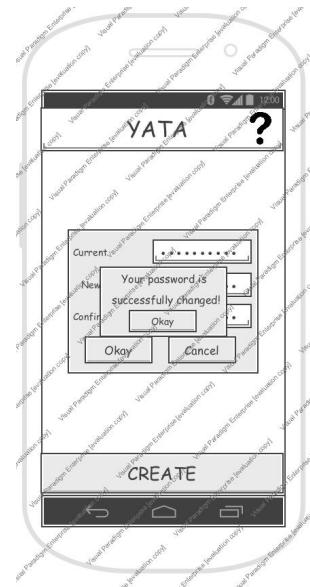


VI. When ‘Okay’ button is clicked

If Okay button is clicked, there are two cases.

• Success

If the current password is right, new password keeps the form of password and it is same with confirmed password, another pop-up “Your password is successfully changed!” comes. If the users click ‘Okay’ button, all pop-ups die and users come back to my page.



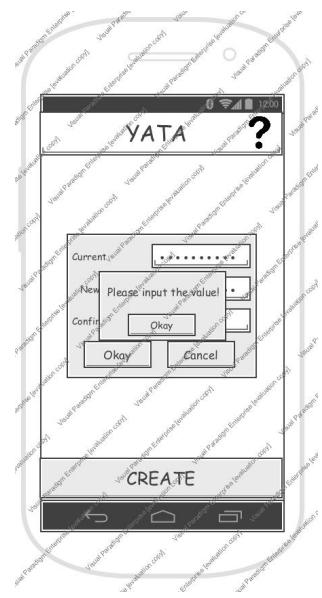
• Fail

There are 4 fail cases. The password is changed when there

is no fail case. Pop-up message will come in following order.

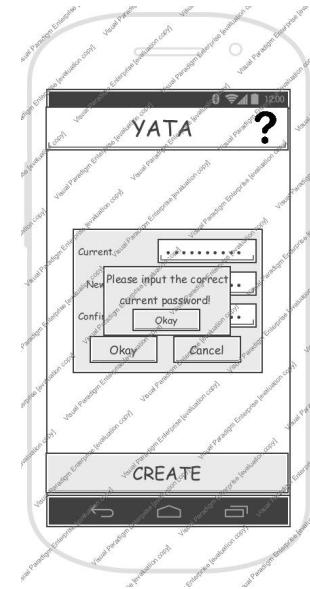
- If there is no value

If there is blank box, there will be another pop-up “Please input the value!”. When the users click ‘Okay’, they will move to changing the password screen again. The cursor will automatically go to blank text box.



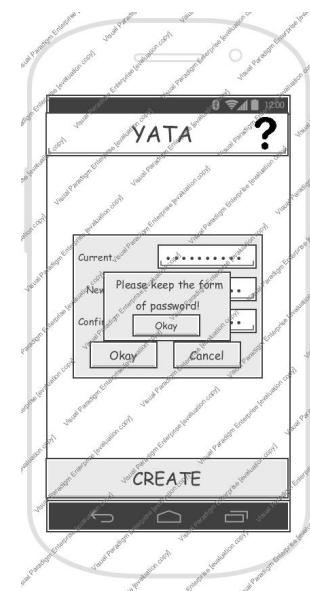
- If the current password is wrong

If the current password is wrong, there will be another pop-up “Please input the correct current password!”. When the users click ‘Okay’, they will move to changing the password screen again. The cursor will automatically go to current password text box.



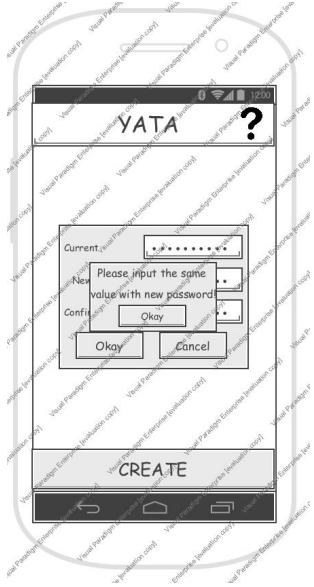
- If new password doesn't keep the form of password

If new password doesn't keep the form of password, there will be another pop-up “Please keep the form of password!”. When the users click ‘Okay’, they will move to changing the password screen again. The cursor will automatically go to new password text box.



- If confirmed password is different with new password

If confirmed password is different with new password, there will be another pop-up “Please input the same value with new password!”. When the users click ‘Okay’, they will move to changing the password screen again. The cursor will automatically go to confirm password text box.



c. Adding time table (optional)

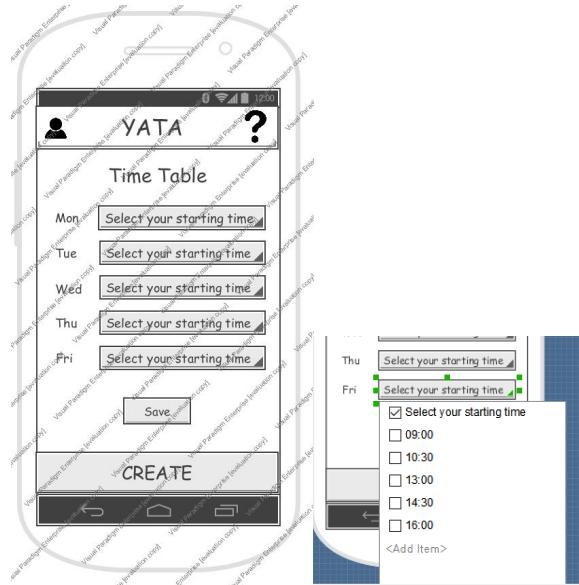
The upper settings in b were required functions. From this part, users have a free to input these things or not because following functions are optional. The users can add their time table into our application. If they click ‘Add the time table’ button, my page screen is changed into time table screen which they can mark their time table.



1. How to add time table

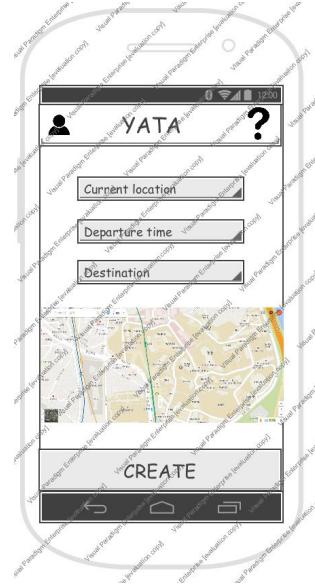
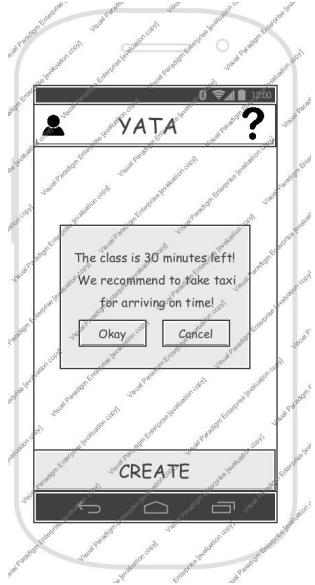
There are 5 options of Monday to Friday. When the students once come to school, they usually do not go their home in empty times although their home is near. Additionally, they usually do not use taxi when they go home. Therefore, we will just let them

input first coming time. They can select the starting time by selection scroll. The default value is ‘Select your starting time.’ Starting time options are ‘09:00’, ‘10:30’, ‘13:00’, ‘14:30’ and ‘16:00’. Users select one of them. If they have empty day, they will keep as a default value. They can change their schedule fluidly by their time table. If they click the ‘Save’ button, their time table is saved.



2. Pre-alarm

On the right bottom part of time table screen, there is ‘on/off’ button which decides pre-alarm. It is set into ‘off’ as a default value. If the users click once and change it into ‘on’. Then the alarm pop-up goes to users like “The class is 30 minutes left! We recommend to take taxi for arriving on time!”. This pop-up will also go to users when they are not in the application. In the below part of sentence, there will be 2 buttons.



(1) When 'Okay' button is clicked

One is 'Okay' button. When the users click this button, pop-up screen disappears and they automatically move to our application main screen.

(2) When 'Cancel' button is clicked

The other is 'Cancel' button. When this button is clicked, pop-up screen disappears, too and nothing happens. The users can do just what they do with their phones.

E. Set and start the matching

When the users come back from my page to main page, there is a picture of a map. The default map value is Hanyang university. There are 3 selection scrolls: 'Current location', 'Departure time' and 'Destination'.

a. How to add map into an application?

We will use Google Maps Android API function.

1. Go to <https://console.developers.google.com/project> and click 'Create the project'.

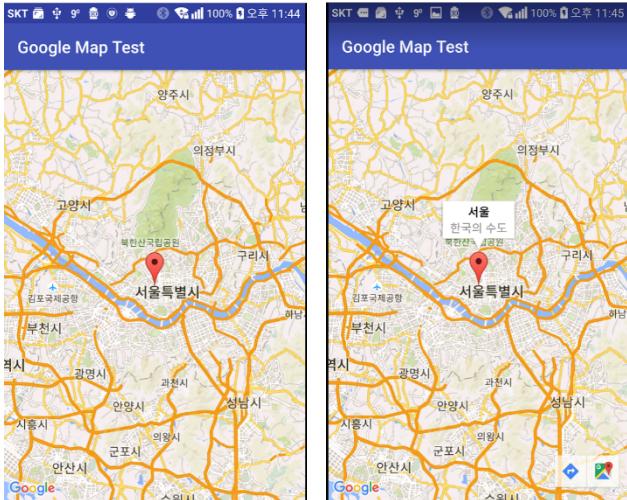
The screenshot shows the Google APIs console interface. On the left, there's a sidebar with 'IAM 및 관리자' and '프로젝트'. Under '프로젝트', there's a button labeled '프로젝트 만들기' which is highlighted with a red box. The main area shows a list of services: IAM, 할당량, 서비스 계정, 라벨, 설정, 암호화 키, and IAP(Identity-Aware Proxy). A message at the bottom right says '종료되어 삭제 대기 중인 프로젝트'.

2. Click Google Maps Android API

The screenshot shows the Google APIs console under the 'Google API' tab. A search bar at the top contains the text '100% 이상의 모든 API 검색'. Below it, there's a section for '인기 API' with icons for Google Cloud API, Google Cloud Vision API, Google Cloud Natural Language API, Google Cloud Translation API, and Google Maps API. The 'Google Maps API' icon is highlighted with a red box.

3. This function will give us API key and help to add map.

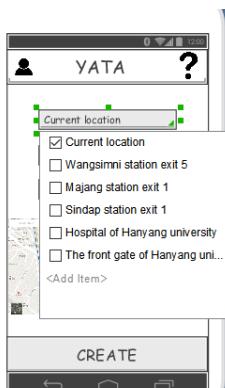
API 키 생성 완료



(application executed screen)

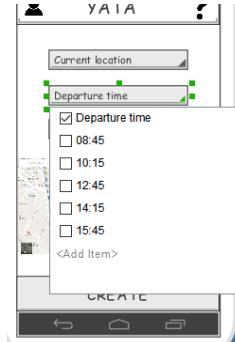
b. Set current location

First button is to set current location. We will give some options of taxi stop. The default value is ‘Current location.’. The other options are ‘Wangsimni station exit 5’, ‘Majang station exit 1’, ‘Sindap station exit 1’, ‘Hospital of Hanyang university’ and ‘The front gate of Hanyang university’. If the users select the option of current location, the map is moved and they can confirm if it is correctly set into current location.



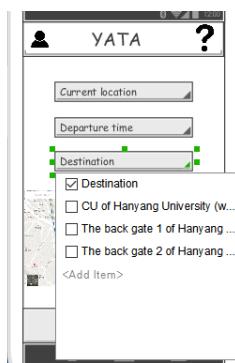
c. Set departure time

The next button is to set their departure time. Our route takes about 10 minutes, so we set departure time before 15 minutes of class starting time. The default value is ‘Departure time’. The other options are ‘08:45’, ‘10:15’, ‘12:45’, ‘14:15’ and ‘15:45’.



d. Set destination

The last button is to set their destination. The default value is ‘Destination.’. The other options are ‘CU of Hanyang University (when the lions eat snacks)’, ‘The back gate 1 of Hanyang University’ and ‘The back gate 2 of Hanyang University’. We chose it by referencing our questionnaire. CU is the center part of our school and the back gates are last part of our school so they can be the efficient destination for the application.



e. If the users have used record

If the users log-in before and share taxi with this application once, their data is saved in the server.

1. If it is same

They don't have to mark their location, time and destination again. These values are set into previous used record automatically. They can click the ‘CREATE’ button in the bottom part of the main page directly.

2. If it is not same

If it is different with current options, users should click the 3 buttons respectively to change into other settings like a, b and c on the upper part.

f. Create the matching

1. Success

After setting, they will click “Create” button. When the button is clicked, there comes a pop-up screen “Your current location is ___. Your time is ___. Your destination is ___. Are these right? (___ values are the users’ input value)”. There are 2 buttons ‘Yes’ and ‘No’.



(1) If the button ‘Yes’ is clicked

When people clicked the button ‘Yes’, then the server starts the matching.

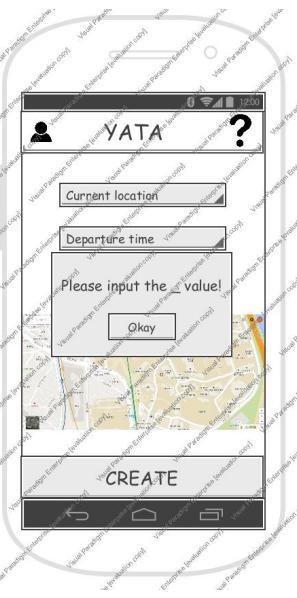
(2) If the button ‘No’ is clicked

When people clicked the button ‘No’, pop-up screen is closed and main page screen is shown again. They can modify their settings which suit their current state. After that, they will click ‘Create’ again.

The server will start to find people who have same current location, time and destination.

2. Fail

If there is a missing part, there comes a pop-up message “Please input the ___ (missing part) value!”. If users click ‘Okay’ button, pop-up message is gone and the main page comes again. Users should fill the missing parts and click the ‘CREATE’ button again. If there isn’t anything missing, the server will start to find people who have same current location, time and destination.

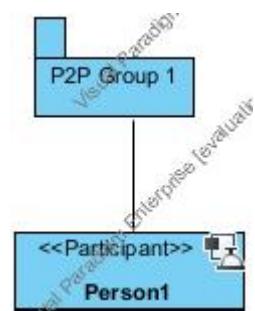


F. Make the chat room

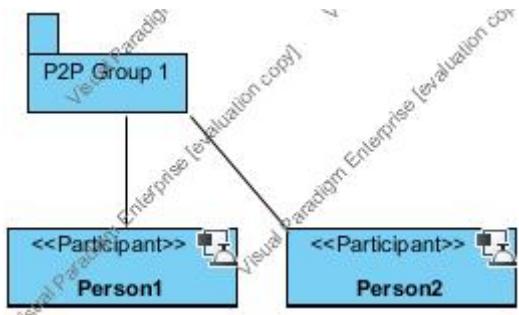
If the server finds the person who has a same current location, destination and time, they are invited to the chat room. If there are more, they will be also invited to chat room. The maximum number of chat room is 4. When more than 4 people are found, the server makes the new chat room and invite likewise. 4 is our ideal number of people because it is easy to pay taxi. We will use ProudNet to implement matching function. This is an engine to develop game server more easily. It provides P2P engines which can connect between clients and allows the client concurrent connection.

A.If the user clicks ‘CREATE’ button, their information go to database.

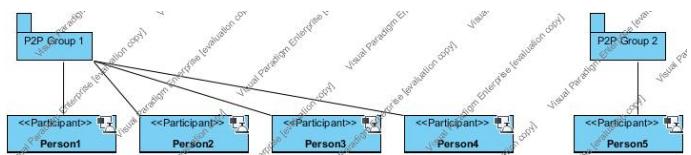
B. Person1 goes into P2P group.



- C. If another user send their information, the server checks if there are same information in the database.
D. If there is, the server instructs to connect person1 and person2.
E. They will know directly that they are connected each other and the server invites the person2 in the chat room (P2P group) which the person1 exists.

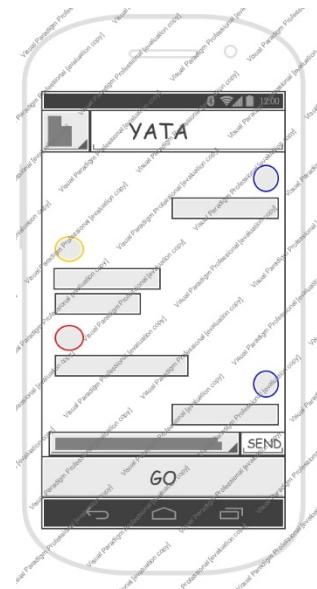


F. Several people can go to the chat room like upper way but if there are 4 in the group already, then the new P2P group is created.



G. In the chat room (chat, go out, penalty)

This is the part which made after matching someone who has same the departure and destination. IN this phase, users can use the application's chat features to exchange simple conversations, arrange meeting places and coordinate their opinions to board a taxi.



a. Chat room service

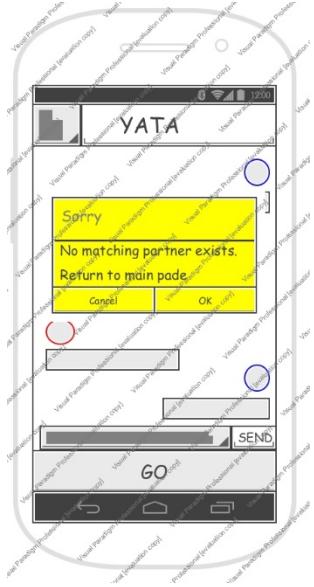
1. When the chat partner enters

When someone enters the chat room, “[ID] has entered” message is shown in the existing chat screen. It shows the message between people. When there is no matching person for the first time, only “[ID] has entered” message is on the top of the screen. The user should wait for the partner's entrance. The user can check this status message if he/she is not offline. If they have more than two people in their chat room, they can always use the "Go" button to take a taxi. If the status of the user in the chat room is before "GO", new users can enter the chat room any time.

2. When there is no matching partner

This application is intended to provide matching for up to 5 minutes. The same origin and destination will be picked up in the chat room within 5 minutes. If users cannot find a matching partner to use taxi sharing in less than 5 minutes, user should return to the previous screen by popping up a popup window titled "No matching partner". If you want to match again, you need to enter the information for matching again and enter the "create" button.

(1) Message when there is no matching partner



In this application, if there is no matching user who can use the taxi-sharing function with same departure, time and destination in 5 minutes. Then it returns to the main page with pop-up massage "No matching partner exists. Return to main page." Pop-up message will ask the user's decision to 'Cancel' or 'Okay'. The user who acknowledged the message will return to the main page, which is where the current location, time and destination can be entered. If the user presses the 'Cancel' button, the application is terminated.

3. Error status of chat partner

The number of people allowed in the chat room is limited to two to four. Once all four are in the chat room, no one can enter the chat room anymore. The chat room is available only when the user is online. The application will check the status of users in the chat room and it will be disabled chat room if users goes offline. If the status of the person in the chat room is offline, "[ID] has exited this chatroom" will be displayed on the screen and will be forced out of the chat room, which will help the user's quick access.

4. Chat contact verification

Users can see the information of person who is matched with in the chat screen. If they click the name of the other person in the chat room, they will get the name, grade, year, student number and profile image information from the user's 'My page', and it is printed on the screen to get the information of the contact person. If the users want to exit this information screen, they should enter the 'X' button to display the chat screen again.

5. Chat room screen message

Users can send and receive messages from a minimum of 1 character to a maximum of 1000 characters. Messages sent by users are displayed on the screen in a conversation format. Messages are output in chronological order.

(1) Sub-condition in chat room display: message bubble regulation

The message will be sent by the user pressing the "send" button. Messages on the screen are divided into 20 characters including blanks in speech bubbles. At this part, messages with more than 20 characters are entered and processed on the next line. Users can freely enter space bar and enter in conversation. If the user enters a blank space or enter key, it will be recognized as a single character and sent as a message. In the chat window, the user ID is displayed, below which are the message bubble and the time the message was sent.

(2) Sub-condition in chatroom display: display regulation

Messages from users are displayed in order from the right, and messages from other contacts are displayed in chronological order from the left.

6. Chat - sending message

In this phase, The development of the process of sending and receiving messages will refer to SmackAPI.

Users in the chat room can all send messages at the same time. Messages can only contain letters and numbers up to 1000 characters and simple symbols from the Android keyboard. The user enters the message that the user inputs into the keyboard and displays it on the screen. We will add the ability to send and receive images and emoticons in the chat room later, if possible.

Messages can be sent in two ways:

- In the form of packets using XMPPConnection sendPacket (Message msg) method
- As a string of Chat Messages using the Chat class. The Chat is a series of messages exchange between two or more users.

Following code snippet shows how to send Message using sendPacket() method of XMPPConnection

The chat class is a convenient way to send messages. The following code snippet shows how to send message using Chat class.

The ChatManager instance is obtained from XMPPConnection using the getChatManager() method.

ChatManager keeps track on all current chats. Chat is created which will now be series of messages exchanged between two users.

The sendMessage(String msg) or sendMessage(Message msg) method is used for sending text messages or message object in the context of a given chat session.

Moreover, MessageListener can be used to get callbacks of notification of Message from other users on chat.

7. Chat – receiving message

When a message is sent from the chat room, data is passed to the server at the same time and can be seen by all users in the chat room. Information about the time the message was sent will also be displayed on the screen. Dates are shown in format h: mm or, if the last message received at least a day in the past MMM DD, YYYY h: mm: ss.

Receiving messages from other user is done using:

- Poll mechanism in the use of the PacketCollector class
- Asynchronous mechanism in the use of the PacketListener (Recommended)

The following code snippet shows an asynchronous way of listening to incoming messages using PacketListener.

8. Exit chat room

If someone does not like the match even though users entered it in the chat room for matching purposes, they can leave the chat room. If the user presses the Back button or forcibly terminates the application, "[ID] exits in this chat room" to the screen. The information of the exited user is still available for viewing by clicking on the name on the chat window screen.

(1) Sub-condition

Users who do not send a message for three minutes with the add-in or have not sent a matching completion status by pressing the Go button will be forced to leave and provide quick access. When the users leave the chat room, they can return to the previous screen, where they can enter the origin and destination, and the user can request the matching again.

9. Chat room complete

After users send and receive a message, they can go to the screen by pressing the "Go" button in the chat room. If more

than one person presses the "Go" button after confirming the user who pressed the "Go" button in the chat room, the screen now displays the message "complete" on the new page. To ensure that the user has successfully completed the match, print a "yes" button under "complete".

10. Chat room complete – error

If one person in the chat room becomes "Go", everyone in the chat room must be "Go" within 5 minutes. Five minutes after one user enters the "GO" status, the chat room is forcibly closed. If a user does not press the "GO" button within 5 minutes, the user will be forced out of the chat room. Users who were ejected will be taken to the previous page. The user who clicked "GO" status will proceed to the next screen, "compete" screen.

(1) Sub-condition 1

If users have more than one GO status, they can take a taxi at any time. This chat room, even if there are not 4 people who want to complete the match, they can still complete the match even if at least 2 users are in the "GO" state. If user does not press the "GO" status, he/she can still stay in the chat room and chat here. However, even if one person in the chat room has checked the "GO" button, the chat room will be forced to close in five minutes.

(2) Sub-condition 2

Users who enter "GO" within 5 minutes will receive the following "Complete" message, otherwise they will be recognized as unmatched and have to go back to the page to enter their previous origin and destination and start matching again.

11. Penalty for users

Within this application, we want to give a certain penalty to the users who are constantly causing problems in the chat room, and to provide services that others can view to verify their information. 1. When user is in the chat room, they are forced to leave the chat room after 5 minutes. 2. If users intercept the "go" button like not clicking the button within 5 minutes. It should be not available.

(1) Black list – Label

In this case, we can force the user out of the chat room. If the forced withdrawal status is more than 5 times, we will show the "BLACK LIST" next to the name of the user's information confirmation window to help other users. The user who issued

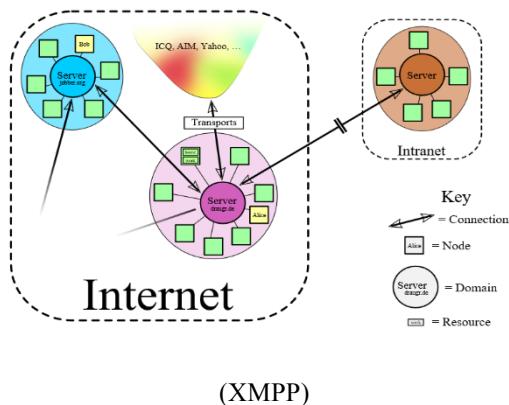
the phrase "[ID] exists in this chat room" more than 5 times through the statement that can count how many times the user has output the message "[ID] exists in this chat room" It provides the ability to display it in user information.

12. Completed chatroom

After users send and receive a message, they can go to the screen by pressing the "Go" button in the chat room. If more than one person presses the "Go" button after confirming the user who pressed the "Go" button in the chat room, the screen now displays the message "complete" on the new page. To ensure that the user has successfully completed the match, print a "yes" button under "complete". Also, if user presses the GO button within 5 minutes, the message "Matching is completed" is displayed on the screen.

b. Chat room environment

From now on, we will explain the part that is the basic environment for implementing messenger chat function in this application. We will develop chat room condition based on smack connecting to an XMPP or Jabber server instance.



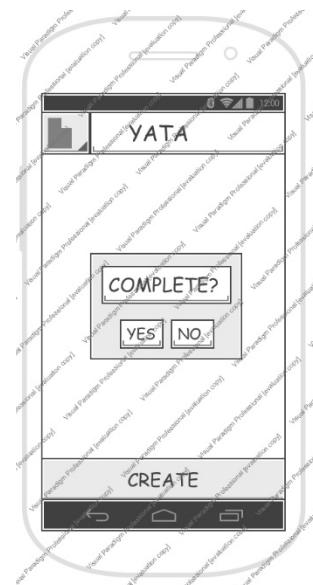
- Green clients are online, yellow clients are writing each other and small green sub-clients are the resources of one user. The brown network is not connected to the internet.

Extensible Messaging and Presence Protocol (XMPP) is a communications protocol for message-oriented middleware based on XML (Extensible Markup Language). It enables the near-real-time exchange of structured yet extensible data between any two or more network entities.

Getting notification on incoming messages, if there are only messages from one contact, open directly the chat windows. Otherwise get a list of all open chats. Messages cannot be sent if account is offline. Dates are shown in format h:mm or, if the last message received at least a day in the past MMM DD,

YYYY h:mm:ss.

H. Finish the matching / Get in the taxi



This part is to confirm that the users who have exchanged messages through chat rooms have completed the conversation and are now ready to board the cab. This application helps the user by creating a page to check if the user has finished the conversation completely and is ready to take the taxi.

a. Matching complete – new page

Now that matching is complete, user should see a new page on the screen, not a chat room. The following screen of the chat room displays "COMPLETE?" As a message on the screen and displays a 'Yes' or 'No' button below it to check the completion status of the user so that the user can recognize the indication they will.

1. No answer situation

If the 'Yes' button is not pressed within 10 minutes, we will assume that the user is in the cab and will proceed to the next page.

2. Return to start – return to main page

Also, in this part, if the user clicks the "create" button below, the message "Do you want to go to the start page?" Is displayed in the window and "Yes" and "No" If the user presses the "Yes" button, the user's screen will return to the start screen to enter the origin and destination. Otherwise, if the user presses the "No" button, it will stay on the additional page screen.

b. Matching complete - state "Yes"

If the user clicks "Go" and sends a confirmation message "complete" and the application confirms the completion of the matching through the "Yes" answer, the user will immediately jump to the new page. Since new pages are additional pages for using applications other than matching, users can force the application to close this time.

1. No answer situation

Even if the user does not press the "Yes" button in 10 minutes, the application recognizes it as a matched state and goes to the next additional page state. If the user goes offline without clicking the answer, the app will be recognized as matched and terminated.

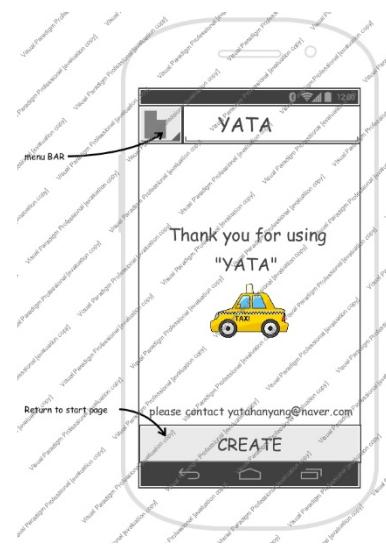
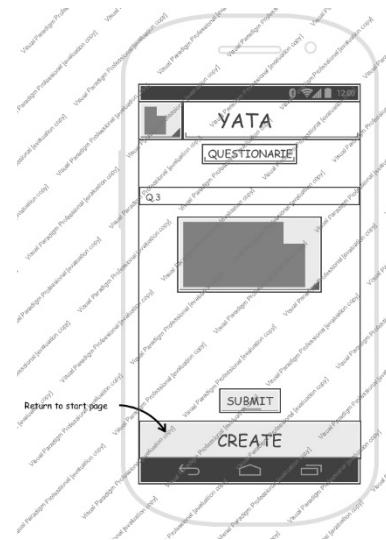
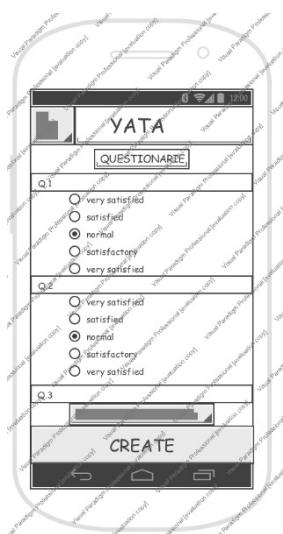
c. Matching complete - state "NO"

If the user who clicked "GO" in the chatting room and went to the "complete" message screen for confirming the completion of the matching presses the button "NO", this means that the user has not been able to board even after leaving the matching chat room. If the "NO" status is entered, the application will go to the page to enter the previous map of the chat room and the destination of the departure point and go through the steps for re-matching.

1. No answer situation

Even if the user does not press the "NO" button within 10 minutes, the application recognizes it as a matched state and goes to the next additional page state.

1. Additional page



This is an additional page user will see after completing the match. This page has been created to maintain this application and to provide a better service experience to users. Show the message "Please input the next button if you want to give a score for this application" to the user and display the "NEXT" button on the screen.

a. Additional page - "YES"

If the user enters the "NEXT" button to ask for additional pages and enters the status of affirmation, he/she will be taken to the questionnaire page. This application can be used to get feedback from users through the following simple questions.

b. Additional page - "NO"

If the user does not press the "NEXT" button when he or she asks if they want to see additional pages, he / she should

recognize that he / she does not want to see additional pages and close the application. If there is no user response for 10 minutes, it will move to the end status of this application. Also, if the user's state goes offline in the middle, the user's state is also switched to the end state.

c. Additional page - Question 1

Q. How much credit would you give to the passenger who boarded the taxi?

This application allows users to select one of five answers using the check box button to answer the above questions. The user must select one of the answers from 1 to 5. The user will choose one answer: 1 point is very unsatisfied, 2 points are unsatisfactory, 3 points are normal, 4 points are satisfied, 5 points are very satisfied.

d. Additional page - Question 2

Q. Are you satisfied with this application? If you are satisfied with this application and are helpful, please enter 5 points.

This application allows users to select one of five answers using the check box button to answer the above questions. The user must select one of the answers from 1 to 5. The user will choose one answer: 1 point is very unsatisfied, 2 points are unsatisfactory, 3 points are normal, 4 points are satisfied, 5 points are very satisfied. Through this question, we will measure user satisfaction with the application and strive for better service.

e. Additional page - Question 3

Q. Please write, if users have any additional request.

The last question is where the user can enter additional requests for this application. We will provide a space to enter the message in this area. Users can enter up to 10000 characters. Here they can enter the alphanumeric and simple symbols provided by the Android keyboard, and can not enter emoticons or special symbols. The screen can only contain up to 100 characters and the requirement message can be viewed and entered by the user using the roll cursor.

f. Additional page - "submit"

1. Question submit

At the bottom of the Additional page screen is a "submit" button where users can submit answers to their questionnaire. If they press the "submit" button while the user is online, the

answer information will be sent to the [yatahanyang@naver.com] mail registered in the server in advance, and the user ID information, the answer time and date, and the answer entered will be sent.

2. Question submit – error

This application will only send the questionnaire to the server when all the answers have been completed to confirm that users have completed all the answers to the questions before completing the submission. If they have not answered all three questions, they will see the message "Please answer all questions" on the screen. At this stage, after confirming that all the answers are completed, press "submit" and the answer will be completed and the user will proceed to the next state.

3. Not submit – return to main page

If the users do not want to submit an answer in the middle, they can go to the start page by typing the "return to the beginning" button pinned on the screen. At this time, if the user clicks the "create" button below, the message "Do you want to go to the start page?" Is displayed in the window and "Yes" and "no" If the user presses the "YES" button, the user's screen will return to the start screen to enter the origin and destination. Otherwise, if the user presses the "NO" button, it will stay on the Additional page screen.

g. Last phase

The screen of the user who clicked "submit" will go to the last page of this application. The message "Thank you for using" is displayed on the screen, and the message "please contact yatahanyang@naver.com" is displayed on the screen at the bottom of the screen. The user can force the application to terminate in the ideal state.

1. Return to main page

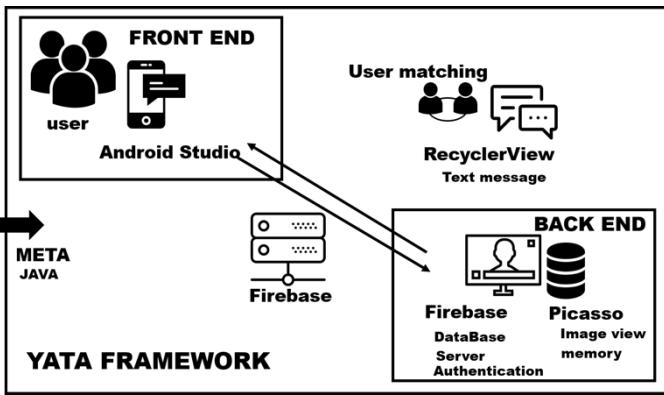
Also, if the users press the "create" button, which is the fixed bottom tab button of the application in this state, at that time, they can go to the start page, which is the page where they enter the start point and the destination point at any time without asking further questions.

V. ARCHITECTURE DESIGN AND IMPLEMENTATION

A. Overall architecture

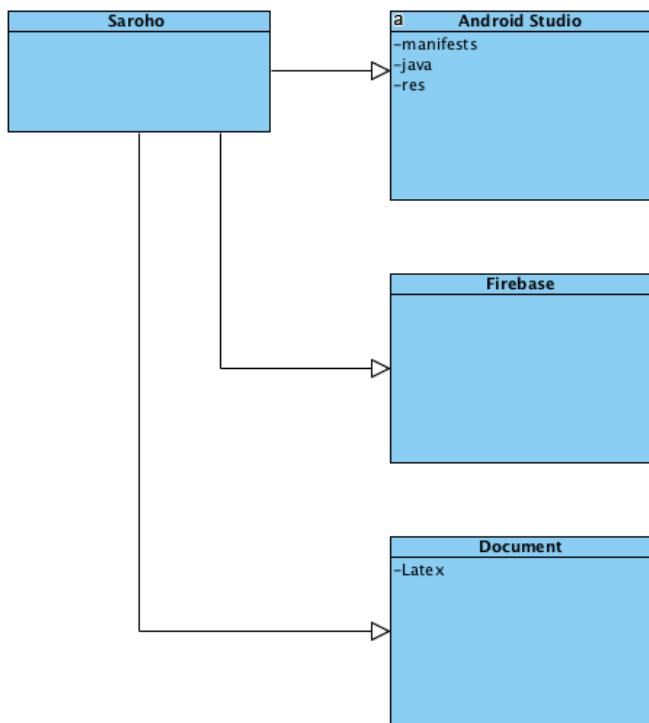
We have three modules consisting our application 'YA!TA!'. First part is 'Android Studio' which is for our front end. It mainly reacts with the client(user) part. This will help us our

overall application containing several functions and designs. For our back end which mainly reacts with the server, we use firebase. Firebase has real-time database and server. The database contains the information of users and it will select the same information for the chat room matching. The users can sign up, log-in or chat and the server will pick their taxi mate using the information in the database. Like this picture, these 2 modules are interacting each other for our application.



B. Directory organization

This picture is our directory organization using UML with visual paradigm. We are using 2 big modules which follow the overall architecture. One is Android Studio and the other one is firebase. Firebase has its directory on-line so there is no separate directory for it. However, we have additional directory for latex which is our project document file.



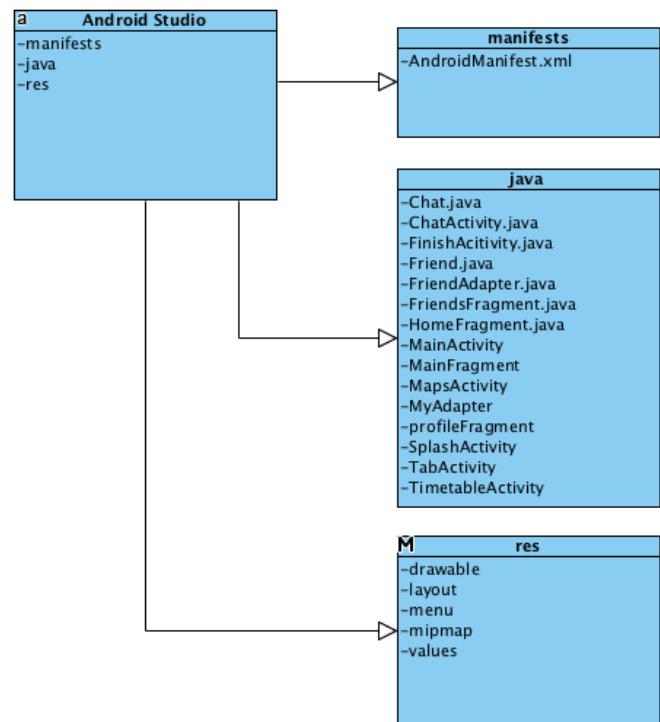
This is the table which shows directory, file name and module name.

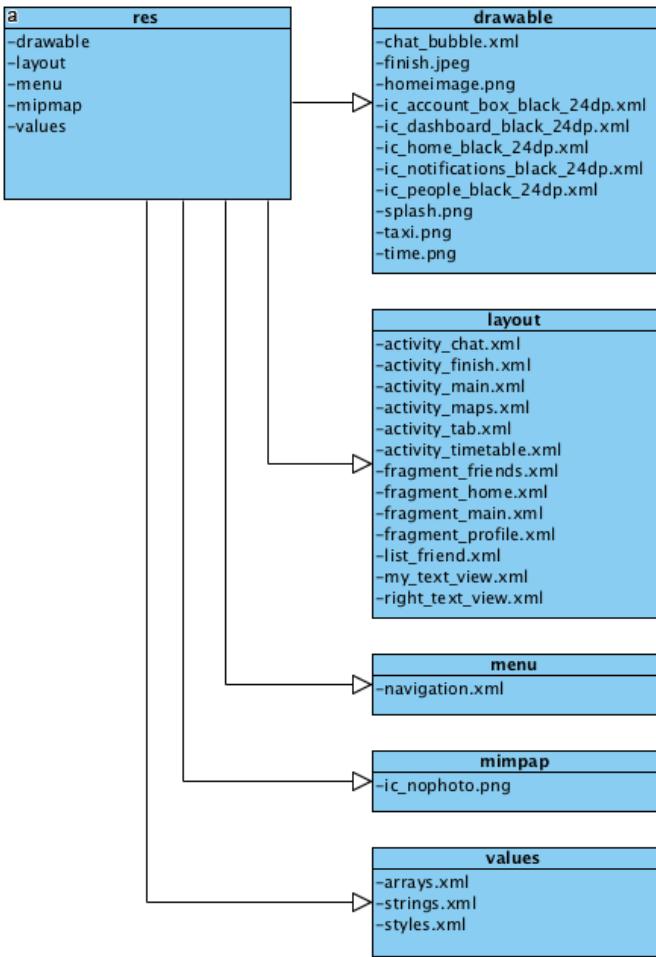
Directory	File name	Module name
AndroidStudioProjects/M yapplication 6/app/src/main/manifests	AndroidManifest.xml	Android Studio
AndroidStudioProjects/M yapplication 6/app/src/main/java/com/ example/kim sohee/myapp lication	Chat.java ChatActivity.java FinishActivity.java Friend.java FriendAdapter.java FriendsFragment.java HomeFragment.java MainActivity.java MainFragment.java MapsActivity.java MyAdapter.java ProfileFragment.java SplashActivity.java TabActivity.java TimetableActivity.java	Android Studio
AndroidStudioProjects/M yapplication 6/app/src/main/res/drawable	chat_bubble.xml finish.jpeg homeimage.png ic_account_box_black_ 24dp.xml ic_dashboard_black.24 dp.xml	Android Studio

	ic_home_black_24dp.xml ic_notifications_black_24dp.xml ic_people_black_24dp.xml splash.png taxi.png time.png	
AndroidStudioProjects/M yapplication 6/app/src/main/res/layout	activity_chat.xml activity_finish.xml activity_main.xml activity_maps.xml activity_tab.xml activity_timetable.xml fragment_friends.xml fragment_home.xml fragment_main.xml fragment_profile.xml list_friend.xml my_text_view.xml right_text_view.xml	Android Studio
AndroidStudioProjects/M yapplication 6/app/src/main/res/menu	navigation.xml	Android Studio
AndroidStudioProjects/M yapplication 6/app/src/main/res/mipmap	ic_nophoto.png	Android Studio

AndroidStudioProjects/M yapplication 6/app/src/main/res/values	array.xml strings.xml styles.xml	Android Studio
document/latex	document.docx document.pdf	Documentation

C. Module 1: Android studio





1. Purpose: We use Android Studio, an integrated development environment, to develop a taxi-sharing application for the Android mobile platform. We use the Android studio provided by Google because we are developing this application for users who use the Android mobile device. Android Studio is the official integrated development environment (IDE) for Android app development, and it provides a basic layout and uses the Java language to help you develop applications easily.

2. Functionality: Android Studio is an environment that is based on all aspects of application development. We build the front-end to be displayed on the user's display through the Android studio. Android Studio also allows you to design applications using the features provided. This module also connects back-end servers and DBs for application-wide development. In addition, you can develop additional functions in your application through various SDK tools provided by Android Studio.

3. Location of Source Code: AndroidStudioProjects /myapplication6

4. Class component: There are several components in the Android studio.

(1) activity_main.java: Set the spinner so that the user can select among the various items that have been set beforehand. The main class will be implemented via the intent function to move from the user's screen to the next ChatActivity.java page.

```

MainActivity.onCreate()
11
12 public class MainActivity extends Activity {
13
14     @Override
15     public void onCreate(Bundle savedInstanceState) {
16         super.onCreate(savedInstanceState);
17         setContentView(R.layout.activity_main);
18
19         Button b = (Button) findViewById(R.id.imageButton);
20         b.setOnClickListener(new View.OnClickListener() {
21             Intent intent = new Intent(
22                 getApplicationContext(),
23                 mypage.class);
24             startActivity(intent);
25         });
26
27         Spinner spinner1 = (Spinner) findViewById(R.id.mySpinner1);
28         ArrayAdapter adapter1 = ArrayAdapter.createFromResource(
29             this, R.array.departure, android.R.layout.simple_spinner_item);
30         adapter1.setDropDownViewResource(
31             android.R.layout.simple_spinner_dropdown_item);
32         spinner1.setAdapter(adapter1);
33
34         Spinner spinner2 = (Spinner) findViewById(R.id.mySpinner2);
35         ArrayAdapter adapter2 = ArrayAdapter.createFromResource(
36             this, R.array.time, android.R.layout.simple_spinner_item);
37         adapter2.setDropDownViewResource(
38             android.R.layout.simple_spinner_dropdown_item);
39         spinner2.setAdapter(adapter2);
40
41         Spinner spinner3 = (Spinner) findViewById(R.id.mySpinner3);
42         ArrayAdapter adapter3 = ArrayAdapter.createFromResource(
43             this, R.array.destination, android.R.layout.simple_spinner_item);
44         adapter3.setDropDownViewResource(
45             android.R.layout.simple_spinner_dropdown_item);
46         spinner3.setAdapter(adapter3);
47
48         ArrayAdapter sAdapter =
49             ArrayAdapter.createFromResource(
50                 ListViewActivity.this, R.array.Departure,
51                 android.R.layout.simple_spinner_item);
52
53         ArrayAdapter sAdapter =
54             ArrayAdapter.createFromResource(
55                 ListViewActivity.this, R.array.time,
56                 android.R.layout.simple_spinner_item);
57
58     }
59
60 }

```

(2) activity_login.java /activity_register.java: After processing the event that goes from the login page to the member registration page, we design the membership registration. One user has information such as ID, password, e-mail, gender, and department. As a result, we have developed a membership design screen. Using firebase, we have built a database and a real app server for membership management projects. After the Log-in function, the function to switch the activity to the main screen is implemented.

(3) UserRegister.php / userValidate.php: UserRequest actually implements the login function after this part of UserValidation has set the PHP web server for login. Android attempts to log in to the web server and displays a message that the account information is not available if the IDs do not match. On the other hand, if the IDs match, you can go to the main screen.

```

UserRegister.php
1 <?php
2 $con=mysqli_connect("localhost", "sarohtoandroid", "sarohto123android");
3
4 $userID = $_POST["userID"];
5 $userPassword = $_POST["userPassword"];
6 $userGender = $_POST["userGender"];
7 $userMajor = $_POST["userMajor"];
8 $userEmail = $_POST["userEmail"];
9
10
11 $statement = mysqli_prepare($con, "INSERT INTO USER VALUES(?, ?, ?, ?, ?)");
12 mysqli_stmt_bind_param($statement, "sssss", $userID, $userPassword, $userGender, $userMajor, $userEmail);
13 mysqli_stmt_execute($statement);
14
15 $response = array();
16 $response["success"] = true;
17
18 echo json_encode($response);
19
20
21 ?>

```

```

ChatActivity.java
14 @Override
15 protected void onCreate(Bundle savedInstanceState) {
16     super.onCreate(savedInstanceState);
17     setContentView(R.layout.activity_chat);
18
19     database = FirebaseDatabase.getInstance();
20
21     FirebaseAuth user= FirebaseAuth.getInstance().getCurrentUser();
22
23     if (user != null){
24         //Name, email, address, and profile photo Url
25
26         email = user.getEmail();
27
28     }
29
30     etText = (EditText) findViewById(R.id.etText);
31     btnSend = (Button) findViewById(R.id.btnSend);
32     btnSend.setOnClickListener(new View.OnClickListener() {
33
34         String stText = etText.getText().toString();
35
36         if (stText.equals(""))|| stText.isEmpty()){
37             Toast.makeText(ChatActivity.this, "내용을 입력해 주세요.", Toast.LENGTH_SHORT).show();
38         }else{
39             Toast.makeText(ChatActivity.this, email + ":" + stText, Toast.LENGTH_SHORT).show();
40
41             Calendar c = Calendar.getInstance();
42             SimpleDateFormat df = new SimpleDateFormat("yyyy-MM-dd HH:mm:ss");
43             String formattedDate = df.format(c.getTime());
44
45             DatabaseReference myRef = database.getReference("chats").child(formattedDate);
46
47             Hashtable<String, String> chat = new Hashtable<String, String>();
48             chat.put("email",email);
49             chat.put("text",stText);
50
51             myRef.setValue(chat);
52         }
53
54     }
55
56     etText.setOnFocusChangeListener(new View.OnFocusChangeListener() {
57
58         @Override
59         public void onFocusChange(View v, boolean hasFocus) {
60             if (!hasFocus)
61                 etText.setText("");
62         }
63
64     });
65
66 }
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82

```

```

UserValidate.php
1 <?php
2 $con = mysqli_connect("localhost", "sarohtoandroid", "sarohto123android", "sarohto456android");
3
4 $userID = $_POST["userID"];
5
6 $statement = mysqli_prepare($con, "SELECT * FROM USER WHERE userID = ?");
7
8
9 mysqli_stmt_bind_param($statement, "s", $userID);
10 mysqli_stmt_execute($statement);
11 mysqli_stmt_store_result($statement);
12 mysqli_stmt_bind_result($statement, $userID);
13
14 $response = array();
15 $response["success"] = true;
16
17 while(mysqli_stmt_fetch($statement)){
18     $response["success"] = false;
19     $response["userID"] = $userID;
20 }
21
22
23 echo json_encode($response);
24
25
26 ?>

```

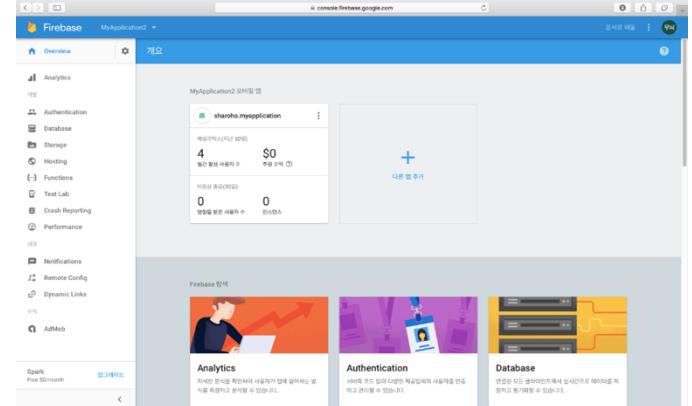
(4) ChatActivity.java: This is a class for chat room that allows users to send and receive messages. This class is linked with DB of the server and the contents of message are stored in DB together with ID. Users' messages appear on the right and left, respectively, and the contents of the message appear in the speech bubble through drawable called chat_bubble.xml.

1. Where it is taken from: we collected information about android studio in Google. Also, many source codes are taken from the library list of Android studio official homepage.

(<https://developer.android.com/studio/intro/index.html?hl=ko>)

2. How and Why we use it: We write the java code using android studio. And we build the project. Using this IDE, we interconnect this with GitHub. It helps our team member share our code and receive public app build function. Android studio provide fast and well-made emulator, so it helps test a system.

D. Module 2: Firebase



1. Purpose: We used Firebase to provide the server and database infrastructure. Firebase is a mobile and web application development platform.

2. Functionality: The information of the user who joined the Firebase and login membership page is stored in the Firebase certificate file. We can see the user's ID and password information through the Firebase console, and also delete the user from the console.

3. Location of Source code: Firebase is the on-line database, so it does not need any location.

4. Class component: There are several components in Firebase.

(1) Firebase Authentication: Firebase Authentication is a service that can authenticate users using only client-side code. It supports social login providers like Facebook, GitHub, Twitter and Google. Additionally, it includes a user management system whereby developers can enable user authentication with email and password login stored with Firebase.

이메일 주소, 전화번호 또는 사용자 이름을 입력	닉네임	생성일	마지막 로그인일	사용자 ID
ya@hanYang.ac.kr	ya	2017. 6. 12.	2017. 6. 13.	56d9qpmCm0BkYVMyH0yghZCnvlWC2
ya@hanYang.ac.kr	ya	2017. 6. 14.	2017. 6. 14.	OutDNg8Q02WV4RkUjLjpwvNz4...
ya@naver.com	ya	2017. 6. 5.	2017. 6. 5.	ZB0ubJzpmPBk3UAn1CqaPNQ2
ya@naver.com	ya	2017. 6. 5.	2017. 6. 5.	deP9mzbphmgn0i0952280W...
ya@naver.com	ya	2017. 6. 8.	2017. 6. 8.	yHcjdMq7XebyrktUcomLwL73

(2) Real-time database: Firebase provides a real-time database and backend as a service. The service provides application developers an API that allows application data to be synchronized across clients and stored on Firebase's cloud. Firebase provide client libraries that enable integration with Android, iOS, JavaScript, Java, Objective-C, swift, and Node.js applications. The database is also accessible through a REST API and bindings for several JavaScript frameworks such as AngularJS, React, Ember.js and Backbone.js. The REST API uses the Server-Sent Events protocol, which is an API for creating HTTP connections for receiving push notifications from a server. Developers using the real-time database can secure their data by using the company's server-side-enforced security rules.

```

https://myapplication-ea5ae.firebaseio.com/users/56d9qpmCm0BkYVMyH0yghZCnvlWC2/messages/chat
+-----+
| 2017-06-12 13:50:02 | x
| 2017-06-12 13:59:49 |
| 2017-06-12 14:02:33 |
| 2017-06-12 14:04:51 |
| 2017-06-12 14:13:07 |
| 2017-06-12 14:25:35 |
| 2017-06-12 14:24:35 |
| 2017-06-12 14:24:35 |
| 2017-06-12 15:04:45 |
| 2017-06-13 01:23:17 |
+-----+
email: "ya@hanYang.ac.kr"
key: "56d9qpmCm0BkYVMyH0yghZCnvlWC2"
photo: ""
  
```

Chat contents of users are stored in DB provided by Firebase. The document is automatically generated according to the user's e-mail information, and the contents of the message and the time are all stored, so the developer can confirm it. The profile pictures provided by the application are also linked using Picasso image, so that the user's ID, email information, and user-registered photos are stored on the Firebase server. Developers can view photos in the database and delete them.

VI. USE CASES

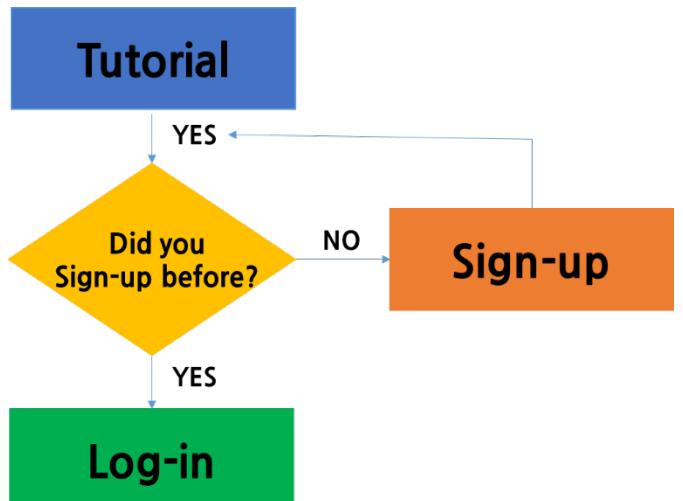
A. Installation

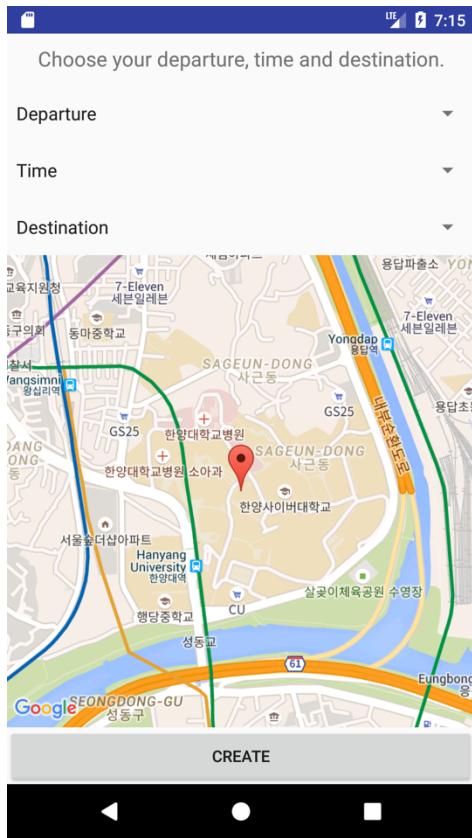
When the user goes to 'Google Play store' and search the words 'taxi', 'car pool' or 'sharing', our application will be recommended, too. When the user downloads it, 'YA!TA!' will be created in their own mobile phone.

B. Turn on the application

If the user clicks the 'YA!TA!' application button, the tutorial page of our application will be shown. When the users' log-in data exists, the main page will be shown instead of tutorial page.

C. Main Page





Main page is the first screen after Log-in. If user run the application, the description of this application will be displayed on the main page first. After users read and understood this, they will see the main screen of the application. Main screen consists of the origin and destination, the time when the user wants to match, a map showing the route briefly, and a 'create' button. Main page consists of menus and application names at the top as shown in the following figure. In the middle, it is composed of a starting point for matching, a destination setting, a desired time, and a map. At the bottom is a button called "CREATE" that starts matching.

a. Set departure, time and destination for matching

1. Departure List

- Wangsimini station exit 5

- Majang station exit 1

- Sindap station exit 1

- Hospital of Hanyang university

- The front gate of Hanyang university

2. Time List

- Now

3. Destination List

- Hanyang University

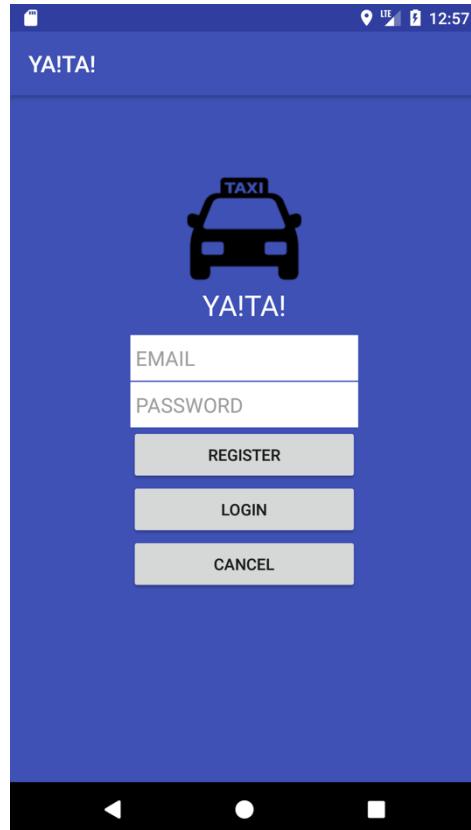
b. Map

The map marker is on 'Hanyang University' as a default value.

c. 'CREATE' button

After the user sets the departure, time, destination, click the 'CREATE' button. Pressing this button means starting the matching.

D. Sign up



a. EMAIL

This is the first screen of our application. When the users sign up, they must enter their e-mail instead of their ID.

b. Password

The password is a combination of English and numbers and can be up to 10 characters long. The password are marked on the screen when typed. If you have entered your e-mail and password to register, you can sign up by clicking ‘REGISTER’ button.

E. Log-in

a. Enter E-mail and password

The user must enter his / her E-mail and password in the ID and password fields for Log-in, respectively. Automatic Log-in is selectable and is selected by default.

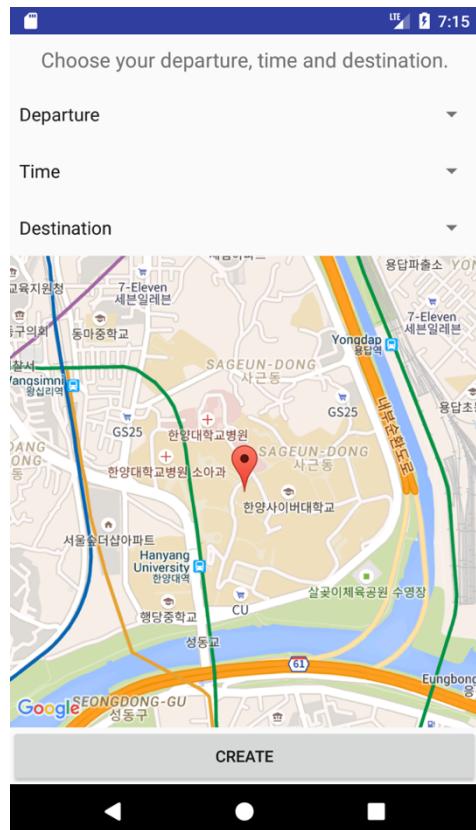
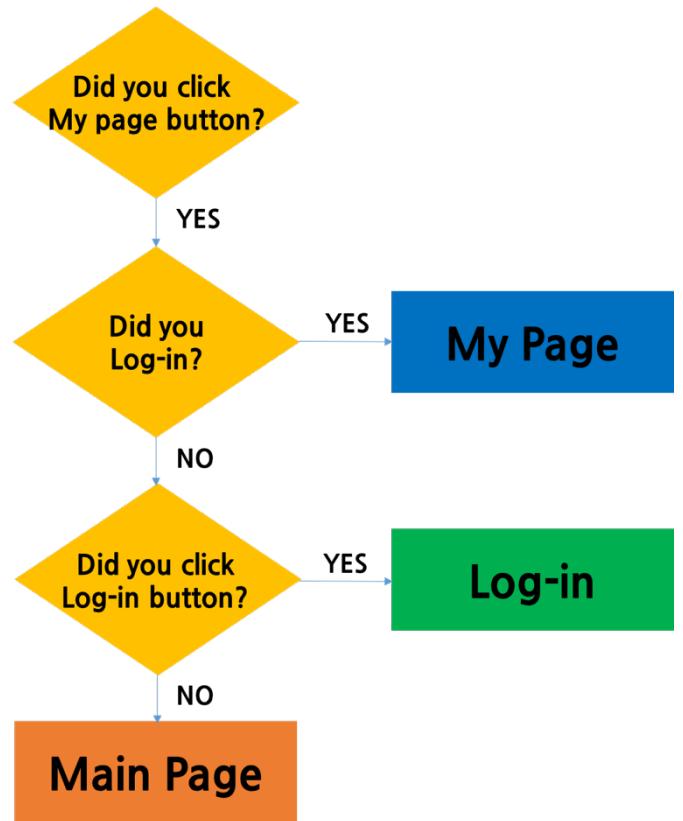
b. Log-in button

1. If the user's ID and password are established: Main page
2. If the user's ID and password are not established: Toast message indicating ‘Authentication failed’.

c. Membership

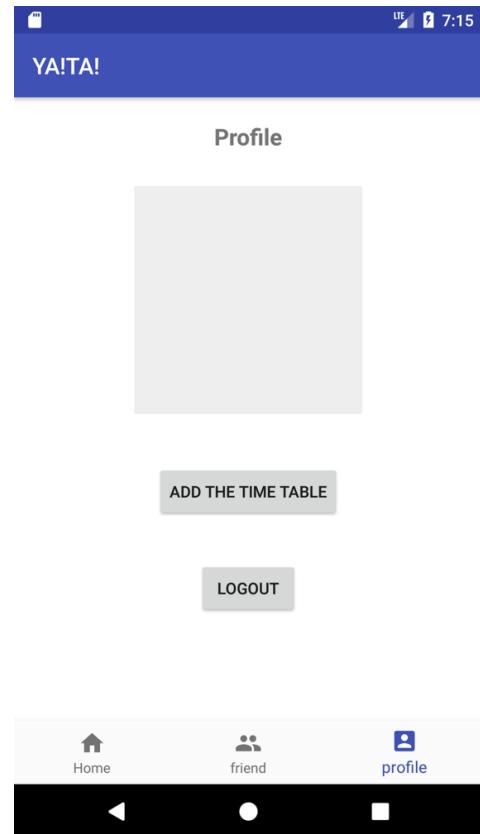
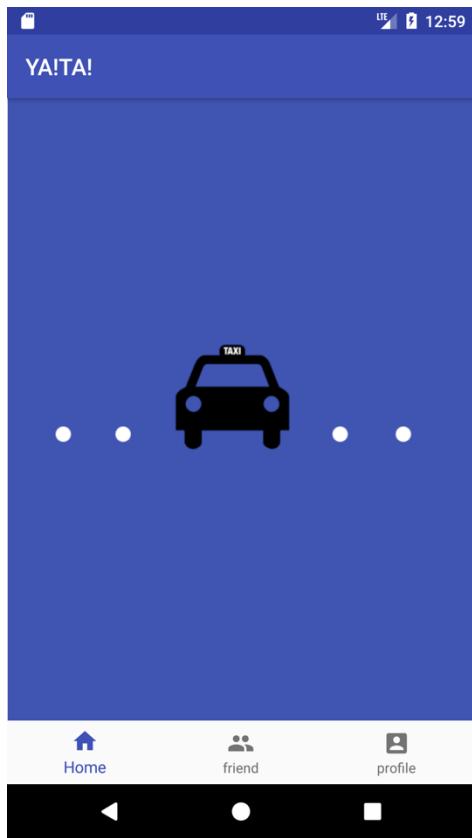
If the user is not registered our application, he/she can sign up. When the user clicks the ‘REGISTER’ button, membership is done if there is no same E-mail in the firebase database.

F. My Page



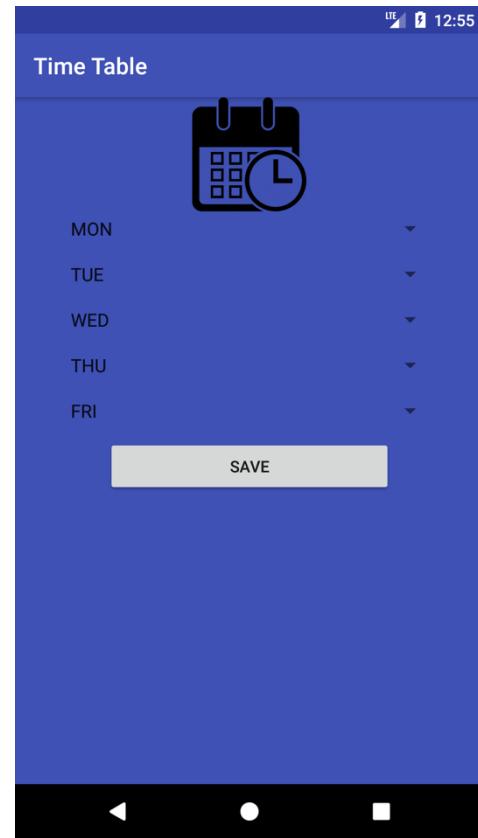
a. Go to My page screen

When the users click the ‘CREATE’ button and click the profile tab, which is on the right below, the screen will change into my page screen.



b. Changing the picture

From this part, users have a free to input these things or not because following functions are optional. When they click the current image, the screen changes to users' gallery and users can choose what picture they want to change.

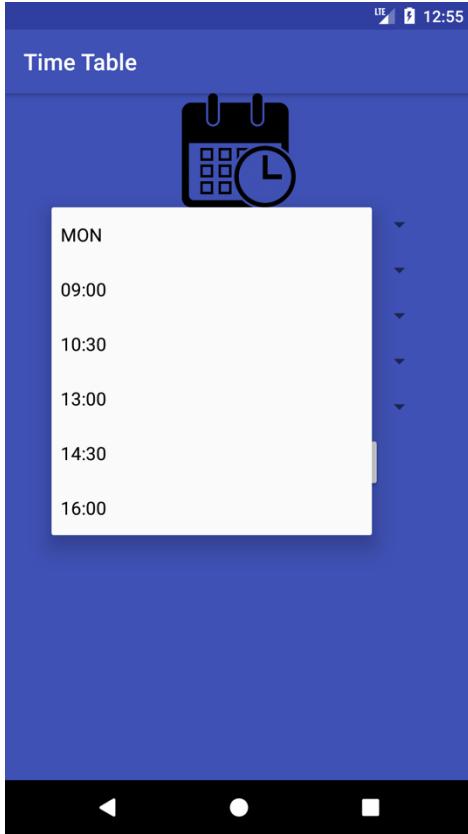


b. Adding the time table

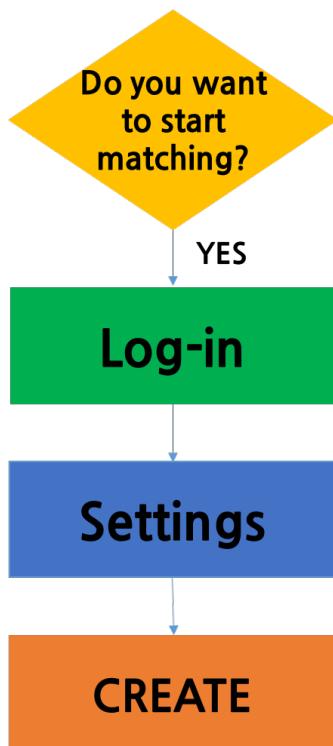
The users can add their time table into our application. If they click 'Add the time table' button, my page screen is changed into time table screen which they can mark their time table.

1. How to add time table

There are 5 options of Monday to Friday. When the students once come to school, they usually do not go their home in empty times although their home is near. Additionally, they usually do not use taxi when they go home. Therefore, we will just let them input first coming time. They can select the starting time by selection scroll. The default value is 'Select your starting time.' Starting time options are '09:00', '10:30', '13:00', '14:30' and '16:00'. Users select one of them. If they have empty day, they will keep as a default value. They can change their schedule fluidly by their time table.



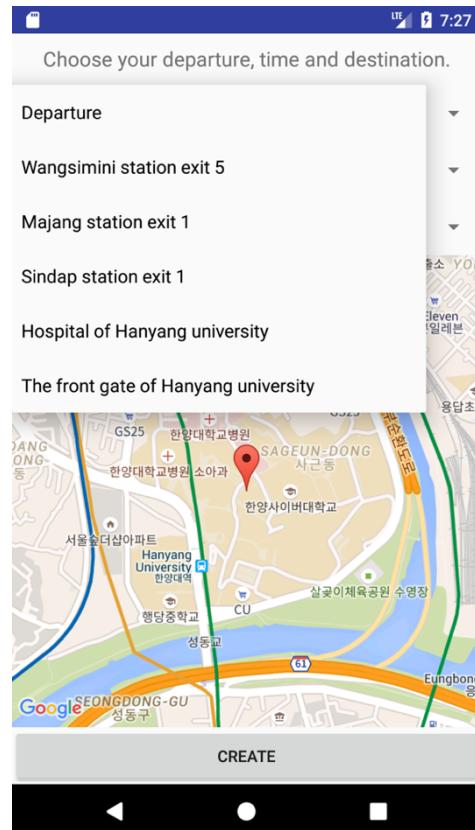
G. Set and start the matching



When the users come back from my page to main page, there are 3 spinners: ‘Departure’, ‘Time’ and ‘Destination’. Also, there is a map. The default map marker is Hanyang university. We implemented it using Google API key and Google map. Users can zoom in or zoom out and get some knowledge about locations around them.

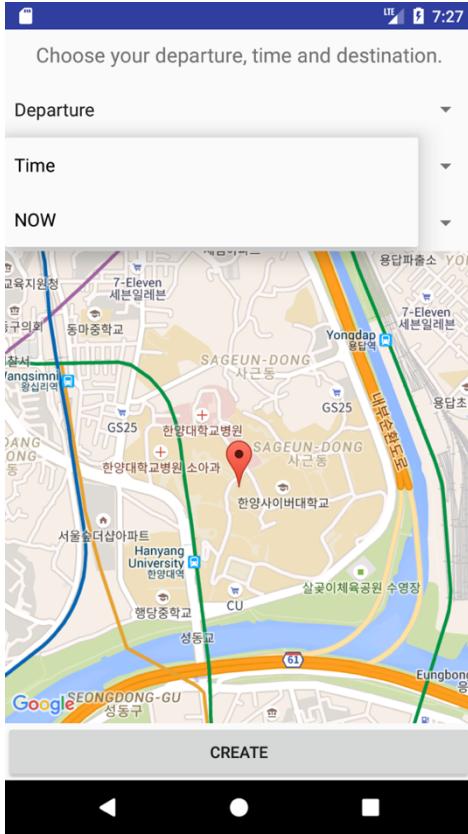
a. Set departure

The first spinner is to set departure. We will give some options of taxi stop based on questionnaire. The default value is ‘Departure’. When the users click this spinner, they can see other options. The other options are ‘Wangsimni station exit 5’, ‘Majang station exit 1’, ‘Sindap station exit 1’, ‘Hospital of Hanyang university’ and ‘The front gate of Hanyang university’.



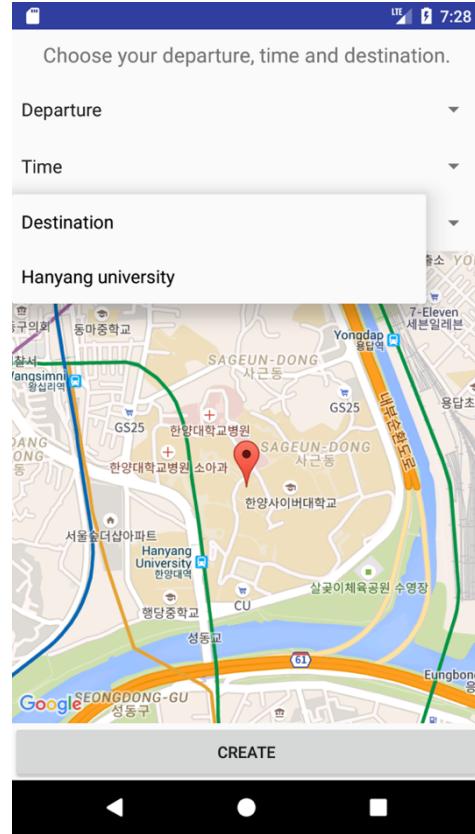
c. Set time

The next button is to set time. The default value is ‘Time’. Our application supports real time. Therefore, there is only one option, ‘now’.



d. Set destination

The last button is to set their destination. The default value is ‘Destination.’. Our application is for Hanyang University students. So, there is only one option, ‘Hanyang University’. Users will decide their destination exactly after they ride taxi.

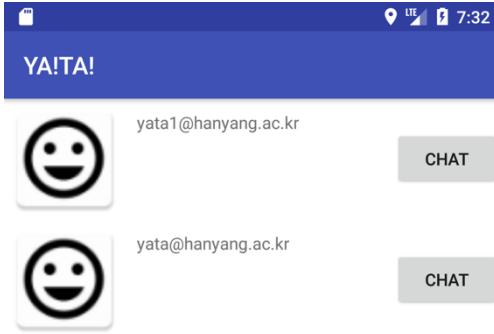


e. Create the matching

If the users terminate their settings, what they should do is just pushing the ‘CREATE’ button.

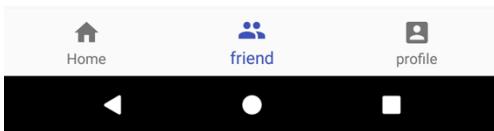
H. Make the chat room

After pushing ‘CREATE’ button, Chat room list screen with list view screen will come up like Kakao Talk chat room screen. All the members will come up at this list. After clicking the ‘CHAT’ button which the users want to chat with, the screen will change into chat room screen.

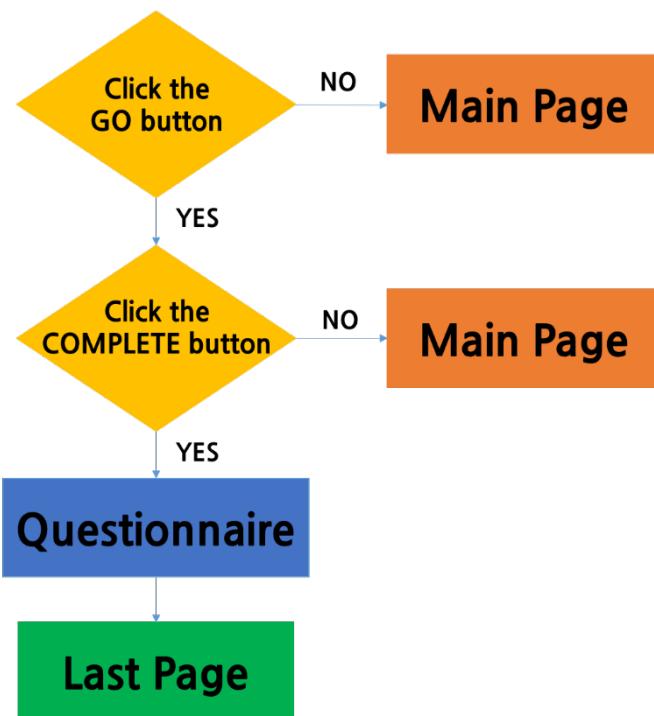


From now on, it is the part where users can select a destination and a departure area, so that they can take a taxi together. The user can talk with the person who selected the same place of departure and destination at the same time as him / herself through the chat room provided in the application. At this point, the user can abandon the match at any time, return to the previous state, or complete the match.

a. In the chat room

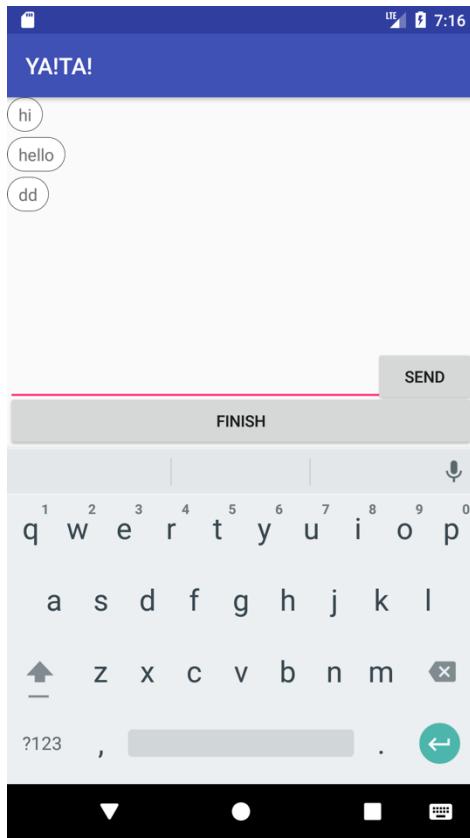


I. In the chat room (chat, go out, penalty)



Our application shows a list of people who have chosen the same origin area and destination. The user can exchange messages with these users. If the users want to exit this information screen, they should enter the 'Back' button to display the chat screen again. In this application, the message on the left shows the message sent by the user himself and the message on the right shows the message sent by the other user. The user has a simple conversation with the server and this information is stored in the developer's firebase database. We have not implemented the ability to send and receive information other than messages.

b. Message bubble



The message will be sent by the user pressing the "send" button. Messages on the screen are divided into 20 characters including blanks in speech bubbles. At this part, messages with more than 20 characters are entered and processed on the next line. Users can freely enter space bar and enter in conversation. In the toast message, the user ID is displayed, below which are the message bubble and the time the message was sent.

c. Chat room environment

The chat server for this application was developed through Firebase. Firebase provides the infrastructure for application servers. Firebase allows you to store user-registered information in your application. The users' ID and Photo are saved. The contents of the message are also stored in the Fire Base database. Messages from users are displayed in order from the right, and messages from other contacts are displayed in chronological order from the left.

d. Chat - sending message / receiving message

When a message is sent from the chat room, data is passed to the server at the same time and can be seen by all users in the chat room. Information about the time the message was sent will also be displayed on the screen. Dates are shown in format h: mm or, if the last message received at least a day in the past MMM DD, YYYY h: mm: ss.

J. Finish the matching / Get in the taxi

a. Final page

We have created a final page to let you know that the chat is complete. This is a page with a thank you note with a logo. After chatting with the user simply by chatting in the chat, the user presses the go button and goes to the finish activity screen. The user can exit the application by clicking again in this screen.





K. Turn off the application

After the users finish the matching, if the users click the 'Exit' button or terminate forcibly, they will go to home page of their own mobile phone.

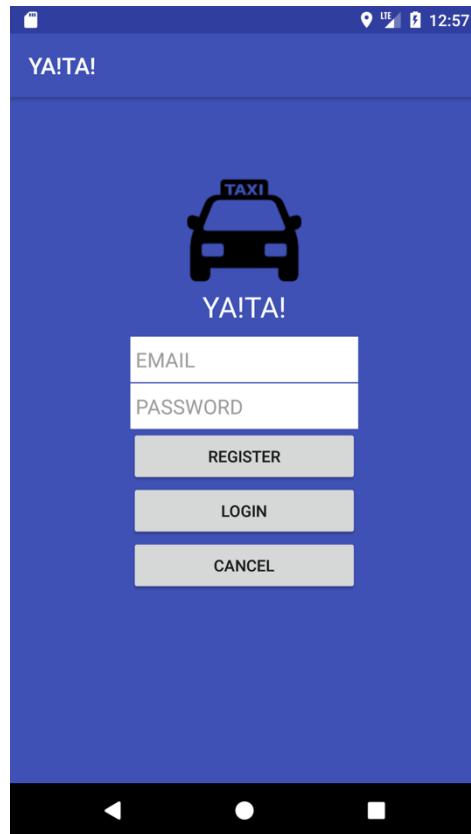
VII. INSTALLATION GUIDE

A. Introduction

This part is for our application users and we will give some information about how to use it. Our first goal in this application is to users match fast because of their comfort. Therefore, this user guide is very simple and the users can follow easily.

B. Installation procedure

Step 1. Sign up



The second picture is the first screen of our application after splash activity. For using our application, users should sign up. Users should input E-mail and password. After that, they should click the 'REGISTER' button.

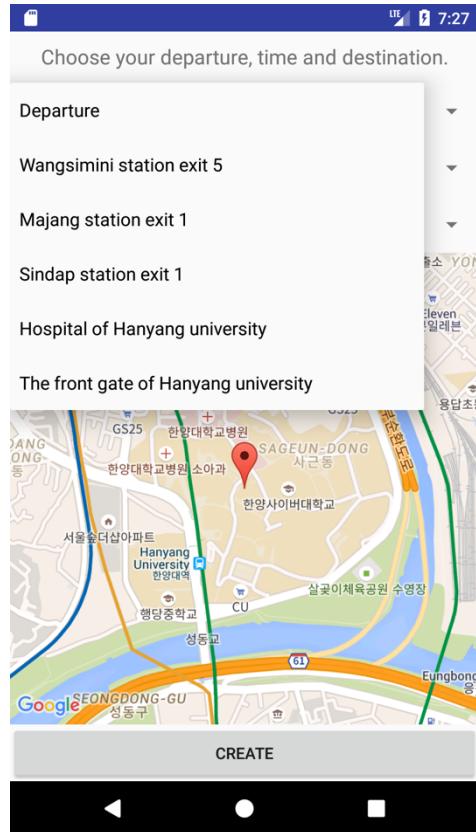
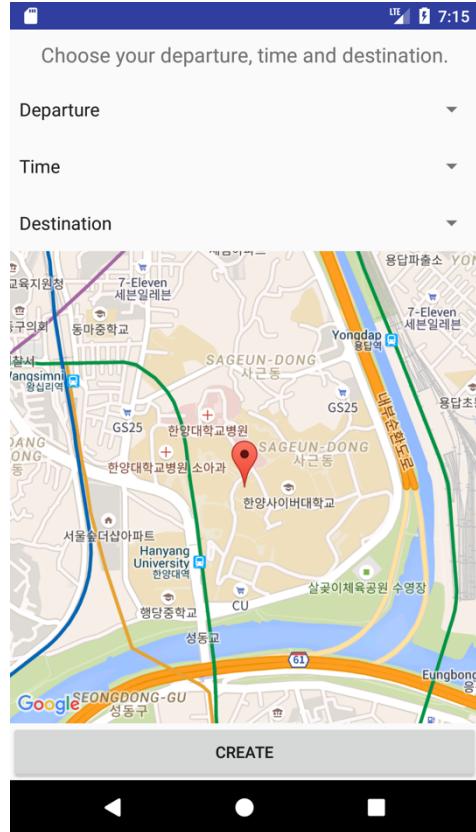


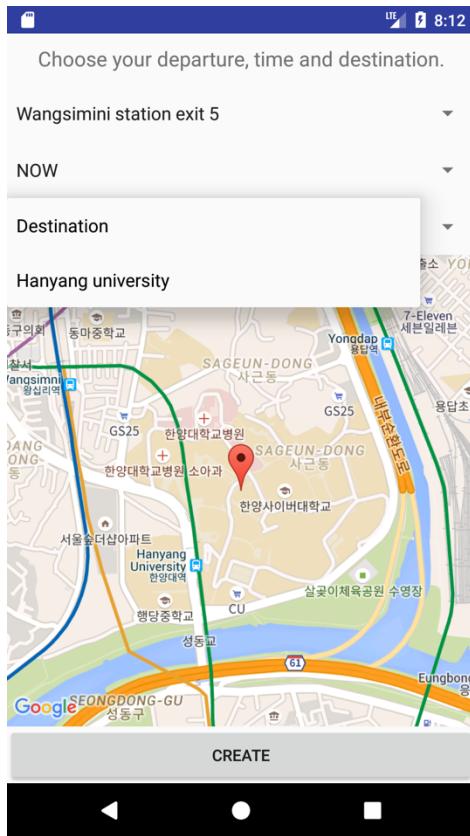
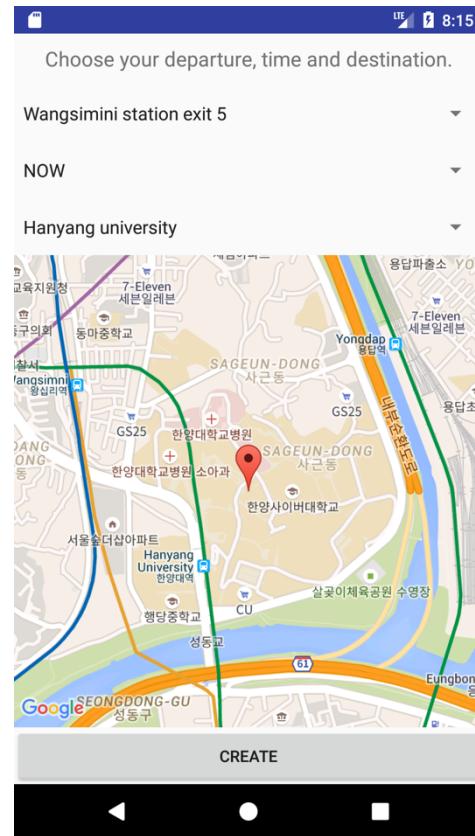
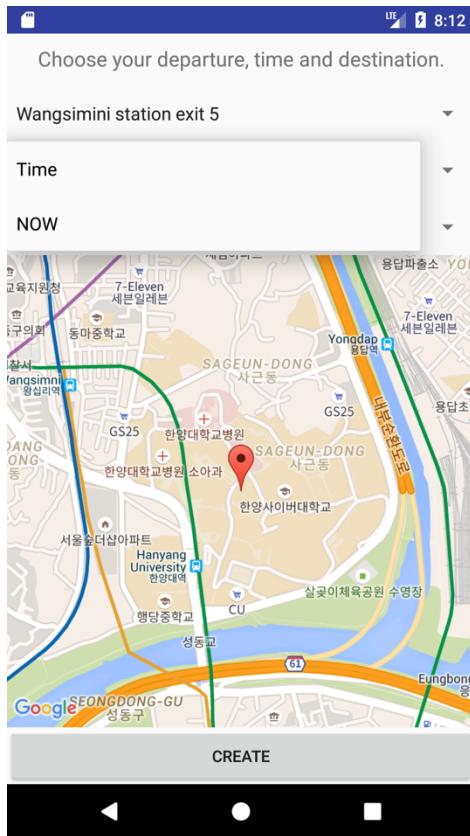
Step 2. Log-in

If the E-mail is not on the firebase database already so the registration is successful, then the users can Log-in.

Step 3. Settings

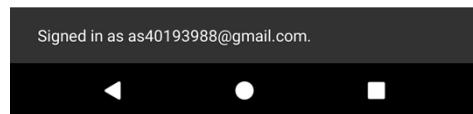
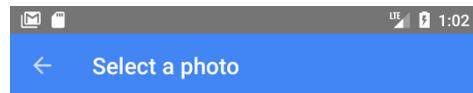
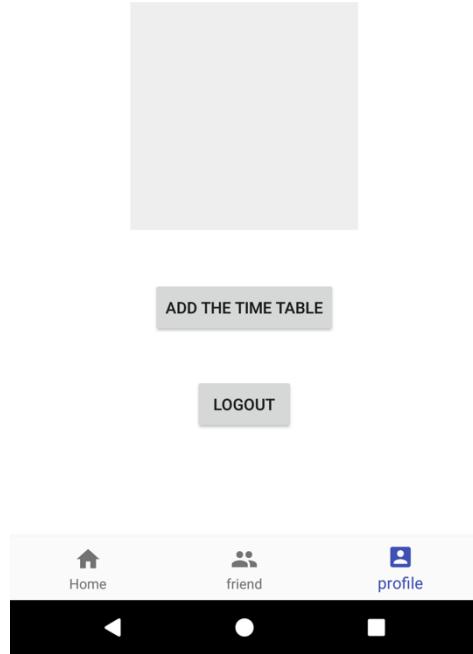
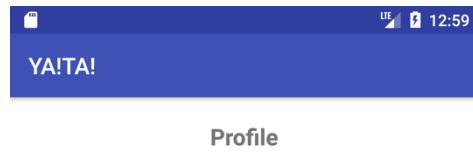
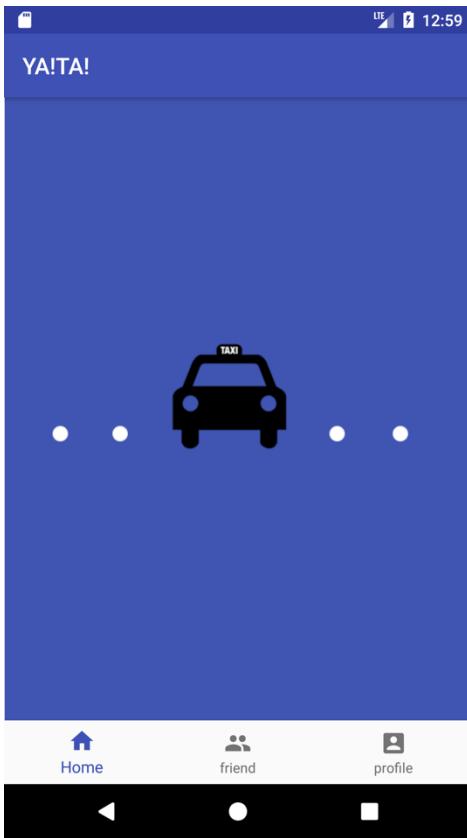
After Log-in, the main page is shown to the users. They can choose their departure, time and destination with 3 spinners in the screen.

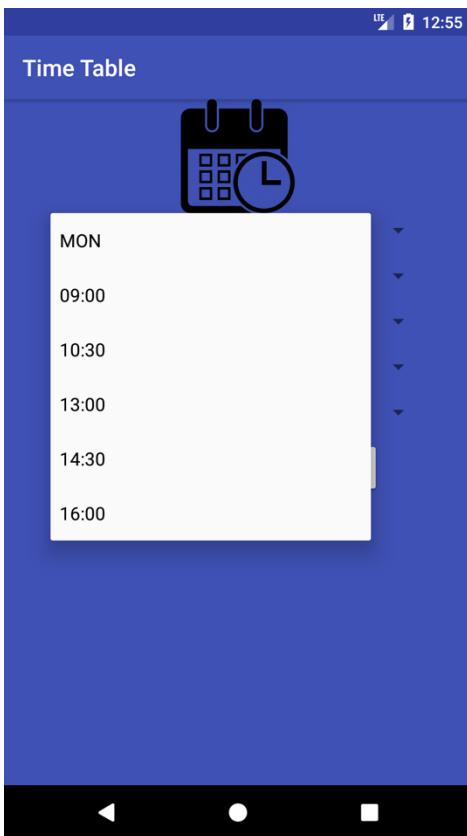
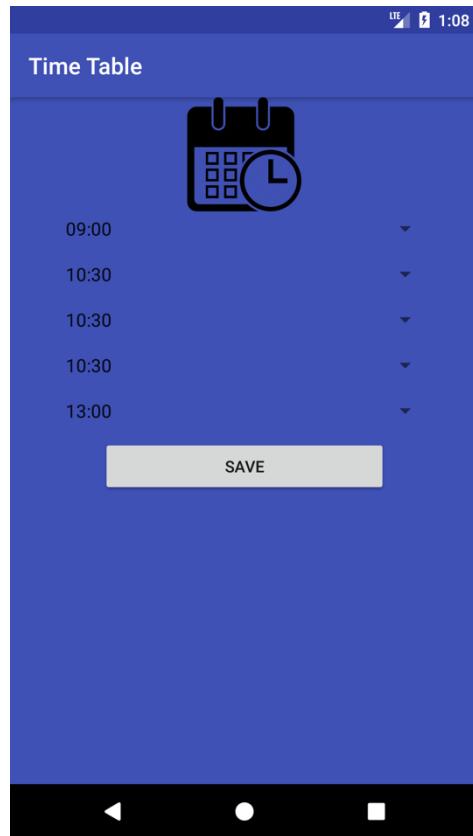
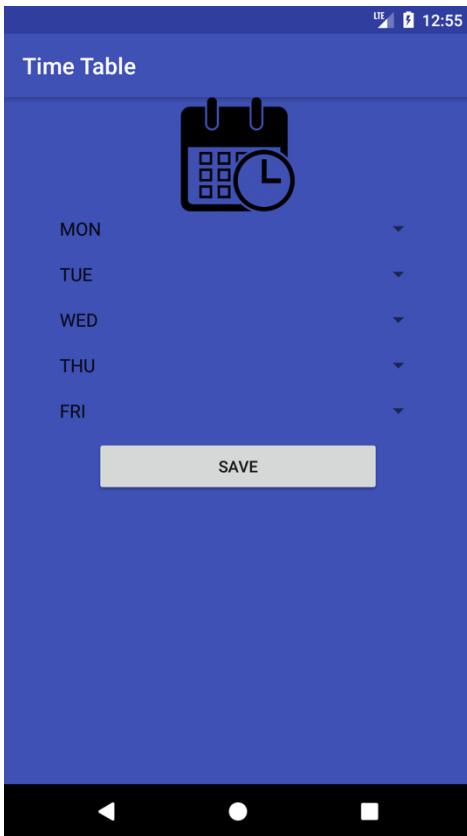




Step 4. My page

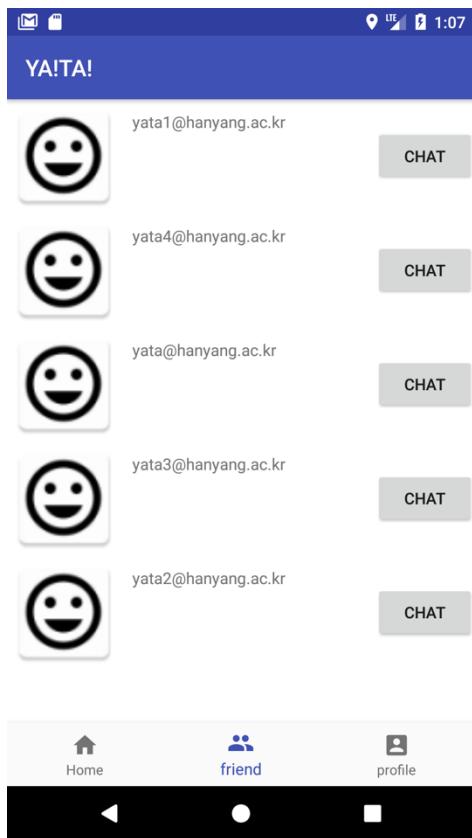
After pushing ‘CREATE’ button, there are 3 tab fragments below. This screen is the home fragment which shows taxi matching.





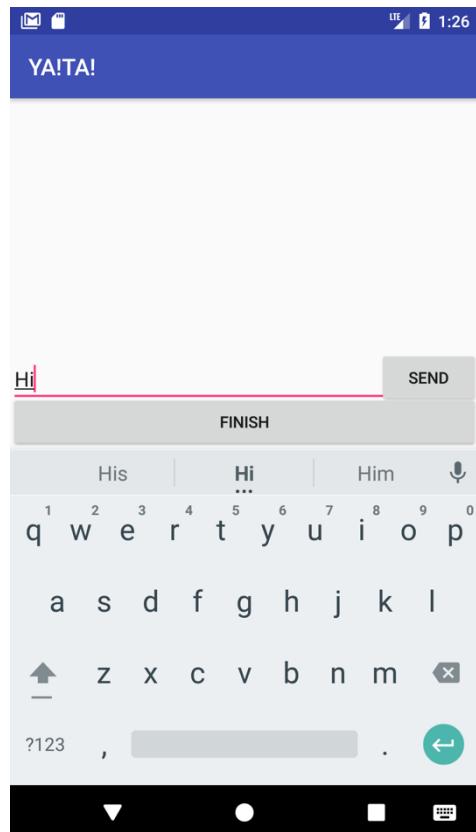
Step 5. Choose the chat mate

When the friend tab is clicked, there are users who have signed up our application. Users can choose the chat mate's ID and click the 'CHAT' button which is next to ID.

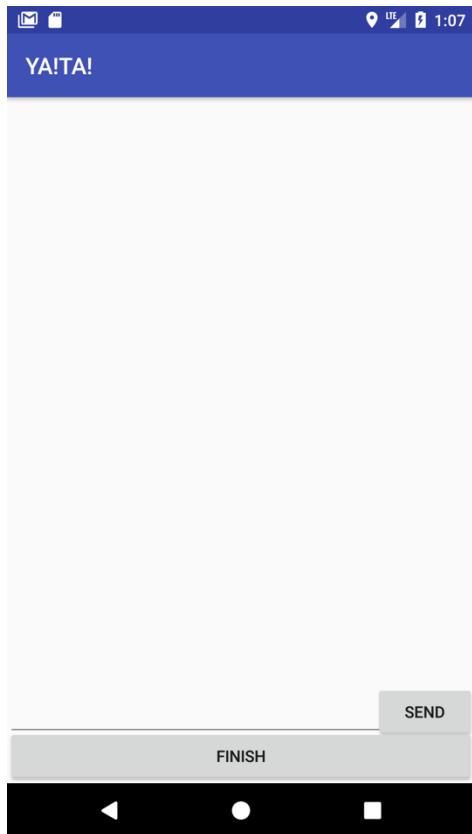


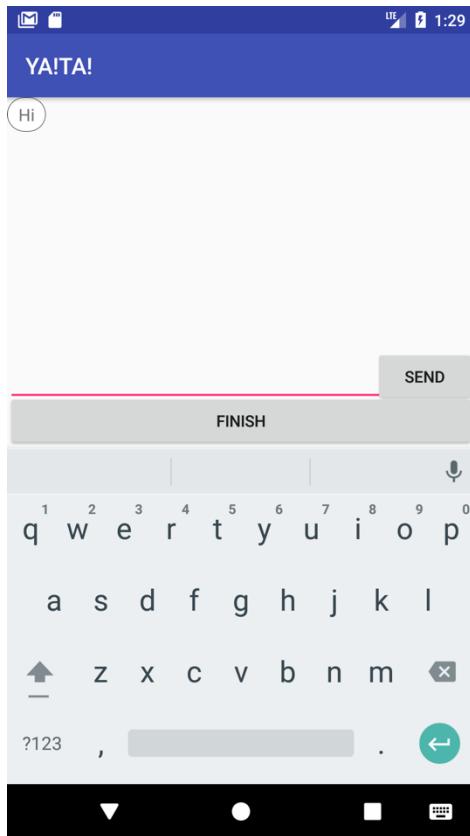
Step 6. Chat

The upper one is the first chat room screen.



After sending message, it is shown on the upper left side.





After chatting and clicking the 'FINISH' button, then the application is finished with this screen.

VIII. DISCUSSION

We have been working on the project for over three months. We looked at our ideas and looked for other examples that could be used to implement our ideas and designed and implemented our own apps. Since it was the first time to implement the software in person, there were many mistakes and many times we made a lot of mistakes and often spent time in inefficiencies. In particular, there was a problem with implementation because of poor connectivity with the server, and it was difficult to write code to match students. But I think we are well worth it for our own implementation of the software, and we learned a lot in the process. We think that these processes are like taking the first step in becoming a software engineer later on and will help us a lot in the future.

<Github repository>

<https://github.com/skh03089/MyApplication6>