# CS Messaging Web App

Thank you for being willing to participate in this exercise! At Branch, our first value is being **passionate about our customers**. This includes offering world-class customer service through our in-app chat.

In this project, you will address the challenge of handling a high volume of customer inquiries while flagging the most urgent issues. The main objective is to build a simple messaging app for Branch that can scale with us as we grow our customer base.

1. Build a messaging app that can be used to respond to incoming questions sent by our customers. Assume that the messages are received from and sent to the Android app through an API endpoint which you can simulate via a simple web form (no need to build an Android companion app). The app should allow a team of agents to respond to incoming messages from (potentially many) customers in a streamlined fashion. Design the app so that multiple agents can log in at the same time and respond to incoming messages (no need to handle authentication).
2. We will provide a set of real customer service messages to you in a CSV file. Store these messages in a database of your choosing. Your app should provide a way to view and respond to these individual messages.
3. Host your app somewhere (your machine is fine as well!). We will go through it during the demo and presentation.

If you have time to spare after implementing these basic requirements, here are some ideas for how to extend the app (we certainly don't expect you to implement all of these, please pick and choose based on your own preferences):

- Figure out a scheme to help agents divide work amongst themselves, and to prevent multiple agents working on the same message at once.
- Explore ways to surface messages that are more urgent and in need of immediate attention. For example, customers who are asking about the loan approval process or when their loan will be disbursed might have more urgency than those asking how to update information on their Branch account.
- Implement search functionality to allow agents to search over incoming messages and / or customers
- Explore ways to surface additional information about customers (e.g. external profiles or some internal information we have about them) in the UI, to provide context to agents.
- Implement a canned message feature that allows agents to quickly respond to enquiries using a set of pre-configured stock messages.
- Make the agent UI (and/or the customer-facing UI) more interactive by leveraging websockets or similar technology, so that new incoming messages can show up in real time.

Create a private repo for your code and commit often. Share the repo with branch-dev on Github.

***Please include clear instructions on how to set up and run your solution (on command line or any other tool/software). Please assume that the local machine does not have any initial setup done, hence do mention any dependencies and installations needed.***

***Also, do mention any additional features from the list you might have chosen to implement. In case not mentioned, it will be assumed that only basic functionality has been implemented.***

We encourage you to ask questions while you work on this project. Email us or write your questions below, and we'll respond as quickly as we can.

Good luck, and we look forward to seeing your work on the project!