

CO-EATER

TEAM 3

Cho Sesang Kim Kiwoong Kook Seokyoung Lee Giha

CONTENTS

A. REFINED PROJECT

1. Target Users
2. Project Scope

B. DEVELOPMENT PROCESS

1. Use Cases
2. System Architecture
3. Technical Challenges

C. FUTURE PLAN

1. Final Deliverable
2. Roles & Timeline
3. Potential Threats



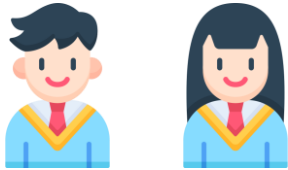
REFINED PROJECT

| Target Users

| Project Scope

Target Users

Prev. : Freshmen & Sophomores



- Deprived of the opportunity to make friends
- Want to learn common senses in university (lecture, career, club activities, etc.)

Eager to get along with new people

New Target User : **Couples** & Friends



- Deprived of the opportunity to meet up (especially **long distance** couple due to self-quarantine)
- Want to have **interesting experience online**

Continue the close relationships

Target Users

Prev. : Freshmen & Sophomores

New Target User : **Couples** & Friends

Service that delivers fun dining experience so that
people can **keep close relationship** and build good memories

activities, etc.)

Eager to get along with new people

self-quarantine)

- Want to have interesting experience online

Continue the close relationships

Project Scope



One on One

- ✓ Send invitation to a friend
- ✓ Add friend with user code
- ✓ Enjoy your meal with gamification features!



Group

- ✓ Select date, time and match based on university and major
- ✓ Create/join Group

ONLINE Less burdensome than meeting offline

GAMIFICATION Emoji&Filter Simple games

The background of the slide is a solid orange color. Overlaid on this is a faint, light-colored image of a technical drawing or blueprint. A pen is visible in the upper right, and a ruler is visible in the lower left, both also in a light color, blending into the background.

DEVELOPMENT PROCESS

| Use Cases

| System Architecture

| Technical Challenges

Use Cases

Downloading the app

1 Direct download



<https://github.com/coeater/coeater-android>

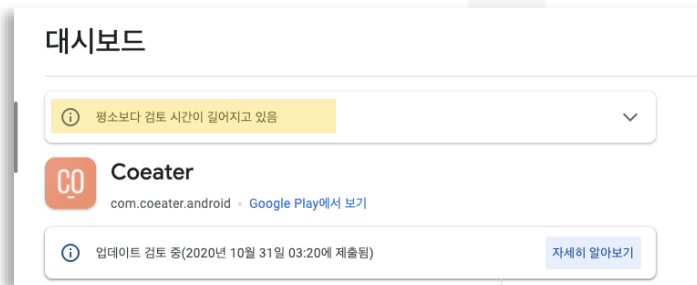


(Link will be offered)

2



Google Play



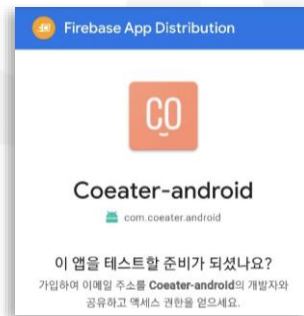
3

Send request to




App Distribution

<https://appdistribution.firebase.dev/i/e1f3f73ac1596247>




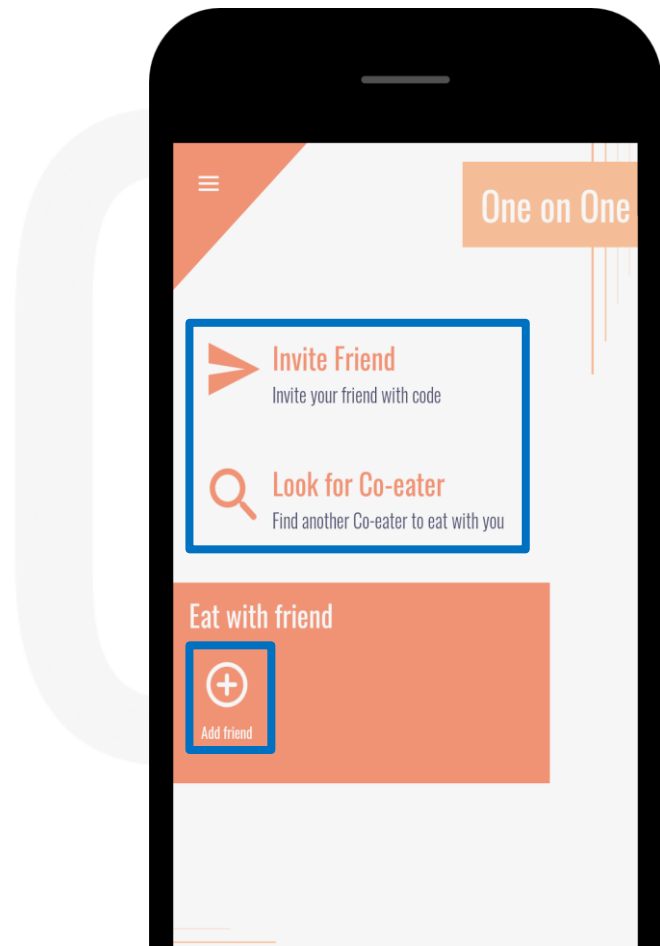
Use Cases 0. Overview

1 Add friend with user code

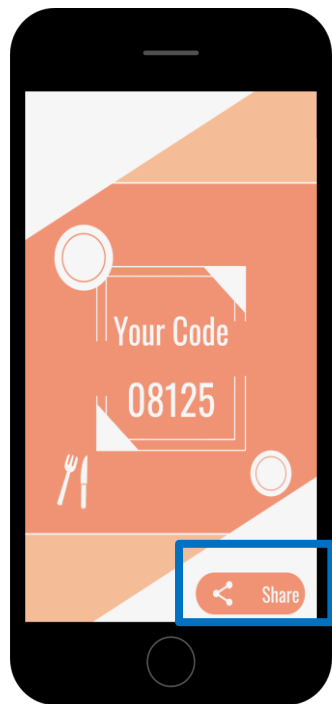
- ✓ Share link to 
- ✓ Eat directly with friend

2 Send or accept invitation and eat together

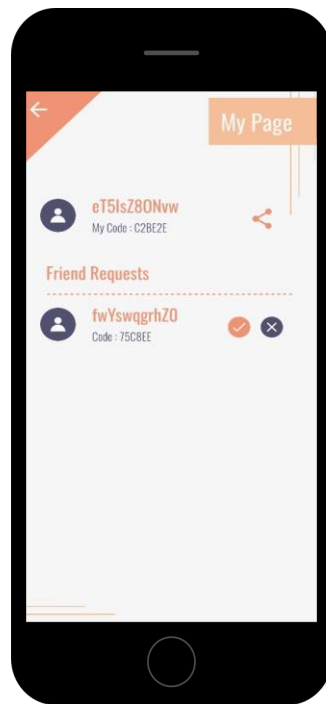
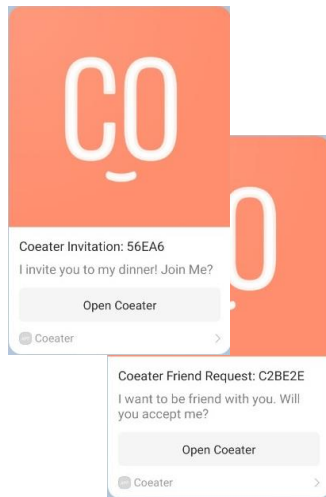
- ✓ Share link to 
- ✓ Able to **accept or decline** the offer before entering co-eating room



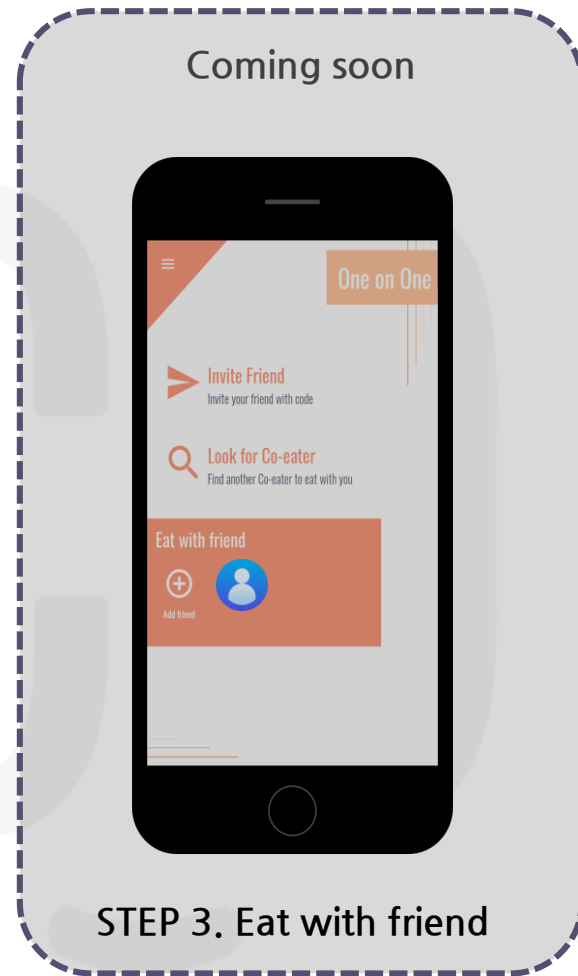
Use Cases 1. Add friend



STEP 1. Share code



STEP 2. Accept request



STEP 3. Eat with friend

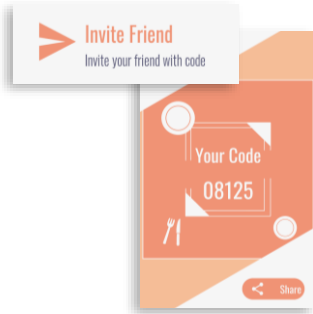
Use Cases

2. Co-eating

STEP 1

Make a room & share

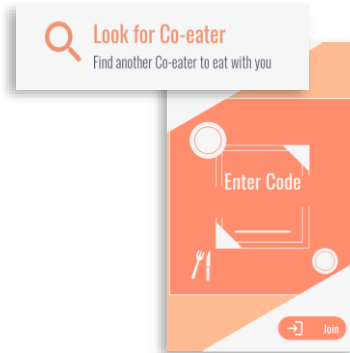
Inviter



STEP 2

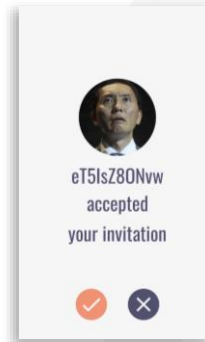
Enter room code

Invitee



STEP 3

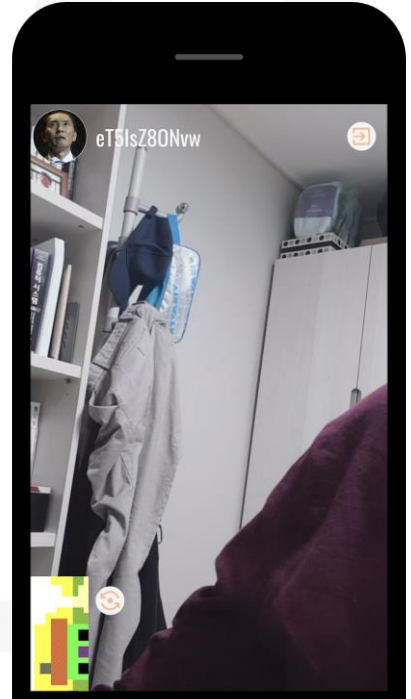
Final accept or decline



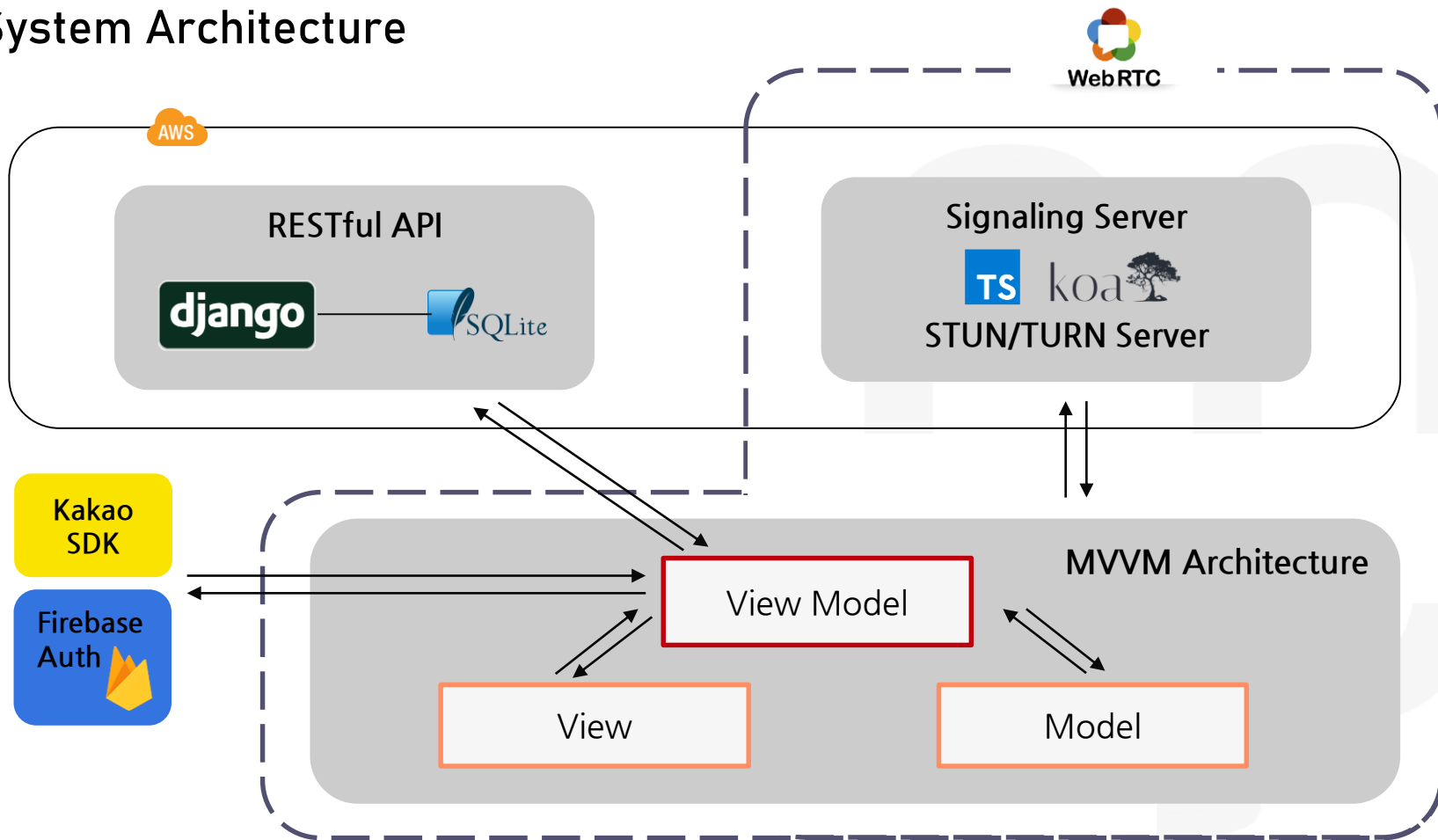
STEP 4

Enter co-eating room

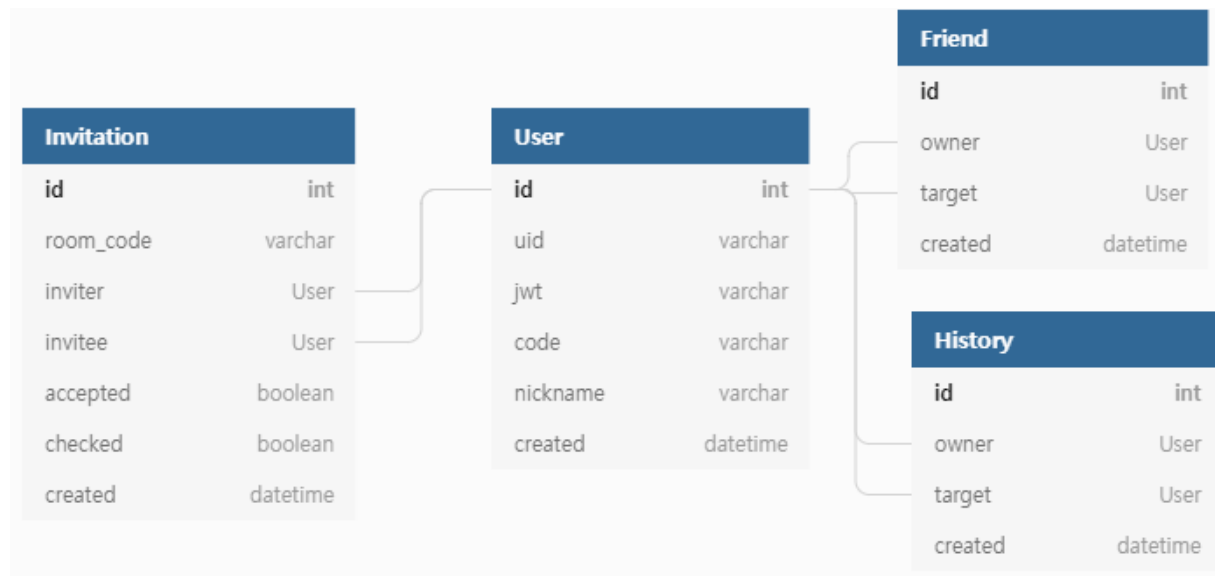
* Toggle camera & screen



System Architecture



System Architecture ERD



Token-based
authentication

Two-way acceptance
(friend & room)

■ Technical Challenges Solved : Server side

1 Database & APIs

- ✓ Developed with **django**
- ✓ Real time modification and upload RESTful API on server wiki

2 Deploy on EC2 instance



2 WebRTC

- ✓ STUN/TURN server to cope with network address translators (NATs) and firewalls
- ✓ Signaling server to coordinate communication before call



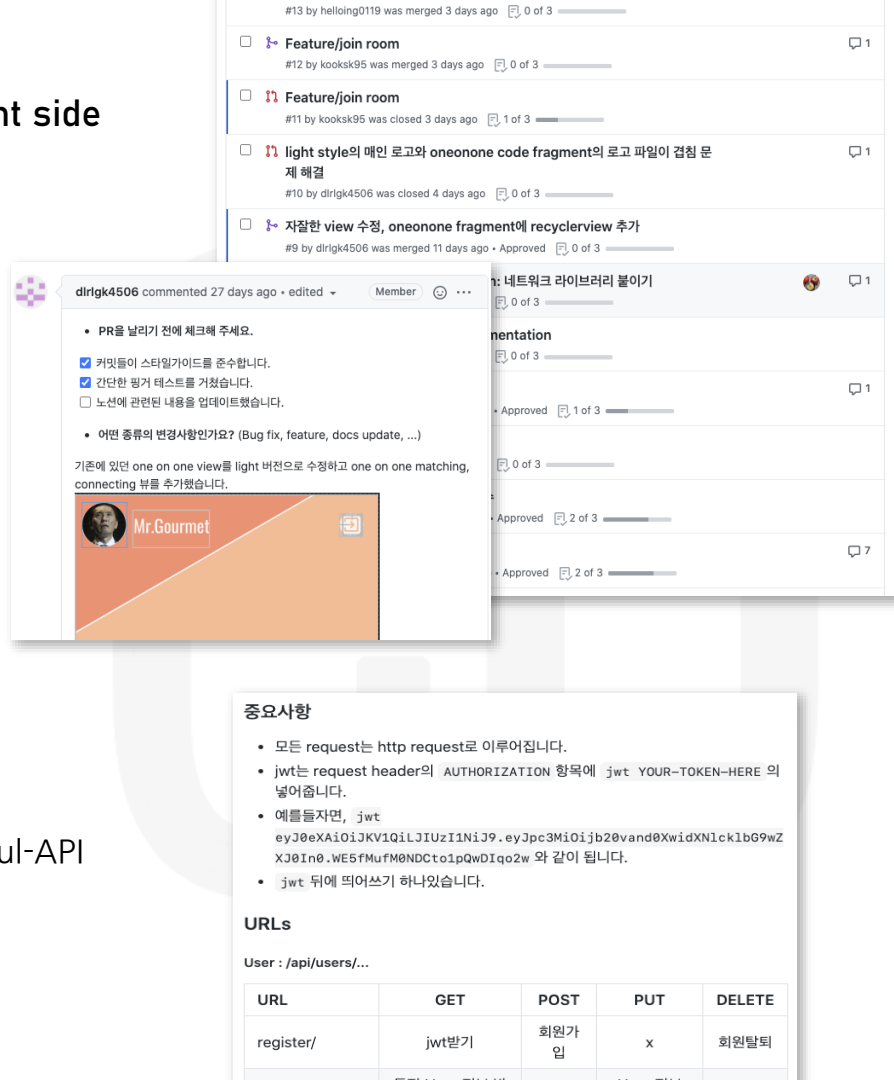
Technical Challenges Solved : Android client side

As developed concurrently...

1 In order to avoid conflict, create a branch for a new feature and open a PULL Request to merge

2 Refer to RESTful API on server wiki

<https://github.com/coeater/coeater-rest/wiki/RESTful-API>



Technical Challenges

Initial evaluation

Success Criteria

1. Development Completion

For pleasant user interaction, latency should be significantly low

→ **Latency < 1s** (while network environment considered)

2. User Satisfaction

Offer smooth co-eating functionally and carefully design UI/UX to deliver a fun dining experience

How to measure

Show synchronized clocks on the screen & measure time difference

Same Wifi

Avg. 0.5 sec

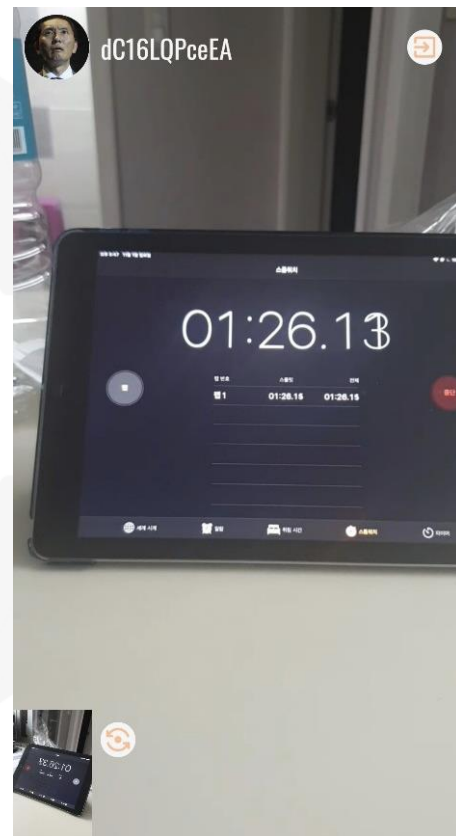
Wifi + LTE (nearby location)

Avg. 0.7 sec

Wifi in distance

Avg. 1 sec

Low latency on same network



Technical Challenges To be solved

Gamification



Simple real-time
game features



Face recognition + Animoji



Uncertainty
HIGH

Inducing Simultaneous Access

Current solution

- Service approach: 'Share with Kakao' automatically enters the code by pressing the access button
- Technical approach: Periodically check the other party's connection and lead to the call immediately.

TODO

- Developing a search function, etc. so that calls can be made without sending invitation link externally



FUTURE PLAN

- | Final Deliverable
- | Roles & Timeline
- | Potential Threats

Final Deliverable and success criteria

What We Deliver

A **videotelephony** android application with **gamification** features, enhancing continuous relationship with fun dining experience

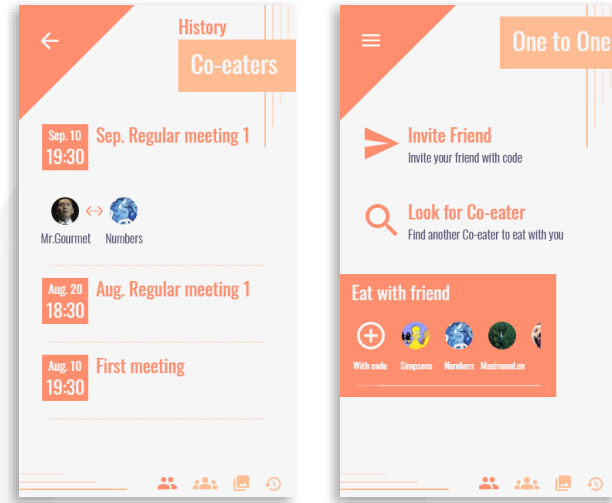
Success Criteria

1. Induce in-app participation

Offer convenient functionalities that naturally lead to co-eating inside the app other than sharing the link

2. User Experience

Add interesting games and improve details of UI/UX to deliver a better dining experience



Roles & Timeline Up to now...



Cho Sesang



Lee Giha



Kim Kiwoong



Kook Seokyoung

Client

Connect WebRTC
Main activity
Share link with Kakao

Main activity
My page
Add friend

UI/UX design
Invitation activity
Match activity

Join Activity

Server

COTURN

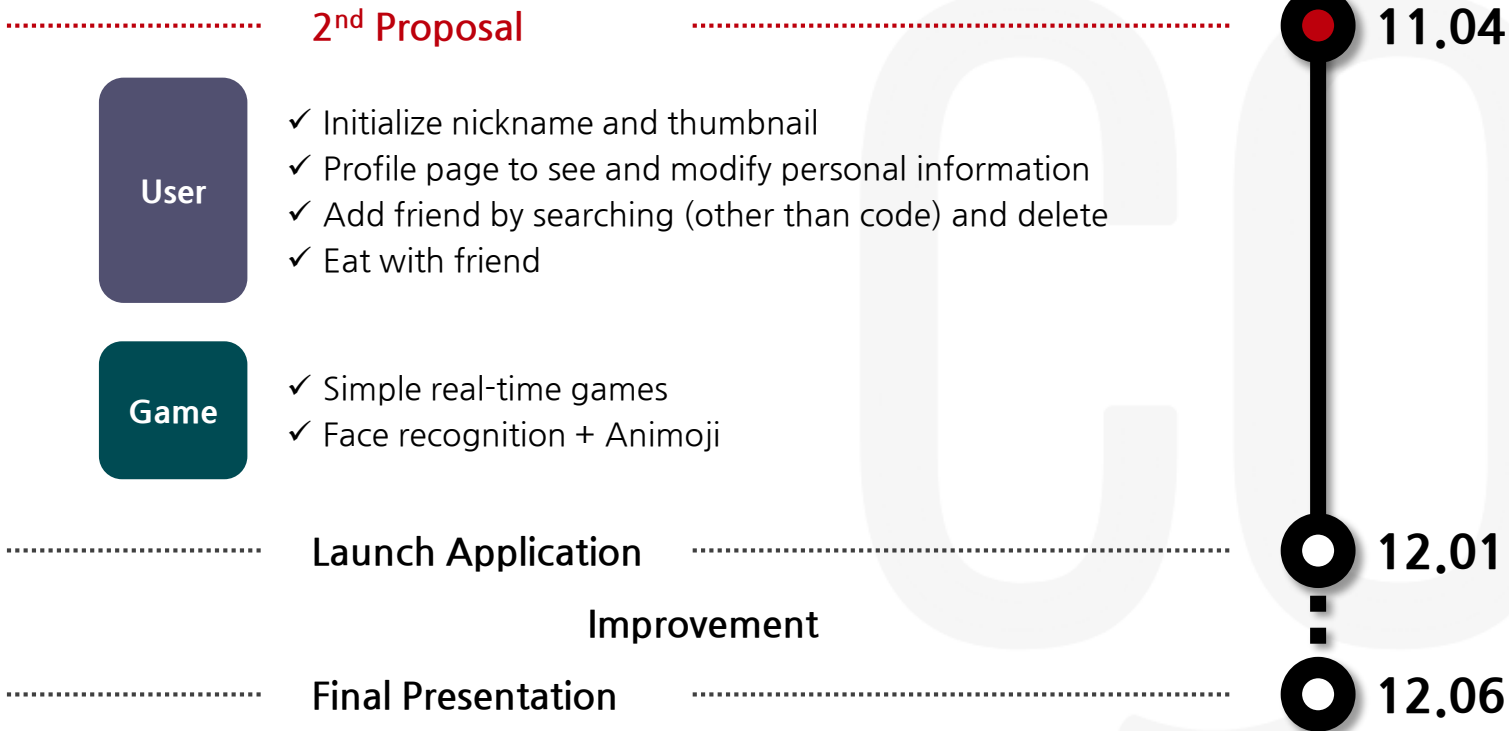
-

RESTful API

Signaling

Roles & Timeline

From now on...

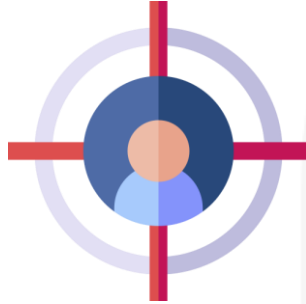


■ Potential Threats



Technical Uncertainty

- ✓ Further research is needed for technical uncertainty of WebRTC



Concentration

- ✓ Focus on dining features for couples



Marketing

- ✓ Need a marketing strategy to promote people to install the app



THANK YOU

Questions?