CS3216 Group 7 - BigSpoon Progress Report 2

Members: Shubham Goyal, Wang Gaoxiang, Yang Zhixing, Yos Riady, Chen Liang

1. Application prototype. Minimally as functional as what was achieved in Assignment 1.

iOS application is in beta, app file is ready to be installed on beta user's iphone.
Staff web application is at http://122.248.199.242/staff/main/, login details:
email - admin@lola.com password - lola

2. Is your project on schedule according to the milestones and timelines submitted in the initial project proposal? If you are on schedule, great! If not, why not? What is your team doing about the slip in your project schedule?

Our team is mostly on track of the milestones and timelines so far, we are going to setup the application and for lola's cafe on 5th November. After gathering feedback, we will be push out the iOS application to users.

By this point in the project schedule, we have written most of the basic functionalities on the staff backend. Features such as menu management, order creation API endpoints, user profile APIs, and general view logic is done. However, we still expect some minor changes in the future in response to user feedback. In general, the overall system is already stable.

This iteration, we have achieved the following successfully:

1. iphone application integration with backend API
2. socket real time system on menu and order system

Because the real time system support is currently in place, and all the major mobile features are working, we believe that we have reached a more stable prototype stage and thus on track in terms of schedule.

3. What were the problems/difficulties your team has encountered? How have you overcome them, or what plans do you have to overcome them?

We need to find more restaurant owners to talk to, so as to get more feedback. However, before we can get restaurant owners to decide to spend time with us, we need to do cold-selling to convince them that our app offers a good enough value proposition that would be worth spending their time to find out more.

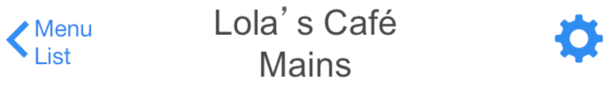
The process of convincing them will take a substantial amount of time, hence we are asking BigSpoon to help connect us to restaurants since they already have made contact with some restaurants. However, BigSpoon's Jay has only met most of the restaurants once and he is somewhat uncomfortable with having us do a live demo and ask the restaurant owner for feedback on a second meeting.

Hence, we may face some setbacks in that our restaurant owner feedback data may be limited.

For the backend server, currently we only have one AWS instance running, whereas in order to separate production data and test data, we may need a staging server as well. Bigspoon is helping us to setup a new AWS instance as staging server.

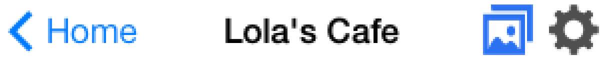
For the iOS app, the biggest difficulty that we faced was the UI. We spent a lot of time discussing with the BigSpoon team about some of the UI elements. Initially the BigSpoon team came up with a mock up for iOS, but there were some UI features that were either discouraged from Apple's convention or difficult to implement exactly as they wanted. So, we had to find viable alternatives.

For example, when the user navigates from outlets page to dishes page, they wanted the back button in the navigation bar to be a toggle button that changes between two view modes of dishes, i.e. photo view and list view. Something like the following:



However, this should not be the case because the back button should always point to the last page, which is a convention of Apple's default navigation bar.

The discussion went on to where to put the toggle of view mode button. We did something like the following:



This way, the back button points to the last page, which obeys the iOS' convention. But they pointed out that the toggle button is confusing because people won't know that it's a toggle button. We agreed and we went on to discuss solutions. In the end, we decided that this toggle button should be a switch button with words in it such as "Photo" or "List" so that it is self-explanatory.

4. Any changes to the application since the initial project proposal?

We have to add Websocket connections from the server to both the restaurant app on the web and diner app on iOS. This is so that the server can push events to both. For the web app, an example will be the server notifying the restaurant web app when a new order is created. For the iOS app, an example will be the server pushing to the iOS client when the restaurant manager has acknowledged the diner's orders. We did not think of the need for websockets earlier when doing the initial project proposal.

Most of the changes are improvements on the user interface after we spoke to users and obtained feedback on it. We have also had a few changes to the general data model structure to facilitate analytics, such as a user's average spending and other types of information.

Based on our discussions with the BigSpoon founders as well as live users, we have made many revisions to our user interface on both the staff side and the diner side. For example, since the staff may be equipped with an iPad to view the web application, we've designed the web app to be responsive and adapt accordingly. We've made design improvements such as making certain types of information more prominent, better alignment, and CSS refactoring.

In response to feedback, we are also considering making the appearance of new orders more prominent for staff through sound or screen flash. This lessens the chance of missing orders.

We expect that the user interface will undergo a process of continuous improvement as we do more user testing.

Since we have gotten the MVP features down, we have also started working on the Report page which requires a number of key analytics such as average monthly spending, top 5 dishes, and other key trends that would be useful for the restaurant owners. As such, we have made some minor revisions to our data model to facilitate this information gathering.

Additionally, we have also changed how notes work, by allowing notes for not only requests but a set of orders so that diners can give specific directions about their dish such as having their steak medium rare and so on.

5. Assuming that your team has already deployed a prototype of your application, how has the response been? Any other new insights, plans or strategies your team has come out with?

We have started testing out our web app with a restaurant. This was what we found out.

<u>Staff Web App</u>

The Report page was positively received. The feedback was that it had a clean design and provided information that was useful for tracking changes.

On the main page, we have Order Cards that shows newly made orders that have yet to be acknowledged by the restaurant manager. We tried taking some orders and in one case, one order card was significantly longer than the other order cards. Due to our current design of one card height per row that takes the longest card, the significantly longer card pushed subsequent rows out of the screen. Hence the amount of useful information shown was drastically reduced.

As such we decided to improve on this flaw. We felt that the significantly longer order card should be truncated. However, if there’s truncation, it means the restaurant manager has to click an expand button when transferring the orders to the POS. It is not just an extra step, but it also has the potential to cause the hidden part of the order list to be missed out during the transfer if the restaurant manager is busy and clicks too fast. That will be very bad.

We rethought about the entire process. The restaurant manager only cares about one order card during the transfer process - the order card that he is keying in, and he doesnt have to care about the rest. On the other hand, when he is not keying in anything, he should have a good overview of many order cards to see the number of orders that have been waiting for a long time. Hence, the solution is to have a button that the restaurant manager can click, whenever he wants to key the orders into the POS system. The first order card will expand itself to take up one entire row and show everything on it. The other orders will be shifted to the next row. Each time he presses acknowledge, we’ll shift the next order into the top row and expand it. After a certain amount of time of inactivity, we’ll switch back to the overview mode. This will provide a much better user experience that gives the best of both worlds.

We also have a header on the Order Card that indicates the amount of time since the order was made, so that the restaurant manager can give priority to orders that have been sitting for a long time. We indicate the day(s), hour(s) and minute(s). After we talked to the restaurant, they felt that almost every order should be fulfilled within the day. Hence, there's no need for the day(s) field. Hence, we we will be removing that field.

Mobile App

The iOS App was reviewed by the co-founders of BigSpoon and they gave us some feedbacks, based on which we discussed and came up with new development plans in the future. We discussed these suggestions and will apply them in the next iteration.

Firstly, the flow of events is quite clear in the iOS app, they feel that diner who uses this app will generally have no problem figuring out its functions and perform their ordering tasks.

Secondly, they gave us some suggestions on how to make the user experiences better. They suggested us to make some animations here and there to make it more user-intuitive. And change the fonds of some buttons to make it more obvious.

Besides, they also pointed out some small issues in the UI and let us fine tune the App, such as some minor misalignment, some colours of label and buttons, and some choices of words in the dialogue box.