Nibble

User Documentation

Sachin Patel

Varun Arora

Jinyan Li

Yasha Taylor

Hairong Wu

John Hynes

User Overview

Nibble is a food delivery solution designed to connect students lacking transportation or time with drivers traveling to nearby restaurants. Users place orders for food from nearby restaurants for drivers to accept. Drivers then pick-up all accepted orders from the restaurant in question and drop-off food at predetermined locations.

Nibble is comprised of three platforms: a user iOS app, a driver iOS app, and a user web application.

User iOS Application

Signing in

When first launching the app, Nibble users are prompted to either create an account or sign in. Users also have the ability to sign up with Facebook.

Choosing a Restaurant

After logging into Nibble, users are able to see a list of restaurants that drivers have opted to deliver from. When tapping on a restaurant, detailed information is presented including possible delivery times and links to view the restaurant's menu and Foursquare page. Tapping on a delivery time allows the user to enter text describing their order, choose a delivery location, and place the order. Confirmation that the order has been placed is provided.

Tracking Your Order

After placing an order, a user can keep track of their order status by tapping on the history icon in the upper left of the home page. This presents a list of all of the user's past and present orders. Tapping on an order shows information about the order including the driver (if assigned at that time),

the restaurant that was ordered from, the location the food will be delivered (if driver is assigned), and the status of the order from Order Placed, Out for Delivery, and Delivered. Users will also receive push notifications as their order status changes.

If a user's order has not yet been assigned a driver, a user can tap the Cancel Order button to cancel the order request. If any issues arise, a user can contact either their driver or the Nibble team by tapping on the appropriately labeled areas in their order history.

Users also have the ability to quickly pinpoint the restaurant location and delivery location by tapping on the appropriate cells in their order history view to open Apple Maps for directions.

Driver iOS Application

Signing in

When first launching the Nibble driver app, a user must first apply to be a driver. Once the driver's application has been processed they may log in with their Nibble credentials.

Starting deliveries / "Going online"

Once a driver has signed in they may tap the button at the top to go online. Going online involves selecting a restaurant to deliver from, a specific delivery location, and a specific time.

While online, drivers will receive incoming orders from users of the Nibble app requesting that driver's restaurant, time, and delivery location. A driver is free to accept or ignore any incoming orders. Once an order has been accepted, that order remains in the driver's dashboard for reference. Driver's will receive push notifications for incoming orders.

When the driver picks up a user's order, they must tap the picked up button. Similarly, when an order is delivered the driver must tap the delivered button. In order to prevent false deliveries, a driver is prompted with the option to contact the Nibble team before truly marking an order as delivered.

While making deliveries, a driver can tap on the header while online to view restaurant information, get directions to both the restaurant or the delivery location, or go offline. Drivers are still held responsible for any orders that they accepted while online.

User Web Application

Users can navigate to http://nble.parseapp.com and log in to view their order history. This order history reflects orders placed by the user with the Nibble iOS app.

Developer Overview

Developers seeking to join the Nibble team ought to be familiar with one or more of the following technologies: Swift 2.0 and Xcode, Parse SDK, Javascript, Backbone.js, and HTML / CSS.

The source code for the iOS application currently implements all features described in the user section of this documentation. It is fully integrated with our Parse database and takes advantages of Parse's cloud code feature as well as third party iOS API extensions. Developers working on the iOS platform ought to be able to read and understand existing code and become familiar with the flow of the app as well as the APIs being used. We are using CocoaPods (http://cocoapods.org) as a dependency manager. Our iOS project has the following third party dependencies:

- Parse SDK – https://github.com/ParsePlatform/Parse-SDK-iOS-OSX
- AHKNavigationController – https://github.com/fastred/AHKNavigationController
- SDWebImage – https://github.com/rs/SDWebImage
- Facebook-iOS-SDK – https://github.com/facebook/facebook-ios-sdk

Developers on the web platform will use HTML, Javascript, and CSS to enhance and complete the user web application. The web application was intended to have full feature parity with the iOS user application but due to time constraints it does not currently implement all iOS features. A developer joining the team would primarily work on completing feature parity. Interacting with Parse's Javascript SDK will require some knowledge of Backbone.js and creating the web application will use HTML/JS/CSS. Alongside completing feature parity, enhancements can also be made to the team's cloud code / backend functions, which are all written in Javascript.

Testing and running our applications is straightforward. Familiarity with Xcode is required to run applications on iOS as well as using Xcode's debug features. The web application can be tested and run by visiting the URL http://nble.parseapp.com and using a browser's built-in web developer toolkit.