



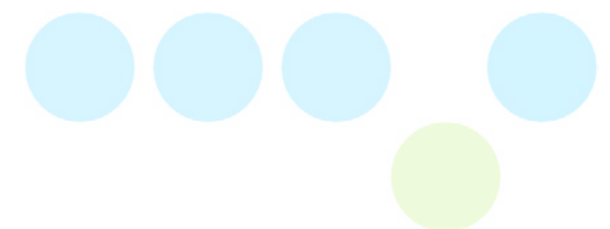
An ATN Mobile App for Silicon Valley Commuters

User Experience Design Specification

Bookapod Confidential

Document Version 0.6

Last Updated: 12/10/2013



Revision History

Date	Author	Version	Description
11/10/2013	Dustin Vaughn-Luma	0.1	- Initial Draft
11/15/2013	Dustin Vaughn-Luma	0.2	- Added Wireframes, Flows, and Interaction Notes
11/25/2013	Dustin Vaughn-Luma	0.3	- Updated wireframes and interaction notes
11/30/2013	Dustin Vaughn-Luma	0.4	- Updated wireframes and interaction notes across spec - Added mockups section - Added style guide section
12/8/2013	Dustin Vaughn-Luma	0.5	- Added additional and updated mockups and refined style guide
12/10/2013	Dustin Vaughn-Luma	0.6	- Finalized formatting and exported for delivery

Notices

- Copyright** Copyright © 2013 Bookapod, Inc. All rights reserved.
- Disclaimer** The information contained in this document is subject to change without notice.
- Trademarks** This document may make reference to other software and hardware products by name. In most if not all cases, the companies that manufacture these other products claim these product names as trademarks. It is not the intention of Bookapod, Inc. to claim these names or trademarks as its own.

Table of Contents

1. Introduction	5
1.1. Document Scope	5
1.2. References	6
1.3. Project Team	7
2. Task Flows	8
3. Bookapod Application	9
3.1. Use Cases: UC01 – Login to application	9
3.2. Use Cases: UC 02 - Map Flow	10
3.3. Use Cases: UC 03 - Book a pod (Reservation System)	11
3.4. Use Cases: UC 04 - Schedule Your Day (Scheduling System)	13
3.5. Use Cases: UC 05 - Favorites	15
3.6. Use Cases: UC 06 - Settings	16
4. Wireframes / Interaction Notes	18
4.1. Link to wireframe assets	18
4.2. Login Flow / First Run Experience	19
4.3. Navigation Flow	24
4.4. Search	28
4.5. Favorites List	29
4.6. Reservation System	31
4.7. Schedule Your Day (Scheduling System)	39

4.8. In-pod navigation	52
4.9. Primary Navigation	54
4.10. Settings	55
5. Mockups and Content	56
5.1. App Icon and Login Screen	56
5.2. Create Account	57
5.3. Main Map Screen (Primary Screen) & Navigation Menu	58
5.4. Closest Station Options	59
5.5. Closest Station Map Popup	60
5.6. Turn-by-turn navigation options	61
5.7. Reservation Screens	62
5.8. Creating New Payment Type	63
5.9. Saved Cards	64
5.10. Confirm Payment	65
5.11. QR Code	66
5.12. Plan Your Day	67
5.13. Favorites	68
5.14. Settings	69
6. Style Guide	70
6.1. Colors	70
6.2. Fonts	71
6.3. Buttons	72
6.4. Screen Dimensions	73

1. Introduction

1.1. Document Scope

This document captures the user experience specification for the Bookapod mobile application. This includes use cases, task flows, interaction wireframes, mock-ups, and style guide.

Purpose of the project

To enable Silicon Valley Commuters to purchase ATN public transportation tickets directly from their mobile device.

Important Features of the application

- Booking a pod at a pod station
- Turn-by-turn navigation
- Scheduling routes
- Saving routes
- Purchasing tickets for a pod
- Saving favorites

End users of the application

Silicon valley commuters, tourists, as well as anyone who requires the service of public transportation.

Target readers of this document

UX Design / Visual Design, Engineering, Product Management, and Marketing

1.2. References

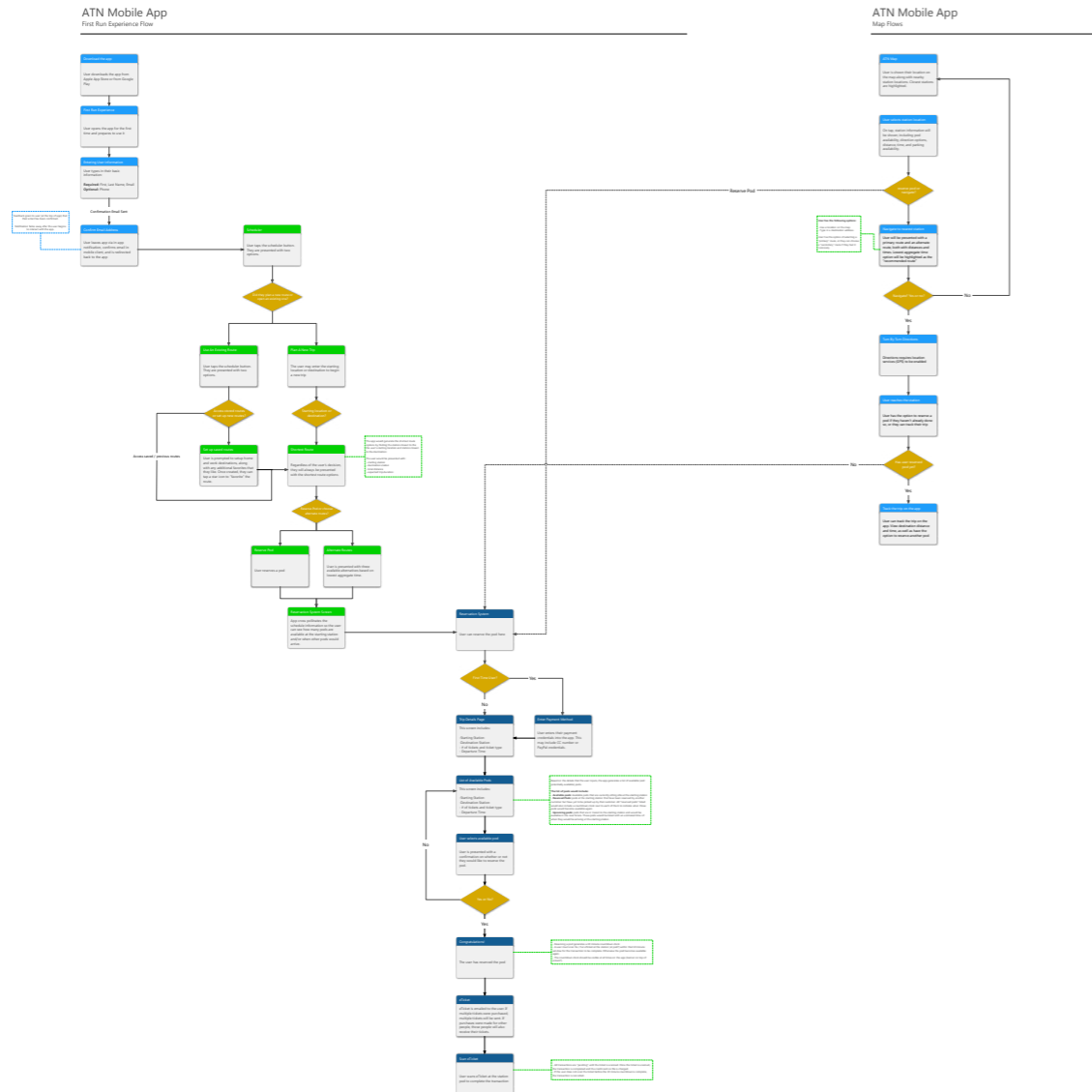
Document	Owner	Link
Midterm project report	Bookapod Team	https://www.dropbox.com/s/e8e5qf0ql4fibnl/dsid-131-midterm-project-documentation-bookapod-nerpio-vaughnluma-vora-wendt-10_28_2013.docx
Bookapod Personas (includes empathy and scenario maps)	Bookapod Team	https://www.dropbox.com/sh/dd57ogq51450nzb/49hyY6HYIF
Bookapod Taskflows	Bookapod Team	https://www.dropbox.com/s/bnbop83dle28skj/dsid-131-project-taskflows.pdf
Bookapod Storyboards	Bookapod Team	https://www.dropbox.com/sh/hsgn27o7mf7cnbt/rpH1gz71IT
Bookapod Wireframes	Bookapod Team	https://www.dropbox.com/sh/a9g8o7tw1n9bs5j/PGLZj9WylP
Axure Prototype	Bookapod Team	http://share.axure.com/2P54LY/login.html Username: tingbin.tang@sjsu.edu Password: tingbin123 After logging in, Double Tap on "B Choose Destination Field" Enter: "Julian Station" in Sentence Case

1.3. Project Team

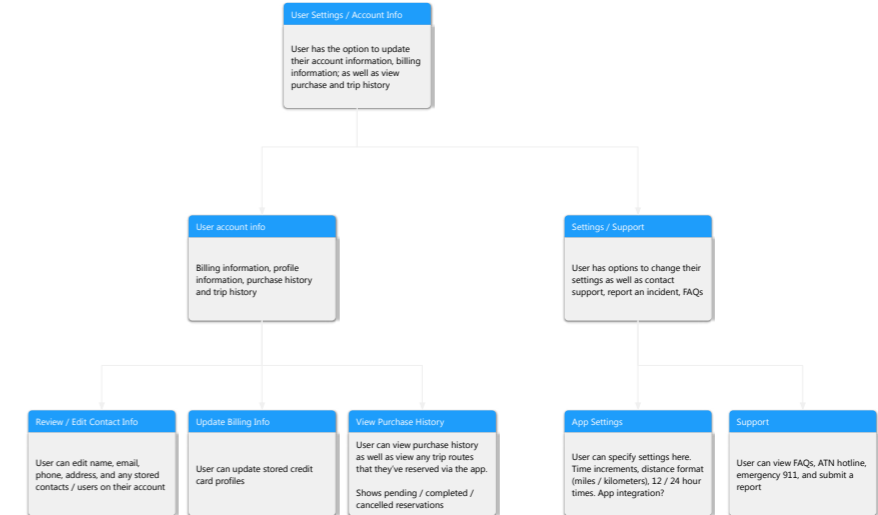
Person	Domain
Nel Nerpio	Senior UX Designer – Lead for the project
Dustin Vaughn-Luma	Senior UX Designer – Lead for the project
Shiv Vora	Senior UX Designer – Lead for the project
James Wendt	Senior UX Designer – Lead for the project

2. Task Flows

Main Application Task Flows (Booking a pod, scheduling a day, purchasing a ticket)



ATN Mobile App
User Menu (Settings / Support)



Note: Flow can be downloaded [here](#).

3. Bookapod Application

3.1. Use Cases: UC01 – Login to application

3.1.1. Description

This use case describes the steps involved during the login process.

3.1.2. Actors

Bookapod customer

3.1.3. Assumptions

User intends to use the app and it has been downloaded to their device.

3.1.4. Pre-conditions

Account creation has already been completed.

3.1.5. Trigger

Actor opens the application on a mobile device.

3.1.6. Main Flow 01: Actor opens the application on their mobile device by clicking on the app icon.

1. Actor triggers the application
2. System displays the login screen
3. User logs in and is taken to the Map Flow

3.1.7. Alternate Flow 01: Actor doesn't have an account yet - First Run Experience

1. Actor triggers the application by tapping the icon after download from Apple app store or Google Play Store
2. System displays the login screen
3. User types in their email address and taps on Continue button
4. Account creation page is displayed
5. Basic user information is entered into the fields and user taps the Continue button
6. Confirm dialog pops up asking the user to confirm their email address
 - a. User confirms email address and is taken to their email client to confirm their address
 - b. User chooses not to confirm their email address and moves directly into the app (the app will not allow them to perform any booking functionality until they confirm their address).
7. Email is confirmed and the user is taken to the primary app screen (Map page)

3.2. Use Cases: UC 02 - Map Flow

3.2.1. Description

This use case describes the steps involved during the map view / turn-by-turn navigation process.

3.2.2. Actors

Bookapod customer

3.2.3. Assumptions

The user has logged into the Bookapod system

3.2.4. Pre-conditions

Account creation has already been completed and the user has confirmed their email address

3.2.5. Trigger

Actor logs into the application on their mobile device

3.2.6. Main Flow 01: User views the map and chooses station location

1. User can reserve a pod at that location from the main screen by choosing the book a pod button
2. User can tap on the station dot on the map to get a pop-up and select the book a pod option from there

3.2.7. Alternate Flow 01: User navigates to the desired station

1. User can navigate to that location via car route / distance / time or by walking route / distance / time
2. The user can access the option to book a pod after the turn-by-turn navigation has been initiated
3. User reaches the desired location and the navigation ends

3.2.8. Alternate Flow 02: User may book a pod by accessing the primary navigation menu and selecting the book a pod option

1. User taps on the main app menu icon in the lower middle portion of the screen
2. User taps on book a pod to begin the reservation process

3.3. Use Cases: UC 03 - Book a pod (Reservation System)

3.3.1. Description

This use case describes the steps involved during the reservation process.

3.3.2. Actors

Bookapod Customer

3.3.3. Assumptions

The user has logged into the bookapod system and has the intention of booking a pod.

The user also has the intention of setting up a payment type, using an existing payment type, or using PayPal.

3.3.4. Pre-conditions

User is logged into the system

3.3.5. Trigger

The book a pod button is tapped from the closest station dropdown, the station pop-up, or from the primary navigation

3.3.6. Main Flow 01: User chooses to book a pod - 1 ticket

1. User is taken to the trip details page where they are given the option to enter destination station and number of tickets for purchase
2. User selects destination and the option to purchase only 1 ticket
3. User is presented with the option to enter a payment type
 - a. User enters a new payment type and stores it
 - b. User enters PayPal account information and stores it
 - c. User uses an existing payment type that they've already set up in the bookapod system
4. User is asked to confirm payment
5. User is placed into queue at the their starting station and is presented a QR code for scanning into the starting station kiosk
6. User arrives as station and scans their QR code at the kiosk or at a customer service desk (a new QR code [pod code] is generated)
7. The station is now aware that the user is at the station and is given a time (and platform number) as to when their pod will be available (3-5 minutes).
8. The user arrives at the platform and scans their QR code on the pod door gaining them entry
9. Their credit card is charged once they have safely entered the pod and it begins moving

3.3.7. Alt Flow 01: User chooses to book a pod - 2 or more tickets

1. User is taken to the trip details page where they are given the option to enter destination station and number of tickets for purchase
2. User selects destination and the option to purchase 2 or more tickets
3. The user is presented with enough fields to match the amount of tickets purchased
4. The user can opt to give themselves one of the tickets or gift them all to their contacts
5. Once the contacts are selected the user is notified that all of their contacts will receive the tickets along with instructions on how to use the ticket
6. User is presented with the option to enter a payment type
 - a. User enters a new payment type and stores it
 - b. User enters PayPal account information and stores it
 - c. User uses an existing payment type that they've already set up in the bookapod system
7. User is asked to confirm payment
8. User is placed into queue at the their starting station and is presented a QR code for scanning into the starting station kiosk
9. User arrives at station and scans their QR code at the kiosk or at a customer service desk (a new QR code [pod code] is generated)
10. The station is now aware that the user is at the station and is given a time (and platform number) as to when their pod will be available (3-5 minutes).
11. The user arrives at the platform and scans their QR code on the pod door gaining them entry
12. Their credit card is charged once they have safely entered the pod and it begins moving

3.4. Use Cases: UC 04 - Schedule Your Day (Scheduling System)

3.4.1. Description

This use case describes the steps involved during the reservation process.

3.4.2. Actors

Bookapod Customer

3.4.3. Assumptions

The user has logged into the bookapod system and has the intention of booking a pod

The user also has the intention of setting up a payment type, using an existing payment type, or using PayPal

The user intends on planning a route for their travels

3.4.4. Pre-conditions

User is logged into the system

3.4.5. Trigger

The scheduler button is tapped from the primary navigation menu

3.4.6. Main Flow 01: Plan a new trip

1. User taps on scheduler button and is taken into the scheduler flow
2. User will already have the closest station populated
 - a. User may change the starting station
 - b. User may leave the starting station in its default state
3. The system defaults to the current day / month
 - a. The user can change the month / day to the date of their choice
 - b. The user may leave the current date as is
4. The user taps into the destination station field and chooses their destination station
 - a. User may choose the destination from their favorites
 - b. User may type in and / or search for a destination station
5. The user may choose to add additional destinations within their scheduled day

- a. The user may edit these destinations (as well as the starting station)
 - b. User may remove these options entirely from their scheduled route
6. The user taps on save route and they will be asked to save the route for quick access later
- a. User can save the route and name it
 - b. User saves the route but opts not to give it a custom name. The system will name it to the current date if a custom name is not specified.
7. User is taken to the reservation system with start / destination station(s) populated
8. User begins reservation flow [\[3.3\]](#)

3.4.7. Alternate Flow 01: Use an existing route

1. User taps on scheduler button and is taken into the scheduler flow
2. If user has previously saved routes, they will be given the option to select one
3. User selects their previously stored route
4. User begins reservation flow [\[3.3\]](#)

3.5. Use Cases: UC 05 - Favorites

3.5.1. Description

This use case describes the steps involved during the reservation process.

3.5.2. Actors

Bookapod Customer

3.5.3. Assumptions

The user intends on viewing a previously favorited station

3.5.4. Pre-conditions

The user has logged into the bookapod system

3.5.5. Trigger

The favorites button is tapped from the primary navigation menu

3.5.6. Main Flow 01: Favorites menu is accessed from the primary navigation

1. User taps on favorites menu icon from within the primary menu
2. User can select their option for starting station
3. User may swipe the favorite item, revealing turn-by-turn navigation options and an option to begin the reservation flow [\[3.3\]](#)
4. User may close the favorites list via a close button

3.5.7. Alternate Flow 01: Favorites menu accessed from the scheduler flow

1. When the user is selecting their destination station(s) from the scheduler flow [\[3.4\]](#), they have the option to access their favorites
2. User may swipe the favorite item, revealing turn-by-turn navigation options and an option to begin the reservation flow [\[3.3\]](#)
3. User may close the favorites list via a close button

3.6. Use Cases: UC 06 - Settings

3.6.1. Description

This use case describes the steps involved during the reservation process.

3.6.2. Actors

Bookapod Customers

3.6.3. Assumptions

User intends to change options within the application

3.6.4. Pre-conditions

User is logged into the application

3.6.5. Trigger

User taps on the settings icon from within the main menu

3.6.6. Main Flow 01: Settings

1. User taps on the main menu icon
2. User taps on the settings icon from within the menu
3. User may modify any available settings
4. User may close the settings screen via a close button

4. Wireframes / Interaction Notes

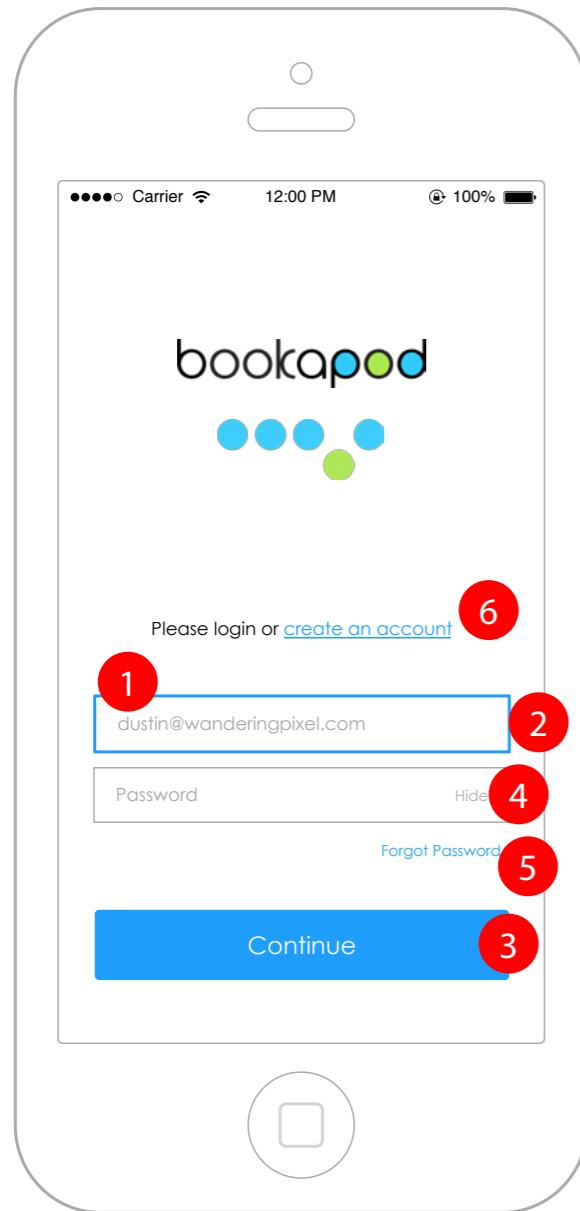
4.1. Link to wireframe assets

<https://www.dropbox.com/sh/a9g8o7tw1n9bs5j/PGLZj9WylP>

4.2. Login Flow / First Run Experience

4.2.1. Login Screen

Wireframe



Interaction Notes

1. When login screen appears, the email field will automatically gain focus.
2. Email field will be outlined with a 2px blue stroke
3. The continue button will be in a persistent enabled state
4. On tap shows secondary password field with password characters masked
5. Takes user to the forgot password flow
6. Takes user to create account flow

Error States

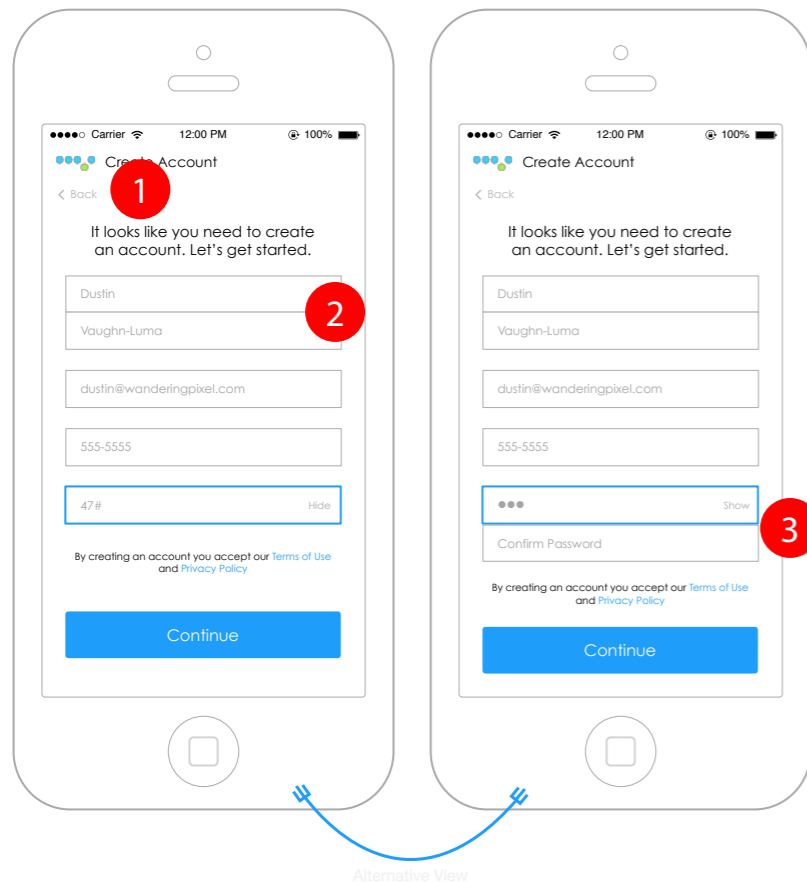
Condition	Notification message
User does not enter an email address - continue button is tapped	<i>Please enter a valid email address</i>
User does not enter a password - continue button is tapped	<i>Please enter a valid password</i>
User's email is not recognized	<i>{email address} is not in our system. Please enter a valid email address</i>
User's password is not recognized	<i>Your password does not match. Please enter a valid password</i>

- Notification message will be green
- Input field error state stroke will match the color of the notification message

4.2.2. Create Account

Wireframe

Interaction Notes



1. Takes user back to the login screen
2. First and last name fields (first name gains focus when the screen appears)
3. Show / Hide button toggles masked password and subsequent password field

Error States

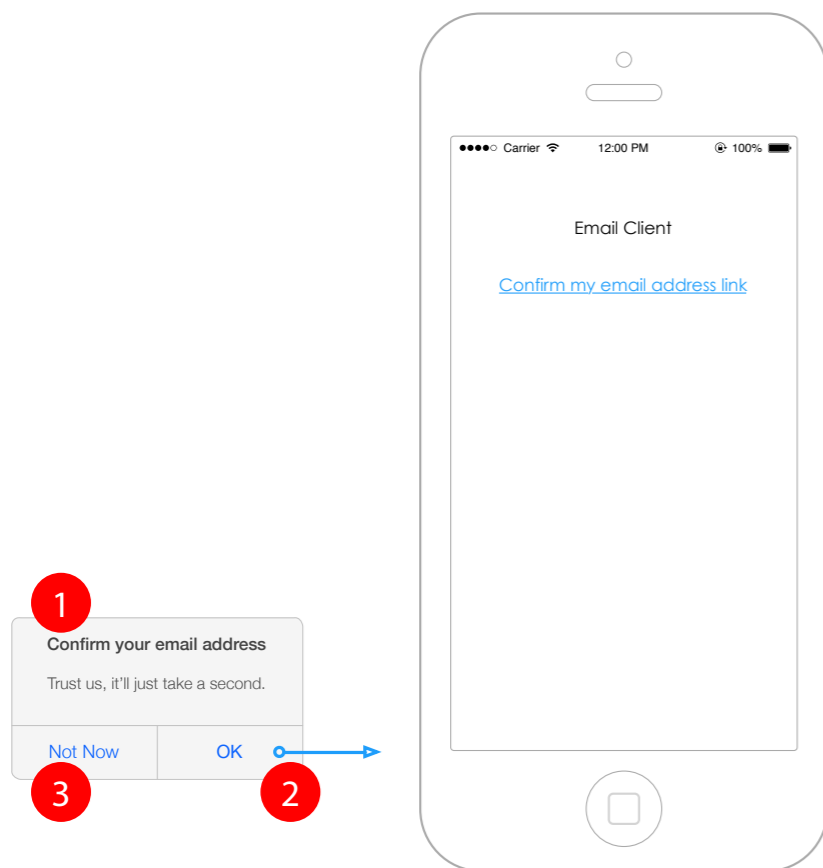
Condition	Notification message
User does not fill in any of the fields - continue button is tapped	<i>Please enter a {field name}</i>
If password is masked and the fields do not match	<i>Your passwords do not match. Please try again.</i>

- Notification message will be green
- Input field error state stroke will match the color of the notification message

4.2.3. Confirm Email Address

Wireframe

Interaction Notes

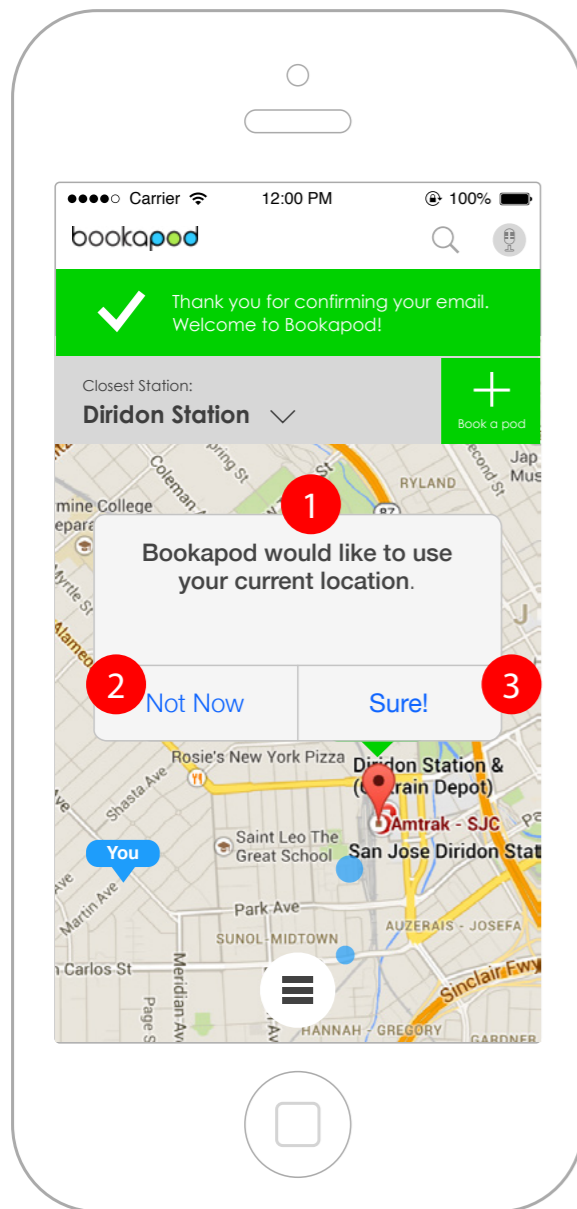


1. User is asked to confirm email address
2. OK - User is taken to their device's email client where they can confirm their address
3. Not Now - User is taken to application but is unable to book a pod until the email address is confirmed.

4.2.4. Location Services

Wireframe

Interaction Notes

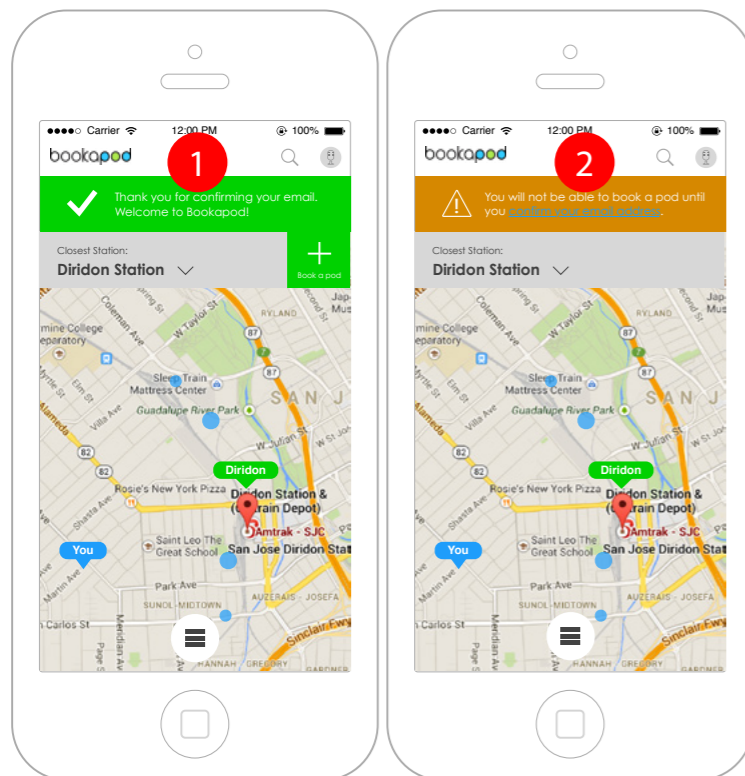


1. Modal is presented asking the user to allow the use of GPS
2. Not Now - User may navigate the app, but navigation will not work
3. Sure - App is ready for use

4.2.5. First run experience notification states

Wireframe

Interaction Notes



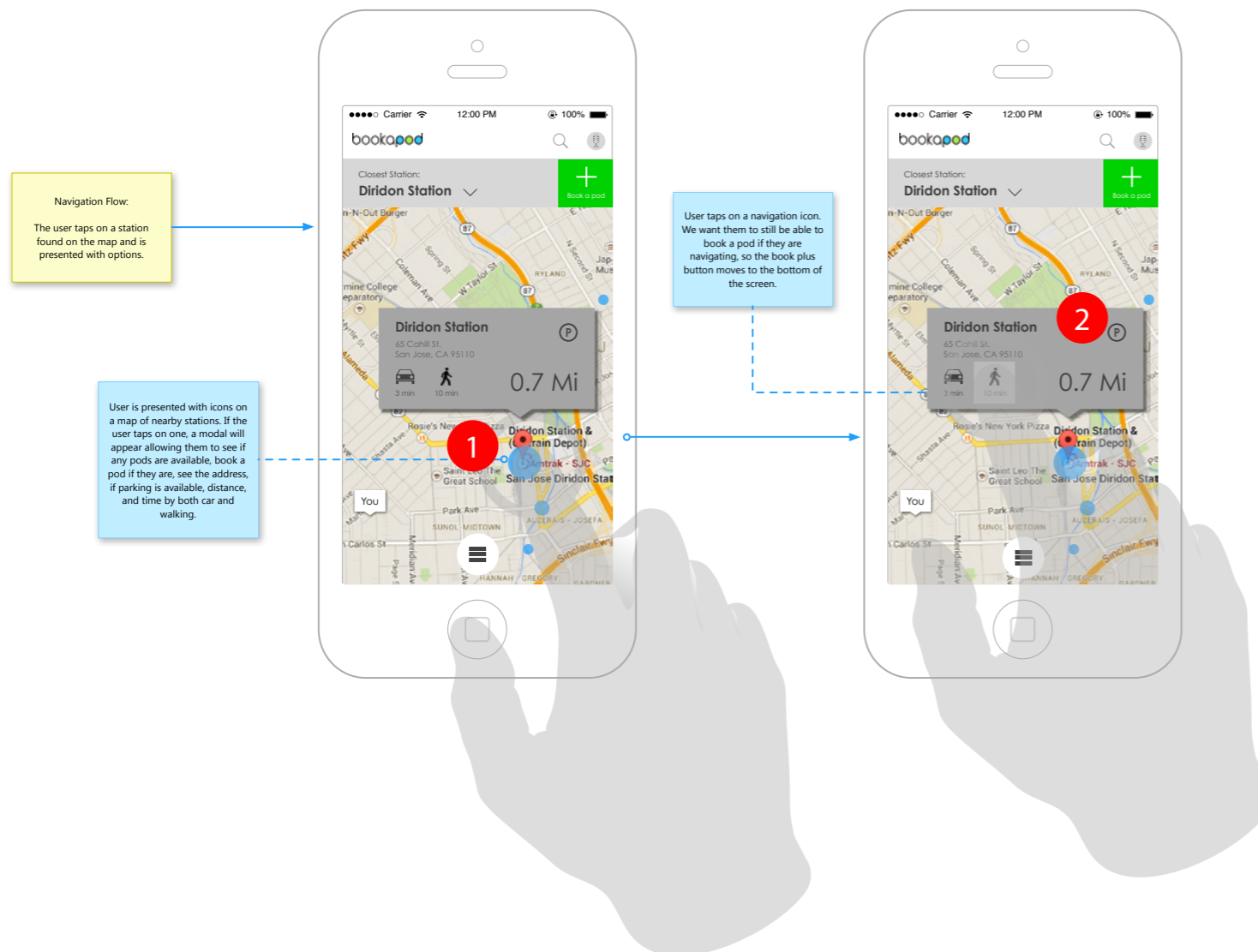
1. Positive confirmation of email address
2. Email address not confirmed - presents user with a persistent notification asking them to confirm their email address. The email address is presented as a link.

4.3. Navigation Flow

4.3.1. Map screen station pop-up

Wireframe

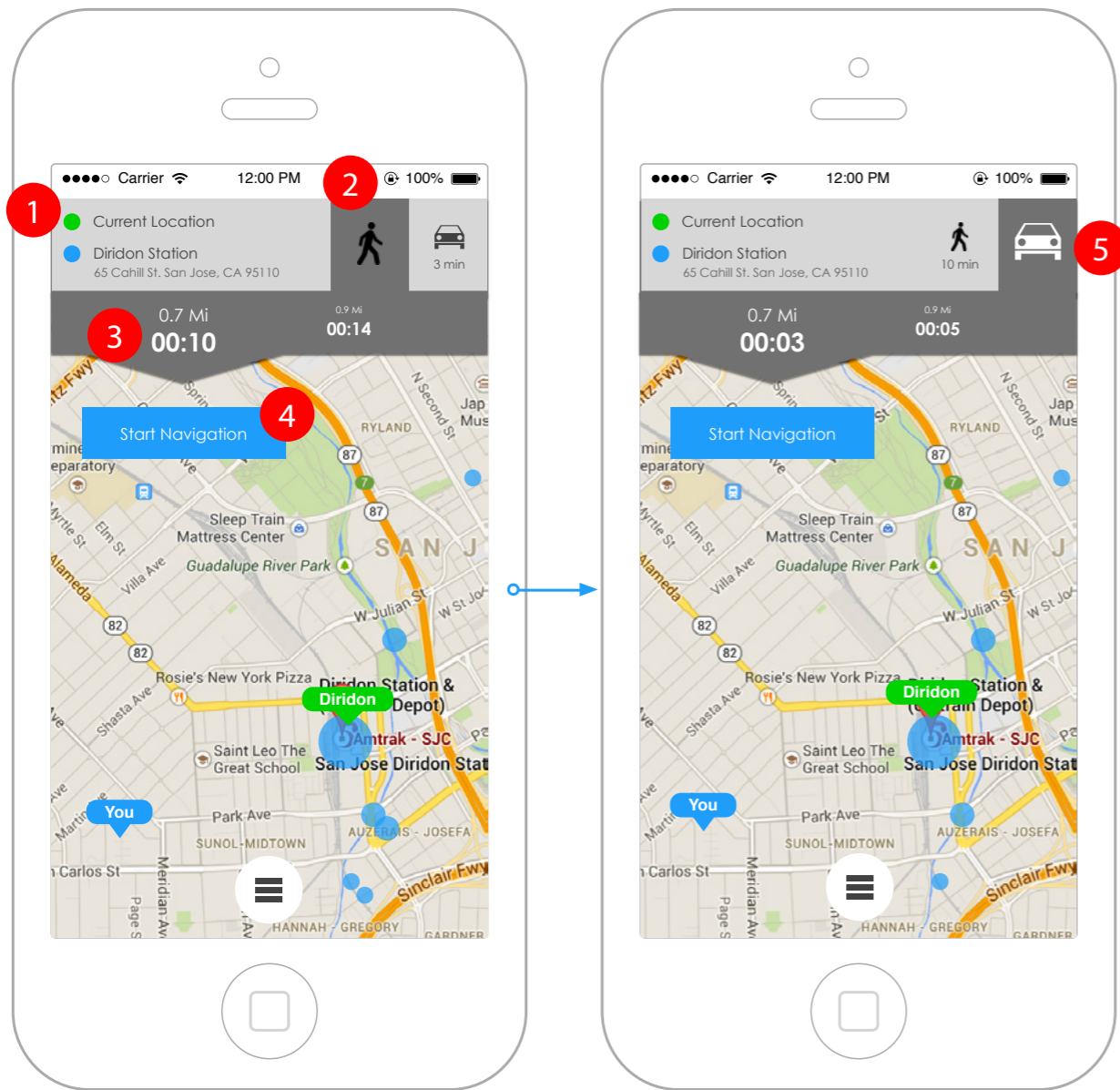
Interaction Notes



1. User taps on any station dot and a pop-up modal will appear showing the station address, parking availability, navigation options for both car and walking, and the distance to the station based on the user's current location.
2. Tap down state for the station pop-up.

4.3.2. Turn-by-turn navigation (dormant state)

Wireframe



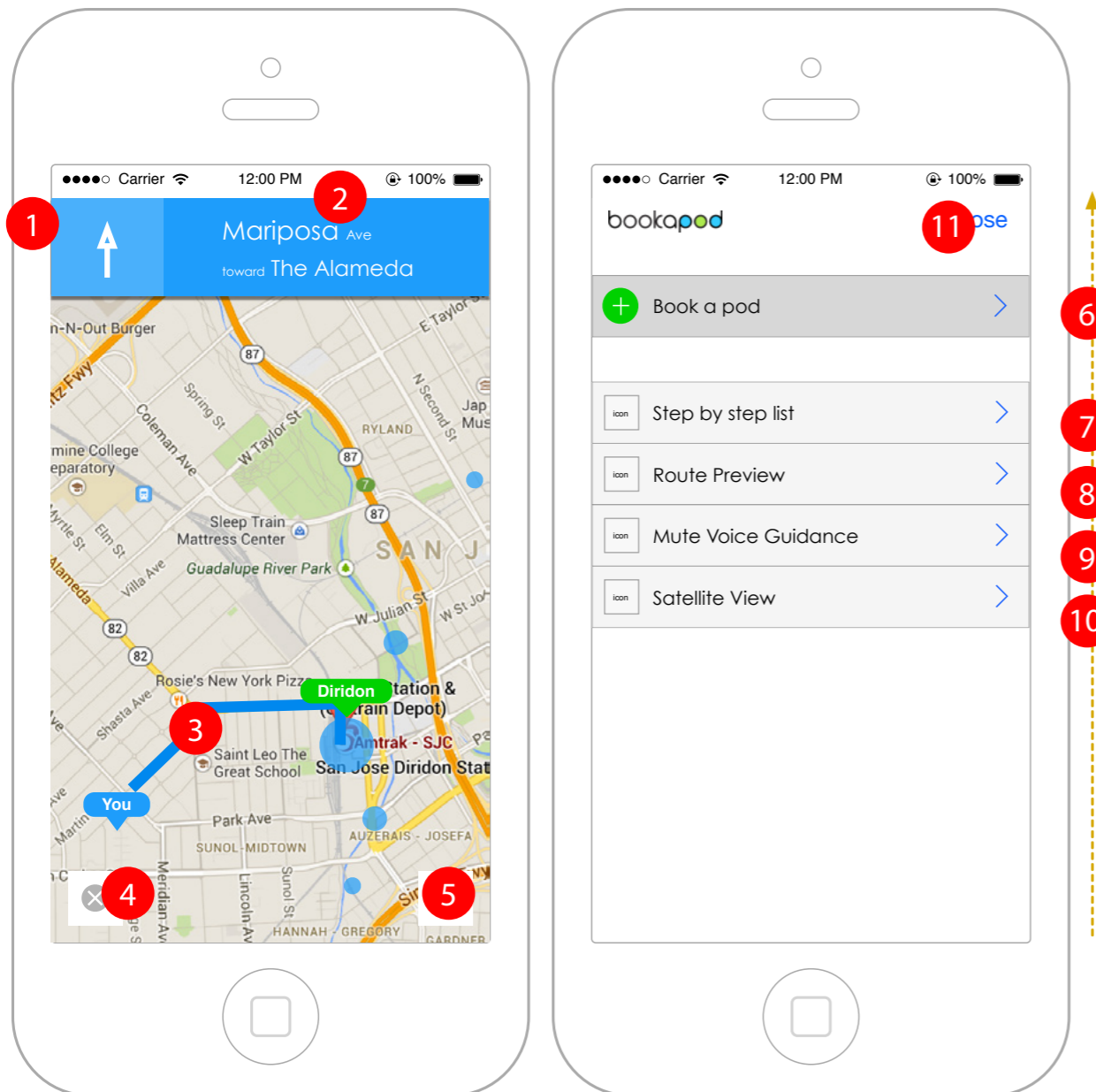
Interaction Notes

1. Current location and destination station are depicted at the top of the map.
2. Walking option is shown. The user will be shown either walking or car times based on the option they select in the prior screen.
3. 2 options are shown. The default option is the best route time. A secondary option will be shown based on the next best route. User can tap either of these and begin navigation.
4. Navigation Start Button
5. Driving tab - adjusted times are shown when this option is tapped.

4.3.3. Turn-by-turn navigation (active state)

Wireframe

Interaction Notes

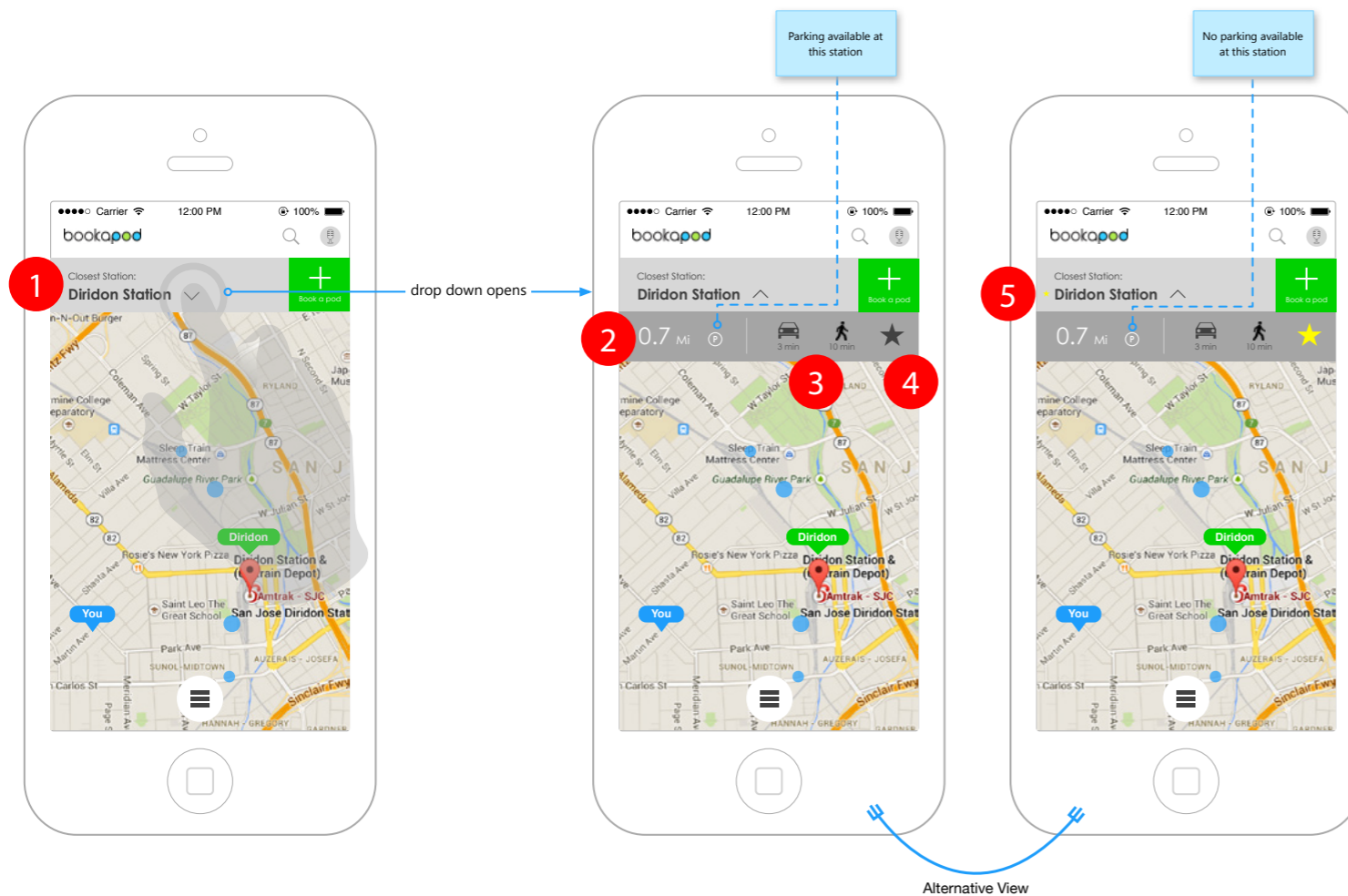


1. Navigation direction
2. Text directions - shows current street and upcoming street
3. Navigation route is highlighted from the users current location to the destination station
4. Stops turn-by-turn navigation
5. Turn-by-turn navigation options
6. Book a pod option - takes user to the reservation system
7. Shows the user a step by step list for their route
8. Show the user a route preview
9. Mutes voice guidance
10. Switches to satellite view of the map
11. Closes navigation options screen and takes the user back to the turn-by-turn navigation

4.3.4. Closest Station Dropdown

Wireframe

Interaction Notes



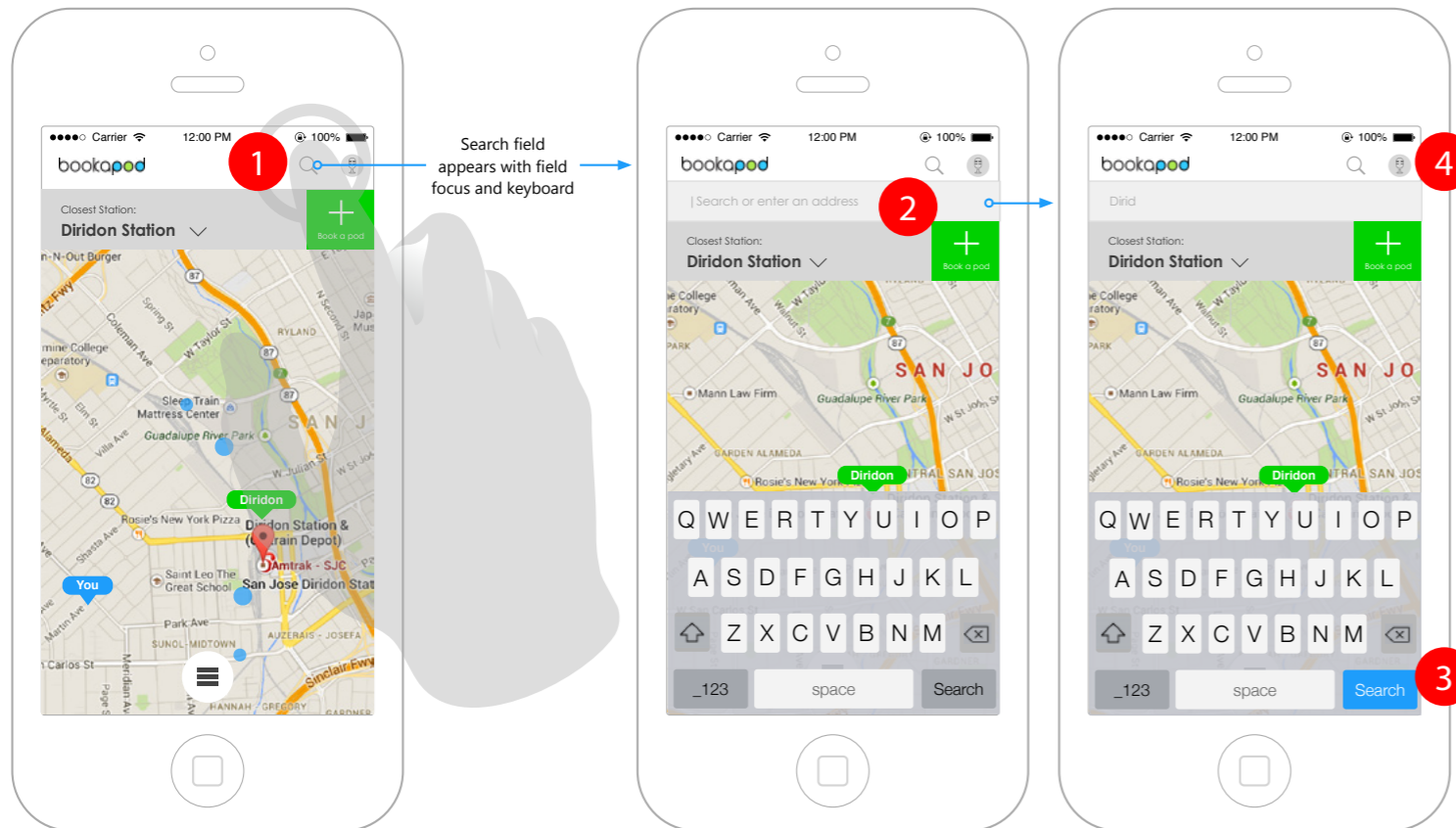
1. Closest Station - always shown on top of the map as a quick way to bookapod
2. Closest station dropdown - shows distance (from the user) to the station and parking status (available / not available)
3. Turn-by-turn navigation options (driving and walking)
4. Option to favorite the station. On tap, the star icon turns yellow
5. Once the station is favorited, a small yellow star icon will appear next to the closest station name

4.4. Search

4.4.1. Search functionality

Wireframe

Interaction Notes



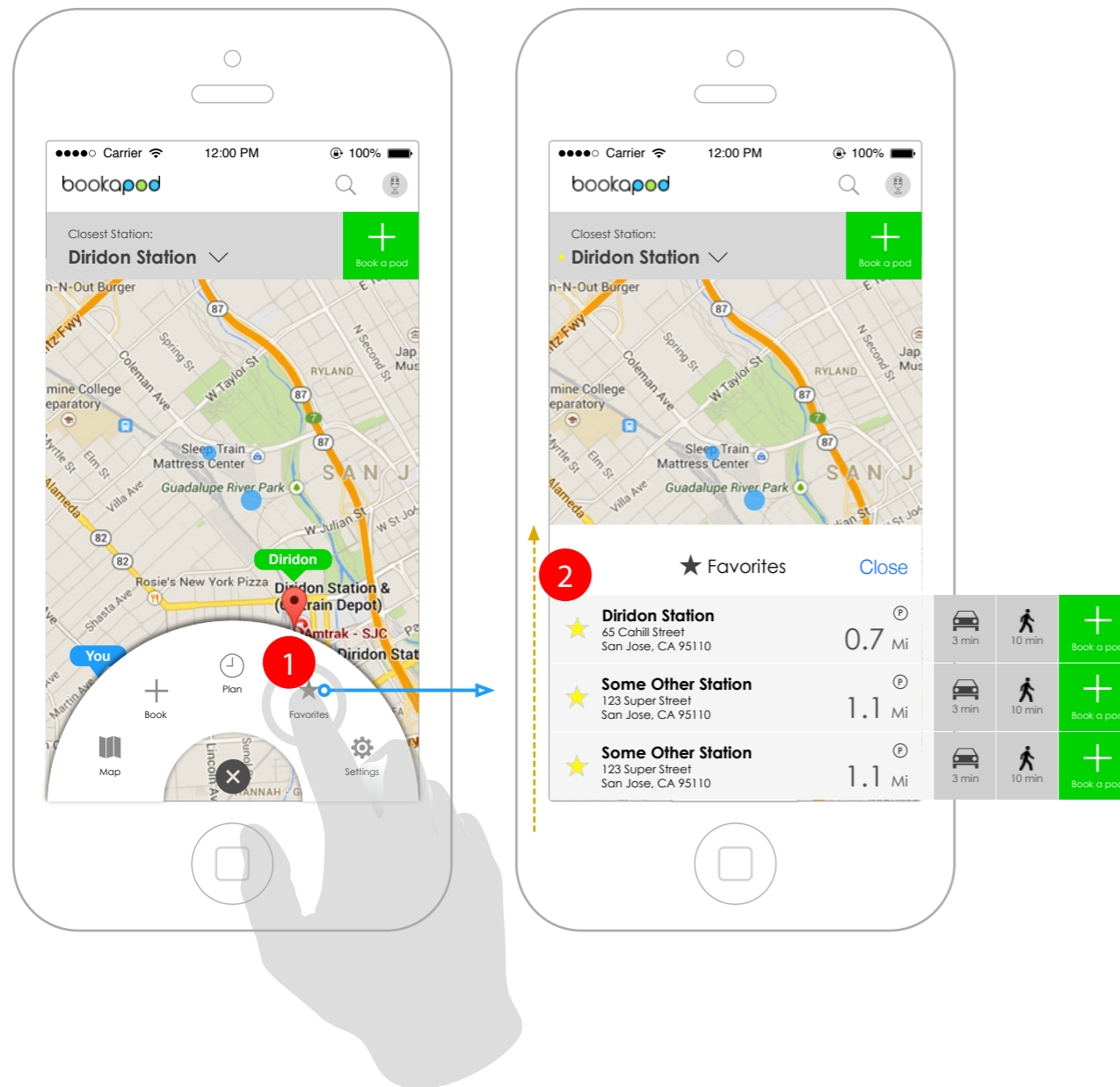
1. Search button - on tap, search field drops down from beneath the search icon. If tapped again, the field will collapse.
2. Search field - on field focus, the device keyboard appears
3. Search button on keyboard becomes active after first character is typed. On tap, the map will relocate to the desired location.
4. Search by voice - on tap, the user will be asked to speak. Results will be shown in a list (BETA)

4.5. Favorites List

4.5.1. Accessing favorites list

Wireframe

Interaction Notes

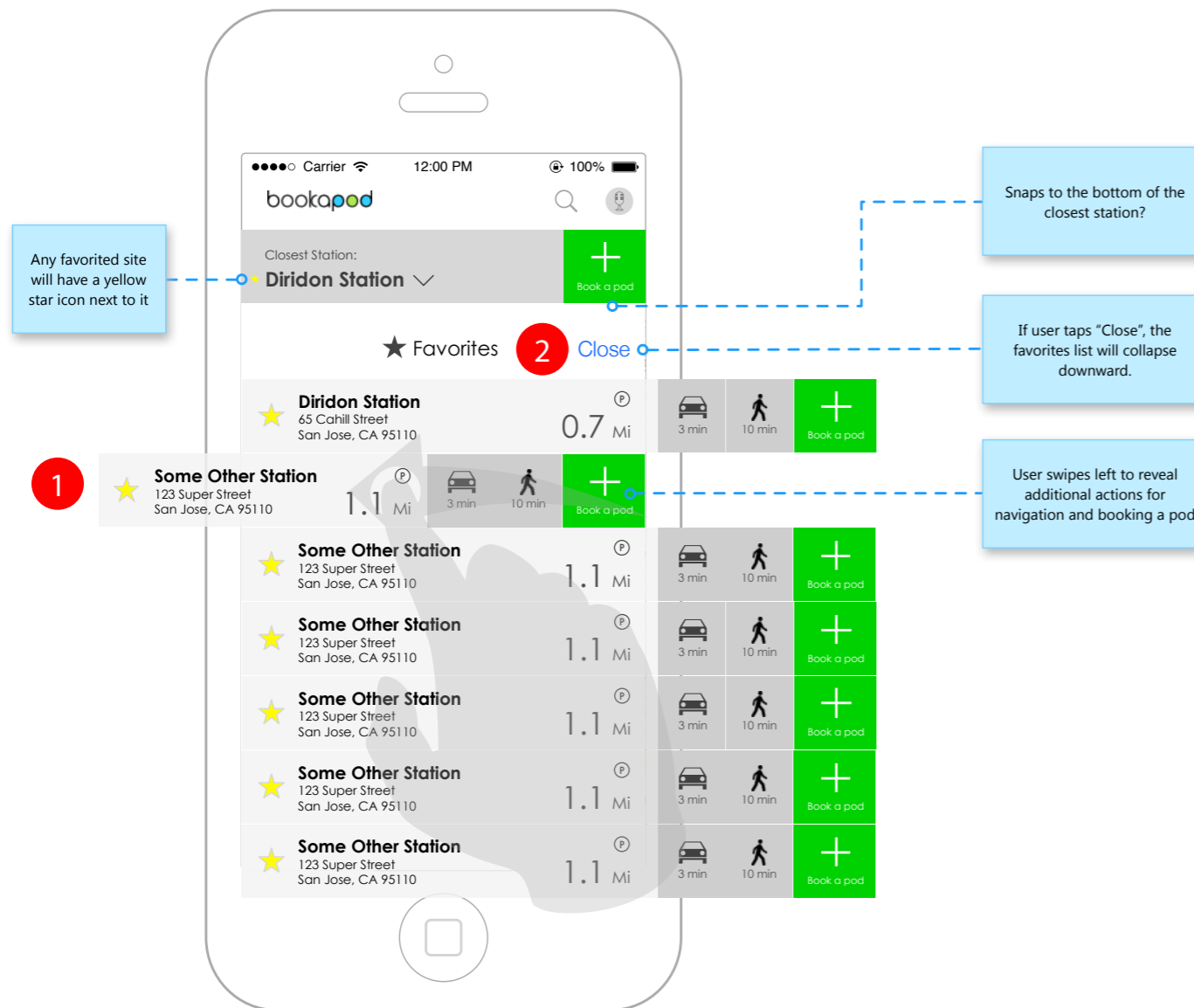


1. User accesses the favorites list from the primary navigation menu
2. On tap, the favorites list slides up from the bottom of the screen and stops just beneath the closest station banner.

4.5.2. Favorites list functionality

Wireframe

Interaction Notes



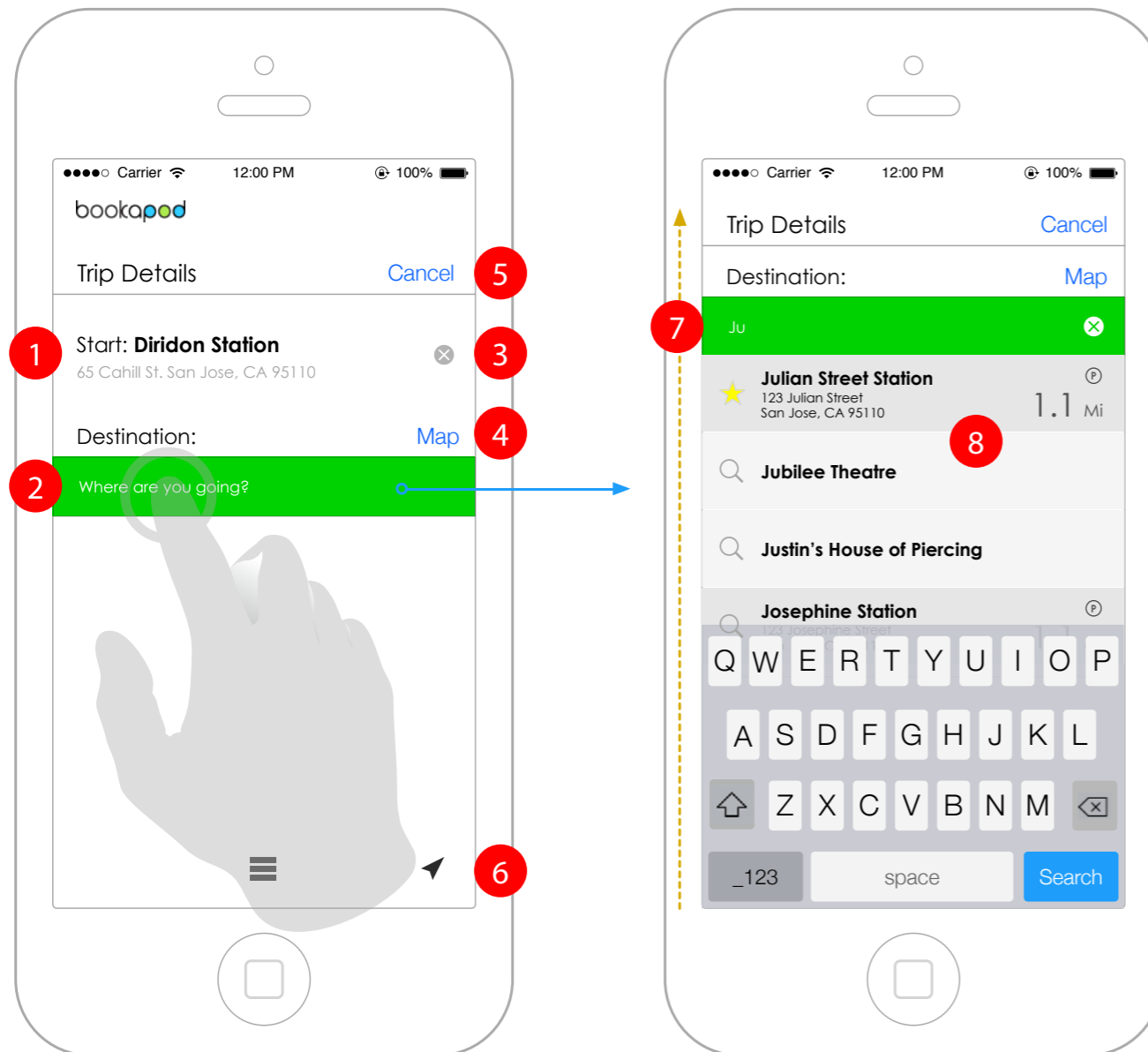
1. User may swipe the favorite item to the left to display the navigation options as well as the book a pod option. If user taps on navigation options, it takes them to the navigation flow. If user taps on book a pod option, it takes them to the reservation flow.
2. Close - on tap, the screen slides back down.

4.6. Reservation System

4.6.1. Start and Destination Stations

Wireframe

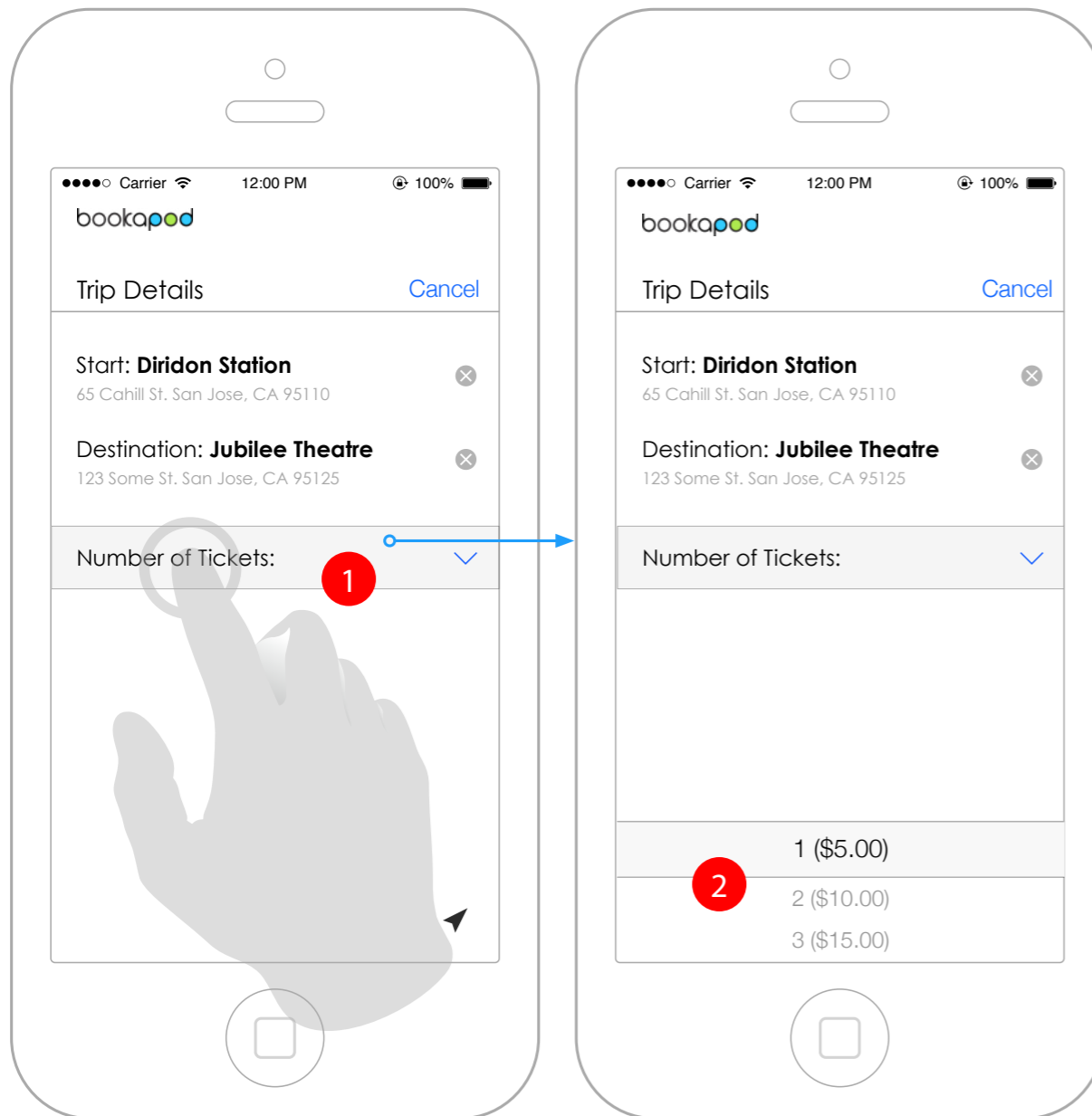
Interaction Notes



1. Start station - shows station name in bold and address of the station beneath it
2. Destination station - highlighted with placeholder text.
3. Small X icon allows the user to clear the field and enter something else. On tap, field has focus.
4. Allows the user to select their destination from the map
5. Cancel - closes the reservation system
6. Nav indicator - only visible if the user has started turn-by-turn navigation before booking a pod. Persistent throughout the reservation flow allowing the user to get back to the navigation system if need be
7. On tap of the destination field, the screen slides up to the top of the viewport.
8. As the user types into the field, a progressive reveal occurs and displays options for users to tap on. Options can be stations (favorited or not) or map locations.

4.6.2. Selecting number of tickets

Wireframe



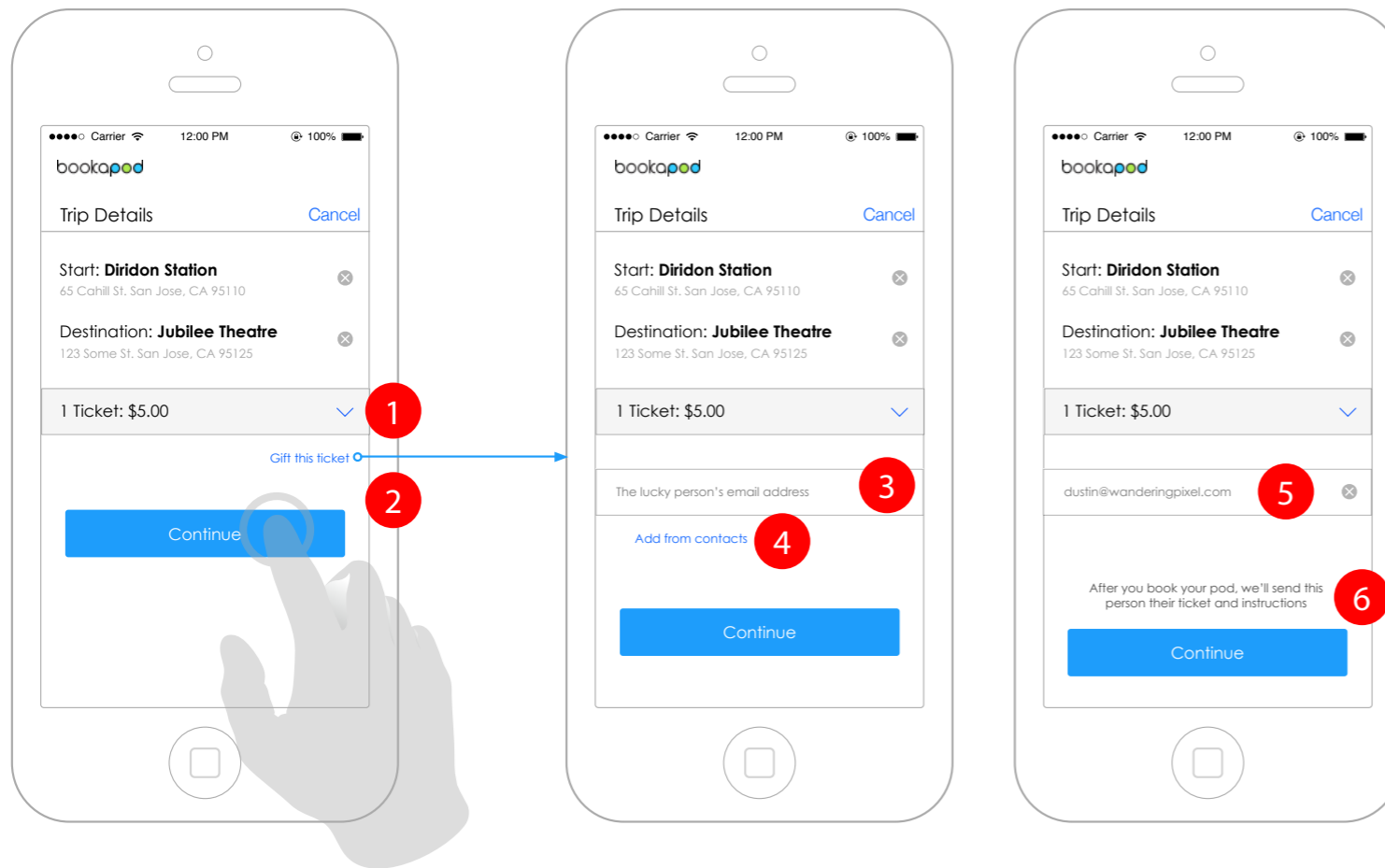
Interaction Notes

1. Once the destination station is selected, the ticket option dropdown appears. On tap, a OS selector will appear
2. Allows the user to select the number of tickets in their purchase (each quantity is paired with a price).

4.6.3. Gifting a ticket

Wireframe

Interaction Notes

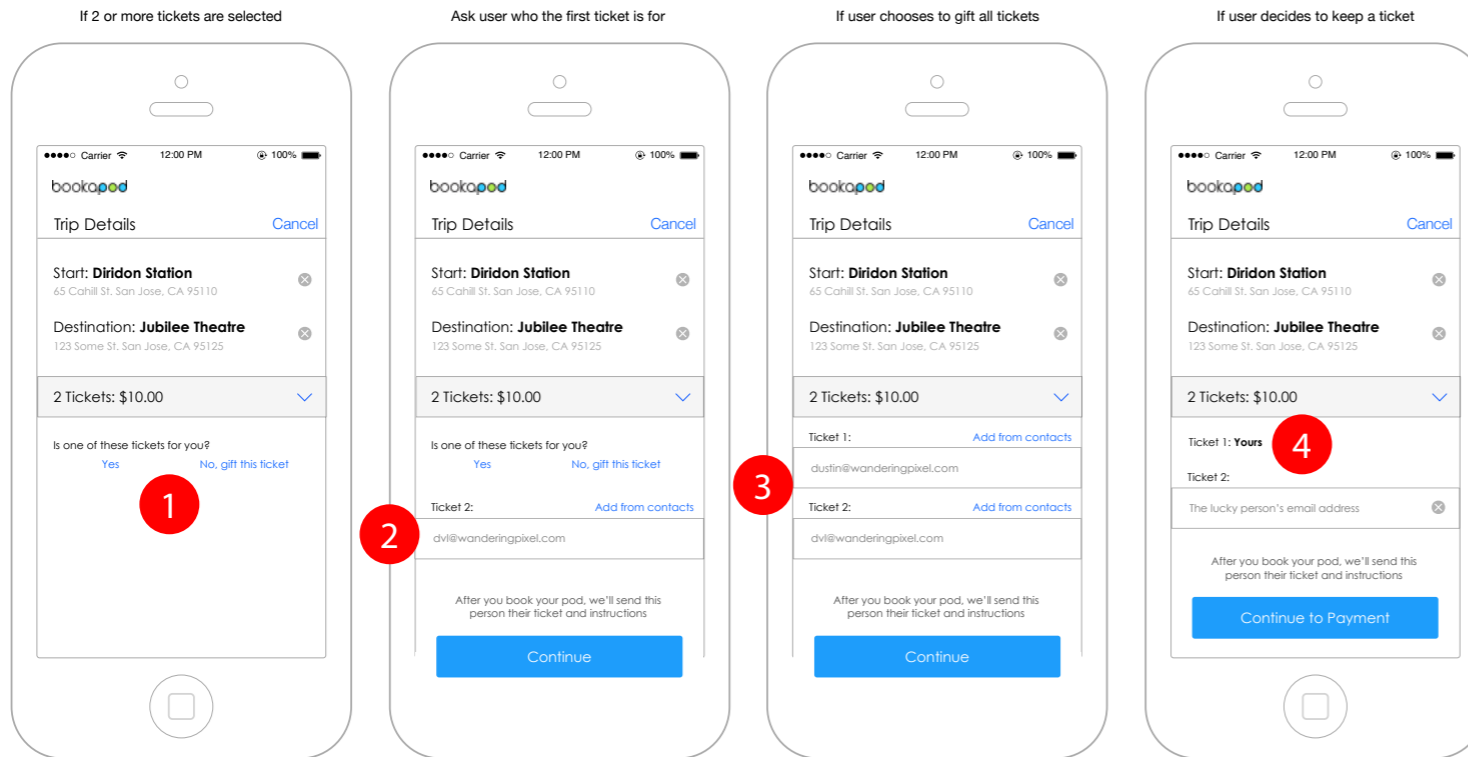


1. Once the ticket quantity is selected, the continue button will appear. The dropdown can still be selected should the user change their mind about the ticket count.
2. A gift this ticket option will also be shown, allowing the user to send the ticket to a contact
3. On tap of the "gift this ticket" link, a field will appear asking for the recipient's email address
4. The user may add the recipient from their device's contact list
5. Once the field is filled with one character, the close button will appear on the right of the field, allowing the user to clear the field if they choose
6. Message to the user appears once a valid email address has been populated into the field

4.6.4. Gifting more than 1 ticket

Wireframe

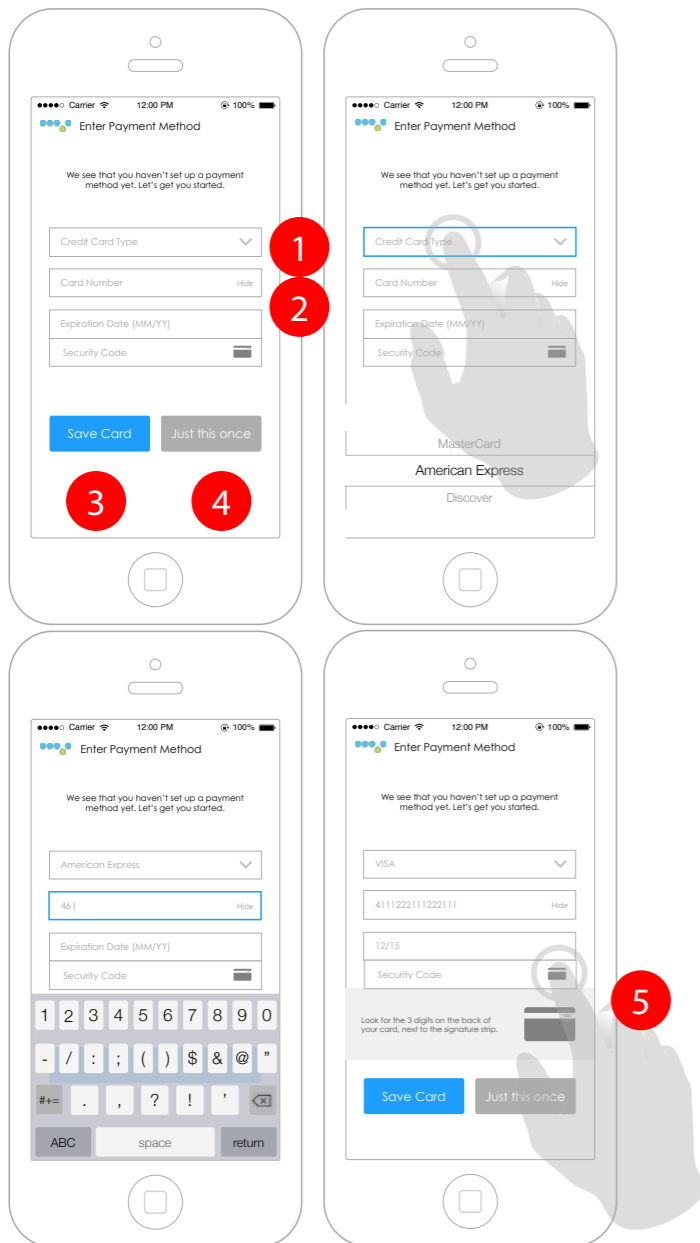
Interaction Notes



1. If the user selects to purchase more than one ticket, then they will be asked if the ticket is for them or if they would like to gift it to another.
2. The first option will always ask the user if the ticket is for them, and any remaining tickets will have their own form field, allowing the user to assign email addresses.
3. Shows the state where the user is gifting all tickets
4. Shows the state where the first ticket has been selected as the user's own

4.6.5. Credit Card / Payment Type setup

Wireframe



Interaction Notes

1. Credit card type selector - on tap, OS selector appears
 2. Credit card number field - show / hide option available for security and will unmask / mask the credit card number on tap
 3. Save card option stores the card for future use
 4. Just this once option will allow payment, but not store the user's credit card information
- Security Code (CVV / CCV) toggle - accordian animation occurs and provides instructions to the user on where to locate the code

Error States

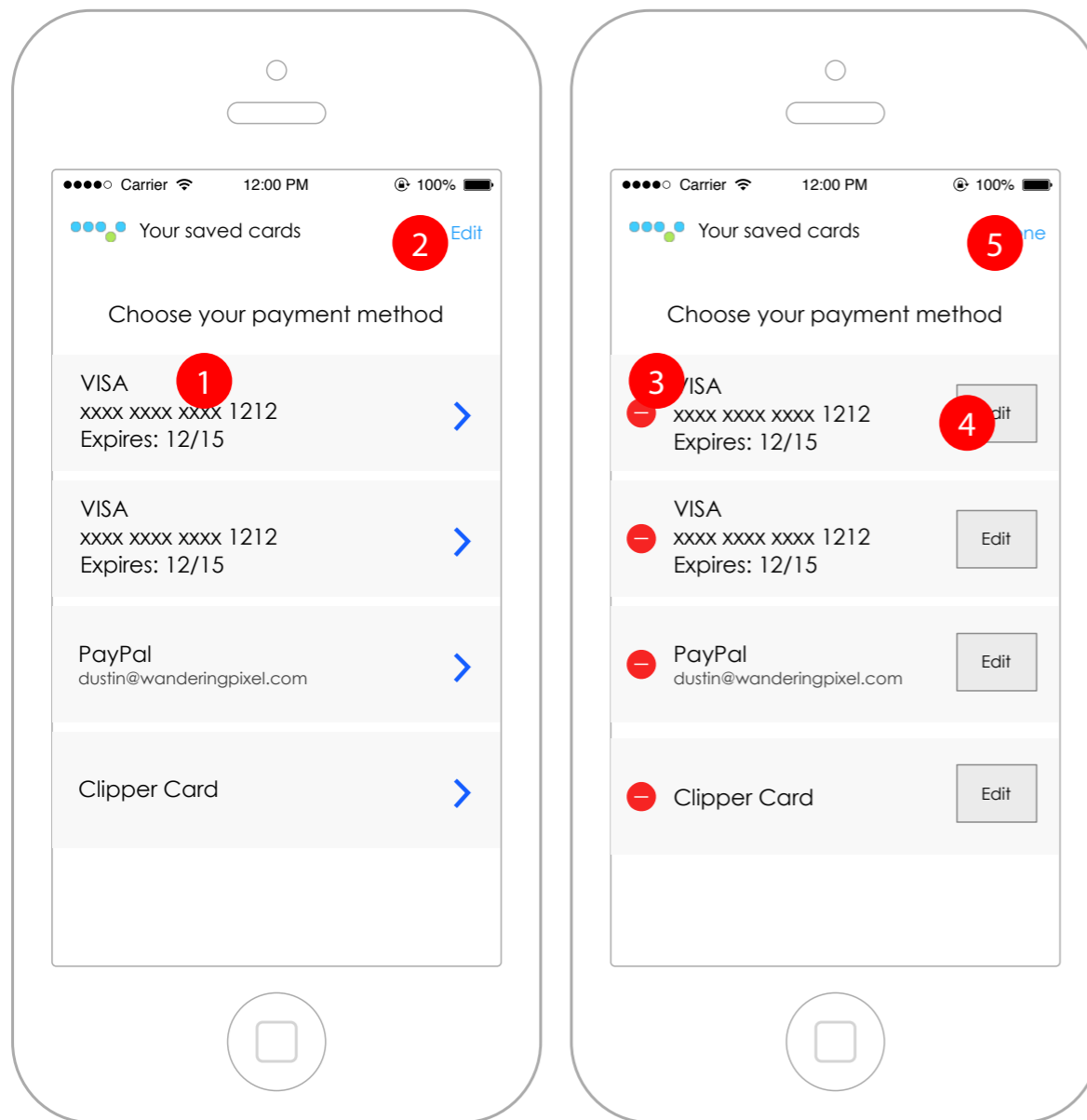
Condition	Notification message
User does not fill in any of the fields - either button is tapped	<i>You must fill out the highlighted fields</i>
Card number, exp date, or cvv code are improperly structured	<i>Please enter a valid {type of number}</i>

- Notification message will be green
- Input field error state stroke will match the color of the notification message

4.6.6. Saved Cards

Wireframe

Interaction Notes



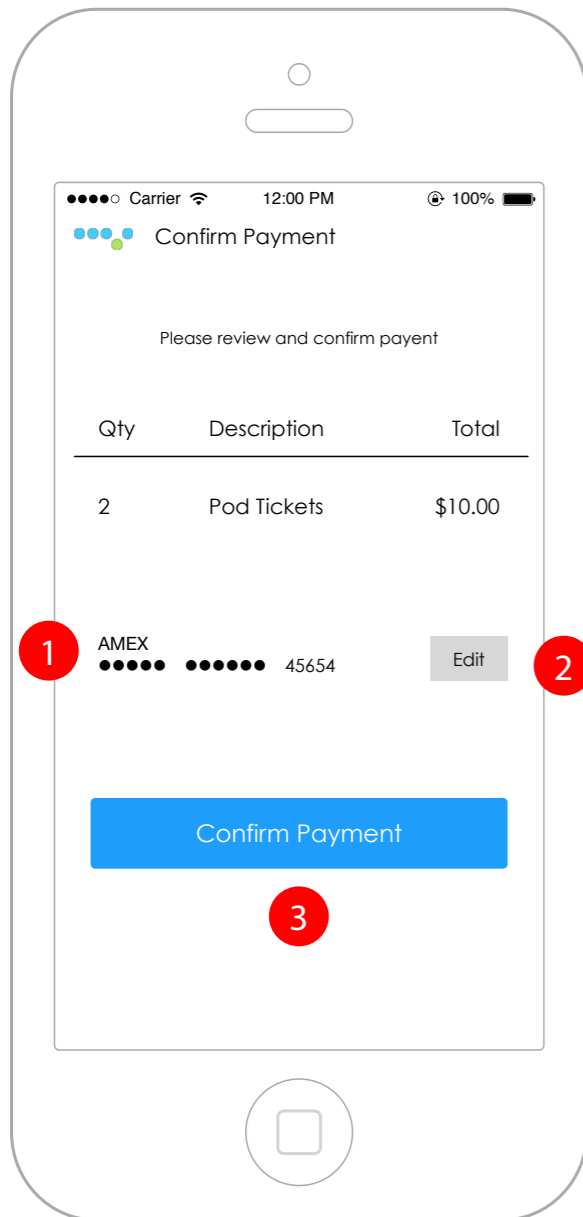
1. Saved card appears with only the last 4 digits in plain text, the rest are masked. On tap, the user will be taken to the confirm payment screen [4.6.7].

Options for payment: VISA, MasterCard, AMEX, Discover, PayPal, and Clipper Card

2. Edit - allows the user to edit each payment type or delete them from the saved cards list.
3. Delete icon - removes card from list
4. Edit button - takes the user back to the payment edit screen [4.6.5].
5. Done - stops the edit state.

4.6.7. Confirm Payment

Wireframe



Interaction Notes

1. Credit card type and number appear - credit card number is masked and chunked; however, the last 4 digits (last 5 for AMEX) appear in plain text.
2. User may tap edit to go back to payment type edit screen
3. Confirm payment - generates QR code and presents user with the QR code screen

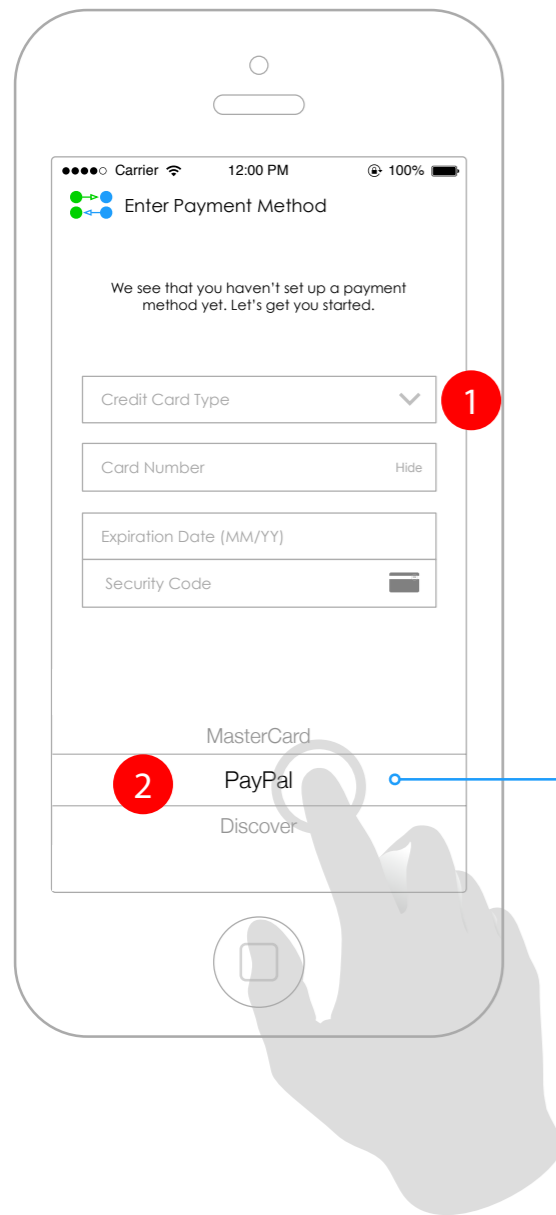
Error States

Condition	Notification message
User's credit card is declined	<i>We were unable to charge your credit card, please check that the card is correct, or enter a different payment type [edit]</i>

- Notification message will be green
- If a notification is presented when the card cannot be charged, and edit link will appear in the notification drop down allowing the user to go back to the payment type screen [\[4.6.5\]](#)

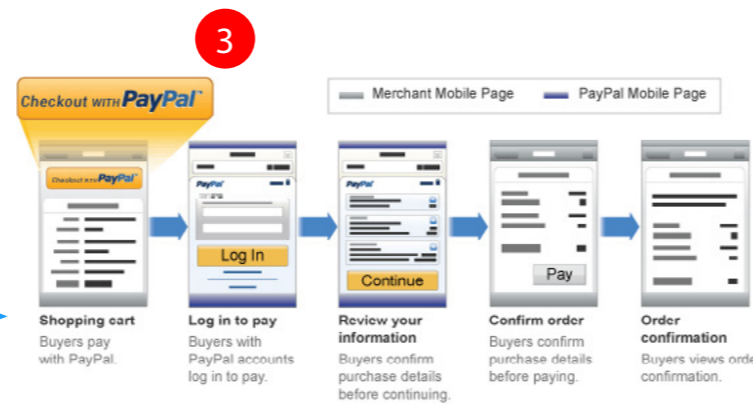
4.6.8. PayPal Payment Type

Wireframe



Interaction Notes

1. PayPal will be presented as an option in the payment type dropdown
2. The OS selector will allow the user to select PayPal. On tap, it takes the user into the PayPal Flow
3. Leads user into the PayPal checkout flow (outside of the Bookapod environment). When completed, the user will be returned to the Bookapod environment.

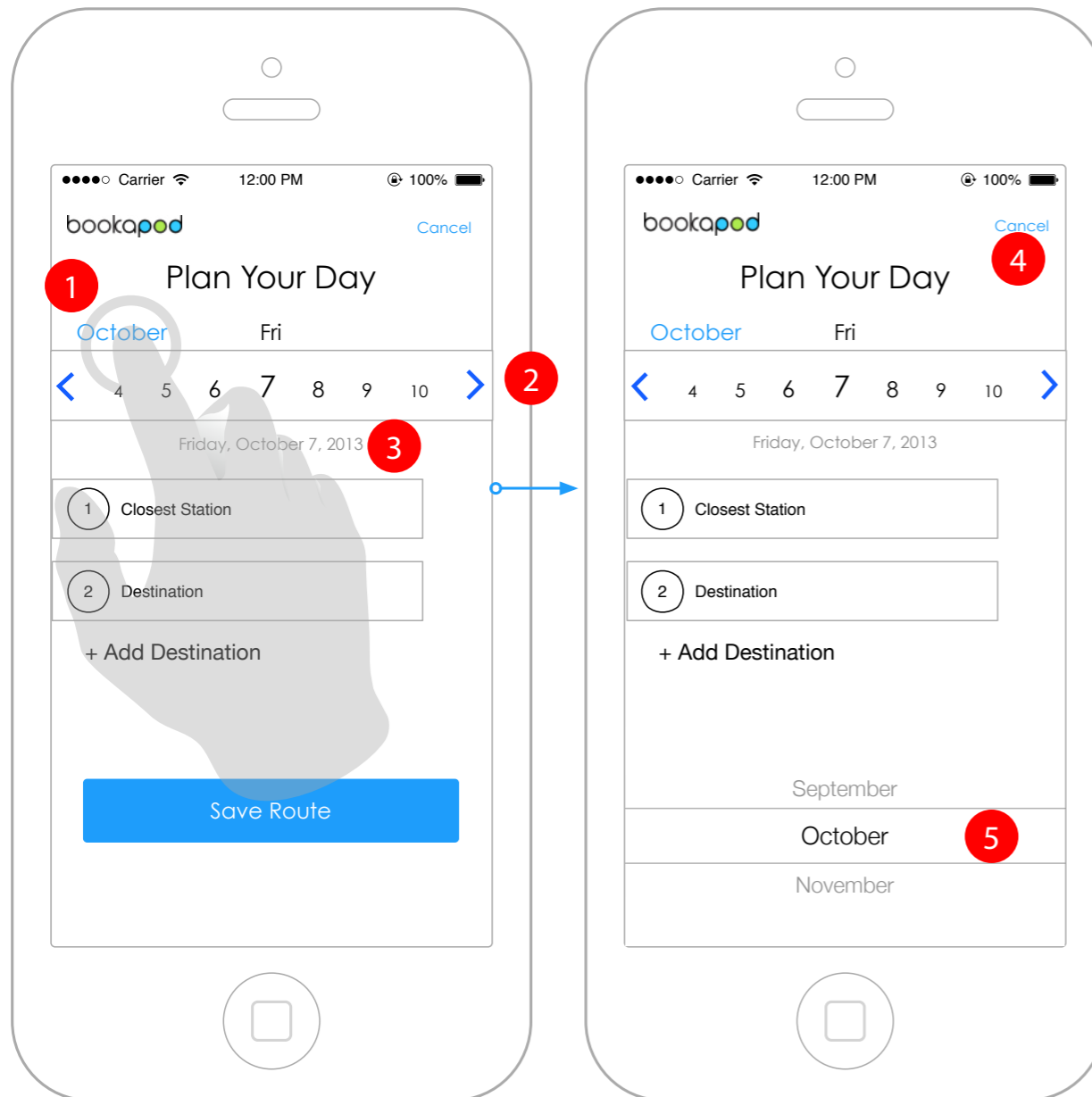


4.7. Schedule Your Day (Scheduling System)

4.7.1. Date Selector

Wireframe

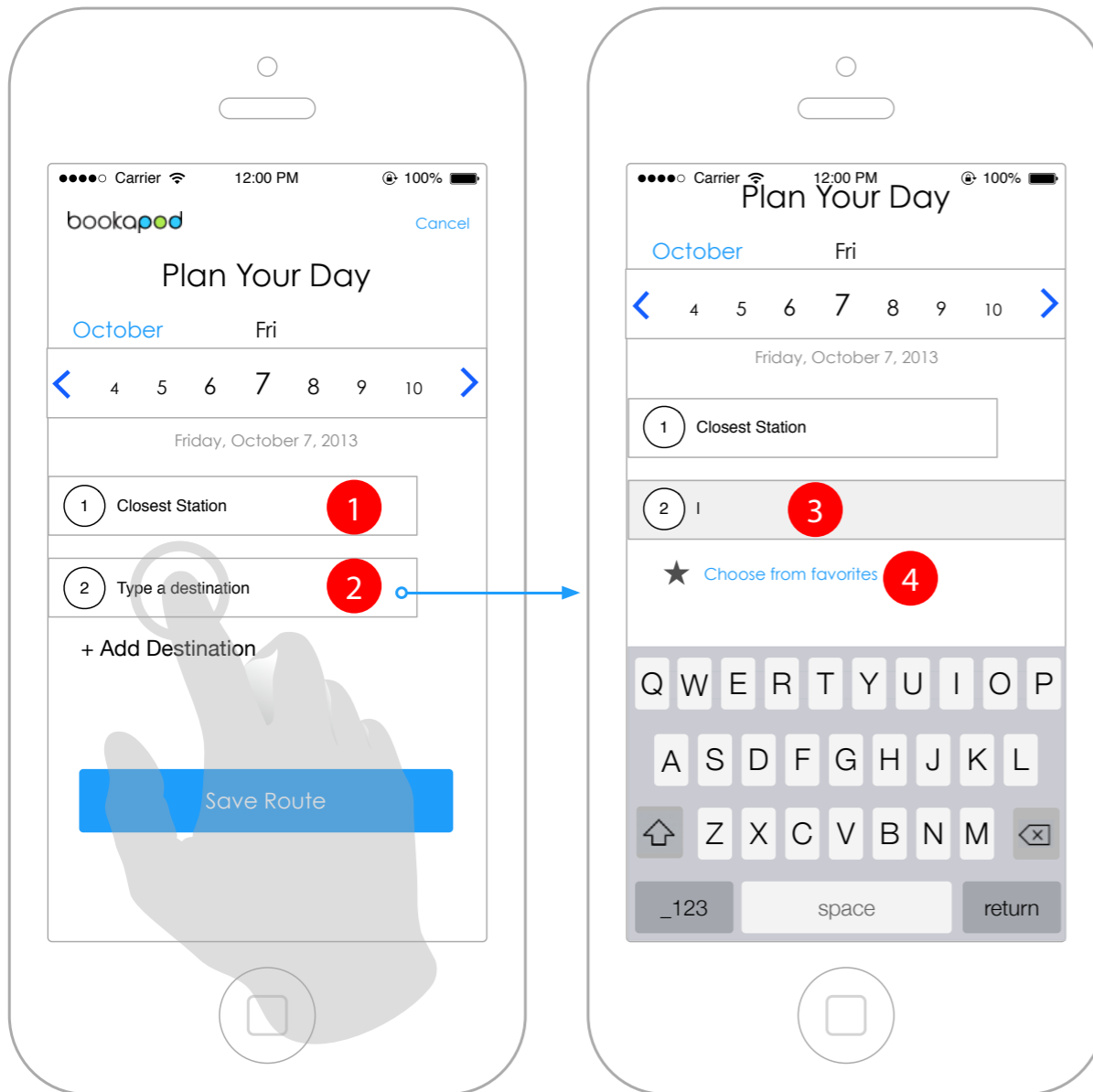
Interaction Notes



1. Month link (plain text) - on tap, shows a selector, allowing the user to change the month should a schedule need to happen in the future.
2. Day selector - this is a scrollable carousel. The user can swipe it or tap on the arrows to control the days. The current day will always show in the middle of the dial and be larger than the rest.
3. The date selected will be shown in full format (Day of week, month, date number, year).
4. Cancel button will close the scheduler and take the user back to the main app screen
5. Date selector (OS specific)

4.7.2. Selecting Destinations

Wireframe



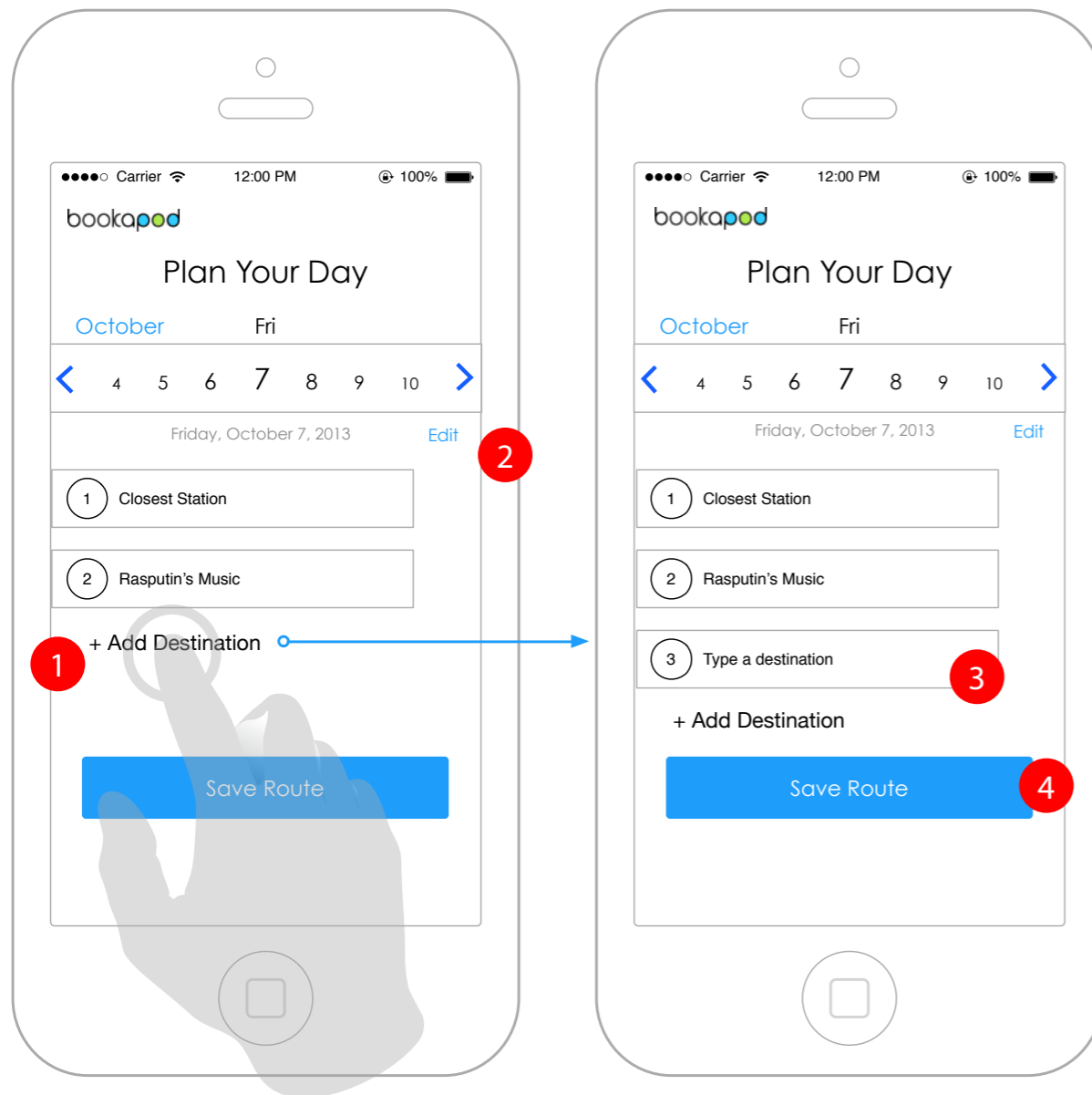
Interaction Notes

1. Closest Station is auto populated with the closest station to the user. The user can tap on this to edit the starting location.
2. First destination station - the user will tap and begin typing their destination. A progressive reveal will occur based on the characters that are typed into the field.
3. Active field state - on field focus, the background and borders of the field are highlighted.
4. Choose from favorites option - on tap, shows the favorites list [\[4.5.2\]](#)

4.7.3. Adding additional destinations

Wireframe

Interaction Notes

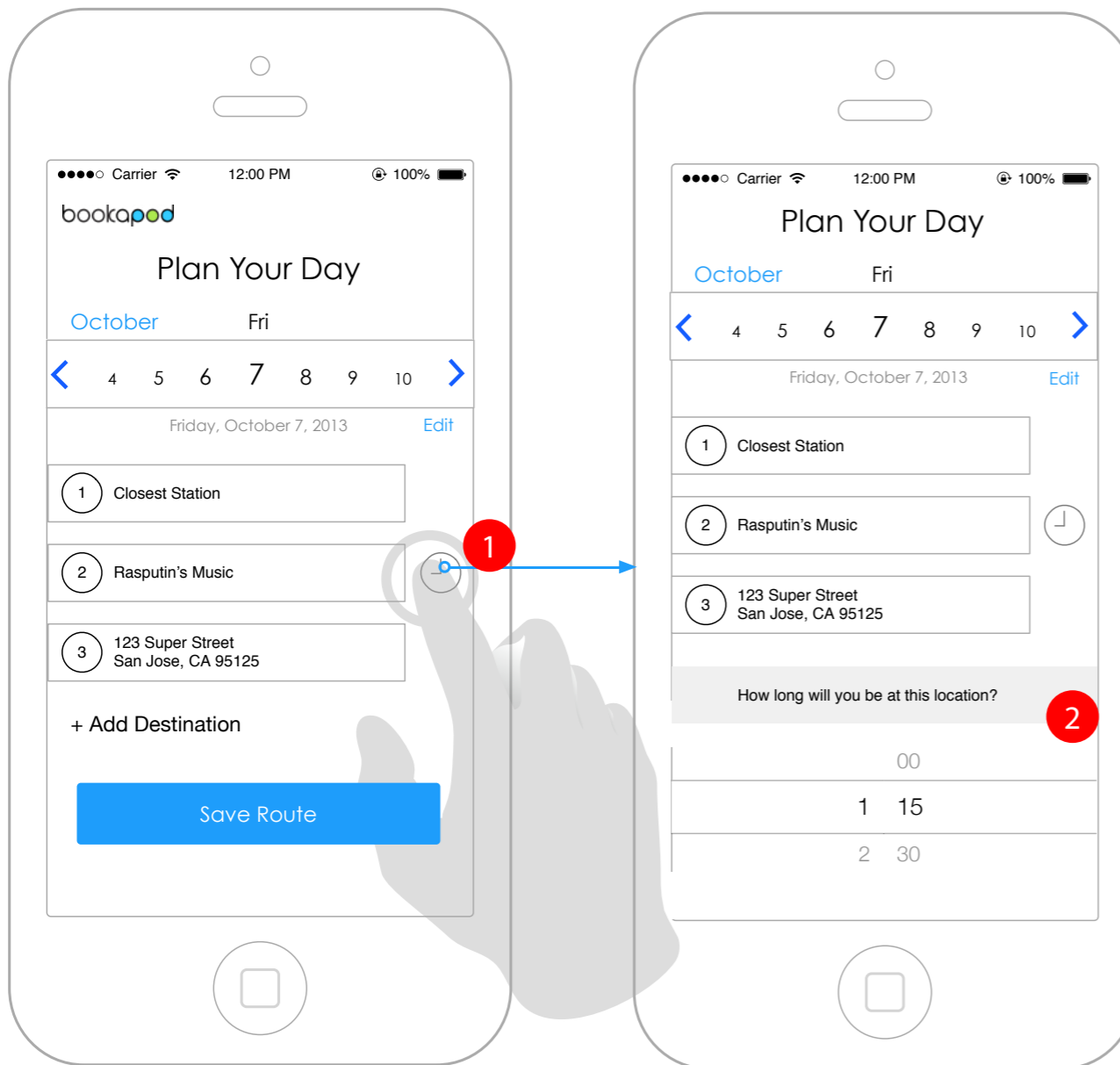


1. Add destination link is always present on screen and will allow the user to add as many destinations as they wish.
2. Once the first destination has been added, the edit button will appear on screen.
3. Subsequent destination fields will be shown as each new destination is added.
4. Save route takes user to [\[4.7.7\]](#)

4.7.4. Destination Buffer

Wireframe

Interaction Notes

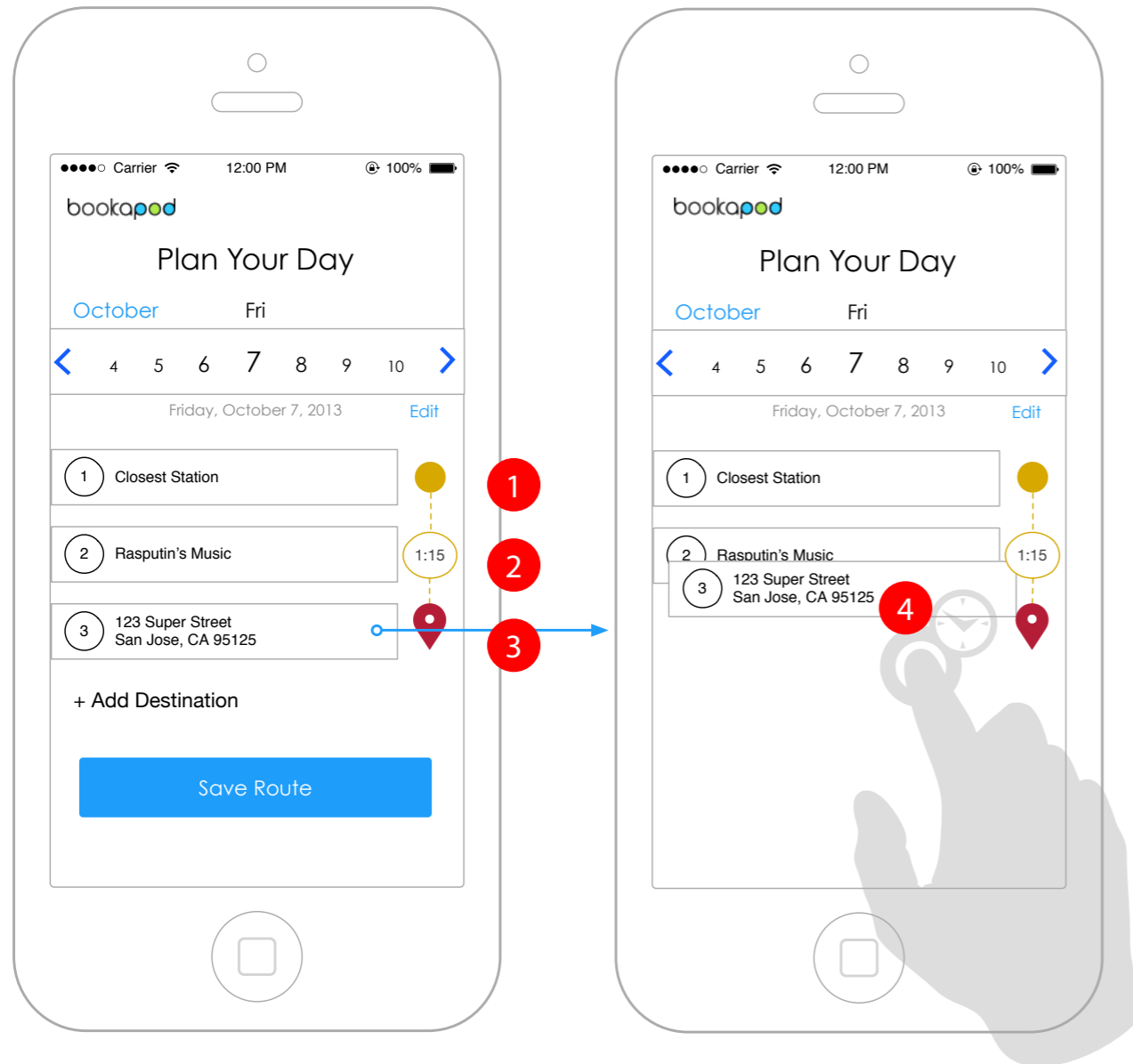


1. Once the third destination is added, a timer icon will appear to the right of the second destination. On tap, a selector will appear asking the user to set the duration at that location. This allows the user to give themselves a buffer at any specific location and the system will give them a notification when they need to book their next pod.
2. Selector - user can scroll in hour increments on the left dial, and in 15 minute increments on the right dial.

4.7.5. Quick destination reorder

Wireframe

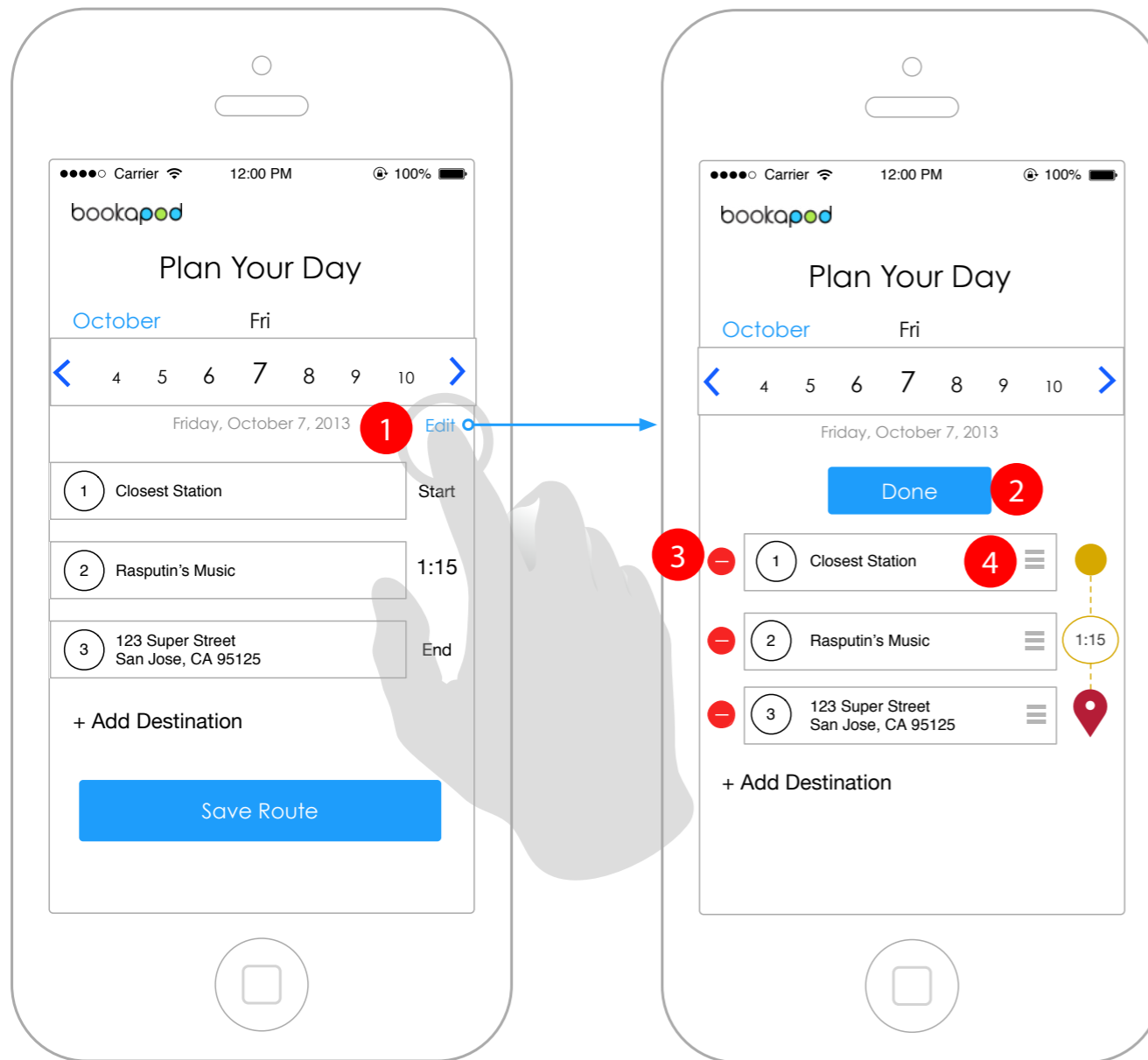
Interaction Notes



1. The scheduler indicates the start location in green
2. The time buffer is set with a green circle around the duration
3. The end destination is indicated in red
4. The user can perform a quick reorder by tapping and holding on the destination and dragging it above or below any adjacent locations.

4.7.6. Destination Edit Screen

Wireframe



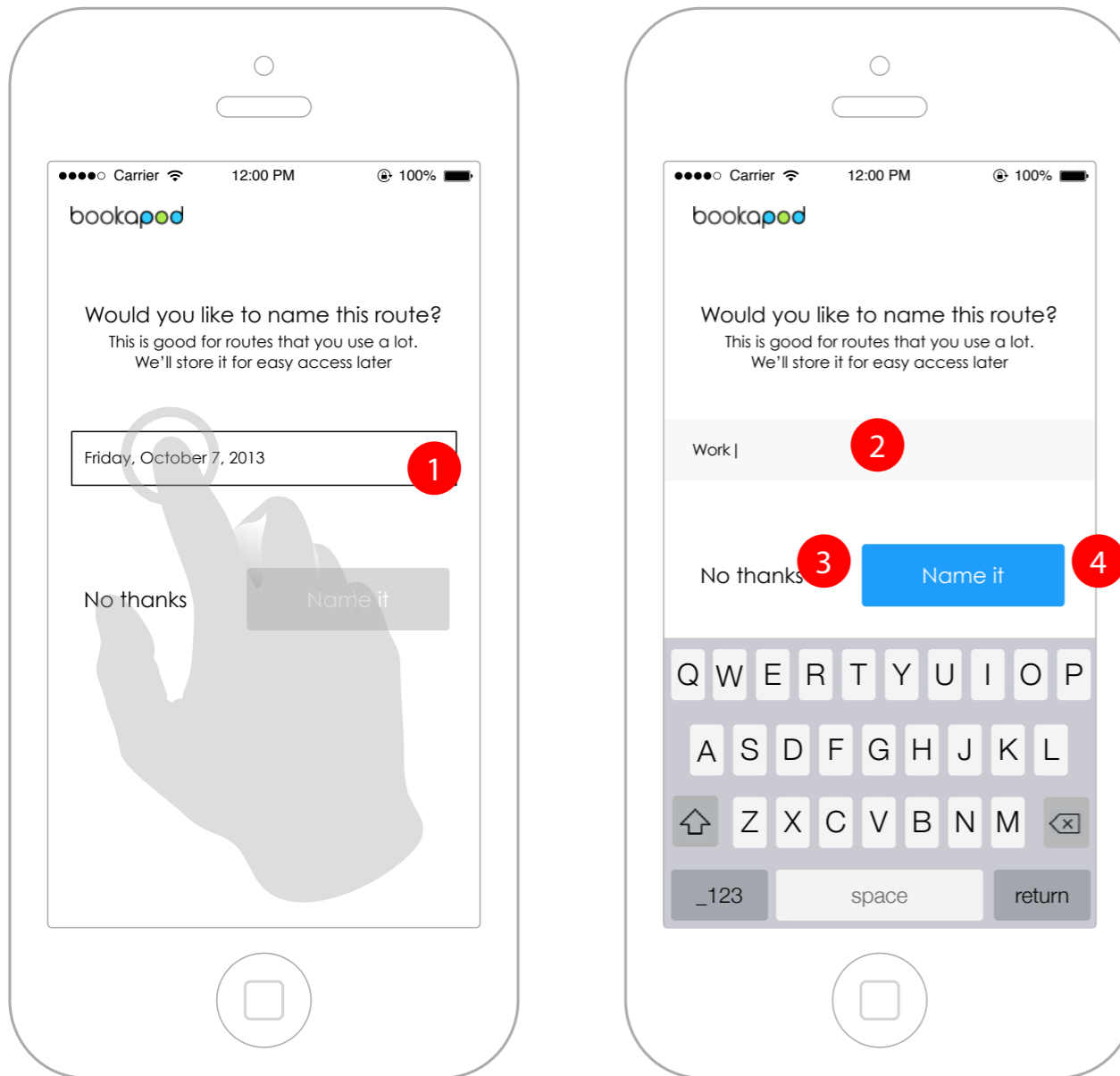
Interaction Notes

1. On tap, the edit button will change the screen so that every destination is editable
2. The Done button terminates the edit session
3. Each destination in the schedule can be deleted
4. Each destination can be tapped, held, and reordered.

4.7.7. Saving scheduled routes

Wireframe

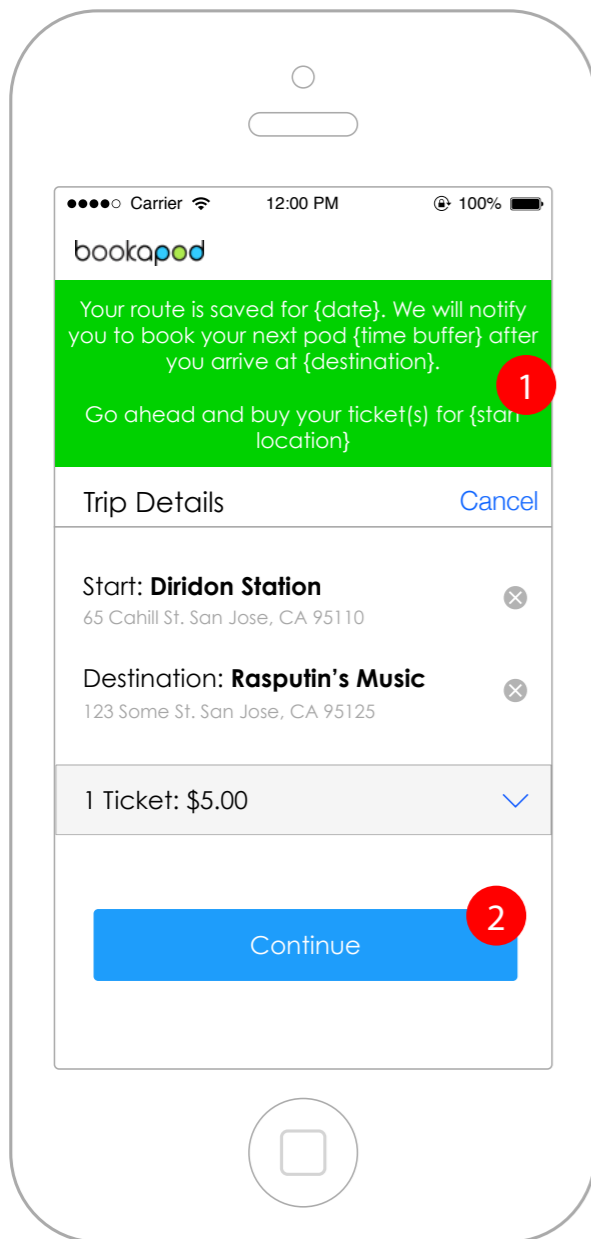
Interaction Notes



1. By default, the saved route name will default to the current date. User can tap to enter route name
2. Edit state of the field - background highlights and field widens
3. No thanks option - will continue to use the default date so the user knows that they are using a scheduled route in the main app screen
4. Name it option - will save the route with a custom name

4.7.8. Route saved confirmation

Wireframe



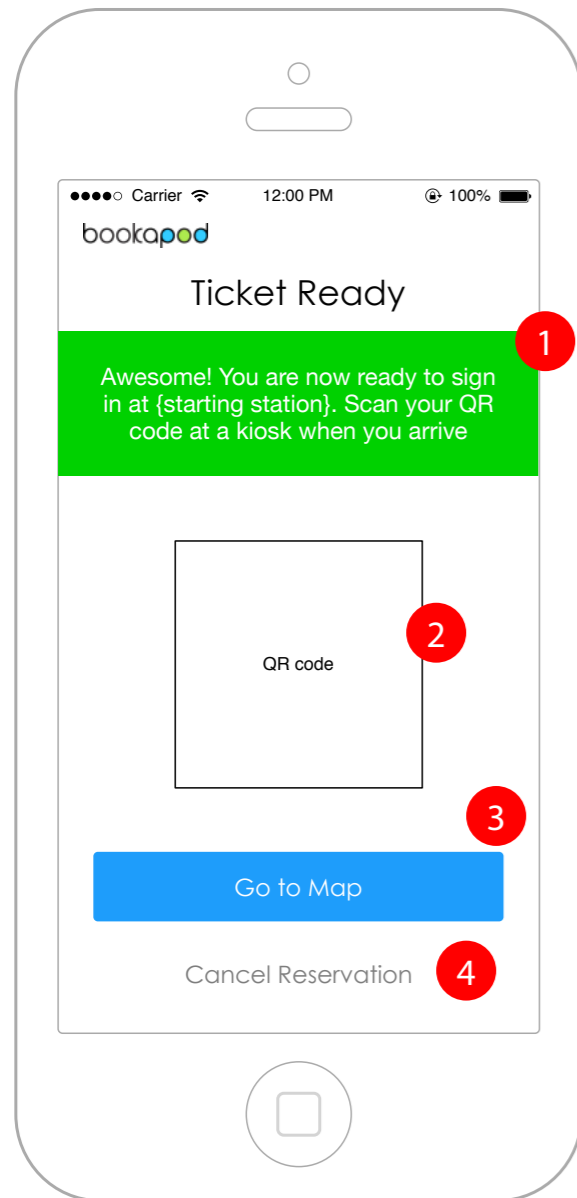
Interaction Notes

1. The user will be notified that their route is saved for {date}, and will ask them to select their ticket quantity.
2. Continue button - on tap, the user will be taken to the payment option (either set up a new card [4.6.5] or a saved cards [4.6.6])

4.7.9. QR code generation

Wireframe

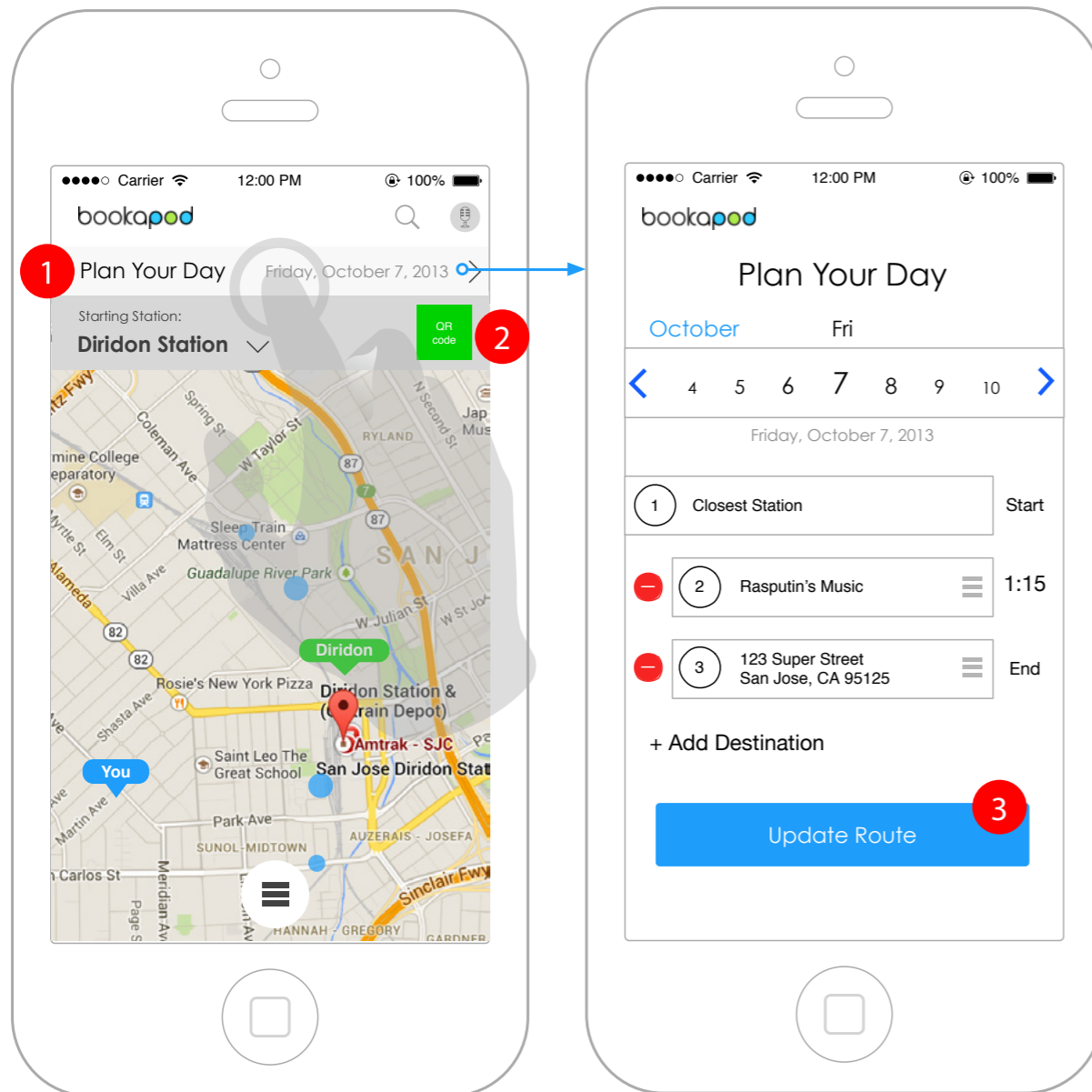
Interaction Notes



1. Confirmation message appears at the top of the screen notifying users that they are now in queue at the nearest station and that they will need to scan their QR code on the kiosk when they arrive.
2. QR code - available to scan on the kiosk
3. Go to map button - takes user to the map screen
4. Cancels the user's reservation and destroys the QR code

4.7.10. Updating Route (Post QR code generation)

Wireframe

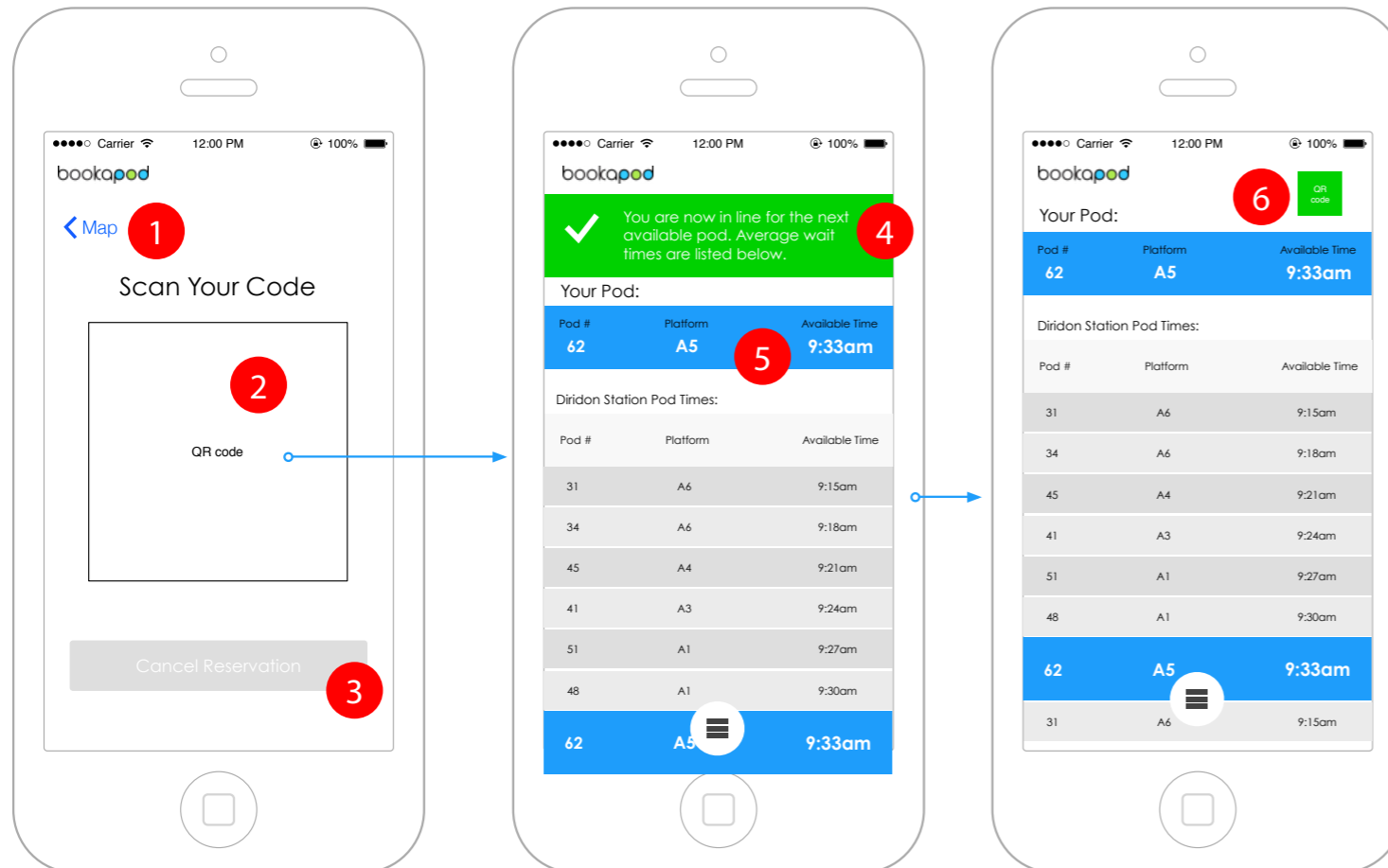


Interaction Notes

1. Plan your day banner - on tap, it takes the user to the edit page, allowing the user to make any changes they need to their route.
2. QR code is always available to the user. They may scan it on this screen or tap it to see the larger QR code screen.
3. Update route - on tap, it updates the user's scheduled route.

4.7.11. Scanning QR code into station queue

Wireframe

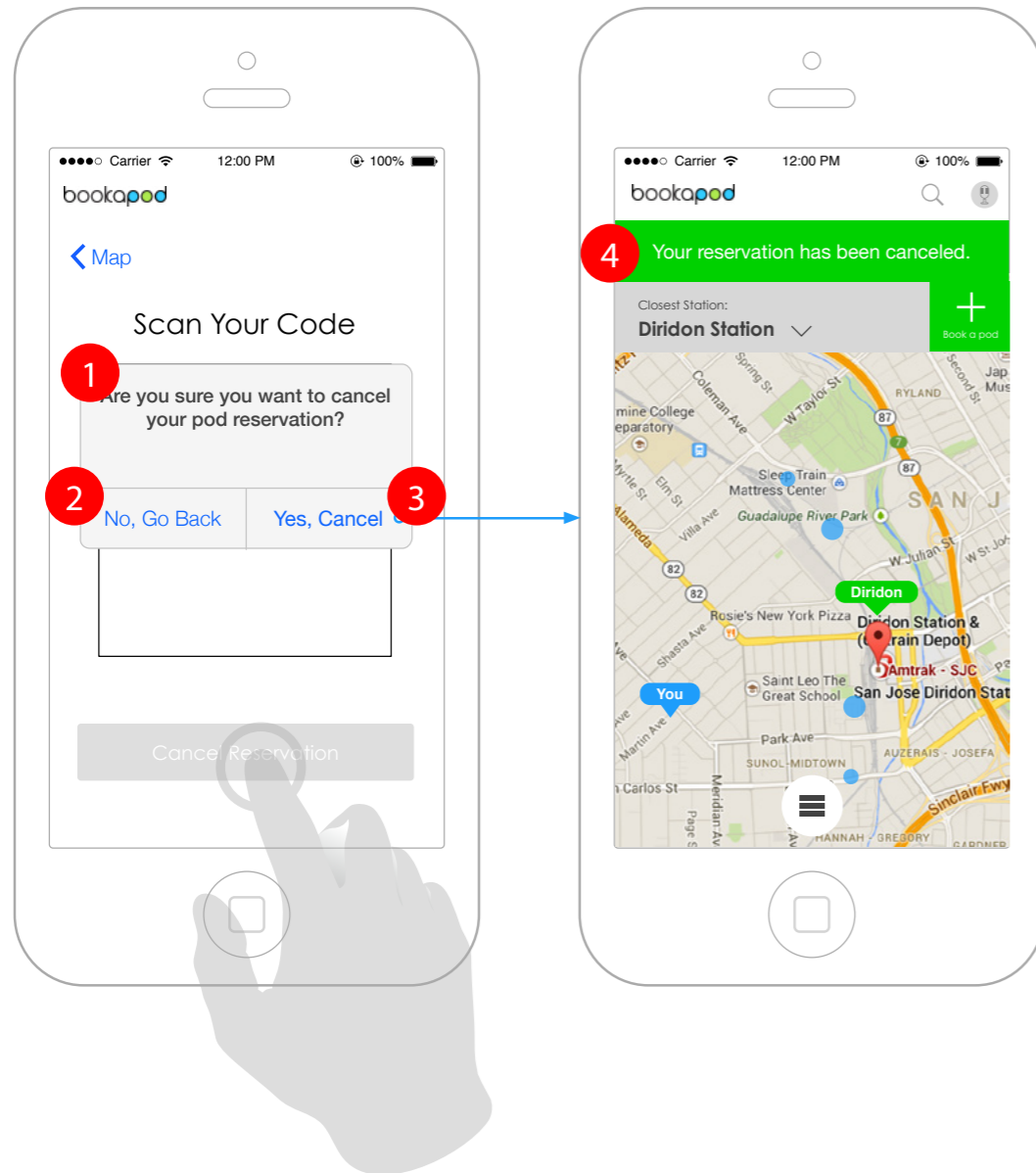


Interaction Notes

1. Map link - takes the user back to the main app screen [4.7.10] should they need to view it (note that mini QR code will always be available on that screen)
2. QR code - scannable at station kiosk. When scanned, it drops the user into the station queue and sets them up for the next available pod (#5).
3. Cancel reservation - terminates the reservation
4. Notification when QR code is scanned at station that the user is now in queue for the next available pod.
5. Station pod listing - shows the user the queue, their pod is highlighted along with the loading platform and time. Note that this same listing will be shown (sans the highlighted user pod) on station screens.
6. Mini QR code - always available to scan on this screen, or the user may tap it and it will show them the larger QR code (#2).

4.7.12. Cancel reservation

Wireframe



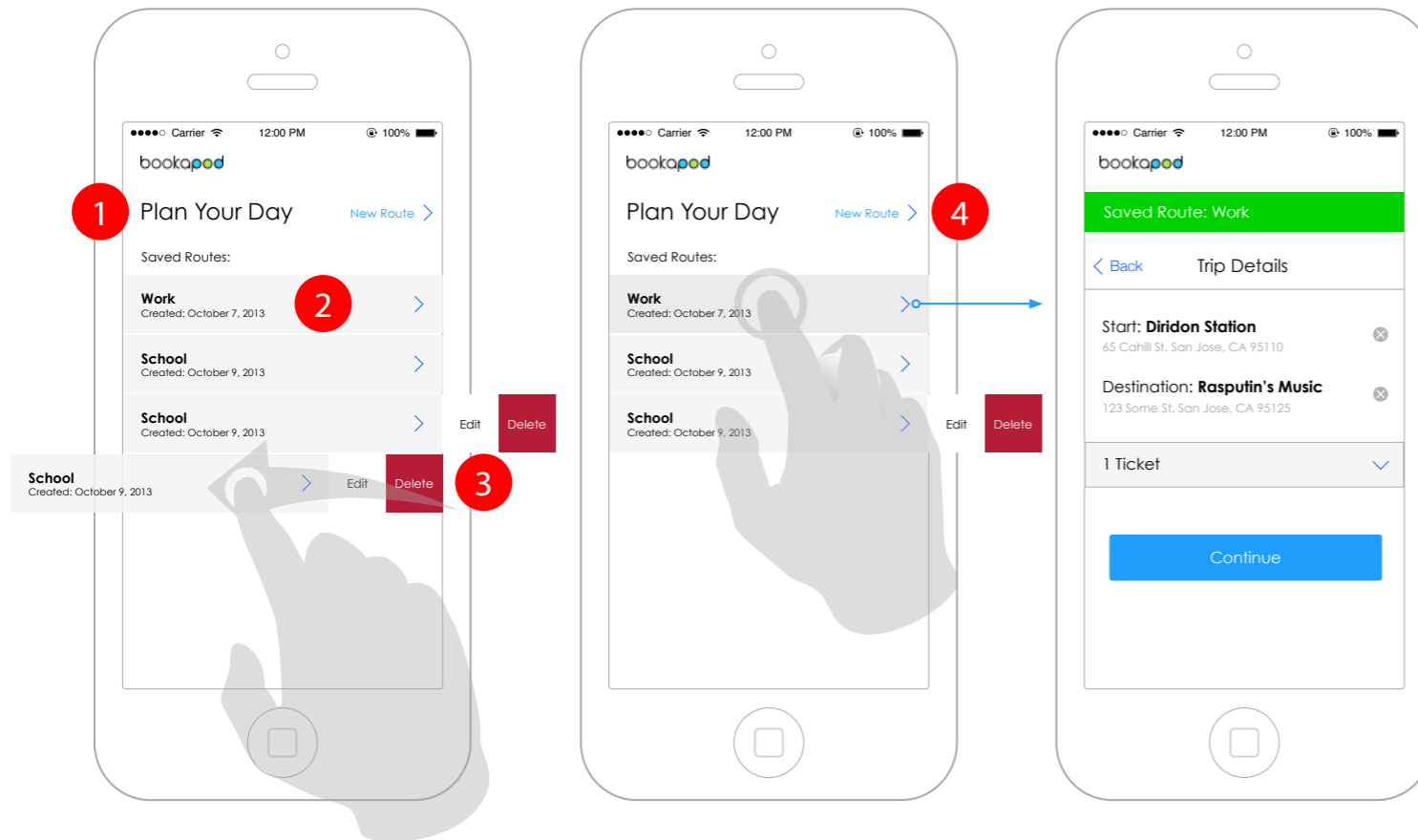
Interaction Notes

1. Cancel reservation OS modal is triggered when the user opts to cancel the reservation
2. No, go back - kills the modal and leaves the user on the QR code screen
3. Yes, cancel - takes the user to (#4)
4. Canceled reservation notification - user is now back on the main app screen and the notification will go away as soon as the user begins to interact with this screen

4.7.13. Saved routes

Wireframe

Interaction Notes



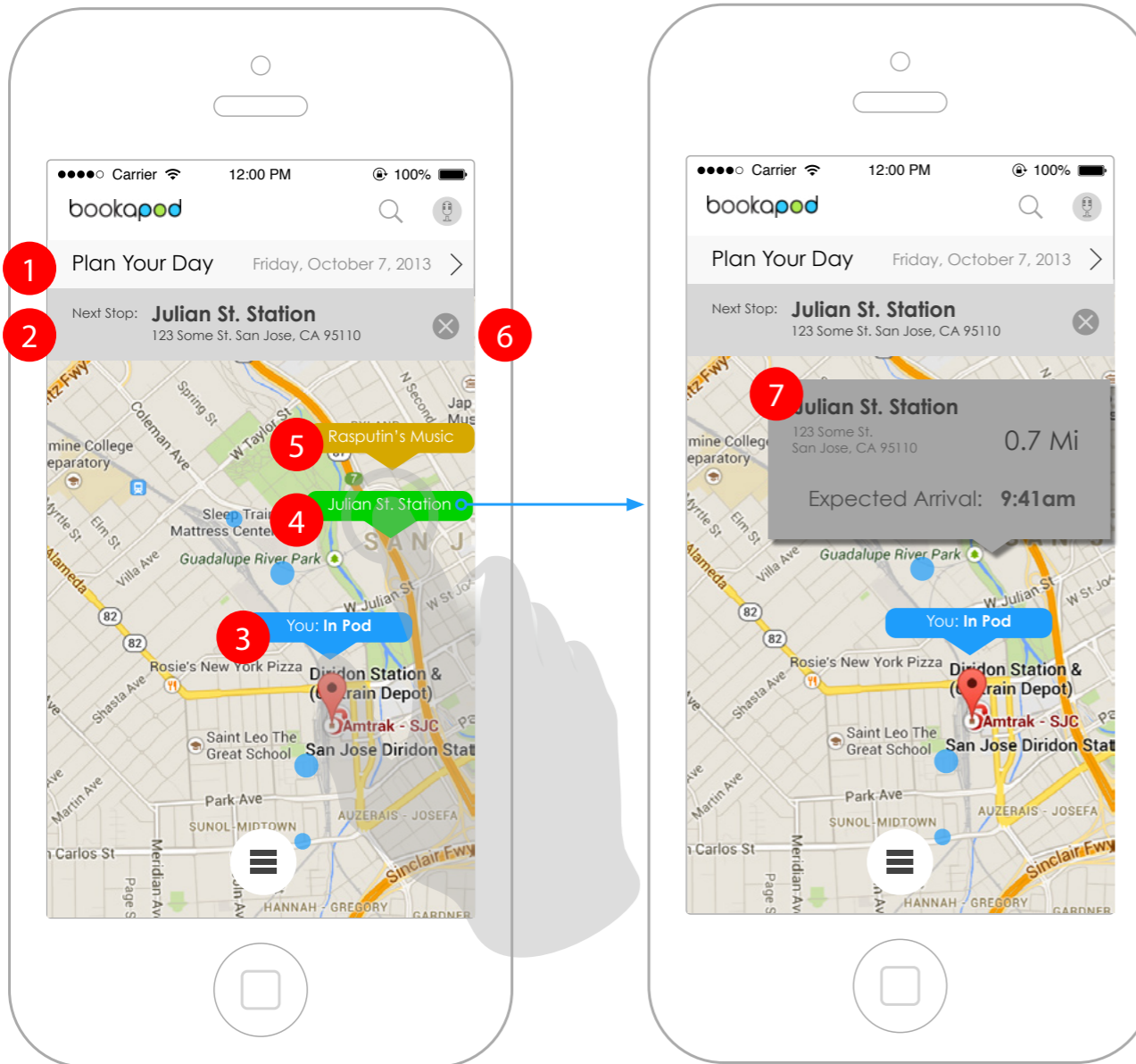
1. If the user has already set up saved routes in their scheduler, this screen will display first when tapping on the scheduler icon from the primary navigation
2. A previously saved route - Shows custom name (if defined) and the date created. If not custom name is defined, the entry will only show the date. On tap, it will take the user to the ticket selection screen.
3. Entries are single finger swipable (to the left), allowing the user to edit their route or delete the entry.
4. Allows the user to create a new schedule or route

4.8. In-pod navigation

4.8.1. In-pod navigation map and destination popup

Wireframe

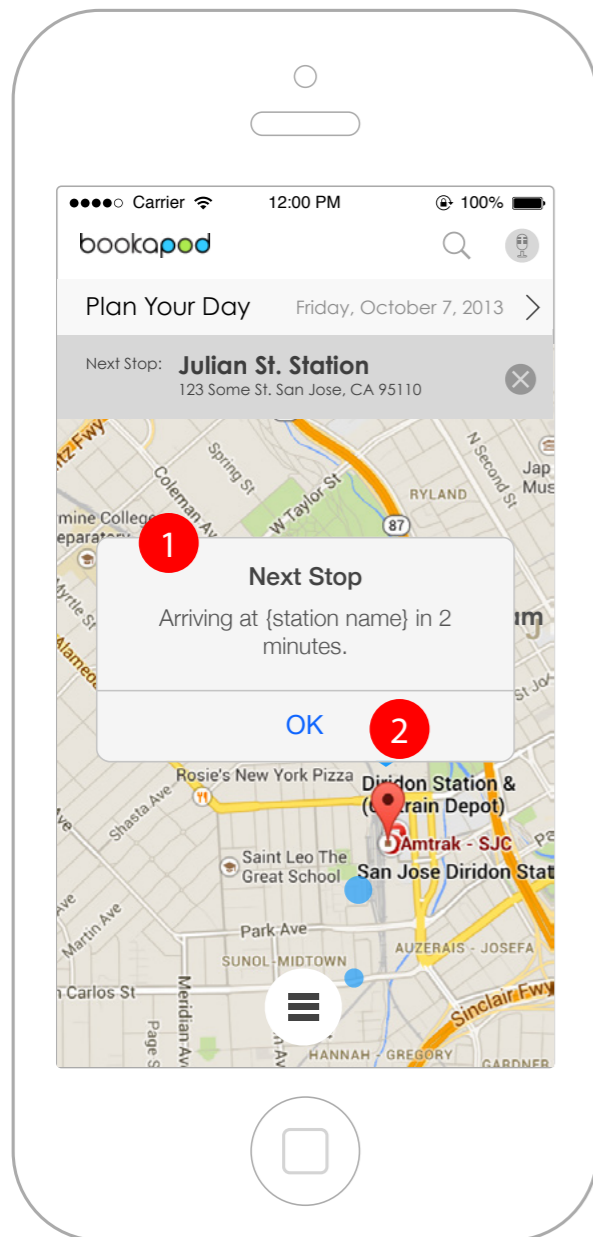
Interaction Notes



1. Plan your day banner is persistent if the user is navigating from a pre-created route. This banner will not appear if the in-pod navigation is started from the basic reservation flow.
2. Next stop banner - user can tap on the X icon to remove this banner from the map screen
3. Shows where the user is in pod on the map
4. Shows the next stop in green
5. Shows any tertiary stops in orange
6. close icon for the next stop banner
7. Next stop pop up - if user taps on any of the upcoming stops, this popup will give details about the stop including distance to the stop and expected arrival time.

4.8.2. In-pod destination alarm

Wireframe



Interaction Notes

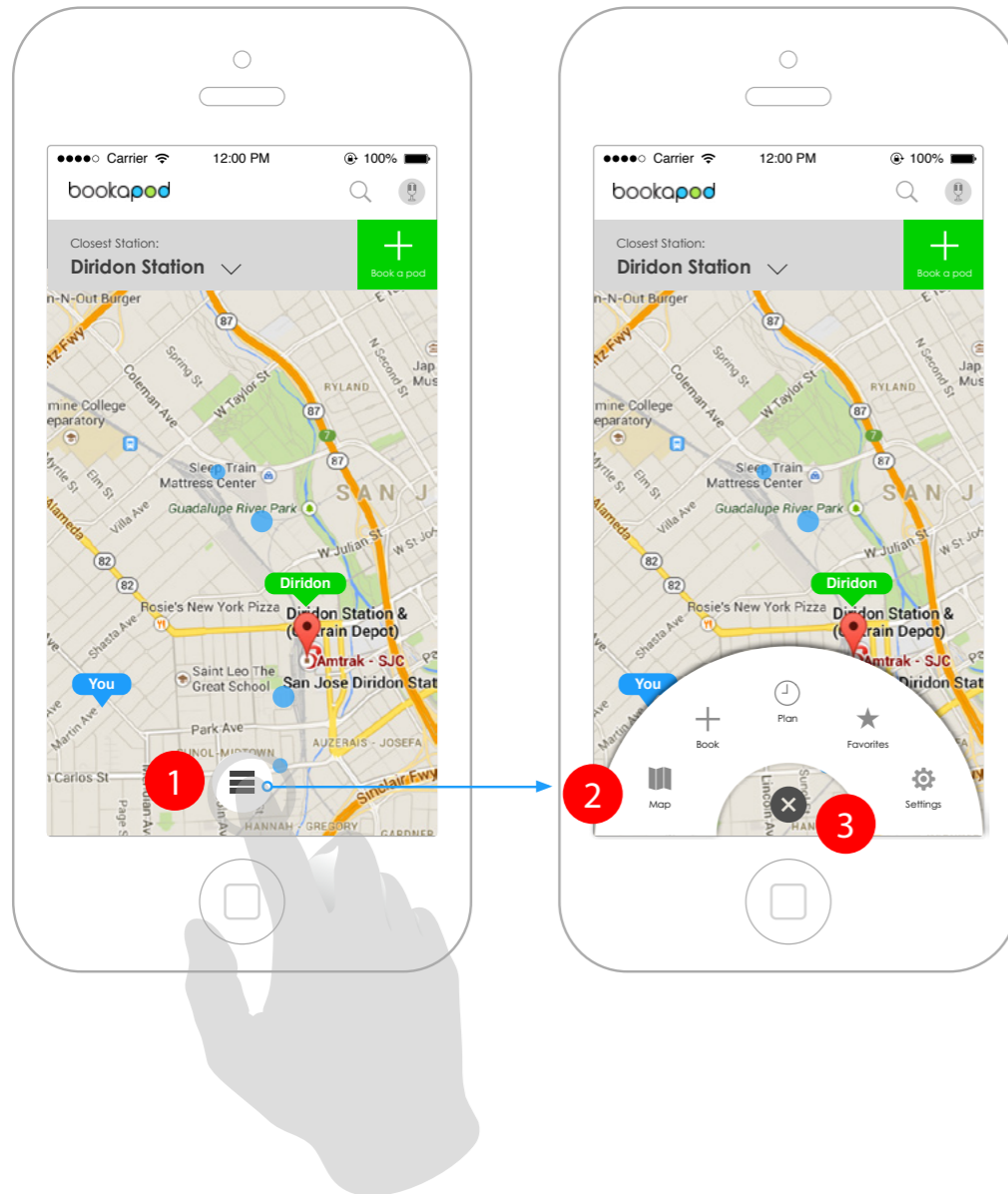
1. Each destination along a user's route will provide them with an alarm notification. This option is configurable in settings and can be set to notify a user at particular times before the destination is reached, or turned off entirely. By default, the alarm notification is set to 2 minutes before the destination is reached.
2. Ok button - terminates the alarm notification and takes the user back to the in-pod navigation map.

4.9. Primary Navigation

4.9.1. Primary navigation control

Wireframe

Interaction Notes



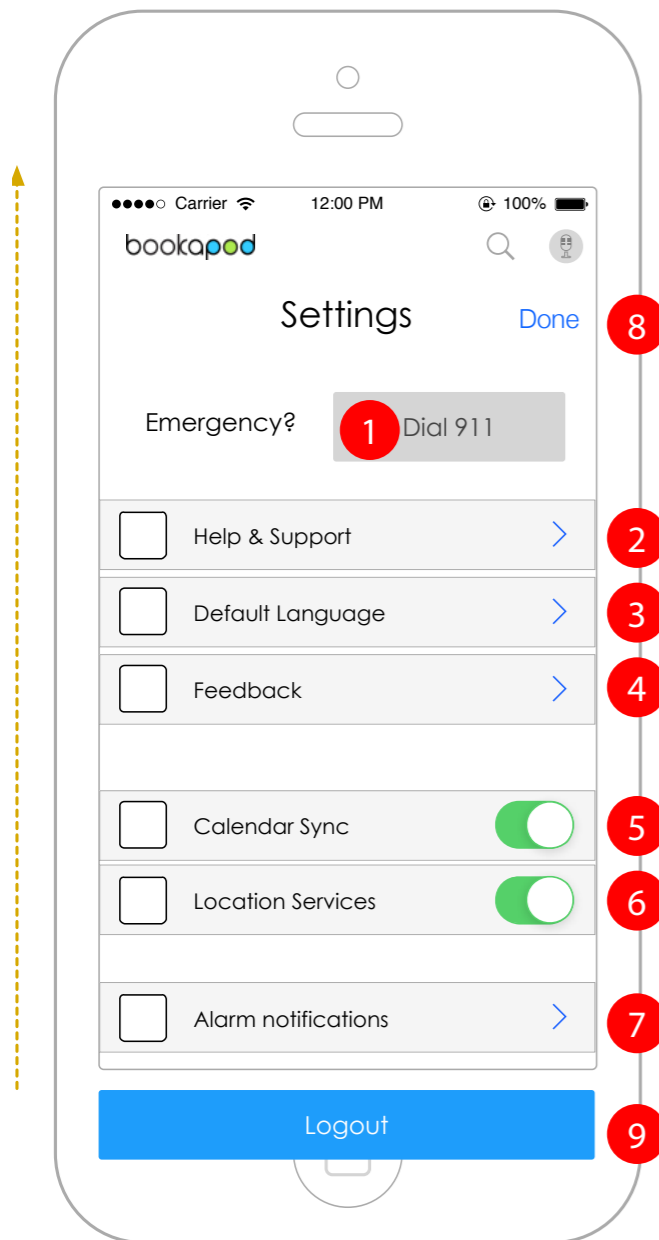
1. Primary navigation menu icon - on tap, the rainbow menu slides up
2. Menu contains 5 icons:
 - Map* - main app screen
 - Book a pod* - books a pod at the nearest station
 - Plan* - starts scheduler
 - Favorites* - shows favorites list
 - Settings* - shows settingsIf any of these icons are tapped, they lead the user to their respective pages; however, the menu will not be shown on those pages.
3. Once the menu slides up, the icon in the middle turns into an X. On tap, it closes the navigation menu.

4.10. Settings

4.10.1. Primary settings screen

Wireframe

Interaction Notes

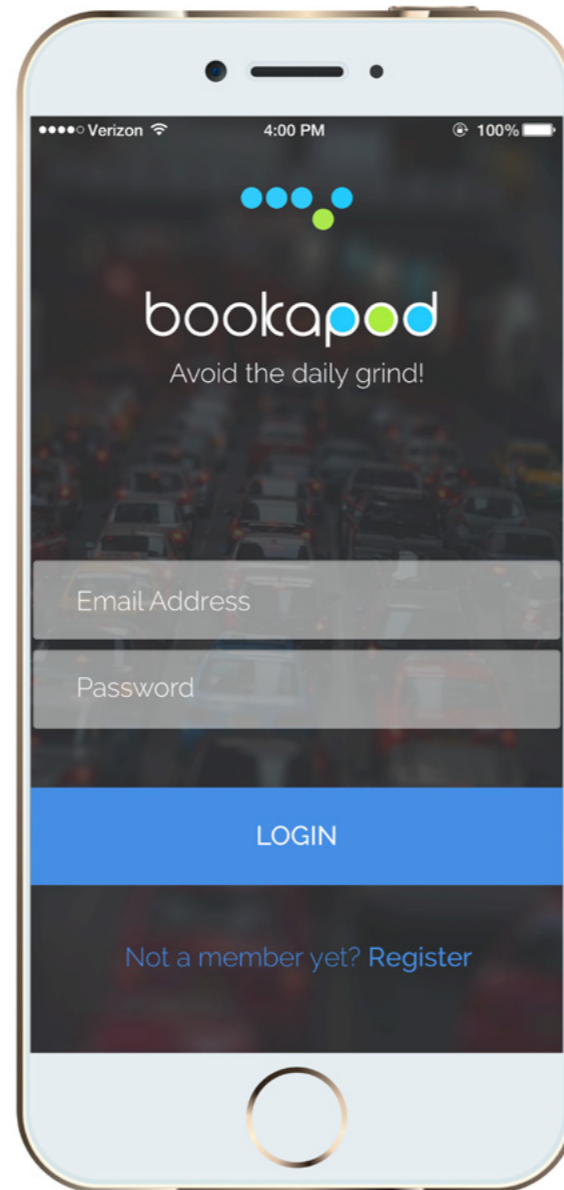


1. Emergency dial - single tap will bring up device phone screen confirming the dialing of 911.
2. Help & Support - a list a help and support information about the app. Includes support website URL and contact numbers.
3. Default language - on tap, the user can select their language
4. Feedback - allows the user to provide feedback about the app
5. Calendar Sync - allows the user to sync their device's default calendar to the app (BETA)
6. Location services - allows the bookapod app to use GPS to track the location of the user's device
7. Alarm notifications - allows the user to set the duration of the alarm notification as well as any ring tones they wish to use
8. Done - closes the setting screen. Screen animates downward on close.
9. Logout - takes user to the main login screen

5. Mockups and Content

5.1. App Icon and Login Screen

Mockups



Content

Device Home Screen App Title:

Bookapod

Login Screen:

Bookapod (logo)

[Avoid the daily grind!]

Email Field:

[Email Address]

Password Field:

[Password]

Login Button:

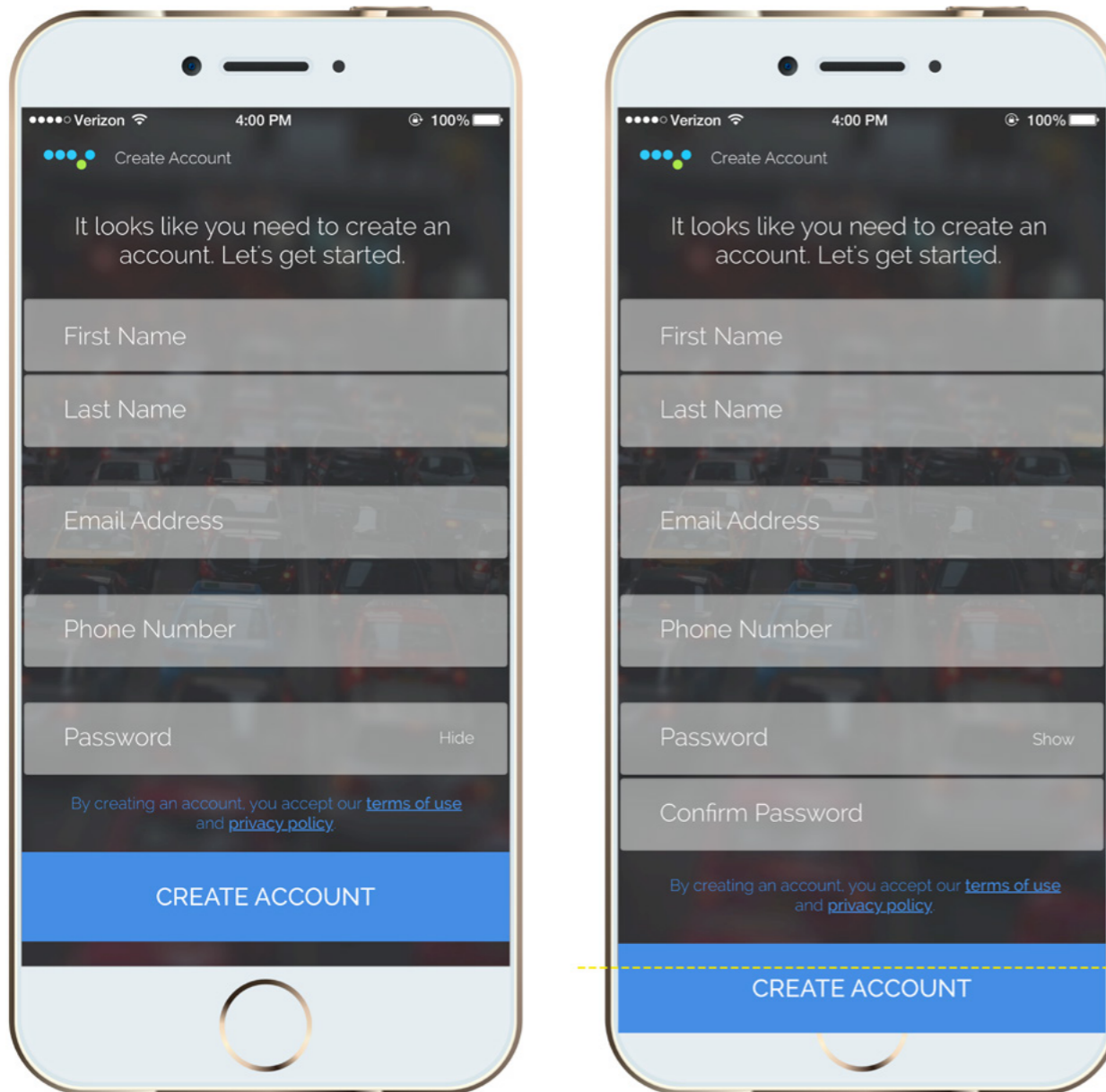
[LOGIN]

Register Link:

[Not a member yet? Register]

5.2. Create Account

Mockups



Content

Create Account Screen:

{logo} Create Account

It looks like you need to create an account. Let's get started.

First Name

Last Name

Email Address

Phone Number

Password

Hide

Alt: Password

Show

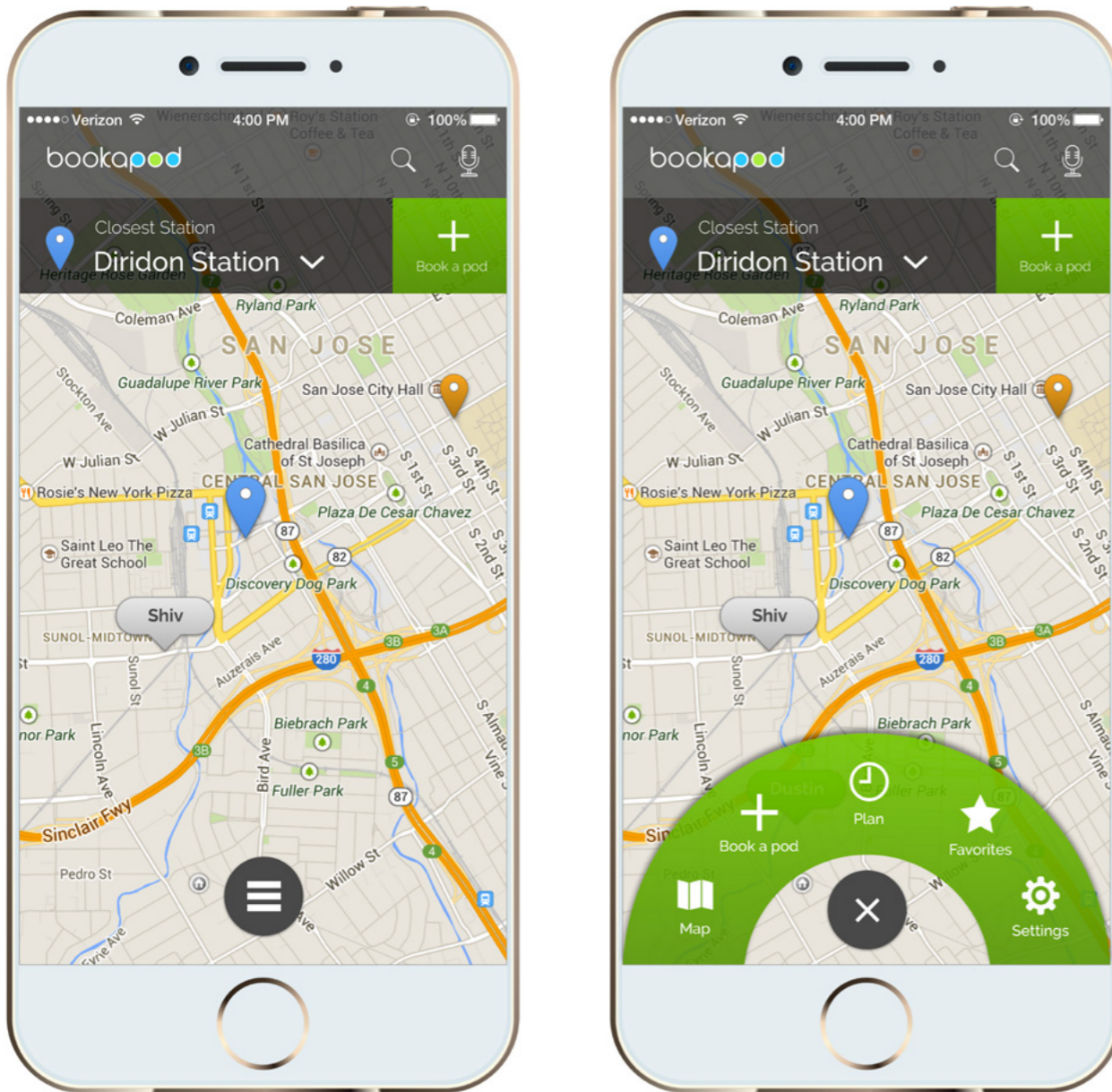
Confirm Password

By creating an account, you accept our [terms of use](#) and [privacy policy](#).

[CREATE ACCOUNT]

5.3. Main Map Screen (Primary Screen) & Navigation Menu

Mockups



Content

Closest Station
{Station Name}

Book a pod

Navigation Menu (Left to Right):

Map

Book A Pod

Plan

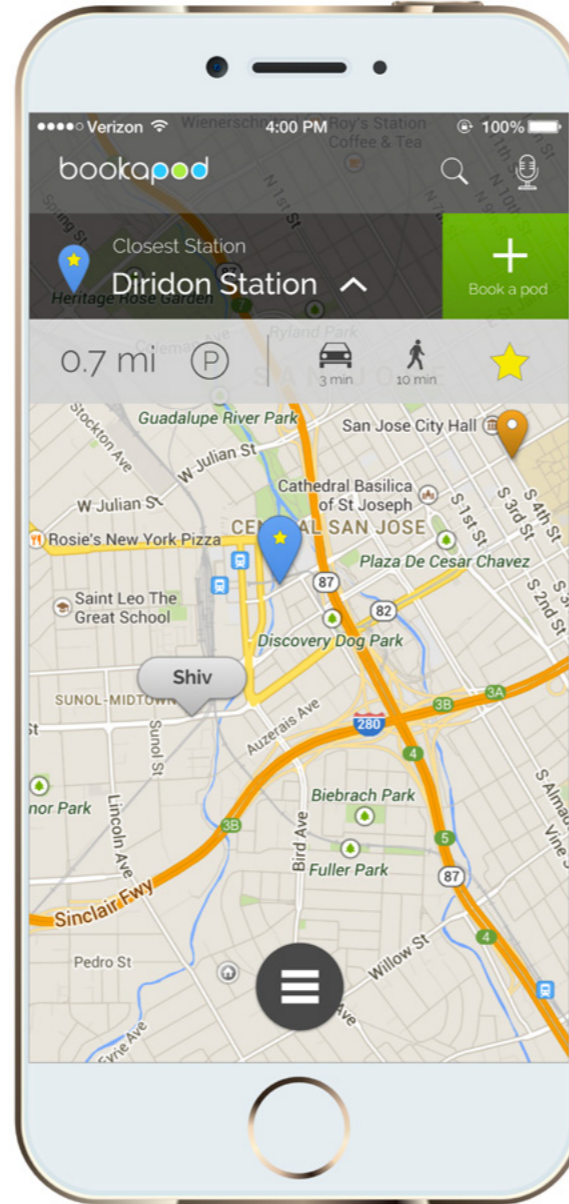
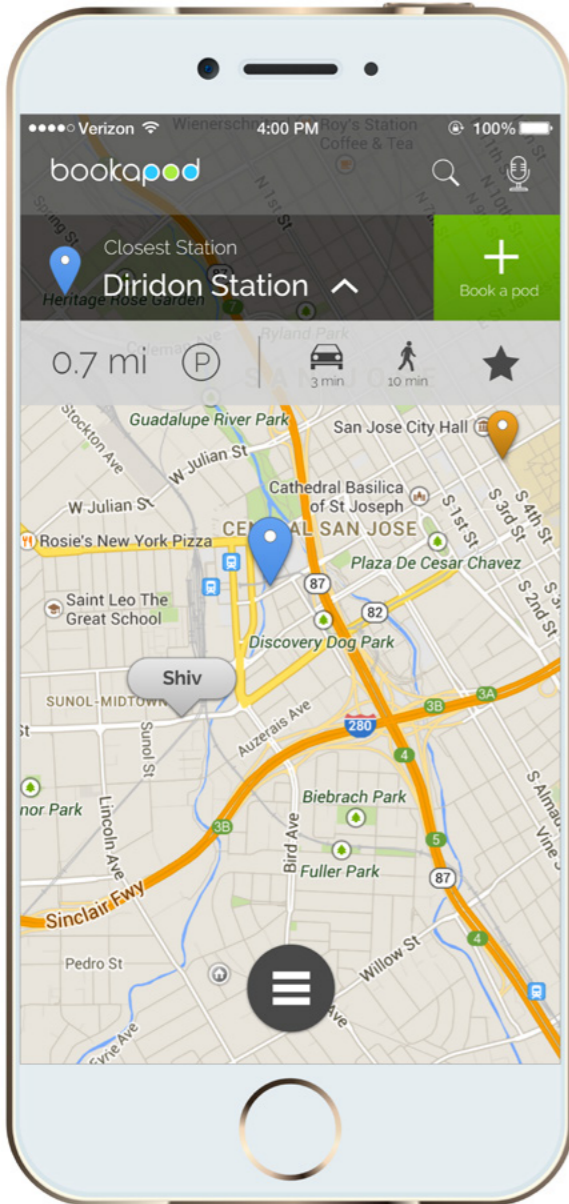
Favorites

Settings

5.4. Closest Station Options

Mockups

Content



Closest Station

{Station Name}

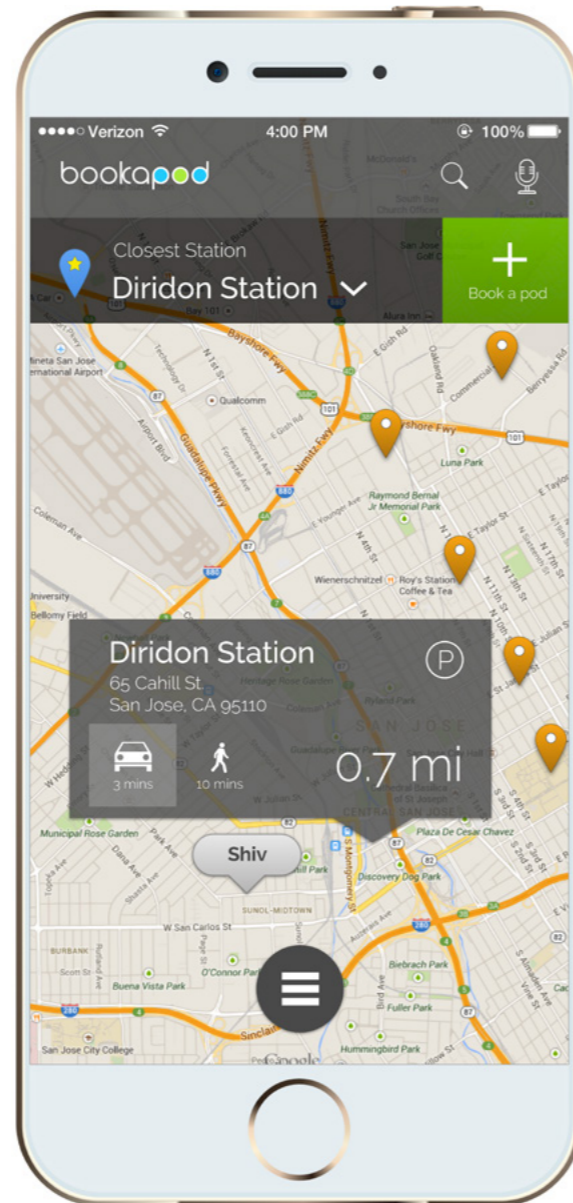
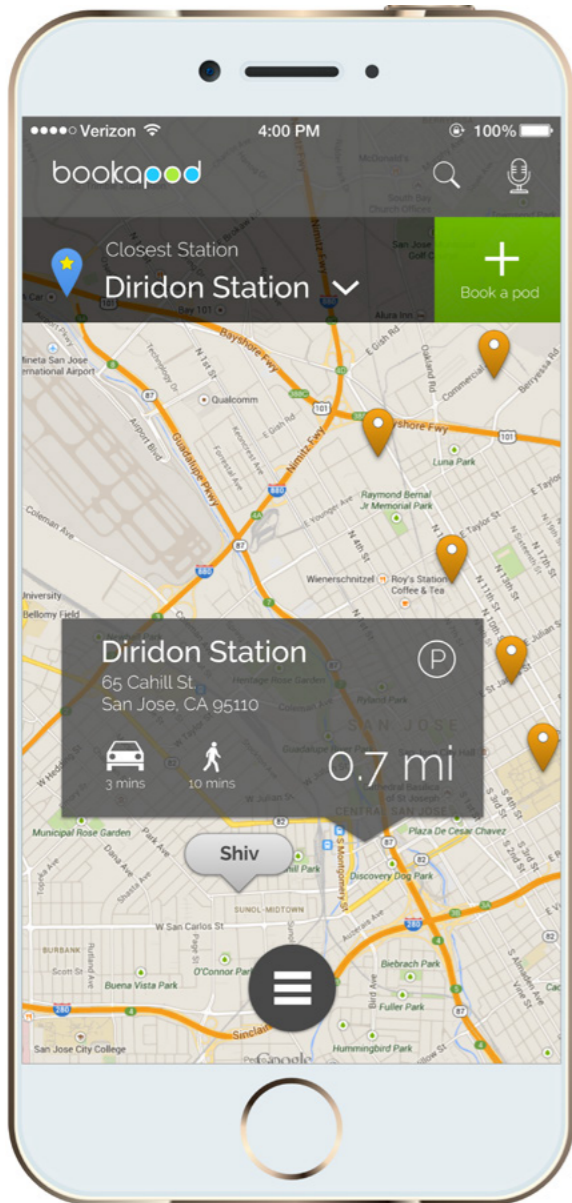
Book a pod

{Distance} {Time Driving} {Time Walking}

5.5. Closest Station Map Popup

Mockups

Content



Closest Station

{Station Name}

Book a pod

Pop Up:

Station Name

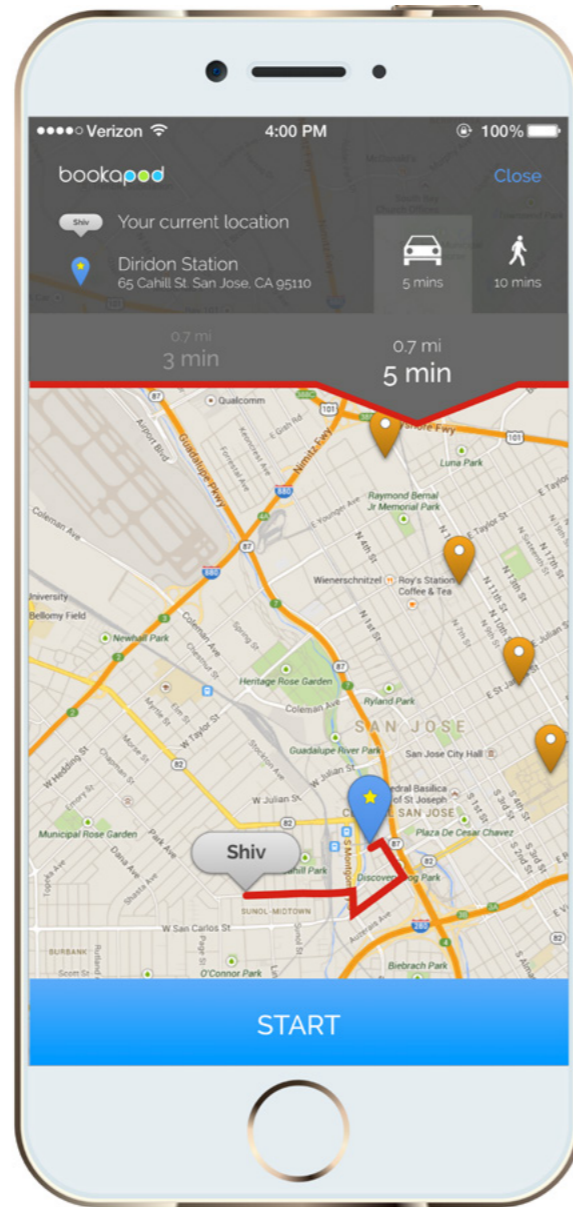
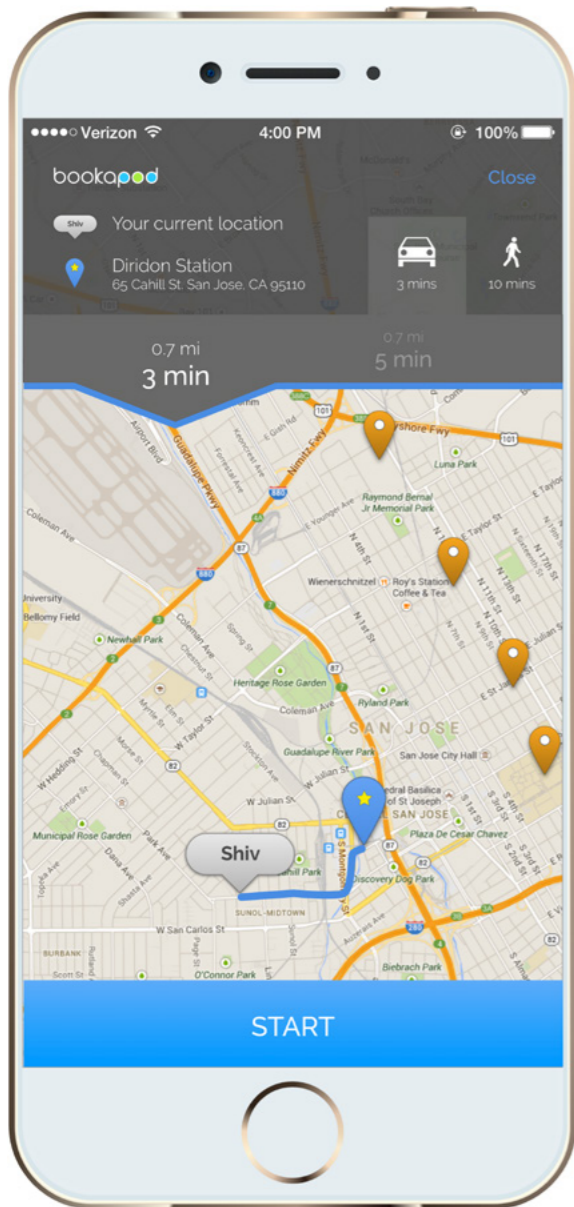
Station Address

{Time Driving} {Time Walking} {Distance}

5.6. Turn-by-turn navigation options

Mockups

Content



{User's Current Location}
{Destination Name} {Time Driving} {Time Walking}
{Destination Address}

Route Option 1:

{Distance}
{Time}

Route Option 2:

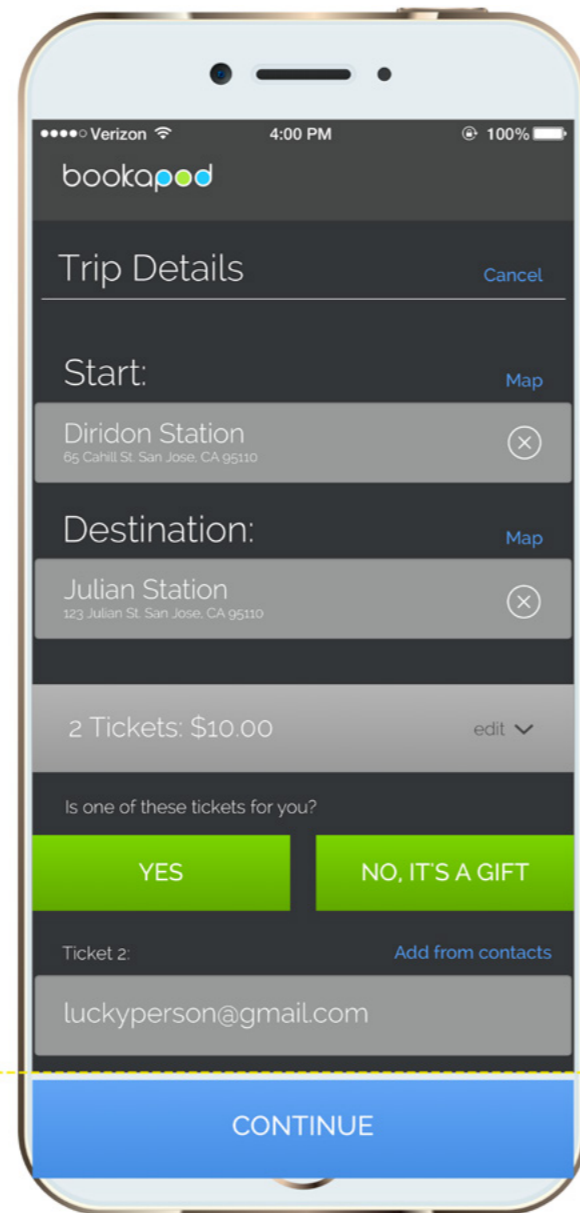
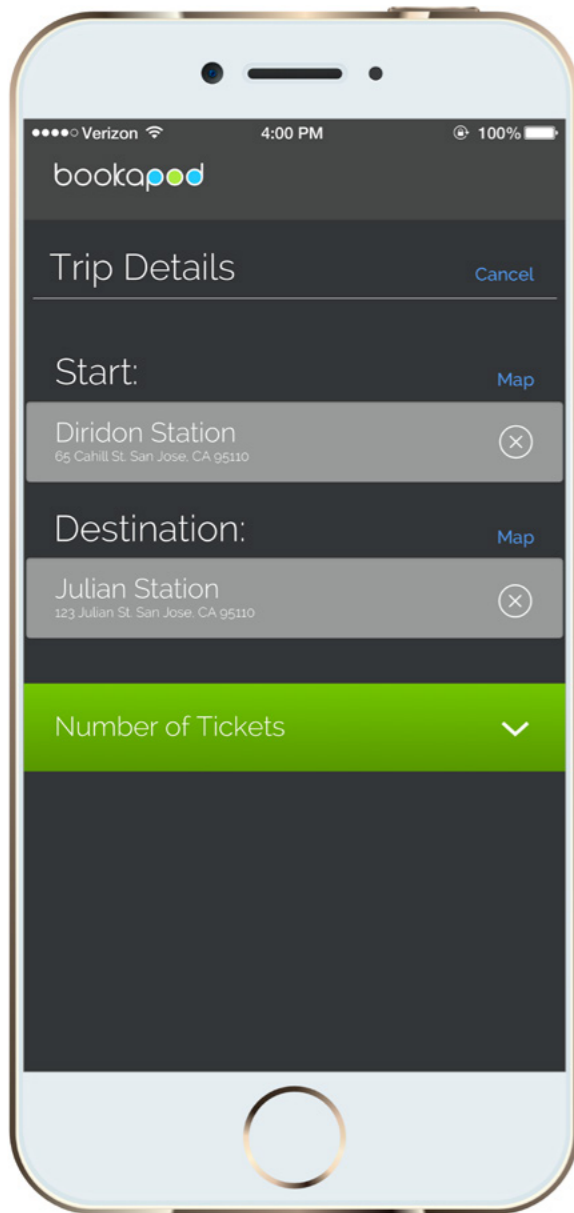
{Distance}
{Time}

[START]

5.7. Reservation Screens

Mockups

Content



Trip Details [Cancel]

Start: [Map]
{Start Location}

Destination: [Map]
{Destination Location}

Selector:
{Ticket Qty and Price} [Edit]

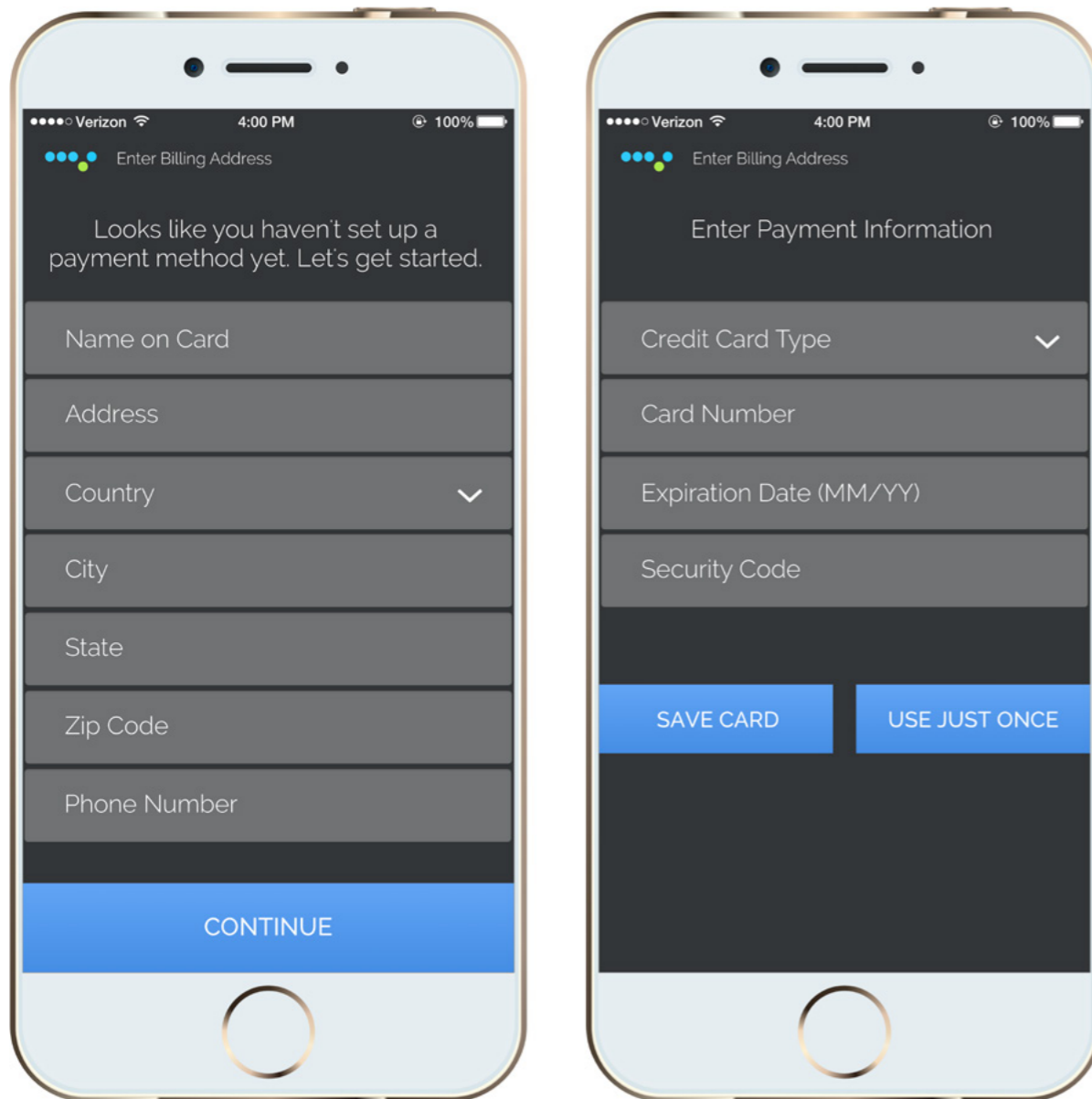
Is one of these tickets for you?
[YES] [NO, IT'S A GIFT]

Ticket 2: [Add from contacts]
{email address}

[CONTINUE]

5.8. Creating New Payment Type

Mockups



Content

User Credentials Screen:

/LOGO/ Enter Billing Address
Looks like you haven't set up a payment method yet.
Let's get started.

Name on Card
Address
Country (selector)
City
State
Zip Code
Phone Number

[CONTINUE]

Payment Information Screen:

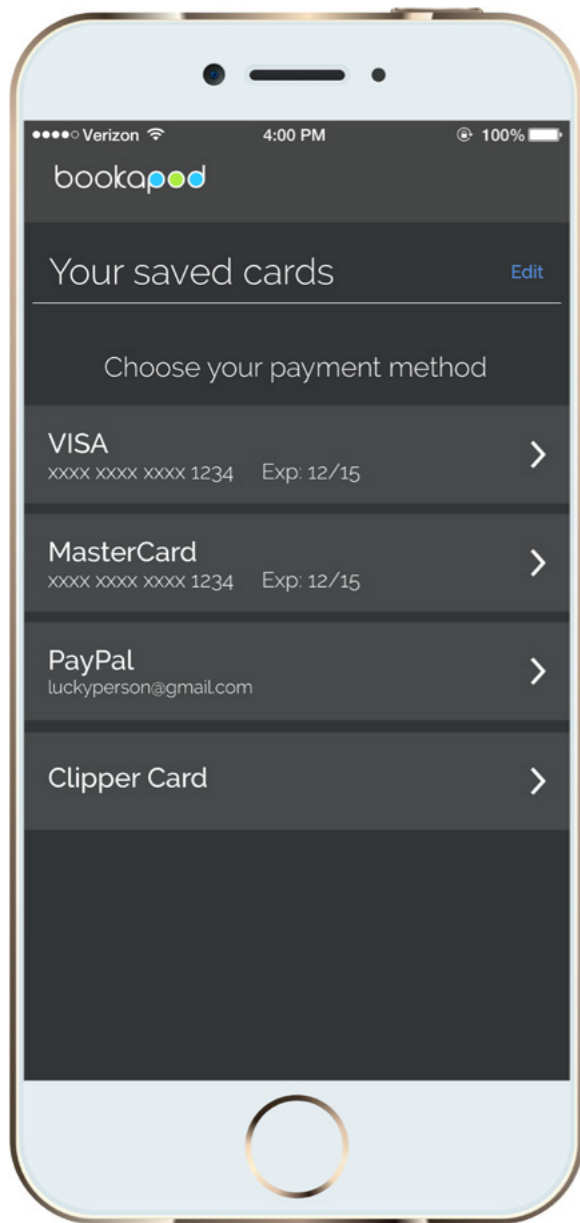
Enter Payment Information
Credit Card Type (selector)
Card Number
Expiration Date: (MM/YY)
Security Code

[SAVE CARD]

[USE JUST ONCE]

5.9. Saved Cards

Mockups



Content

Your Saved Cards [Edit]

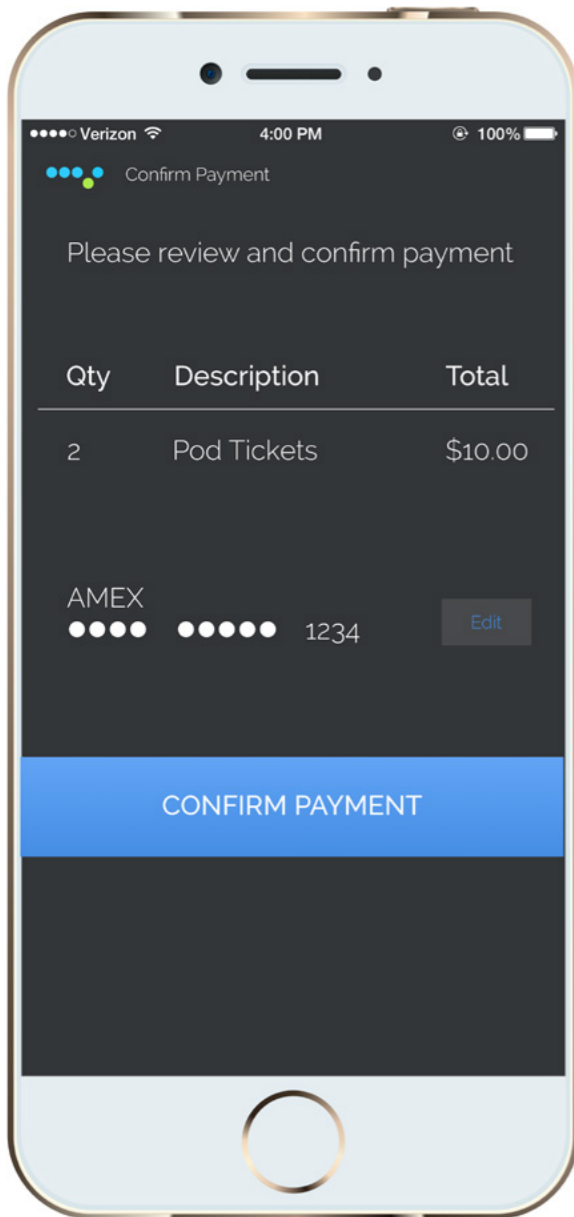
Choose your payment method

{Card Type}

{Card Number} {Expiration Date MM/YY}

5.10. Confirm Payment

Mockups



Content

/LOGO/ Confirm Payment

Please review and confirm payment

Qty	Description	Total
{#}	{Description}	{Total Cost}

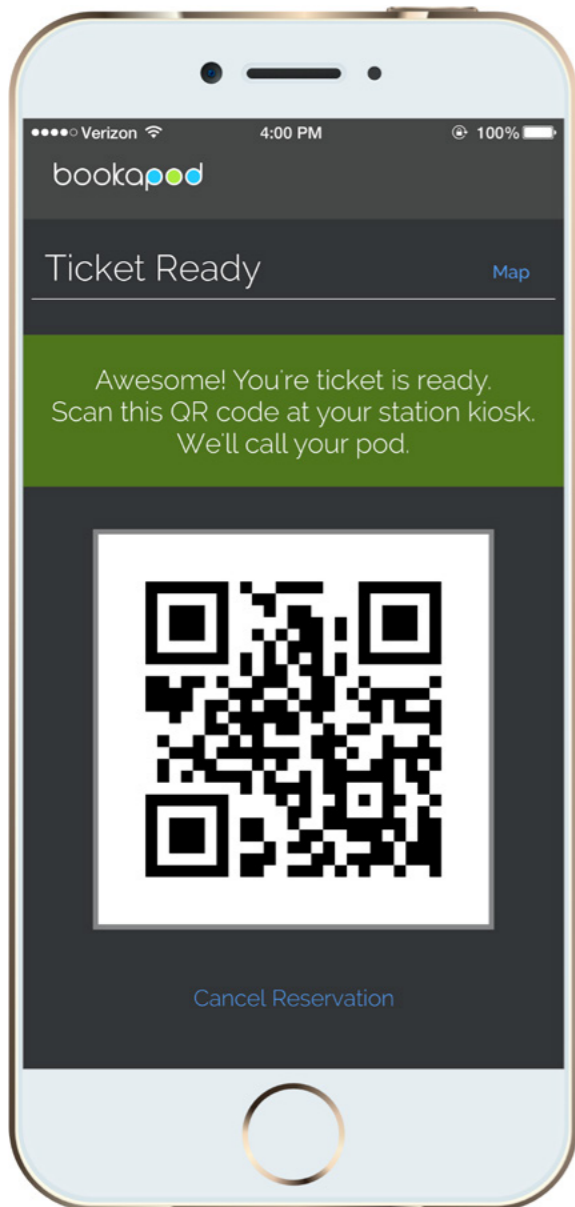
{Card Type}

{Card Number} (Masked) [Edit]

[CONFIRM PAYMENT]

5.11. QR Code

Mockups



Content

Ticket Ready

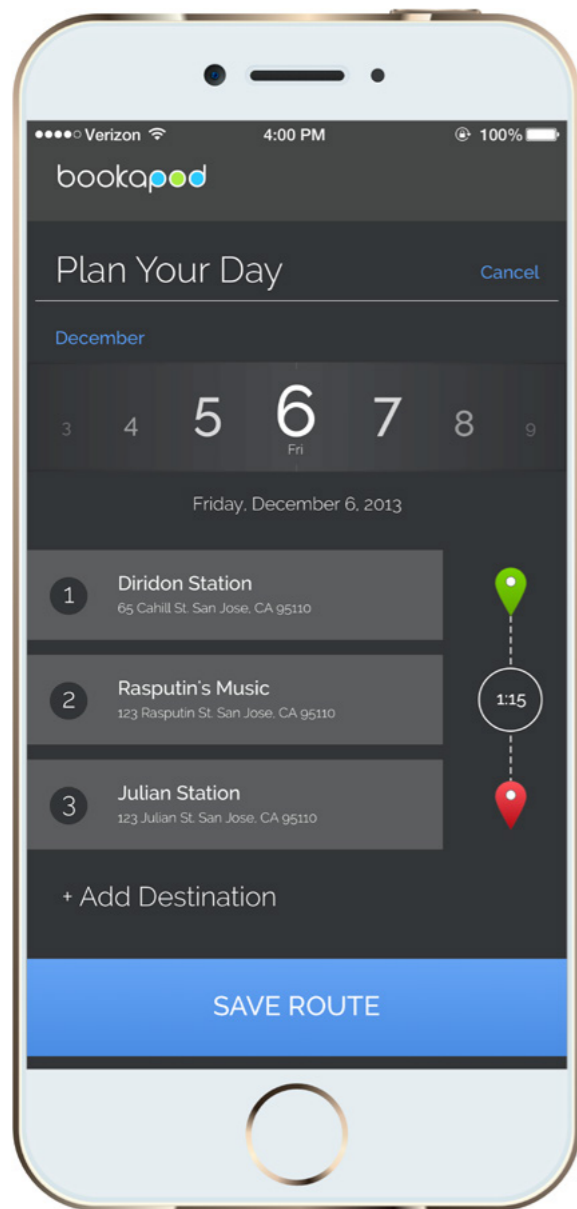
[Map]

{Notification Message}

[Cancel Reservation]

5.12. Plan Your Day

Mockups



Content

Plan Your Day [Cancel]

[{Month}]

{Days}

{Selected Date (Day of week, Month, Day, Year)}

{Starting Location & Address}

{First Destination Location & Address}

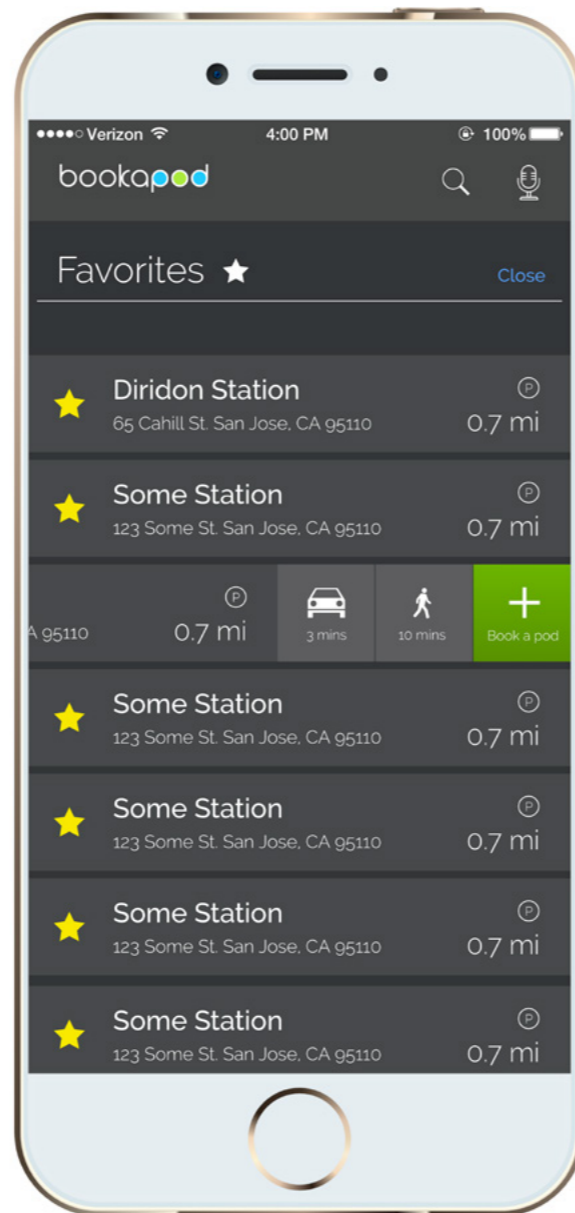
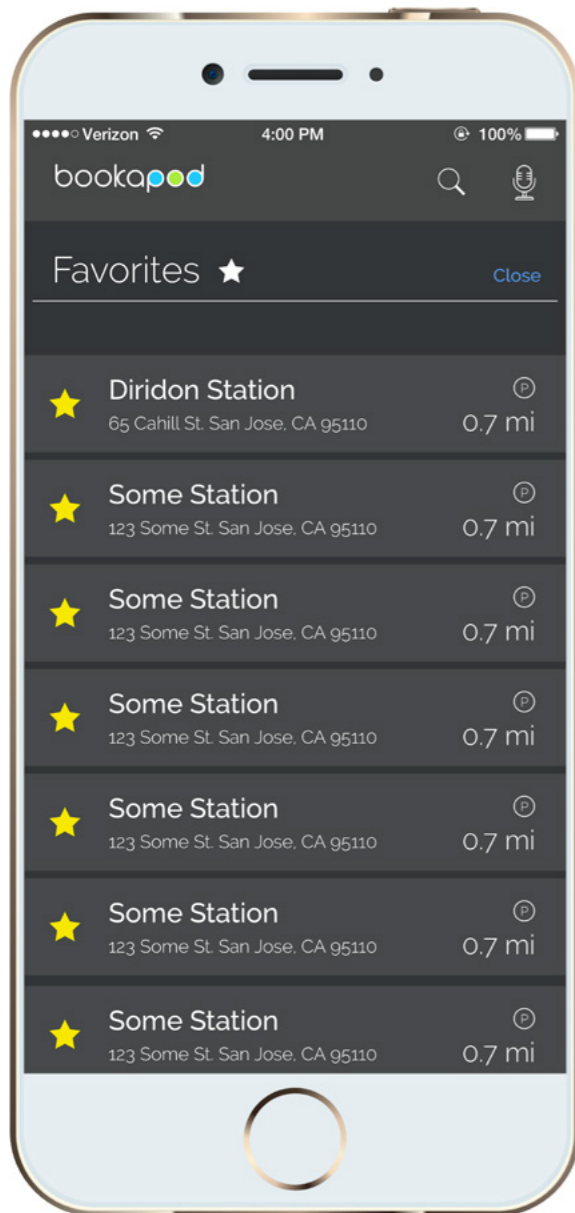
[+ Add Destination]

[SAVE ROUTE]

5.13. Favorites

Mockups

Content



Favorites

[Close]

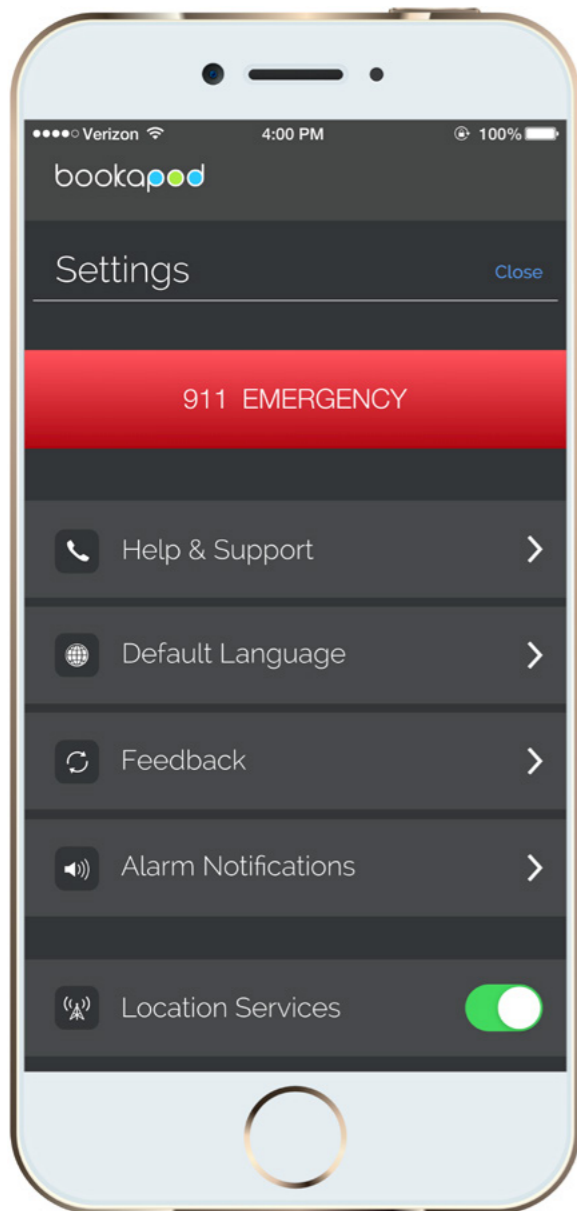
{Station Name}

{Station Address}

{Distance}

5.14. Settings

Mockups

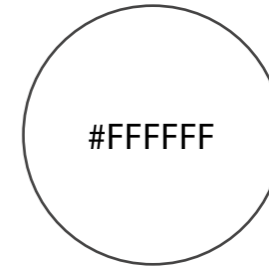


Content

- Settings [Close]
- [911 Emergency]
- Help & Support
- Default Language
- Feedback
- Alarm Notification
- Location Services
- Calendar Sync
- Units

6. Style Guide

6.1. Colors



6.2. Fonts

H1

Font Family: Raleway

Font Size: 24px

Font Weight: Regular

Color: #FFFFFF

H2

Font Family: Raleway

Font Size: 18px

Font Weight: Regular

Color: #FFFFFF

BODY

Font Family: Raleway

Font Size: 18px

Font Weight: Regular

Color: #FFFFFF

BODY (*Disclaimer*)

Font Family: Raleway

Font Size: 12px

Font Weight: Regular

Color: #589DED

FORMS (*Placeholder*)

Font Family: Raleway

Font Size: 16px

Font Weight: Thin

Color: #FFFFFF

BUTTONS

Font Family: Raleway

Font Size: 16px / Uppercase

Font Weight: Regular

Color: #FFFFFF

6.3. Buttons



PRIMARY

Gradient Color: #68A7EE (top)
#4F92F3 (bottom)

Border Radius: 0

Width: 640px (retina)

Height: 130px (retina)



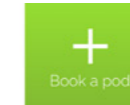
EMERGENCY

Gradient Color: #FE525D (top)
#B00B1A (bottom)

Border Radius: 0

Width: 640px (retina)

Height: 130px (retina)



BOOK A POD

Gradient Color: #73B332 (top)
#71B527 (bottom)

Border Radius: 0

Width: 130px (retina)

Height: 130px (retina)

6.4. Screen Dimensions

