# Team Project - Deliverable 2

## Due: Friday, March 9 @ 23:59

The goals of this deliverable/milestone are as follows:

- Get you to build the first presentable version of your prototype
  (i.e. Demonstrate working software)
- Require you to prioritize tasks and make product decisions
  (the deadline is fixed, you get to choose what to build by the deadline)
- Encourage you to reflect on your team (and individual) work habits

# YOUR PRODUCT/TEAM NAME

> _Note:_ This document is meant to be written during (or shortly after) your initial planning meeting.
> It does not really make sense for you to edit this document much (if at all) while working on the project - Instead, at the end of the planning phase, you can refer back to this document and decide which parts of your plan you are happy with and which parts you would like to change.

## Iteration XX

* Start date: FILL IN THE DATE WHEN YOU HAD YOUR FIRST TEAM PLANNING MEETING
* End date: FILL IN THE DATE WHEN YOU ARE PLANNING TO HAVE YOUR REVIEW MEETING

## Process

(Optional:) Quick introduction

#### Roles & responsibilities

Describe the different roles on the team and the responsibilities associated with each role.

#### Events

Describe meetings (and other events) you are planning to have:

* When and where? In-person or online?
* What's the purpose of each meeting?
* Other events could be coding sessions, code reviews, quick weekly sync' meeting online, etc.

#### Artifacts

List/describe the artifacts you will produce in order to organize your team.

* Artifacts can be To-do lists, Task boards, schedule(s), etc.
* We want to understand:
  * How do you keep track of what needs to get done?
  * How do you prioritize tasks?
  * How do tasks get assigned to team members?

#### Git / GitHub workflow

Describe your Git / GitHub workflow.
Essentially, we want to understand how your team members share a codebase and avoid conflicts.

 * Be concise, yet precise.
For example, "we use pull-requests" is not a precise statement since it leaves too many open questions - Pull-requests from where to where? Who reviews the pull-requests? Who is responsible for merging them? etc.
 * If applicable, specify any naming conventions or standards you decide to adopt.
 * Don't forget to **explain why** you chose this workflow.

## Product

#### Goals and tasks

 * Describe your goals for this iteration and the tasks that you will have to complete in order to achieve these goals.
 * Order the items from most to least important.
 * Feel free (but not obligated) to specify some/all tasks as user stories.

#### Artifacts

List/describe the artifacts you will produce in order to present your project idea.

 * Artifacts can be text, code, images, videos, interactive mock-ups and/or any other useful artifact you can think of.
 * Make sure to explain the purpose of each artifact (i.e. Why is it on your to-do list? Why is it useful for your team?)
 * Be concise, yet precise.
   For example: "Build the website" is not precise at all, but "Build a static home page and upload it somewhere, so that it is publicly accessible" is much clearer.

# YOUR PRODUCT/TEAM NAME

> _Note:_ This document is meant to be written during (or shortly after) your review meeting, which should happen fairly close to the due date.
>
> _Suggestion:_ Have your review meeting a day or two before the due date. This way you will have some time to go over (and edit) this document, and all team members should have a chance to make their contribution.

## Iteration XX - Review & Retrospect

* When: FILL IN THE DATE WHEN YOU ACTUALLY HAD YOUR REVIEW MEETING
* Where: PHYSICAL LOCATION AND/OR ONLINE

## Process - Reflection

(Optional) Short introduction

#### Decisions that turned out well

List process-related (i.e. team organization) decisions that, in retrospect, turned out to be successful.

* 2 - 4 decisions.
* Ordered from most to least important.
* Explain why (i.e. give a supporting argument) you consider a decision to be successful.
* Feel free to refer/link to process artifact(s).

#### Decisions that did not turn out as well as we hoped

List process-related (i.e. team organization) decisions that, in retrospect, were not as successful as you thought they would be.

* 2 - 4 decisions.
* Ordered from most to least important.
* Feel free to refer/link to process artifact(s).

#### Planned changes

List any process-related changes you are planning to make (if there are any)

* Ordered from most to least important.
* Explain why you are making a change.

## Product - Review

#### Goals and/or tasks that were met/completed:

* From most to least important.
* Refer/link to artifact(s) that show that a goal/task was met/completed.
* If a goal/task was not part of the original iteration plan, please mention it.

#### Goals and/or tasks that were planned but not met/completed:

* From most to least important.
* For each goal/task, explain why it was not met/completed.
  e.g. Did you change your mind, or did you just not get to it yet?

## Meeting Highlights

Going into the next iteration, our main insights are:

* 2 - 4 items
* Short (no more than one short paragraph per item)
* High-level concepts that should guide your work for the next iteration.
* These concepts should help you decide on where to focus your efforts.
* Can be related to product and/or process.

TA’s beedback：
https://docs.google.com/document/d/1zbQ-8OLhiaLocrX-kx4eQDMXzanKOb3cBkwZAulBvSg/edit?usp=sharing

Gramar check：https://app.grammarly.com

Demo video script 1：

Step1:
(Background: the Instaline icon)

Hello, welcome to team 1, deliverable 2, demo video.

Have you ever found calling for reservation is just too inconvenient and inflexible?
Have you ever found that you have spent too much time waiting for a seat in the restaurant?

Or, you spend too much effort on managing customers who crowding at your restaurant entries?

Instaline, our android application will provide a virtual lining service to solve variety of lining up inconveniences. While online/phone reservation is not flexible and waiting in a single restaurant is too frustrating, Instaline 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.

Step 2:
(Background: screen shot video)

(Click instaline app)
We use Firebase to manage user accounts. Now, lets pick a handsome profile picture for ourselves.

(After login, the background should be the Explore page, the main functions are in Explore/Content/My line pages)
Entering the app, we provide users a list view and a map view of nearby restaurants. Let us go with list view this time. A search is provided to filter restaurants by keywords.

(Content page)
Once the user selected a restaurant, detail information of the restaurant including the price range, gallery and the number of people in front of you are shown. Simply click on the “Instaline” button to join in the line of such restaurant. A user can enqueue up to 5 restaurants.

Navigate to My Line to check ranking status. Pull to refresh to keep updated. Click the restaurant for more details, such as ranking, and direction to the restaurant. If users decided not to continue line up for such restaurant, click “InstaQuit” to exit the line. Whenever the rank is approaching, the application will send users a notification.

With all of these provided, users could do what had to be done with quite a long time by just simply clicking on their phone. The users no longer need to be in the restaurant for the whole line up process, they only need to show up whenever it is their turn.

(Conclusion)

During this demo, we shown all user interfaces for customers, demonstrated basic functions we designed to solve the real world problem. In the next deliverable, our team is going to refine functionalities on the user side and implement the restaurant side of the application, which allows restaurants to access to the database and manage their own restaurant lineup easily.

Moreover, users could also line up for more than one restaurant at the same time. Users could choose not to go to one of their queueing restaurants even if it is their turn. Such feature provides selectivity, which also saves a lot of time for users.

Demo video script 2 for deliverable 3：

Step1:
(Background: the Instaline icon)

Hello, welcome to team 1, deliverable 3, demo video.

Have you ever found calling for reservation is just too inconvenient and inflexible?
Have you ever found that you have spent too much time waiting for a seat in the restaurant?

Or, you spend too much effort on managing customers who crowding at your restaurant entries?

Instaline, our android application will provide a virtual lining service to solve variety of lining up inconveniences. While online/phone reservation is not flexible and waiting in a single restaurant is too frustrating, Instaline 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.

Step 2 Customer side:
(Background: screen shot video)

(Click instaline app)
We use Firebase to manage user accounts. Now, lets pick a handsome profile picture for ourselves.

(After login, the background should be the Explore page, the main functions are in Explore/Content/My line pages)
Entering the app, we provide users a list view and a map view of nearby restaurants. Let us go with list view this time. A search is provided to filter restaurants by keywords.

(Content page)
Once the user selected a restaurant, detail information of the restaurant including the price range, gallery and the number of people in front of you are shown. Simply click on the “Instaline” button to join in the line of such restaurant. A user can enqueue up to 5 restaurants.

Navigate to My Line to check ranking status. Pull to refresh to keep updated. Click the restaurant for more details, such as ranking, and direction to the restaurant. If users decided not to continue line up for such restaurant, click “InstaQuit” to exit the line. Whenever the rank is approaching, the application will send users a notification.

With all of these provided, users could do what had to be done with quite a long time by just simply clicking on their phone. The users no longer need to be in the restaurant for the whole line up process, they only need to show up whenever it is their turn.

Step 3 Restaurant side:

(Background: Enterprise version icon)
Also, we developed the enterprise version of Instaline for restaurants to manage their Instaline waitlist.

(Click instaline enterprise app)
The login/signup process is similar to the customer side application. The only difference for enterprise version is that users need to provide more information that relates to the restaurant, such as price range, rating, etc.

(After login, the background should be the Waitlist page, show how to delete customer)
After entering the app, we provide users with a waiting list of customers. Users can know the necessary information of customers, including customer images, names, and current rank status. If a customer “Bill” is ready to be seated, or not shows up after a time limitation, then the waitlist manager of the restaurant is able to delete this customer by a simple swiping. And the rank of customers is also changed due to the deletion.

(Show that the database changes will affect the rank, 最好同时展示两个 app)
The rank status of customers will dynamically change by observing the changes of the database from firebase. If a customer “Jerry” quit his queue by his Instaline application, then, in the waitlist of Instaline Enterprise, “Jerry” will disappear, and the rank will be rearranged.

Step 4:
(Conclusion)

During this demo, we shown all user interfaces for both customers and restaurants, displayed how user actions affect the database and how database changes affect the front-end, demonstrated basic functions we designed to solve the real-world queuing problem.

---------- TEMP----------
Step1

Hello, welcome to team 1, deliverable 3, demo video.

Have you ever found calling for reservation is just too inconvenient and inflexible?
Have you ever found that you have spent too much time waiting for a seat in the restaurant?

Or, you spend too much effort on managing customers who crowding at your restaurant entries?

Instaline, our android application will provide a virtual lining service to solve variety of lining up inconveniences. While online/phone reservation is not flexible and waiting in a single restaurant is too frustrating, Instaline 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.

Step 2 Customer side: 跟step3 合起来展示
(Background: screen shot video)

(Click instaline app) 另外一边click enterprise
We developed two separate version  for customer and restaurants and use Firebase to manage user accounts.

Now, lets pick a handsome picture and add some detailed information for our profile.

(After login, the background should be the Explore page, the main functions are in Explore/Content/My line pages)
For customer side, a list view and a map view of nearby restaurants are provided. Let us go with list view this time. A search is provided to filter restaurants by keywords.

(Content page)
Once a restaurant is selected, detail information of the restaurant including the price range, gallery and the number of people in front of you are shown. Simply click on the “Instaline” button to join in the line of such restaurant.

Simultaneously, restaurant side receives the request and the client will be shown on the waiting list.

Navigate to My Line to check ranking status. Pull to refresh to keep updated. Click the restaurant for more details, such as ranking, and direction to the restaurant. If users decided not to continue line up for such restaurant, click “InstaQuit” to exit the line. Whenever the rank is approaching, the application will send users a notification.

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

With all of these provided, users could do what had to be done with quite a long time by just simply clicking on their phone. The users no longer need

to be in the restaurant for the whole line up process, they only need to show up whenever it is their turn.

Step 4:
(Conclusion)

During this demo, we shown all user interfaces for both customers and restaurants, displayed how user actions affect the database and how database changes affect the front-end, demonstrated basic functions we designed to solve the real-world queuing problem.

# CSC301, Final Team Demo

## Monday, April 2nd - Tuesday, April 3rd

Your final demo will be similar to the videos you submitted for deliverables, but there are a few differences:

- It is a live demo, not a recorded video.
- It will be evaluated by the instructor(s), not the TA's.
- It is slightly longer and should include additional discussions (see details below).

To be clearer, your presentation should include the following:

- Demo (max' 5 minutes)
    - Introduce the problem you are solving (and the users you are solving it for).
      Our android application project, Instaline, will provide a virtual lining service to solve variety of lining up inconveniences. While online/phone reservation is not flexible and waiting for a single restaurant is too frustrating, Instaline 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 restaurants they wish.
    - Demonstrate the solution you built.
      Follow the video script.
    - This part should be similar to the videos you submitted for previous deliverables.
- Technical Discussion (max' 3 minutes)
  (从这里开始往下，用ppt演示的效果应该会比直接讲来得好)
    - Describe 1 - 3 technical items that came up while working on your project.
    - A technical item can be an interesting bug, design change, lesson you learned from using a certain technology/framework, etc.
    - Be specific and feel free (and encouraged) to include useful artifacts (e.g. code snippets, screen shots, etc.)

First, the android studio environment setting. We met various android studio environment setting problems.

Second, the front-end design changes during the development. 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 change lots of details of the front-end design, you can see the difference between

Third, the connection between customer side and restaurant (enterprise) side.

- Process Discussion (max' 3 minutes)
    - Describe your team's workflow: How did you communicate? How did you use Git/GitHub? How did you divide tasks? etc.
    - Highlights significant decisions (good or bad) that you've made along the way, and explain the rationale behind them.
- Individual Contribution (max' 1 minute per team member)
    - At the end of your presentation, every team member will be asked to (very briefly) describe their main individual contribution(s) and/or lesson(s) learned.

When preparing your demo, think about your audience (i.e. the instructor(s)).
We don't know your project as well as your TA, and we are interested in:

- Understanding who your target users are.
- Understanding why your software is valuable to its users.
- Seeing your prototype in action and understand the user experience.
- Understanding how you worked together as a team.

During your demo, we may ask you a few questions. You can expect

- Technical questions about your software
- Questions about how you worked together as a team

## Logistics

Total time (including questions from the instructor(s)): 15-20 minutes

When: Monday, April 2nd - Tuesday, April 3rd
Please register your team in this spreadsheett. It will be open for editing **Friday, March 30th at 5:00pm.**

You will be able to connect your laptop to the projector via VGA and to the audio system (headphones jack).
If you need any additional audio/video equipment, please let us know in advance.

## Evaluation

We will evaluate the following aspects of your demo:

- Clarity & focus
  - We should have a very clear idea of *what* you built, *who* you built it for and *why*.
  - The value to the users should be clear.
- Quality of presentation
  - Practice your demo in advance, and make sure it has a coherent and logical flow.
  - When answering questions, make sure you understand the question first. When in doubt, ask us to rephrase the question (instead of spending time answering the wrong question.
- Overall evaluation of the project
  - Effort that was put in.
  - Quality of the work (planning & implementing).
  - Insights (technical or non-technical).