

2018.04.02

INSTALINE ANDROID APPLICATION DEMO

DEVELOPERS:

YICHONG GUAN, CHENJIE NI, HAO WANG, ZHIHONG WANG, JIAXIN WU, JIALIANG YI

Contents



- * Problems we are solving
- * The solution we build
- * Technical Discussion
- * Process Discussion
- * Individual Contribution

Problems we are solving



- * Have you ever found calling for reservation is just too inconvenient and inflexible? **(Customer Side)**
- * Have you ever found that you have spent too much time waiting for a seat in the restaurant? **(Customer Side)**
- * Have you spend too much effort on managing customers who crowding at your restaurant entries? **(Restaurant Side)**



Target Users

- * **Customers** who want to line up for restaurants, but it is **not** convenient for them to wait **at the restaurant**, or they want to line up **multiple restaurants**.
- * **Restaurants** do **not** want too many customs **crowd** at the entries.

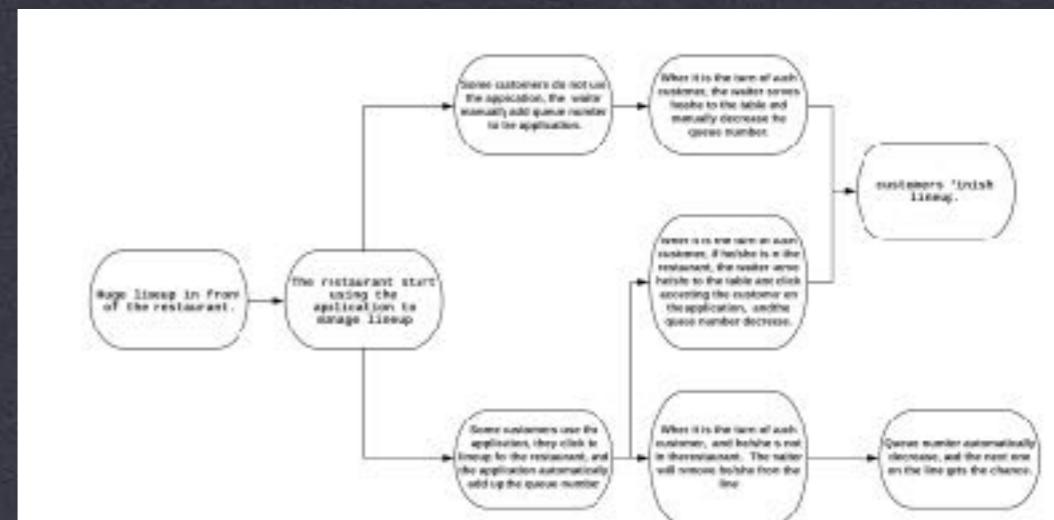
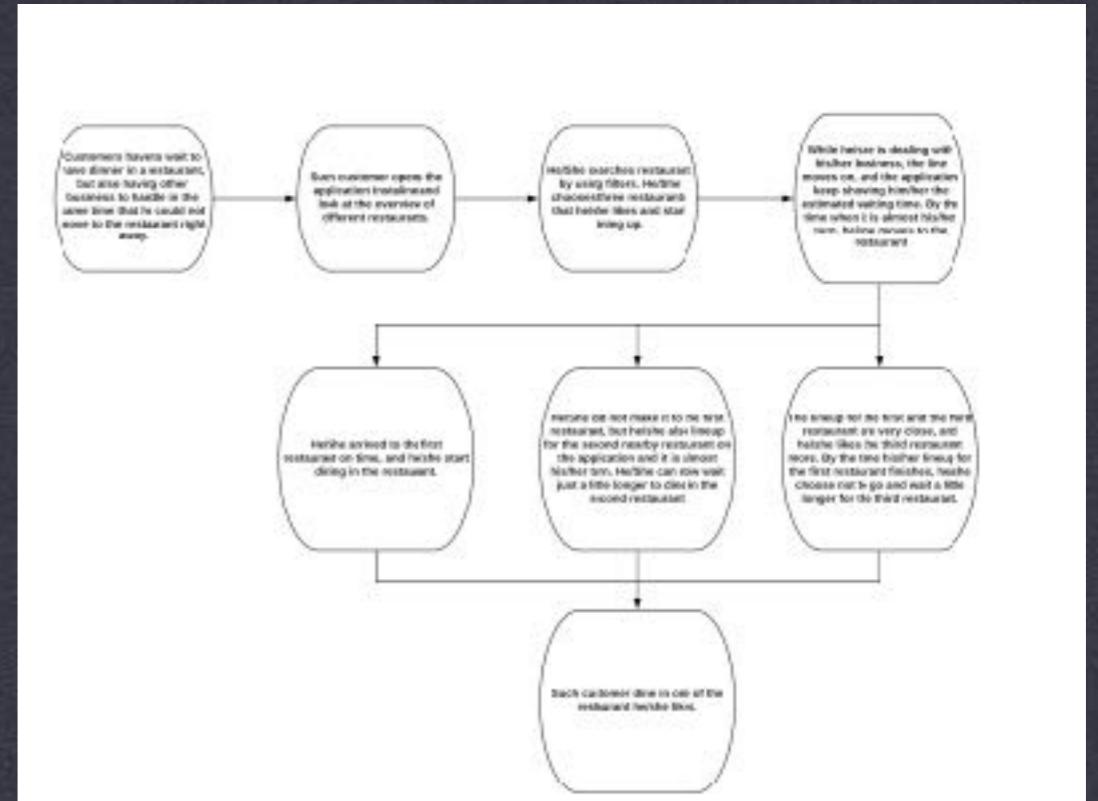
Solution to the problems

- * **INSTALINE (ENTERPRISE)**, our android application will provide a **virtual lining service** to solve variety of lining up inconveniences.
 - * It intends to **free users** from being physically lining up in the restaurant. Users can start or quit lining up at **anytime**, in **anywhere**, for **whatever stores** they wish.
 - * It intends to **give restaurant** convenient management of waiting customers by **a few clicks** on the app.

We developed two separate version for customers and restaurants.

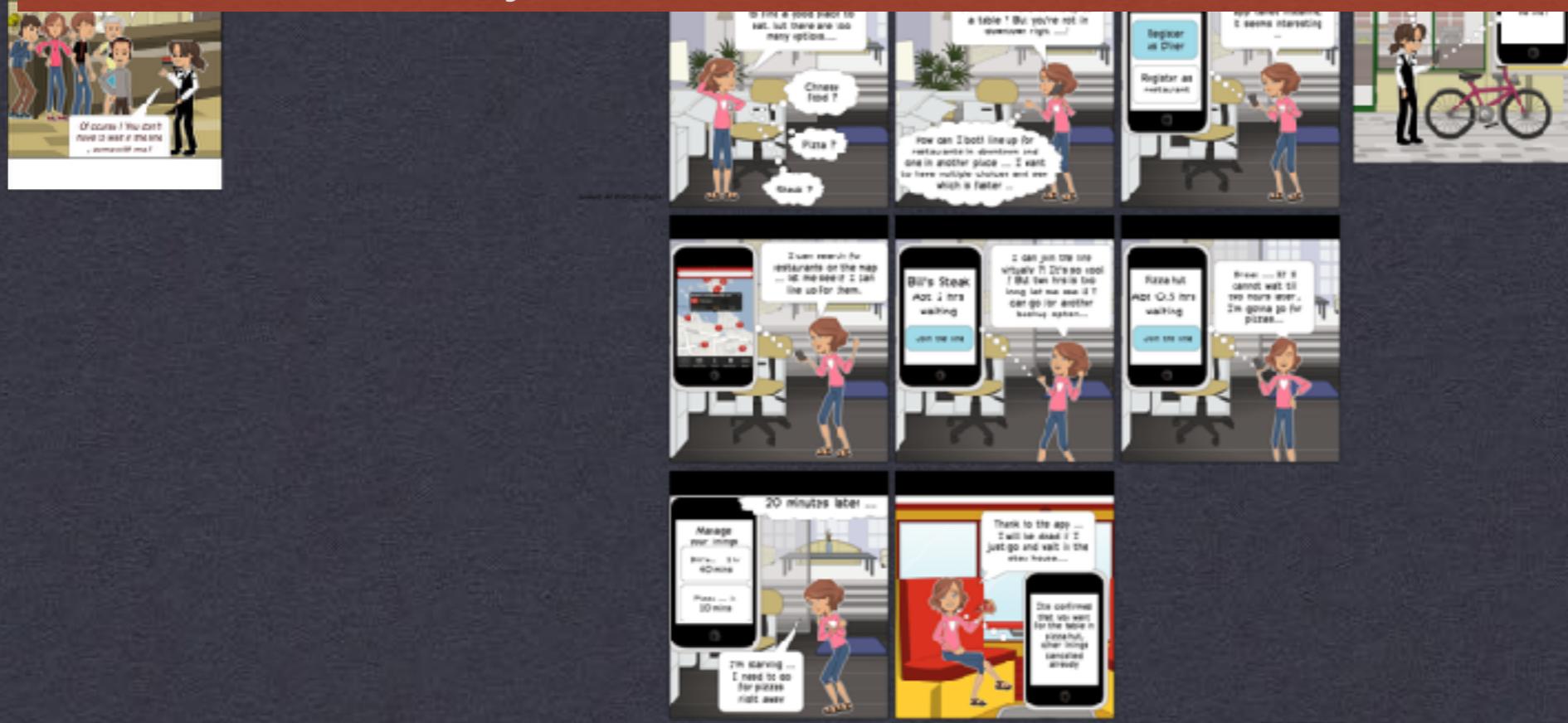
TWO USER WORKFLOWS

CUSTOMER SIDE RESTAURANT SIDE





Why Instaline is valuable to its users?

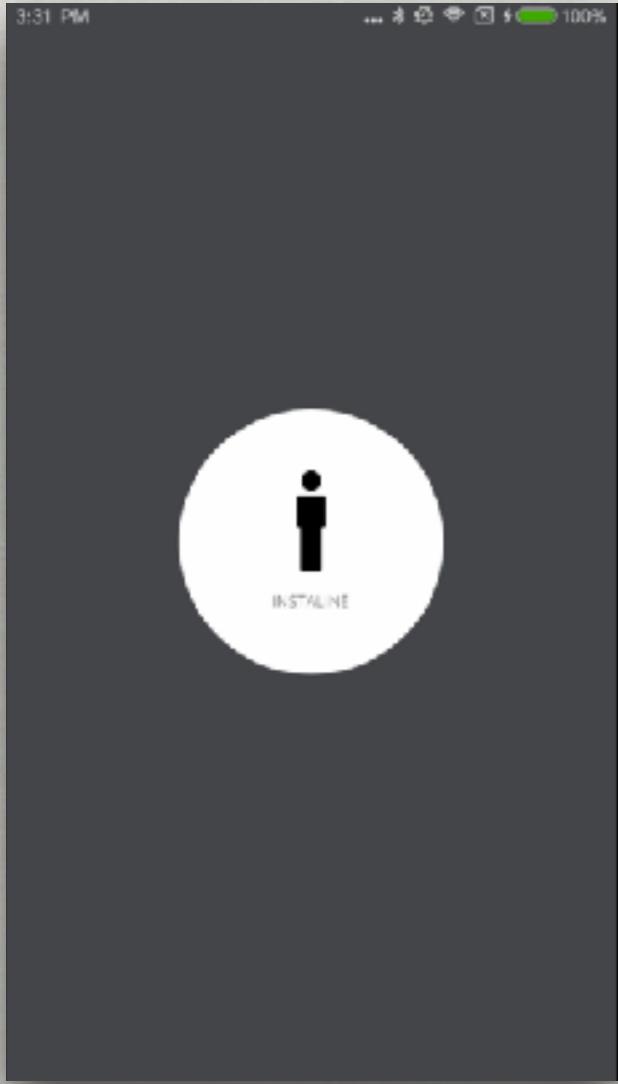


THREE USER STORIES

DEMONSTRATING HOW INSTALINE SOLVES THE PROBLEMS

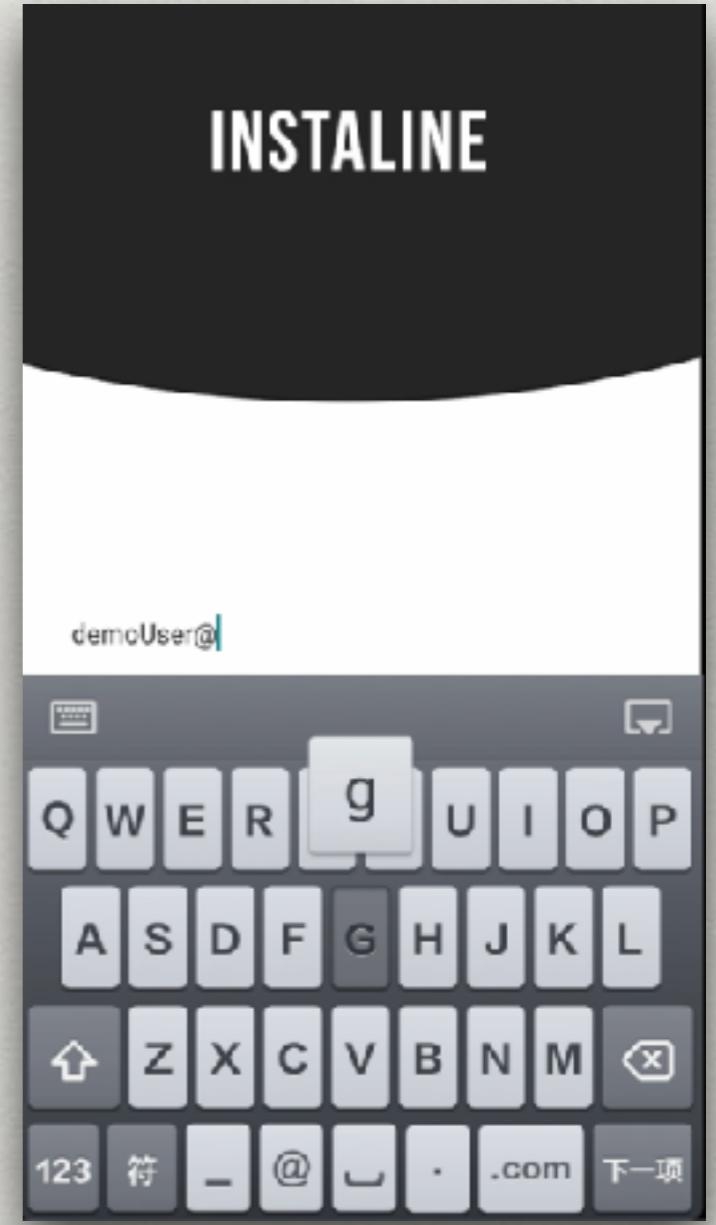
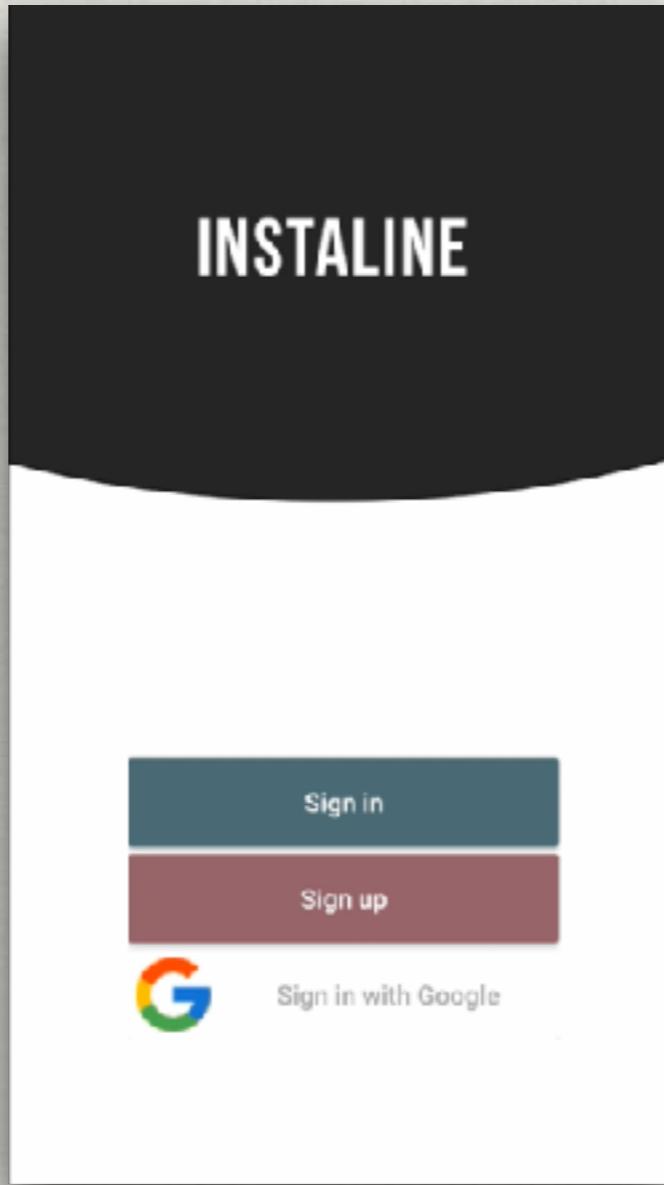
INSTALINE (CUSTOMER SIDE APP)

link to our demo: https://drive.google.com/file/d/11u_5dN_5G838QwiL5LBC5pMv6oLWMppN/view

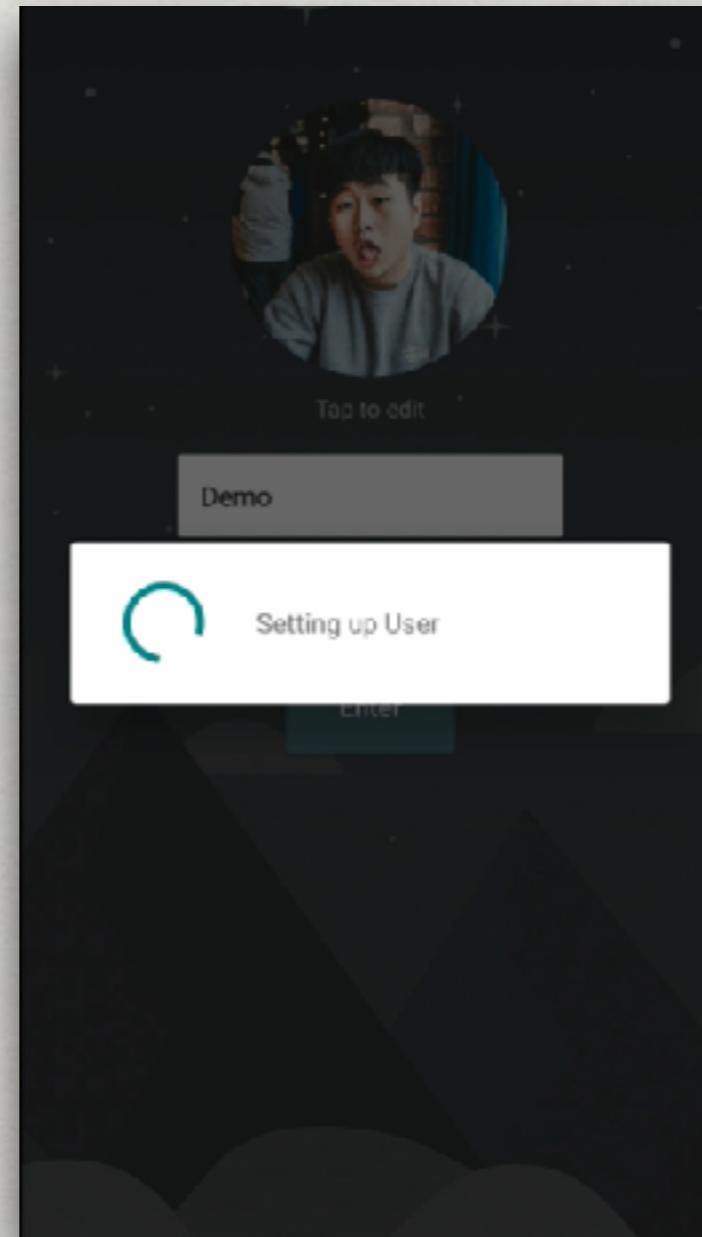
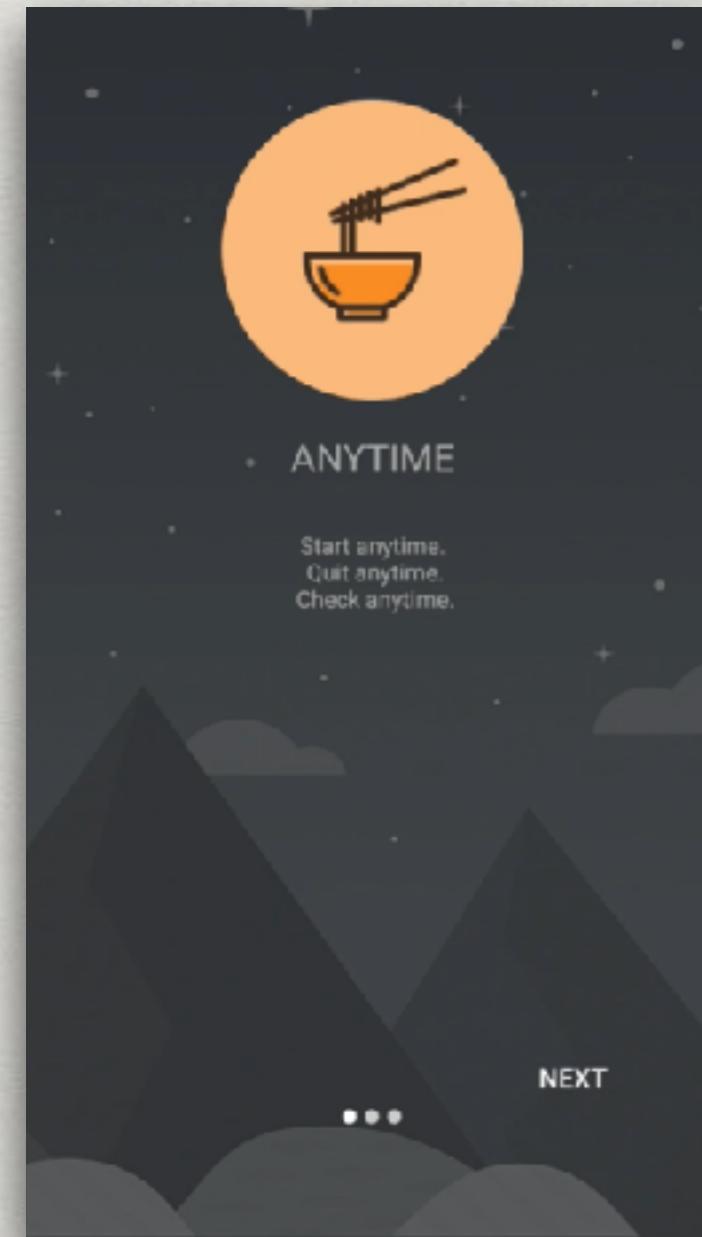


1. Open the app

We use Firebase to manage user accounts



2. Sign up

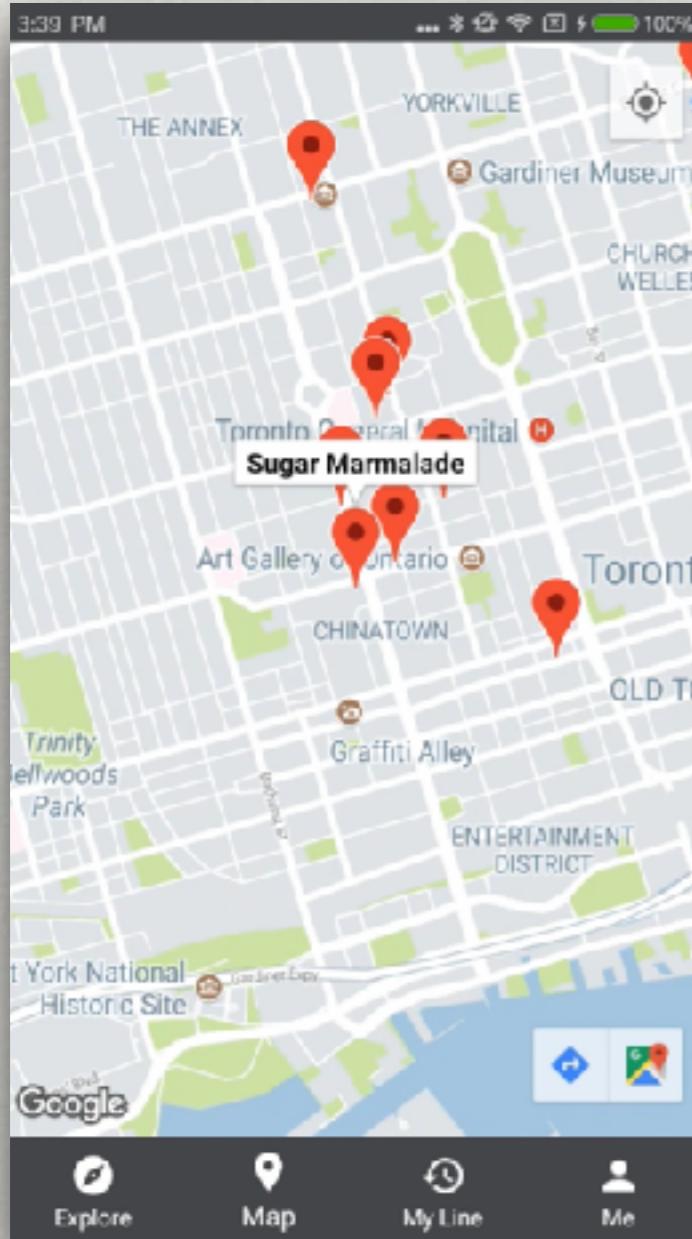


3. Sliding pages introducing our app

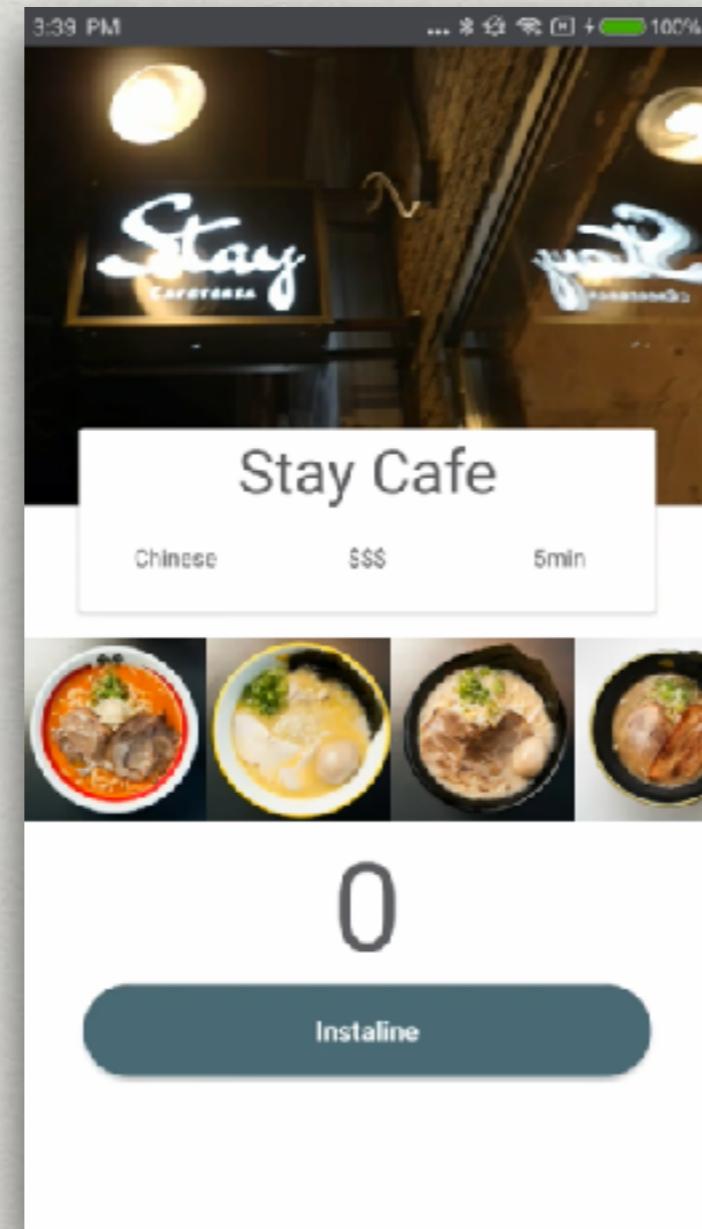
4. Setting up

5. Entering Explore page

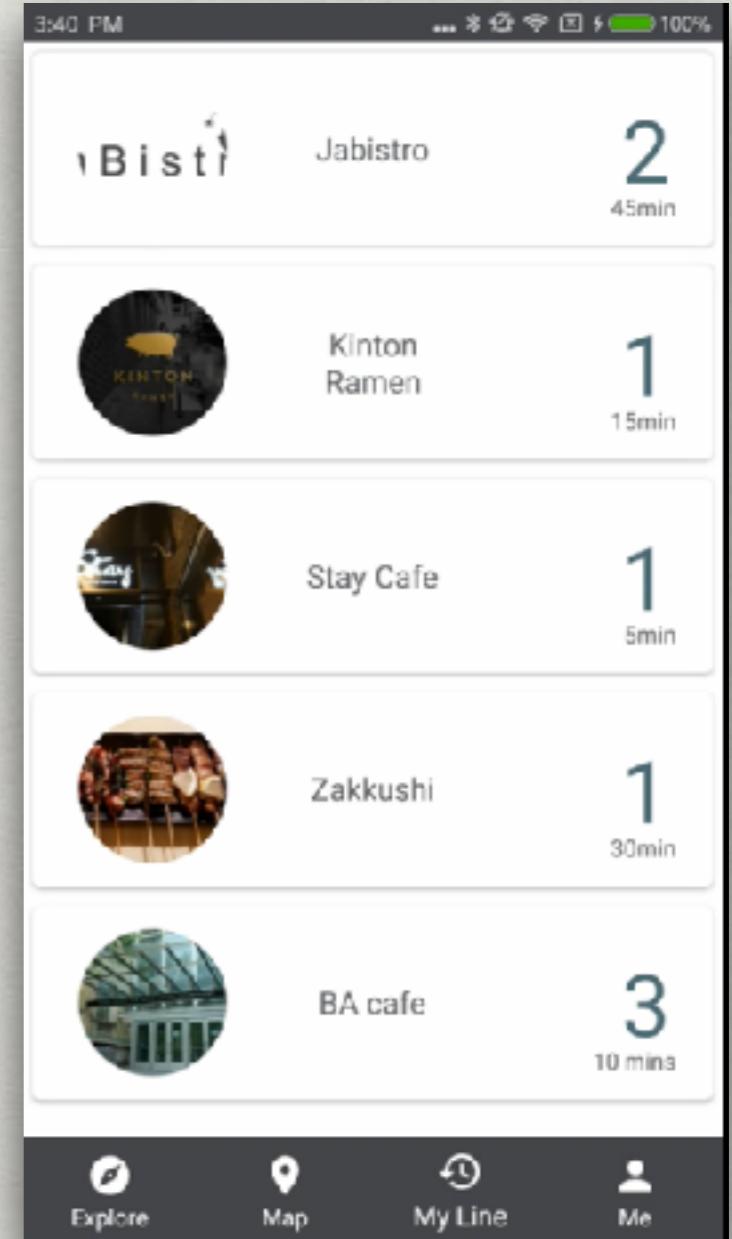
You can always view your up-to-date ranking and time left for waiting!



6. You can see the locations of the restaurants on Map



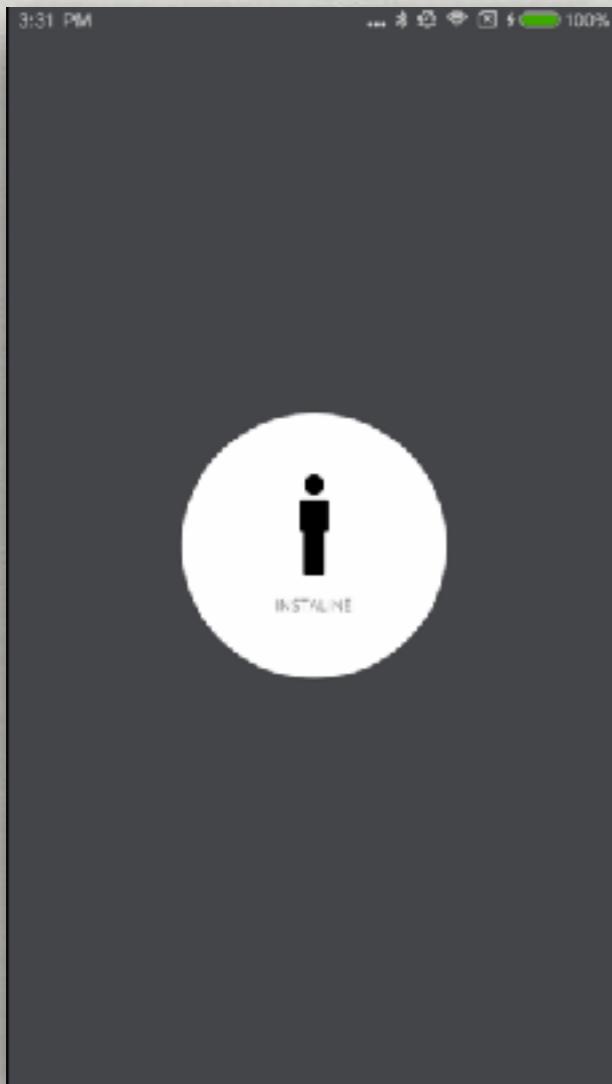
7. Click the restaurant you want to line up on the explore page, then click Instaline



8. You can at most line up for 5 restaurants at a time, which can be viewed on My Line page

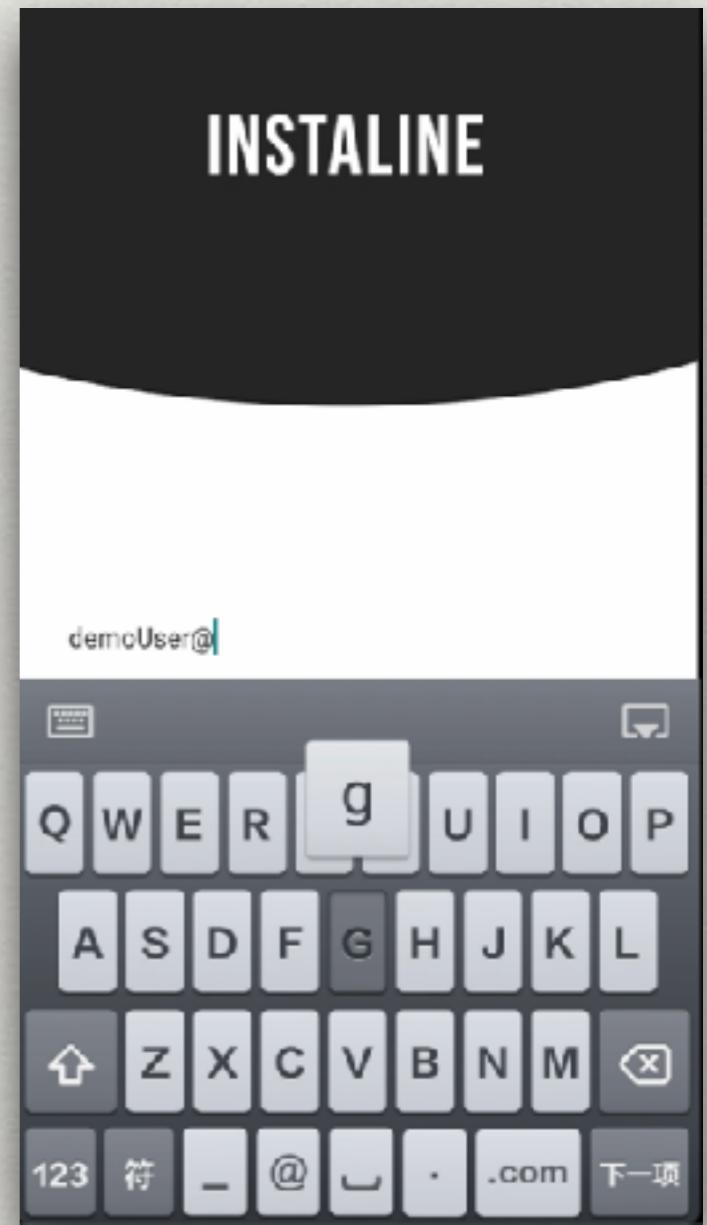
INSTALINE ENTERPRISE

(RESTAURANT SIDE APP)

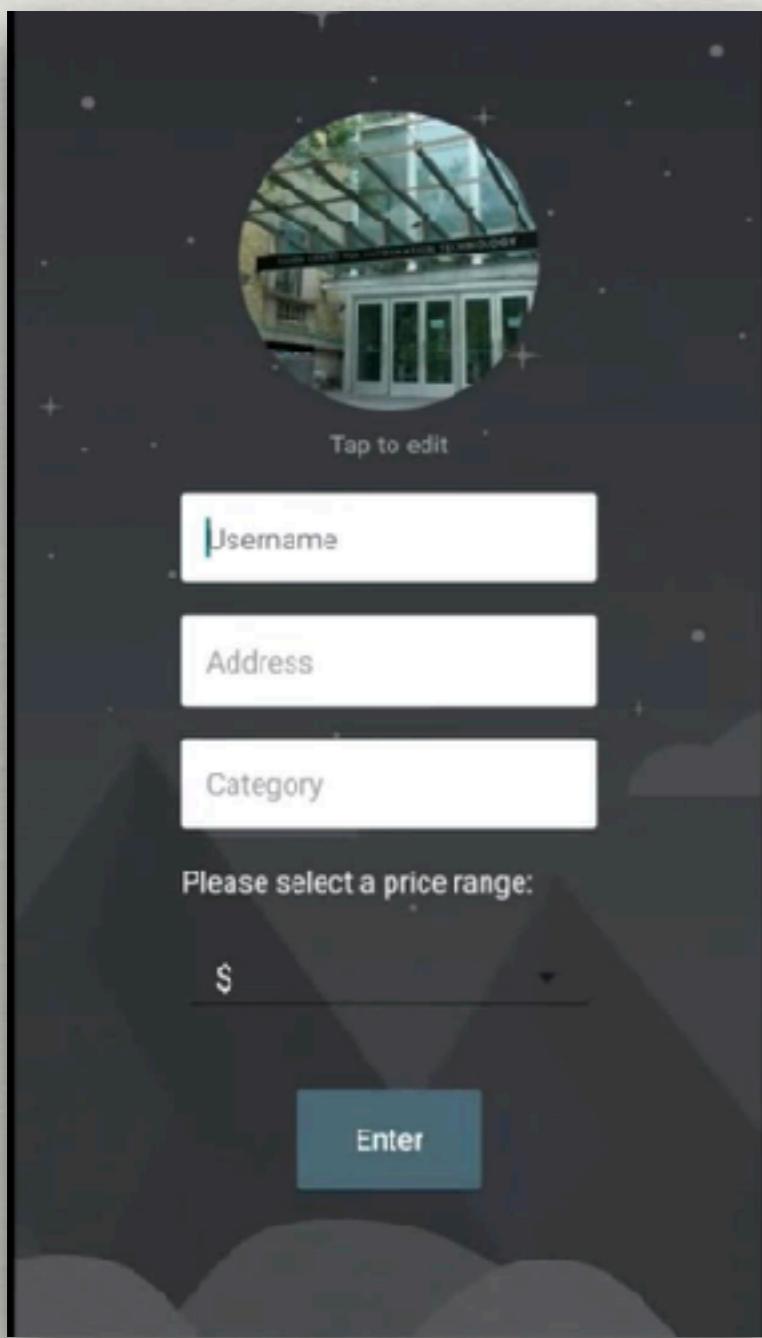


1. Open the app

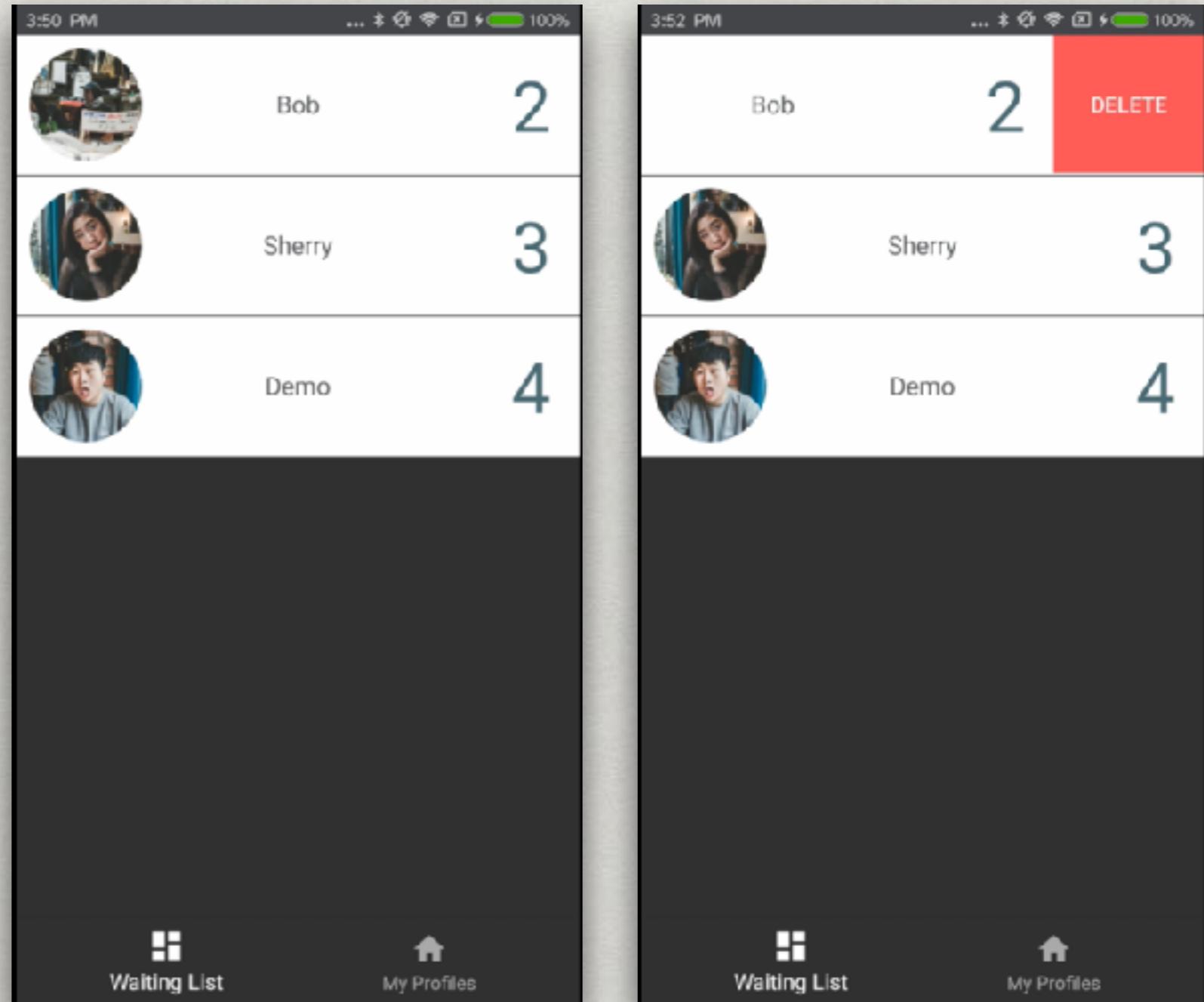
We use Firebase to manage user accounts



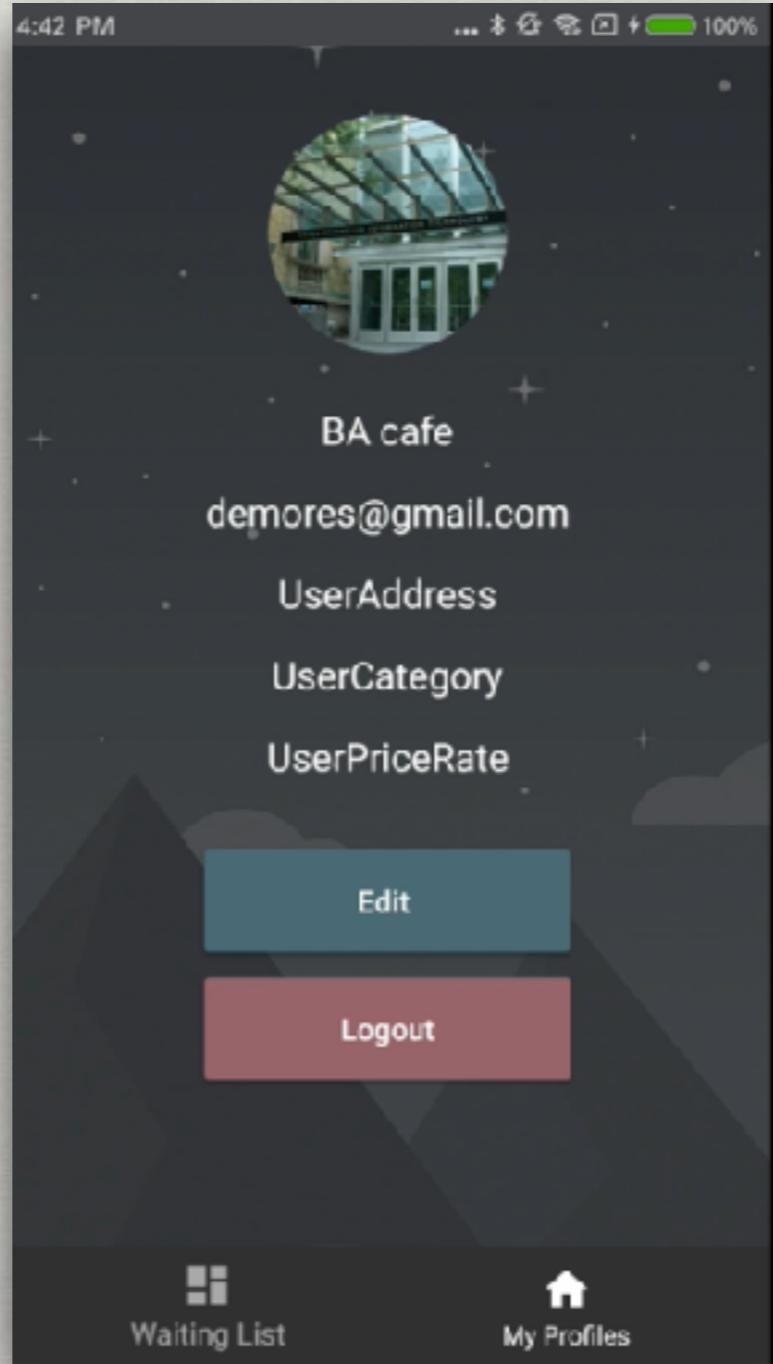
2. Sign up



3. Fill in the info including address, category and price range



4. Waiting List in the restaurant side will be update instantaneously whenever the clients arrives, the restaurant also can remove the client from the list



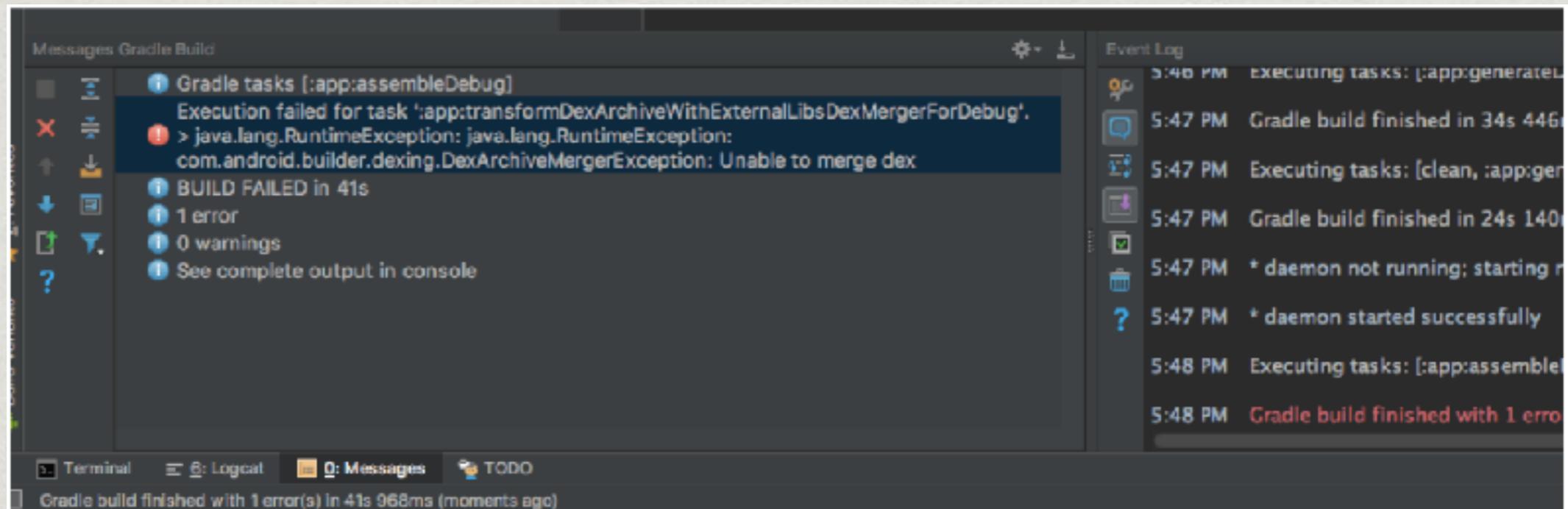
5. You can always edit the profile as well as log out the app

Technical Discussion

- * Android Studio Environment settings
- * Frontend Changes through the iterations
- * Connection between two apps
(Instaline and Instaline Enterprise)

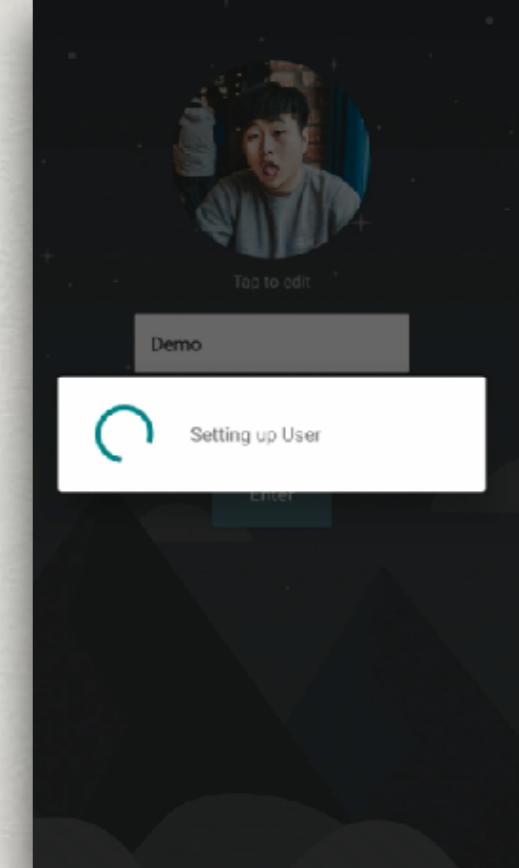
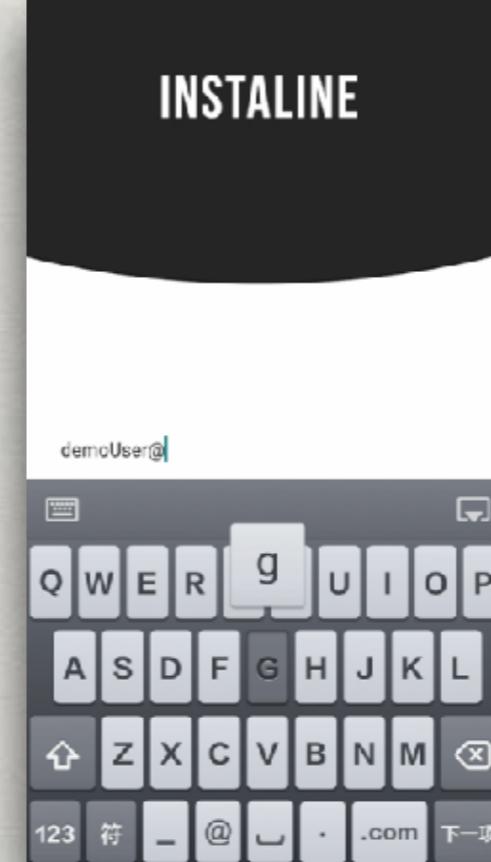
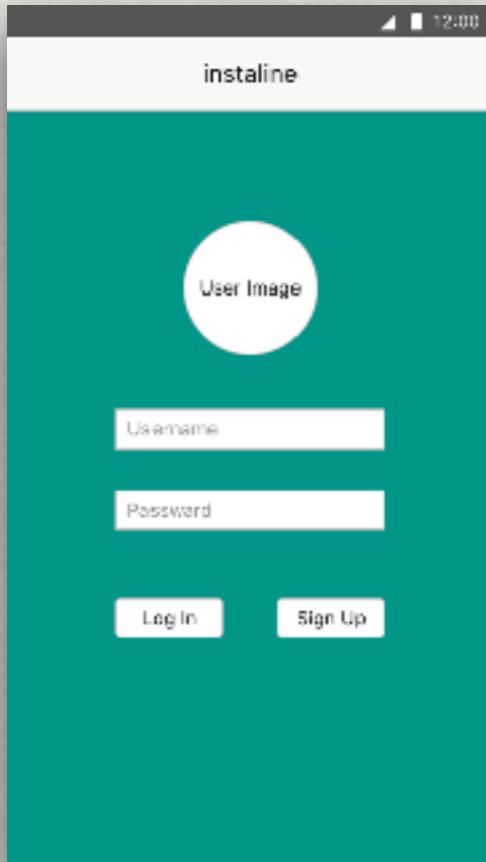
Android Studio Environment settings

- * Android studio is a highly **advanced android IDE**
- * **BUT** when it comes to a team of 6 members all new to it, things getting interesting(horrible), especially for environment settings(dependency conflict), we spent quite a long time to get a bug free gradle file. In the first few weeks, we solve this problem by cleaning and rebuilding repeatedly , then more features added, we have to created a new project with old code, restarted android studio and work on the new project we created. At the end, by the help of stack overflow, we add dependencies one by one and make sure that versions are in consistency.



Frontend Changes through the iterations

- * For example, after completed the restaurant list of customer side, we found out that if there was no search function, it would be hard for users to find their ideal restaurant(s) once there existing too many restaurants in the list. So we **added the search function** for the customer side app. Except this, we also improved some details of the front-end design, you can see the difference between the **mock-ups and the real application**.



Connection between two apps

- * We use **firebase as a server** to manage the database between two apps, which means this database should **monitor data changes from different apps accurately**. Thanks to this development, we get more familiar with the firebase server and back-end data manipulation.

Database

instaline-b3b17

Users

- 7NnAp58X8dRMibiOxDjp8I1zRWD2
 - profileImage: "https://firebasestorage.googleapis.com/v0/b/instaline-b3b17.appspot.com/..."
 - username: "Bob"
 - imageURL: "Bop"

User

Restaurant

instaline-b3b17

Restaurant

- 1LGKfASAc1NLZUp6ANxtW6ehHm2
- Jabistro
- Kintonramen
 - TimeStamp: 7
 - UserQueue
 - address: "51 Baldwin St, Toronto, ON M5T 1L1"
 - category: "Japanese"
 - coverImage: "https://firebasestorage.googleapis.com/v0/b/instaline-b3b17.appspot.com/..."
 - lastDelete: 0
 - name: "Kinton Ramen"
 - price: "\$\$"
 - queueNumber: 3
 - waitTime: "15min"
 - waitTime: "15min"
 - queueNumber: 3

profileImage: "https://firebasestorage.googleapis.com/v0/b/instaline-b3b17.appspot.com/..."

username: "Bob"

imageURL: "Bop"

Process Discussion

- * Artifacts
- * Github Workflow
- * Significant Decisions

Artifacts

- * We have a **Scrum board** in **Google Doc**, which is used to keep track of the work that needs to be done in each deliverable.
 - * We **prioritize tasks** shown on the board based on the importance relative to **the user stories**.
 - * Tasks are **assigned based on the job, interest and time schedule** of each individual in our team.
 - * We also created **burndown charts** to visualize the progress we made during this term.

ITERATION 3: Installing Scrum Board			
Note: You can click on the story images to view details.			
STORIES	TO-DO	IN PROGRESS	TESTING
	Jerry: Fixing bugs of the Google Map display in the Fragment	Jack: Dependency problem that leads to compile error	Eason: Real data fetched from Firebase
		Jerry: Refresh feature for installed	
<img alt="Story image			

Scrum board



February



March

Github Workflow

- * Each new feature has its own branch. Master keeps the stable version.
- * **Frontend UI and Backend Firebase.** Once the work is done, he/she will do pull requests: Feature branch -> Master. In the end, Jerry and Jack will **review the pull-requests** then **merge them into master with conflicts resolved**.
- * This workflow **minimizes conflicts** caused by everyone directly pushing to master. Also, one **advantage** is that we can always **keep a working version of this app**.

Significant Decisions

- * Good:
 - * Plan meeting early and make meeting agenda in advance
 - * Sketch UI design before implementation
 - * Weekly online sync meeting
- * Bad:
 - * Technical session
 - * Individual separate branches
 - * Test on different versions of Android devices

Individual Contribution

- * Eason(Yichong Guan)
- * Jack(Chenjie Ni)
- * Hao(Hao Wang)
- * Bill(Zhihong Wang)
- * Jasmine(Jiaxin Wu)
- * Jerry(Jialiang Yi)

Eason (Yichong Guan)

- * Backend Firebase Designer
- * Decide how the data is stored and retrieved in a feasible way
- * I learned how to design a good database with good structure and simple data, how to see further to design a good database in the early development stage so that the database does not have to be changed all the time as more feature is being added.

Jack (Chenjie Ni)

- * Product Manager
 - * Knowing customers' needs and main features of our app
 - * Assign general tasks to every team member
- * How to organize a team
- * How to describe/explain customers' needs to development team
- * Monitor the quality of our product

Hao (Hao Wang)

- * Frontend Designer
- * Frontend logic on how data is observed and what actions it will perform
- * Mainly focused on user registration and profile update
- * Learned:
 - * Collaboration with teammates
 - * How to get familiar with new stuff in a short time
 - * Find out what I need to improve in future terms: clean architecture and code, effective database design...

Bill (Zhihong Wang)

- * Frontend Designer
- * Frontend structure view design
- * User input gesture
- * I learned how to design user views by linear layout, relative layout, text view, image view, and card view; how to setup an activity files to connect the data with xml files; how to transfer the activity file to a fragment file by searching stack overflow; how to find and use helpful api and library, like the google map api and the swipe library we used; how to monitor user gestures by using android built in attributes.

Jasmine (Jiaxin Wu)

- * Scrum Master
- * Coordinate with the availability of each team member
- * Overall testing
- * I learned how to communicate with all team members as well as coordinate everyone's availability. For the development, I learned that using online resources is really useful, like using knowledge on Stackoverflow to solve bugs. Also, some existing API is really useful that we can directly use like the SearchView. For the testing, I learned that it is always a good idea to test backend and frontend separately which helps locating the bug quickly.

Jerry (Jialiang Yi)

- * Frontend UI Designer and some full stack
- * Leader of the frontend group
- * Make decisions on how each page should look like
- * Connect front end with database
- * Learn android development from scratch. Get familiar with android ui components and their lifecycle. Worked with firebase database, specifically, its authentication functionalities and data listeners. Communicate with backend designer to retrieve data from backend and display them accordingly in the front end. Moreover, I learn to use libraries from the internet that developed by others, import and modified them in order to use them in our project.