

MukGo

...

Team 6

Suhyuk Lee

Jaeho Shin

Gina Sohn

Taekmin Kwon

TARGET USER



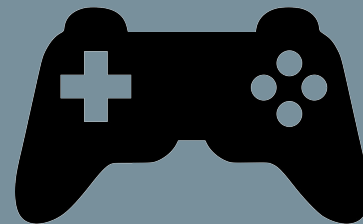
Want to show off their restaurant reviews



Interested in game



Don't want unhealthy reviews



Want to see categorized reviews

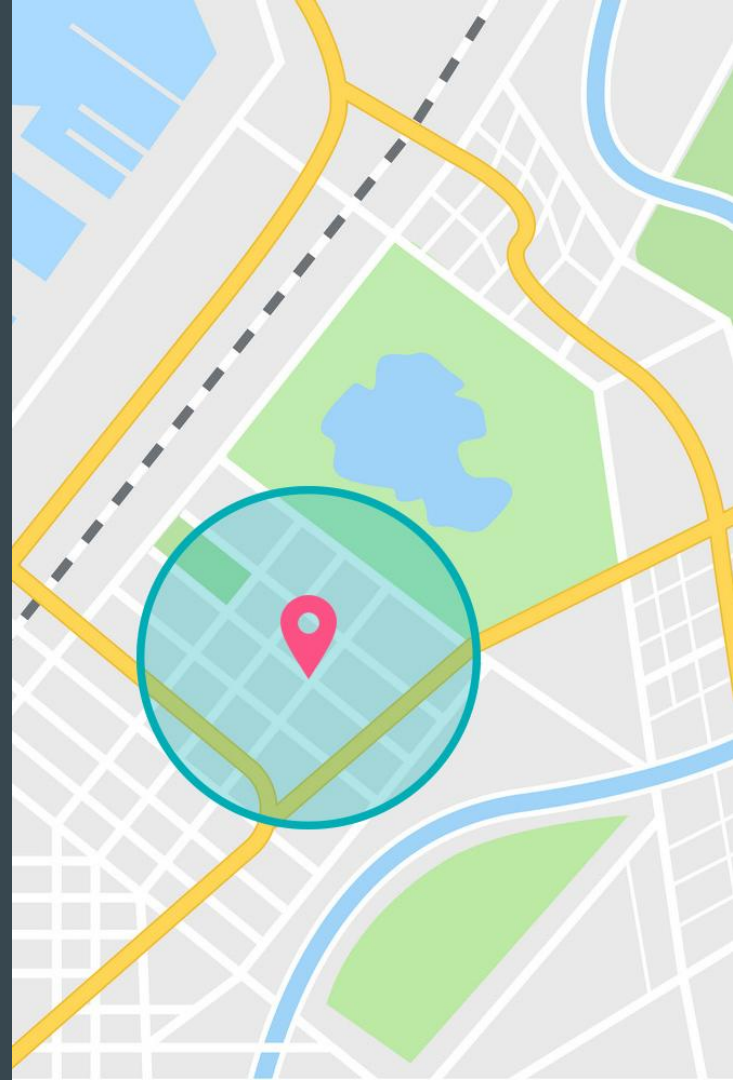


KEY IDEA 1: Location Based

Users can only find restaurants within their **sights**.

Users can only leave reviews on places they **actually visits**.

Get recommendation based on current location.



KEY IDEA 2: Game Characteristic

Motivate users to record and to leave their visits.

Provide gradual services along with users' **level**.

Each restaurants or places has their own history, so there can be **rankings** on some metrics.

Levels

12



다음 레벨까지 12/20 리뷰

Rankings

King in the Mcdonalds (신림점)



100 리뷰



50 리뷰



25 리뷰



10 리뷰

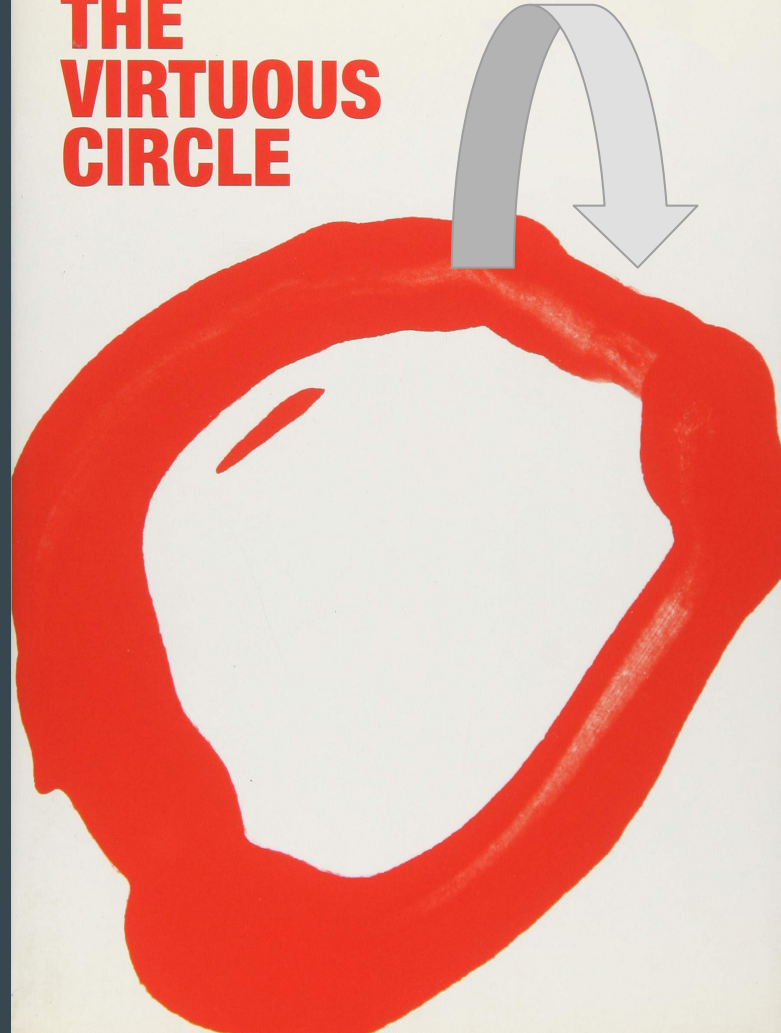
KEY IDEA 3: Personalization

With key idea 1 & 2, meaningful review data will be **accumulated**.

Use the data for making **personal** or **regional** focused information.

The more information users can get, the **more visiting and reviews** they will make.

**THE
VIRTUOUS
CIRCLE**

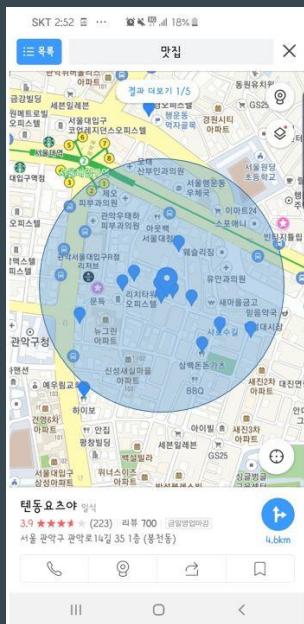


Competitive Landscape

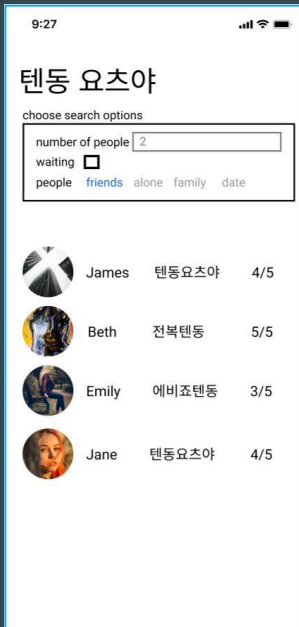
	Google Maps	Dining Code	Mango Plate	MukGo
Concept of level	X	O	O	O
Different services according to level	X	X	X	O
Is the review written right after the user ate?	X	X	X	O
filtering service (for reviews of a single restaurant)	△	△	△	O
Ranking service (among users)	X	X	X	O

USER SCENARIO 1

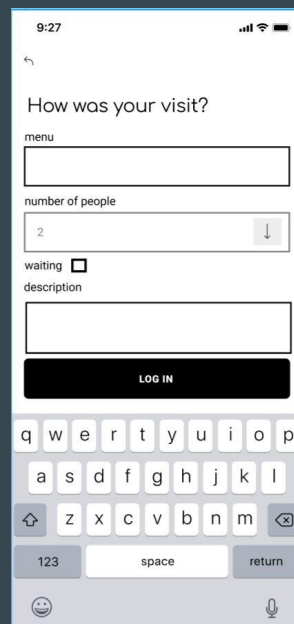
An user is going to have dinner at 샤로수길 but has not decided where to go.



search for restaurants
in her *'sight'*



read reviews by
setting options

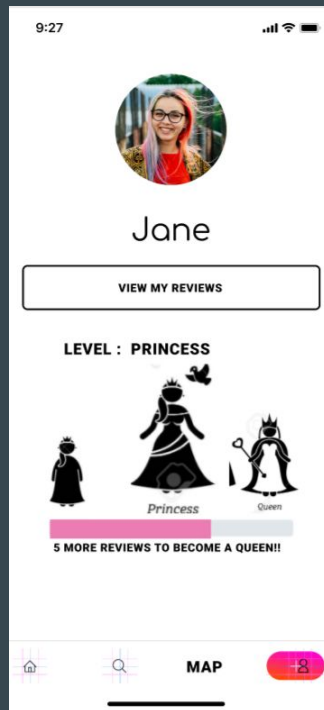


after enjoying dinner
leave a review about the restaurant!

USER SCENARIO 2

An user has left a lot of reviews → **level up !!**

wider '*sight*'
: able to see more restaurants!

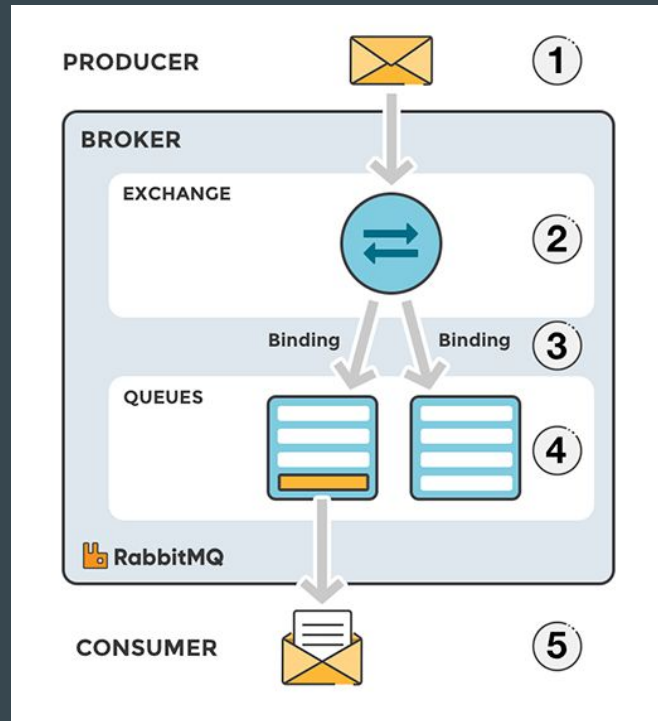


check your **new level!**

TECHNICAL CHALLENGES

- **Multiple servers** running on AWS EC2 instances need to **communicate** each others without worrying about shutdown.
- Multiple clients need to **subscribe** messages from the server.
- AMQP with RabbitMQ

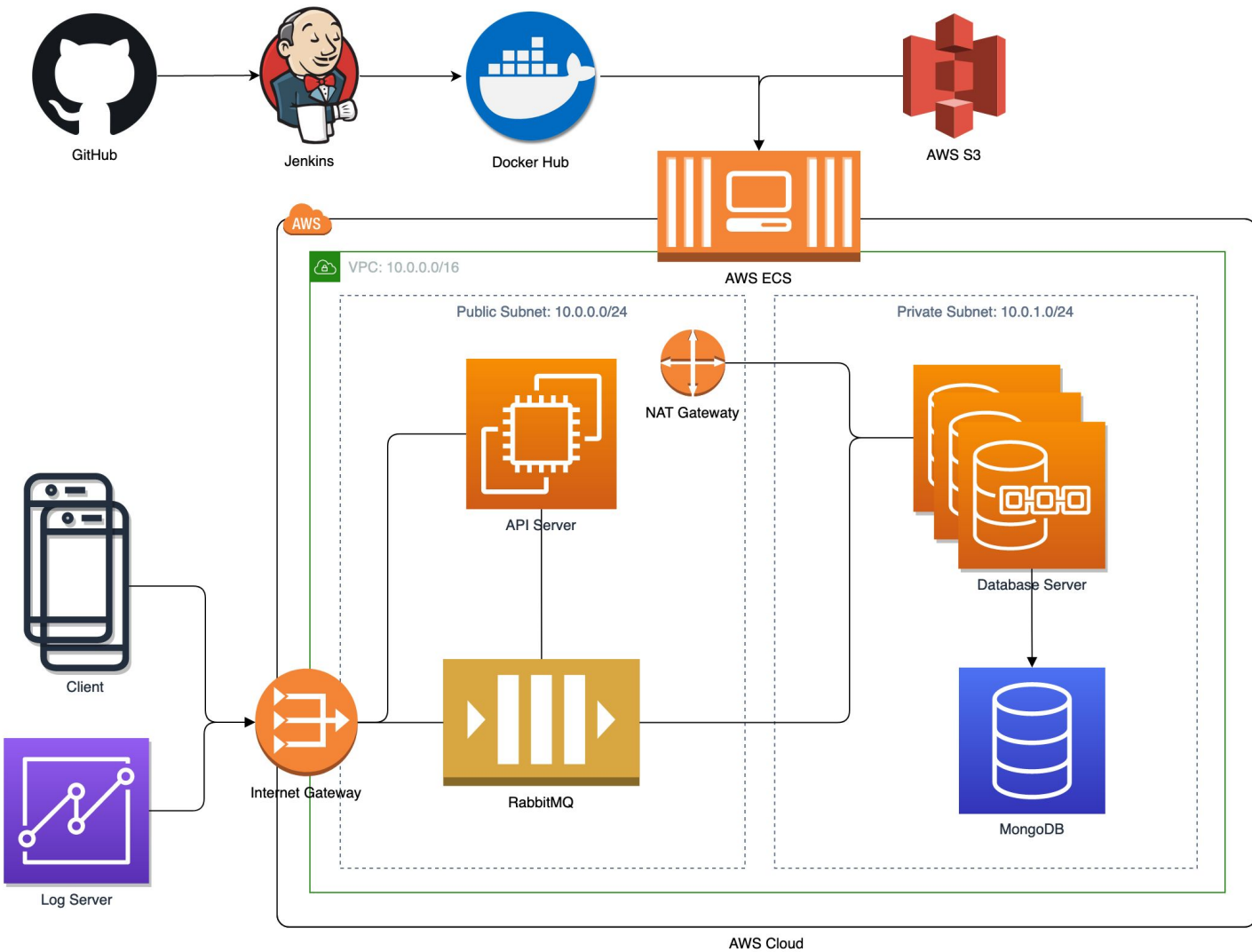
 RabbitMQ



TECHNICAL CHALLENGES

- Need to **deploy** multiple servers into AWS EC2 instances.
- Containerization & CI/CD





PROJECT PLAN

[illegible]

FINAL DELIVERABLE

- MukGo Android application.
- Provide and store a restaurant and game database with stable service.
- Location-based service operation.



SUCCESS CRITERIA

- Development Completion
 - Implement what we want to implement.
- Target User Satisfaction
 - Users can write review based on location.
 - The application has game characteristics. (character, level, ranking, etc.)
 - Users can get high-quality reviews.