

# GoParking



Jia Yao

BU MET CS683  
Fall 2016

# GOALS



## GoParking

GoParking is essentially a communication platform for connecting people who want to rent extra parking spaces and people who want to reserve parking spots in advance.

**My original goals were the following features:**

**=From Guest Perspective=**

1. search by location and time and see parking space listings that are near that location and available during that time. Display results on a map or as a list with different sorting options (by distance, by price, etc)
2. make a reservation request for a listing
3. bookmark a listing for easy access later
4. see which requests have been approved
5. leave star ratings and comments
6. “watch” a location and receive notifications when new listings open up in that location

**=From Owner Perspective=**

1. create a new listing of parking space
2. modify or delete a listing created by him/her
3. see users who are interested via notification of requests
4. approve or reject requests

## Accomplishments

**Due to time constraints I could not implemented all of the features I planned: \*Unimplemented shown in pink**

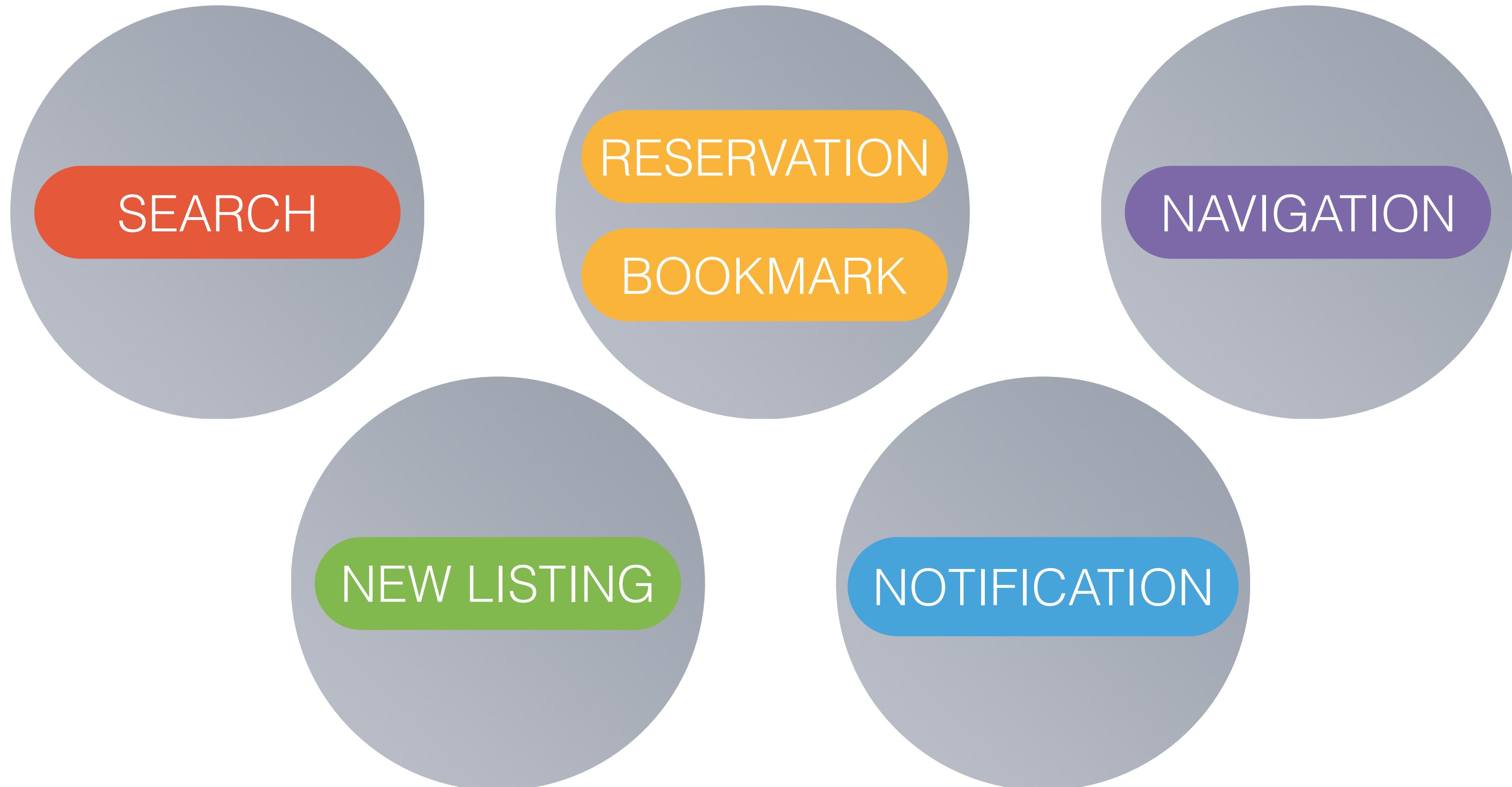
**=From Guest Perspective=**

1. search by location and time and see parking space listings that are near that location and available during that time. Display results on a map or as a list with different sorting options (by distance, by price, etc)
2. make a reservation request for a listing
3. bookmark a listing for easy access later
4. see which requests have been approved
5. leave star ratings and comments
6. “watch” a location and receive notifications when new listings open up in that location

**=From Owner Perspective=**

1. create a new listing of parking space
2. modify or delete a listing created by him/her
3. see users who are interested via notification of requests
4. approve or reject requests

# USE CASES



# SCREEN FLOW



## AUTHENTICATION

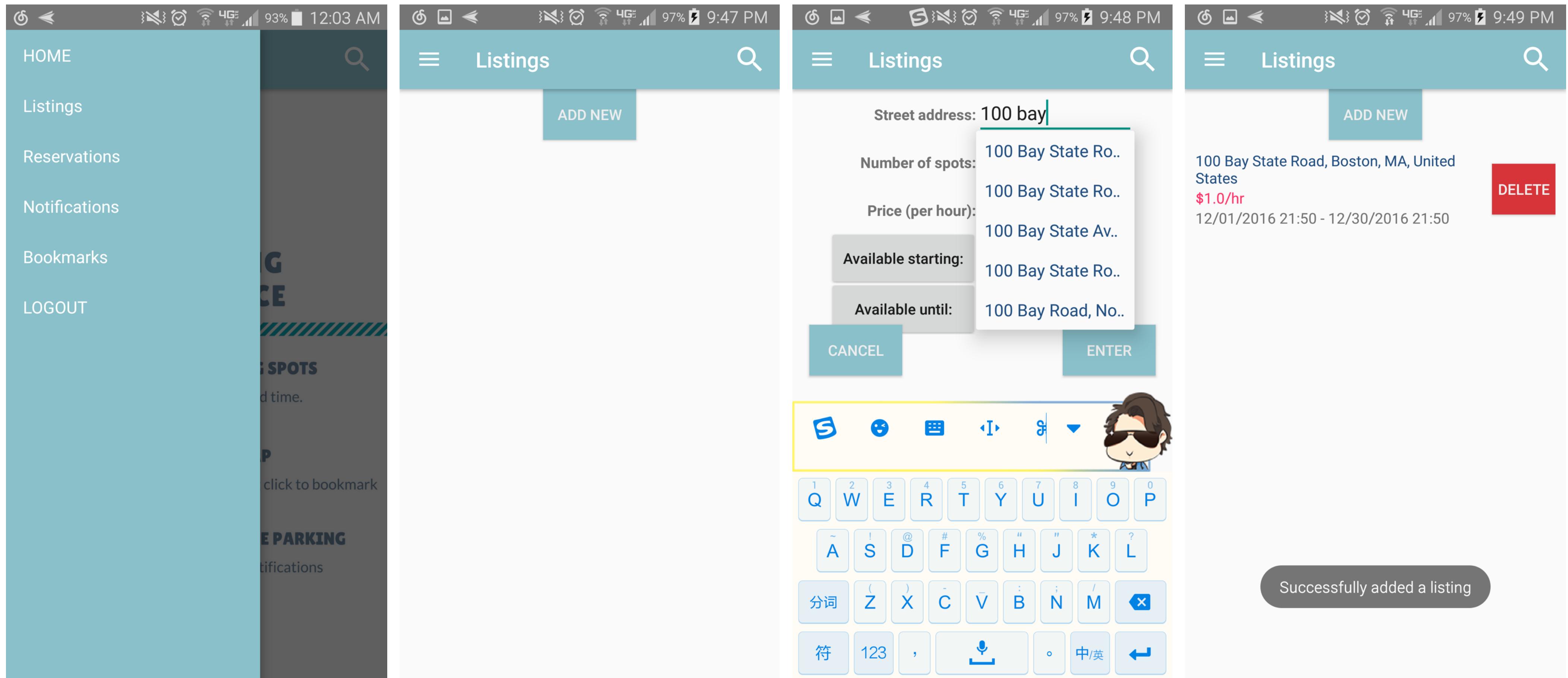
The figure displays four screenshots of the GoParking mobile application interface, showing the authentication flow:

- Welcome Screen:** Shows the text "Welcome to GoParking" at the top. At the bottom are two buttons: a teal "LOG IN" button on the left and a teal "CREATE ACCOUNT" button on the right.
- Create Account Screen:** Shows the title "Create account" with a back arrow. It contains three input fields: "User name:", "Email:", and "Password:". Below the password field is a teal "ENTER" button.
- Log In Screen:** Shows the title "Log in" with a back arrow. It contains two input fields: "Email:" and "Password:". To the right of the password field is a teal "ENTER" button.
- Profile Screen:** Shows the title "GoParking" with a search icon. It displays a user profile icon and the username "user1". Below this is a section titled "GOPARKING AT A GLANCE" with three items:
  - SEARCH FOR PARKING SPOTS:** Accompanied by a magnifying glass icon. Description: "at your chosen location and time."
  - MARKER INFO ON MAP:** Accompanied by a hand icon pointing at a marker. Description: "short click to reserve, long click to bookmark"
  - RENT OUT YOUR SPARE PARKING:** Accompanied by a clipboard icon. Description: "see who's interested in notifications"

# SCREEN FLOW

— — — — —

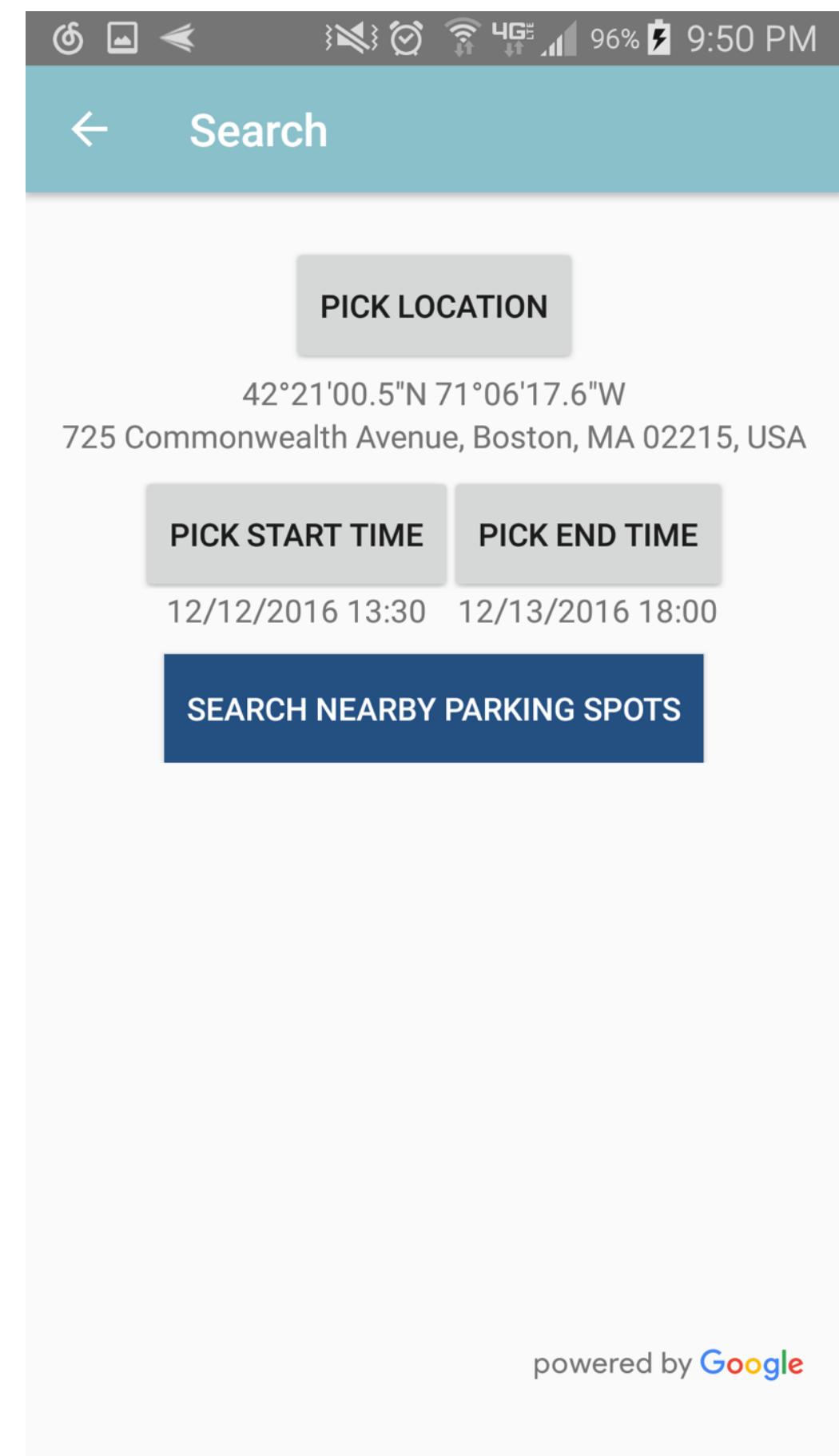
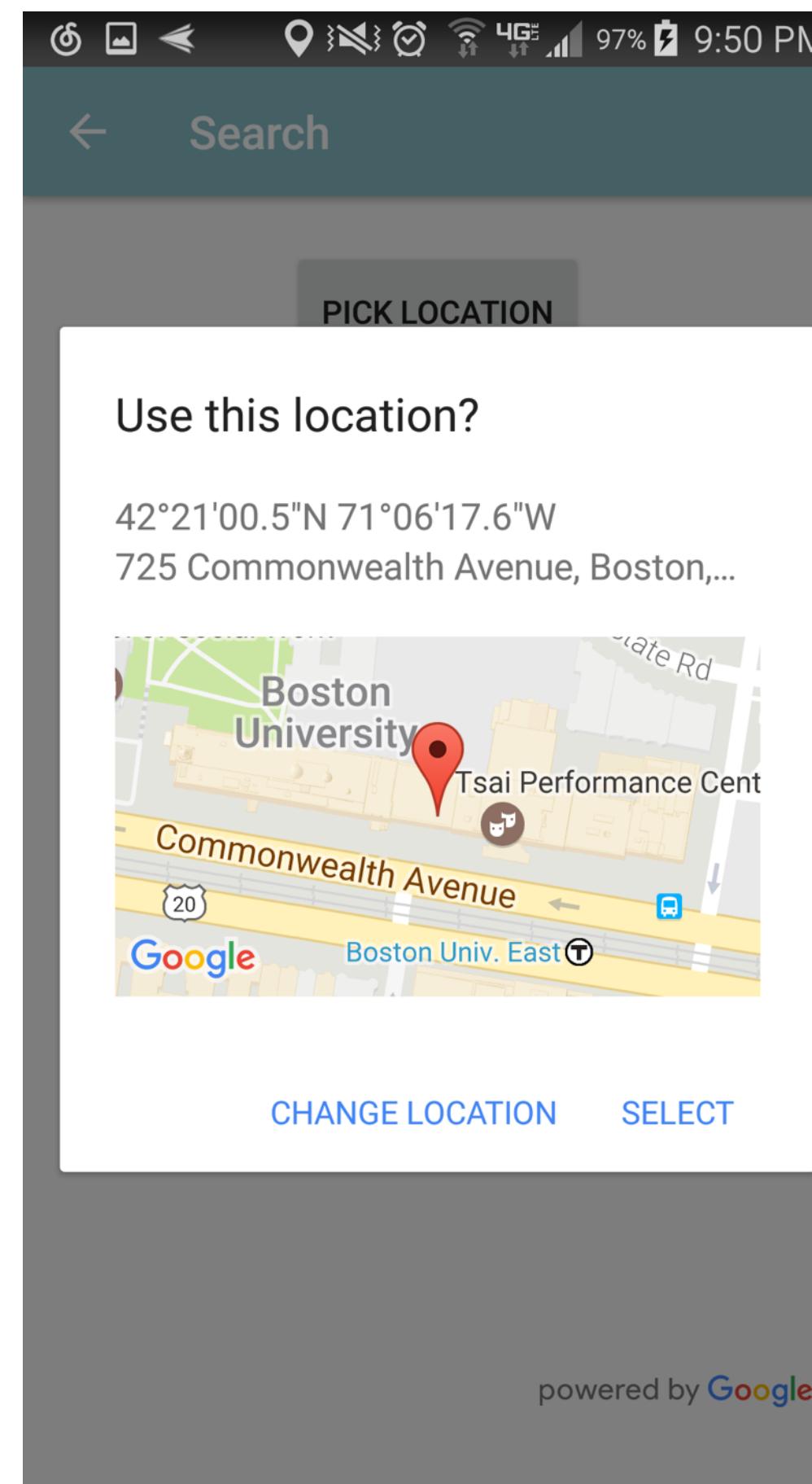
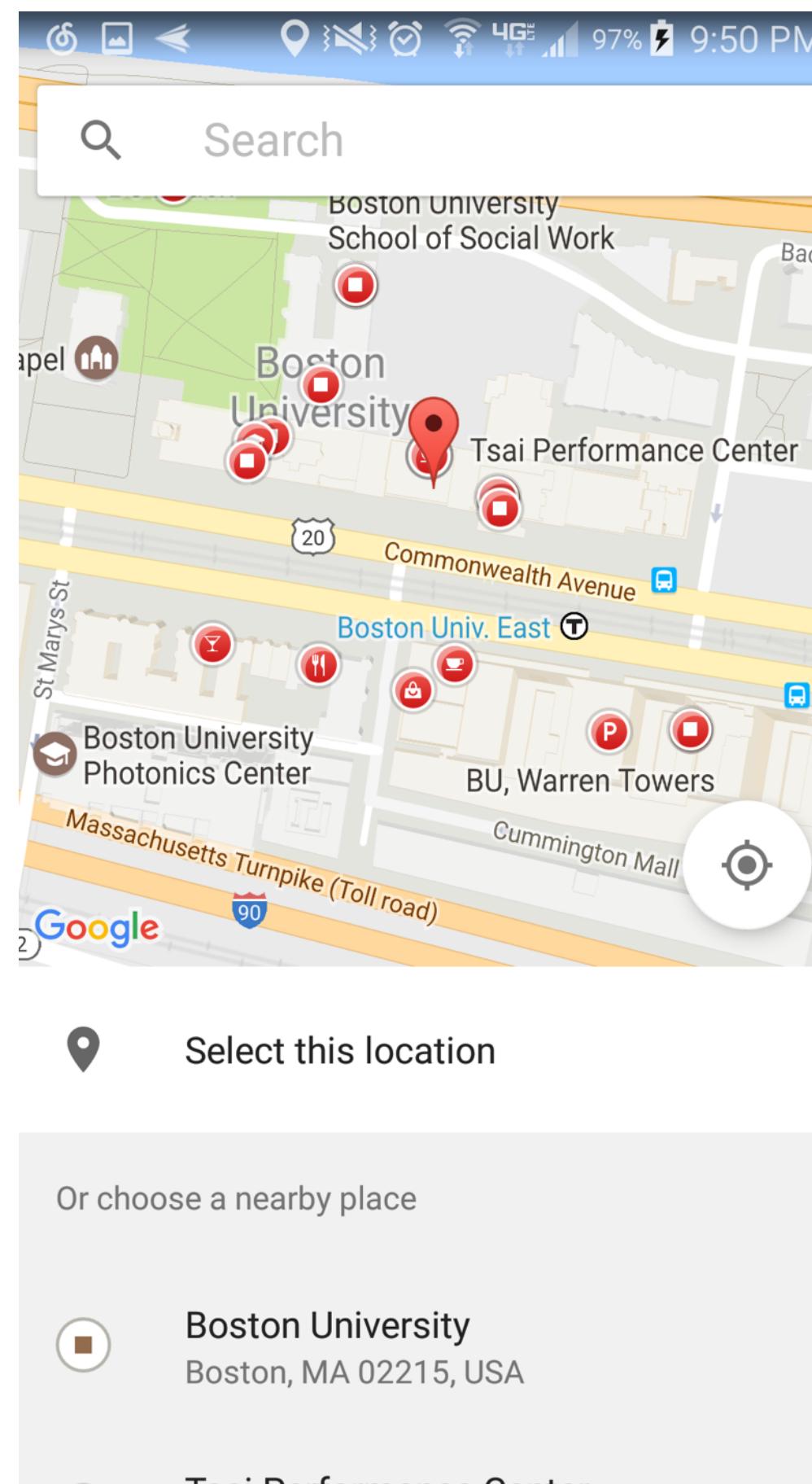
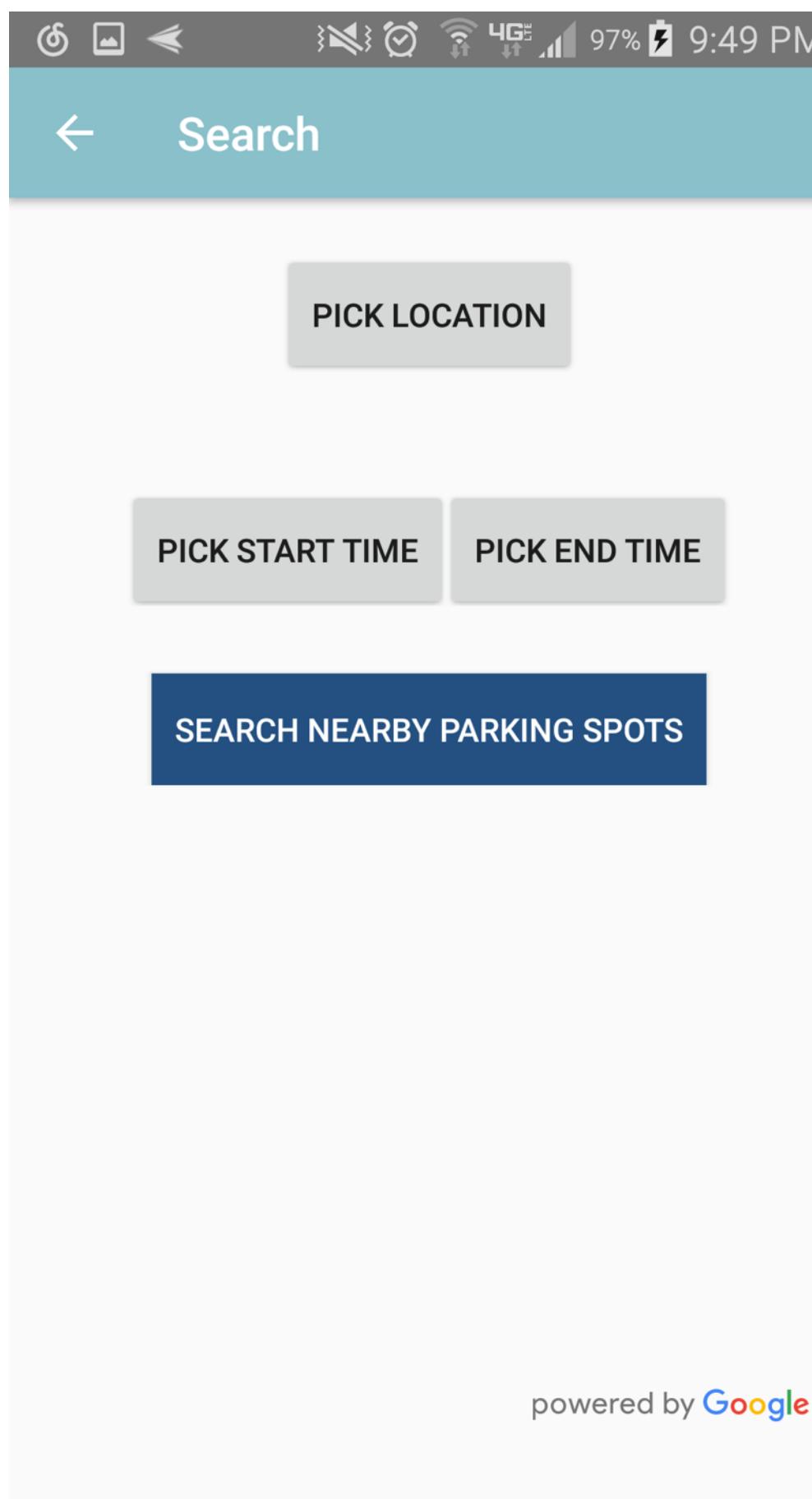
## NEW LISTING



# SCREEN FLOW

— — — — —

SEARCH

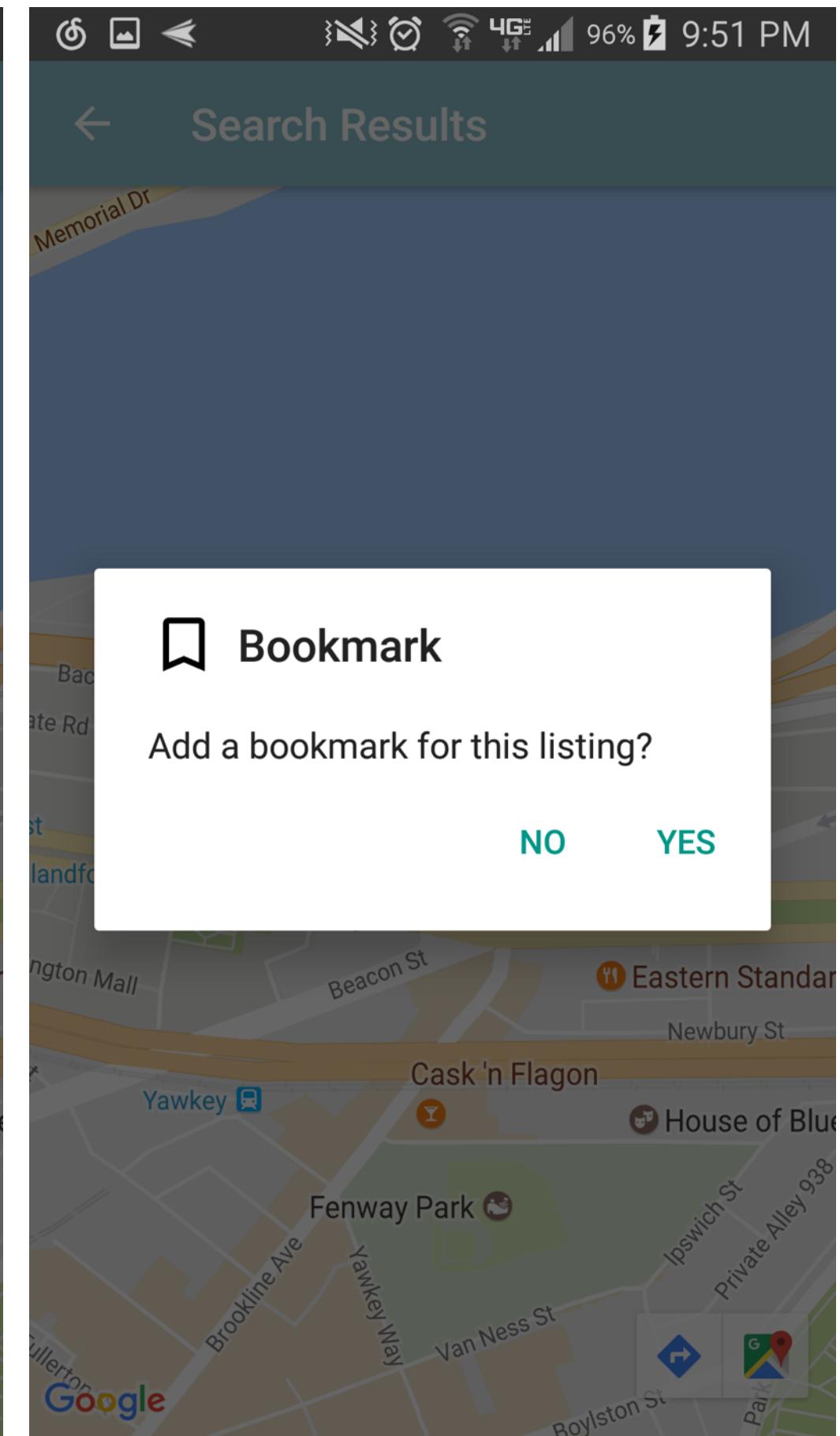
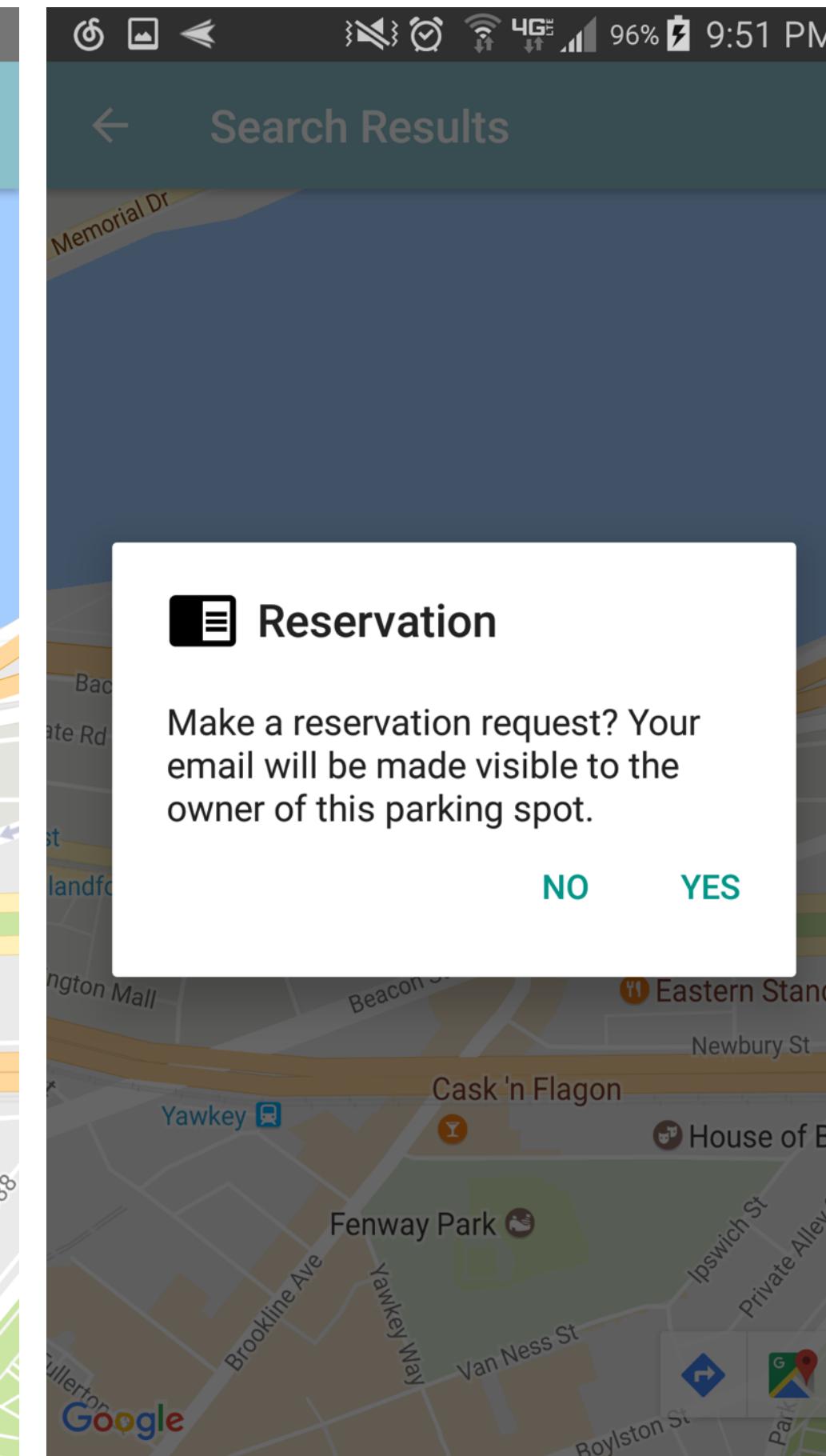
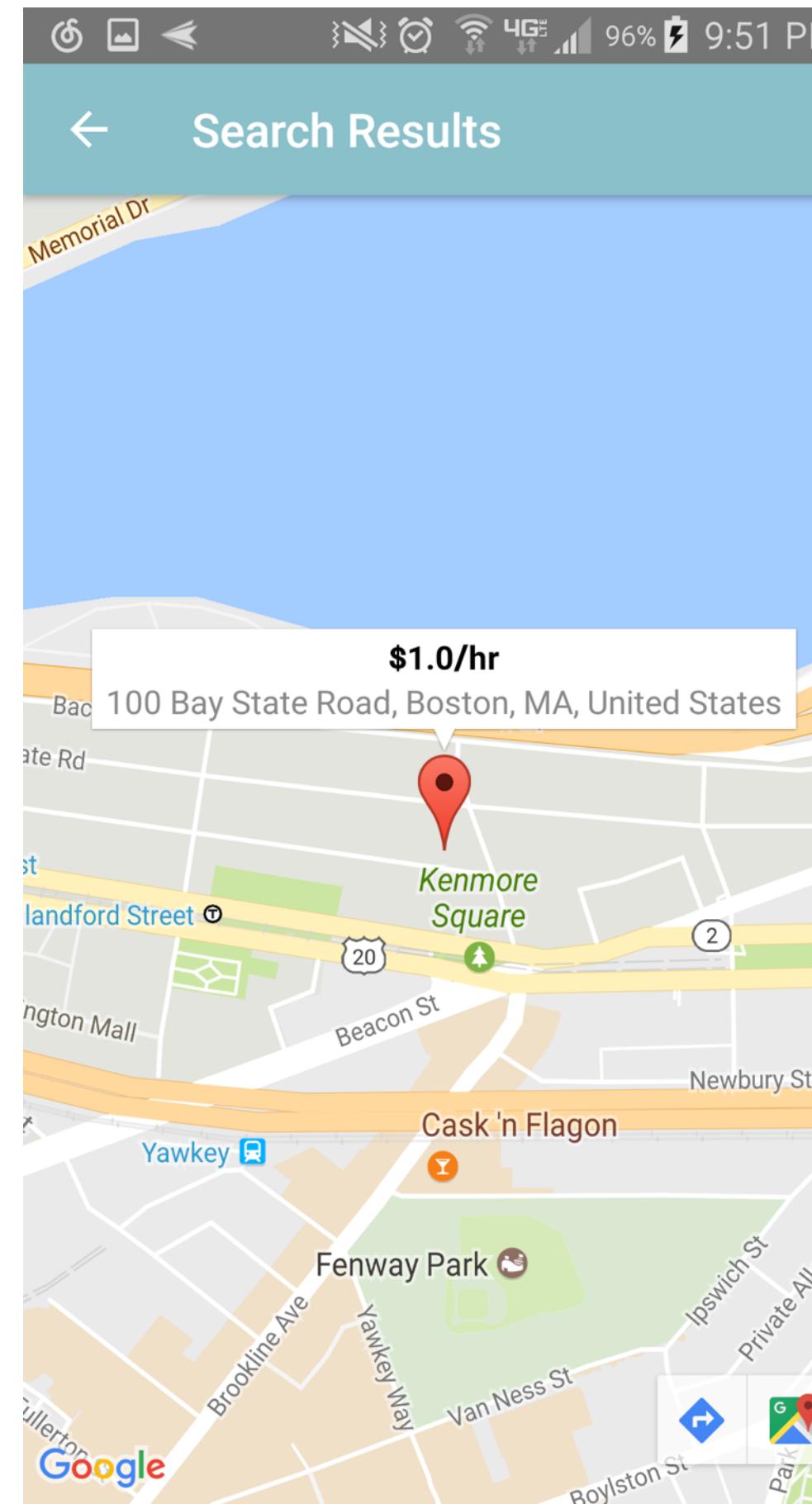
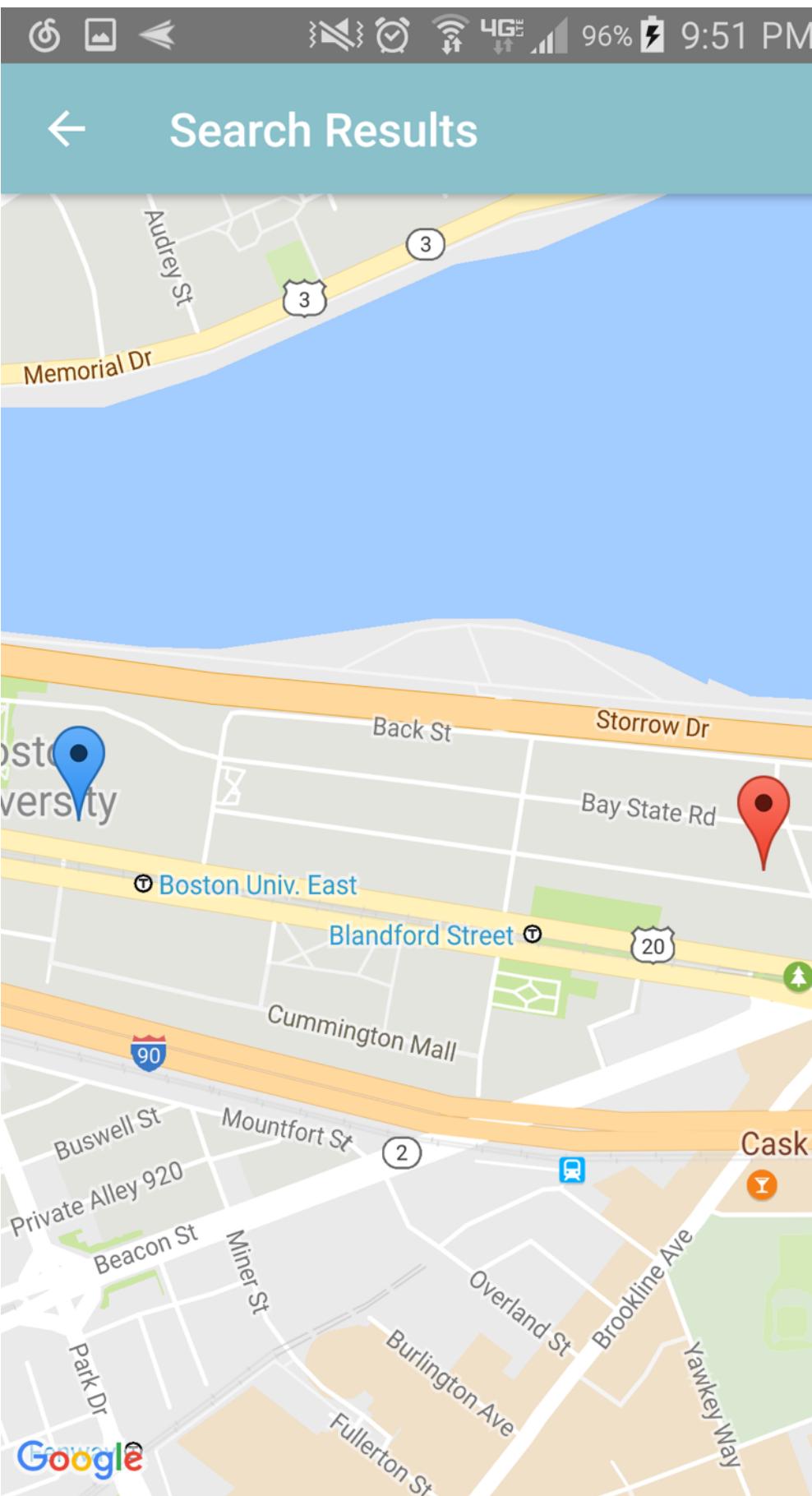


# SCREEN FLOW



RESERVATION

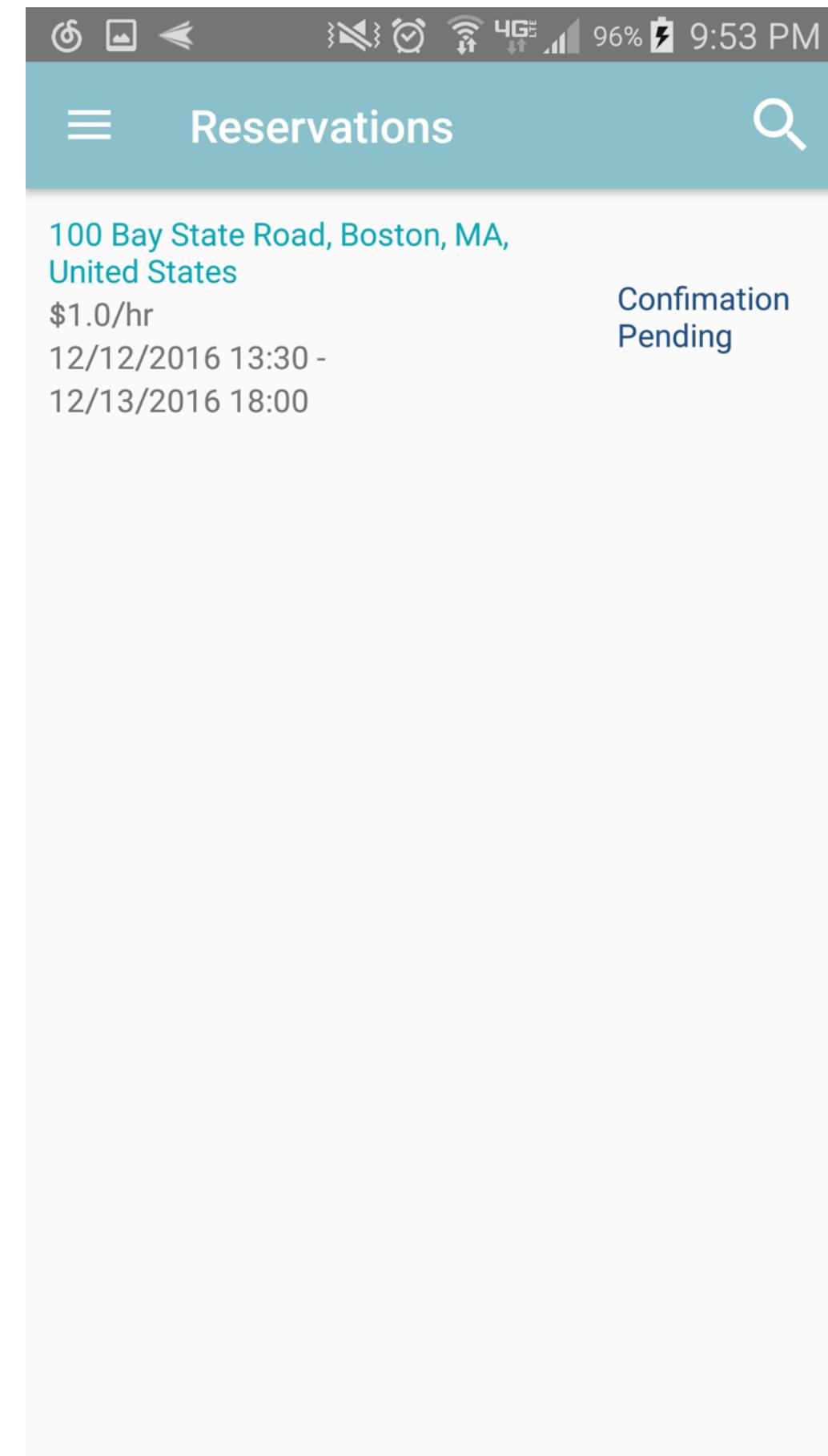
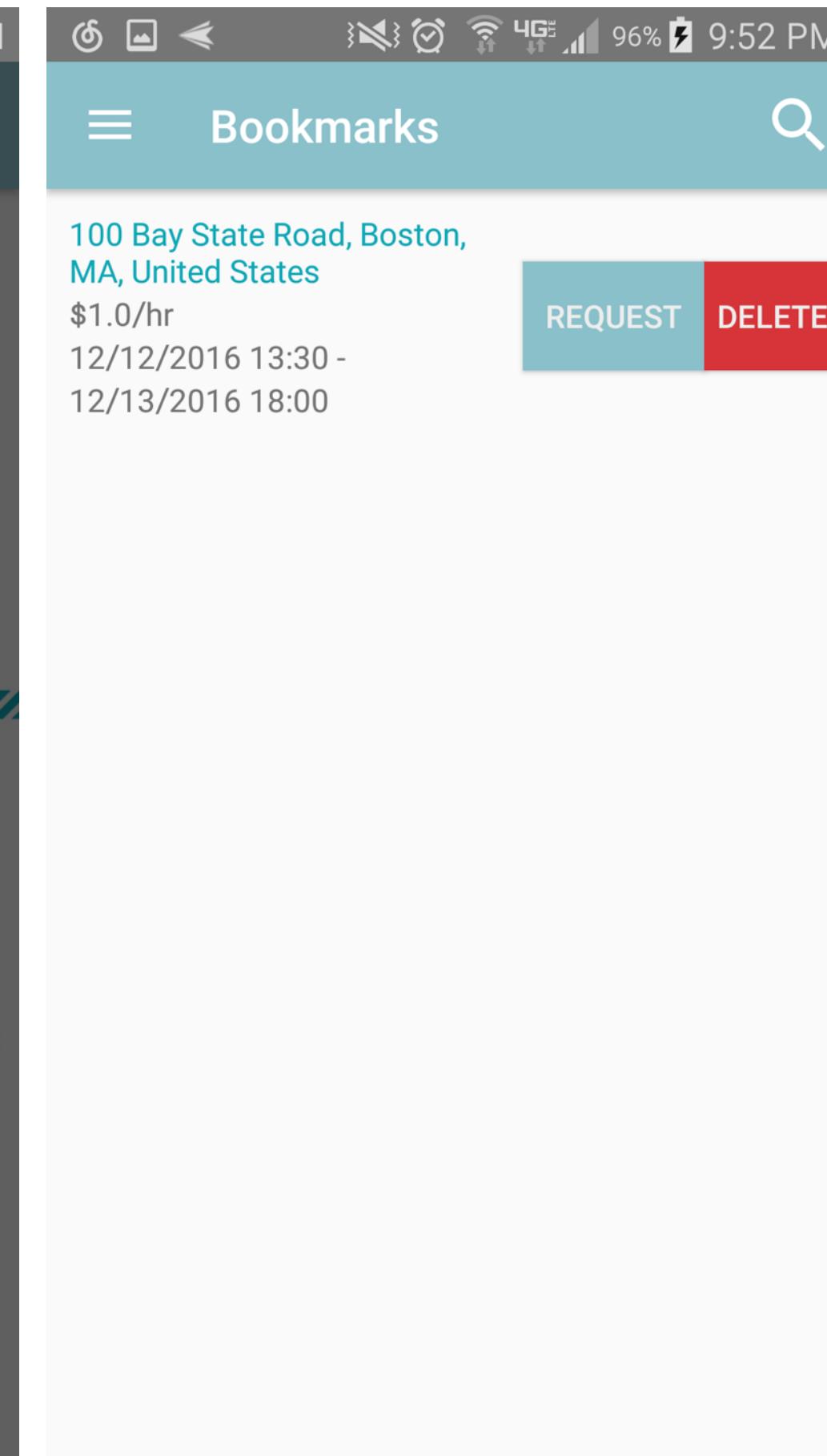
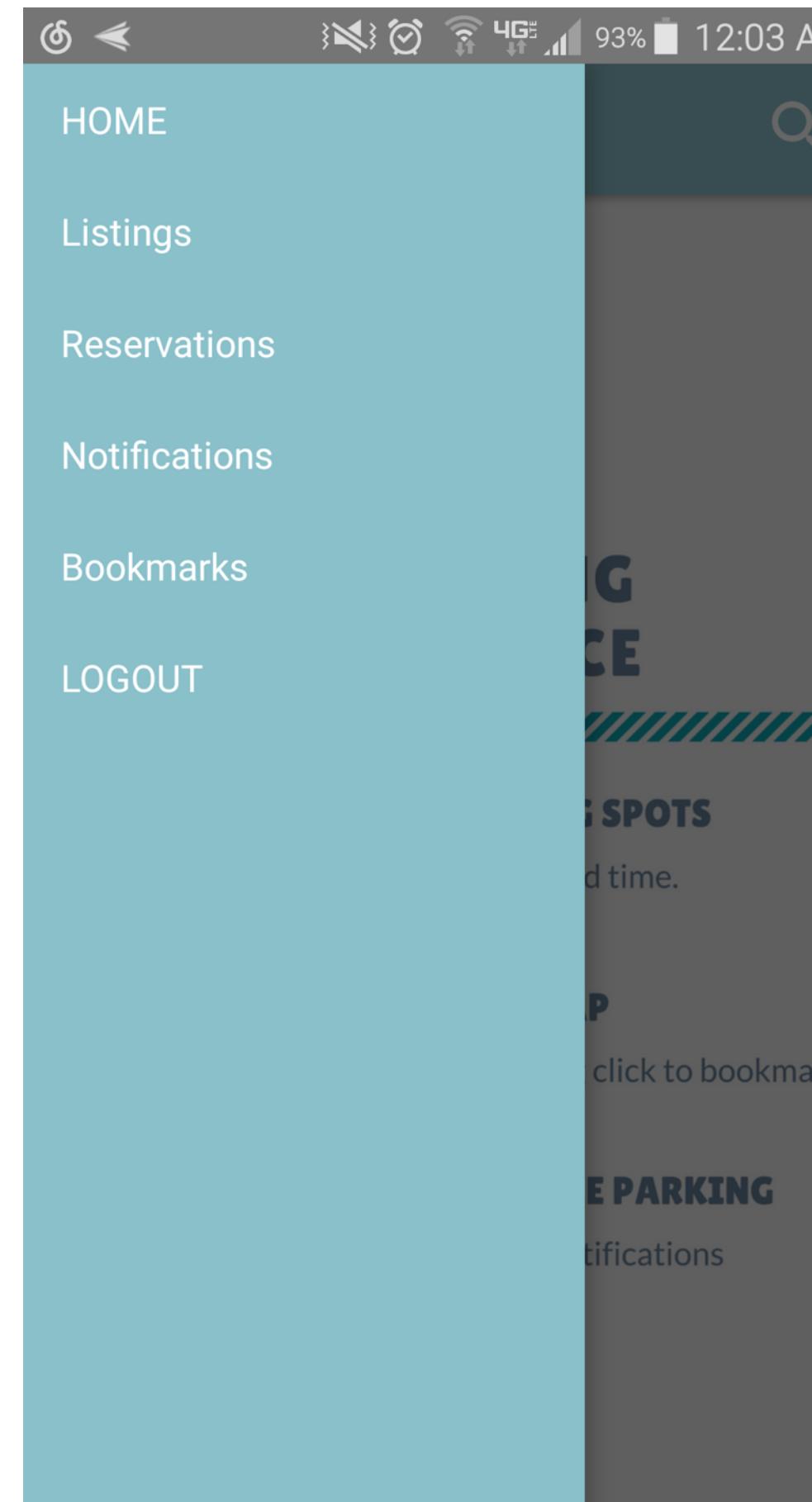
BOOKMARK



# SCREEN FLOW

RESERVATION

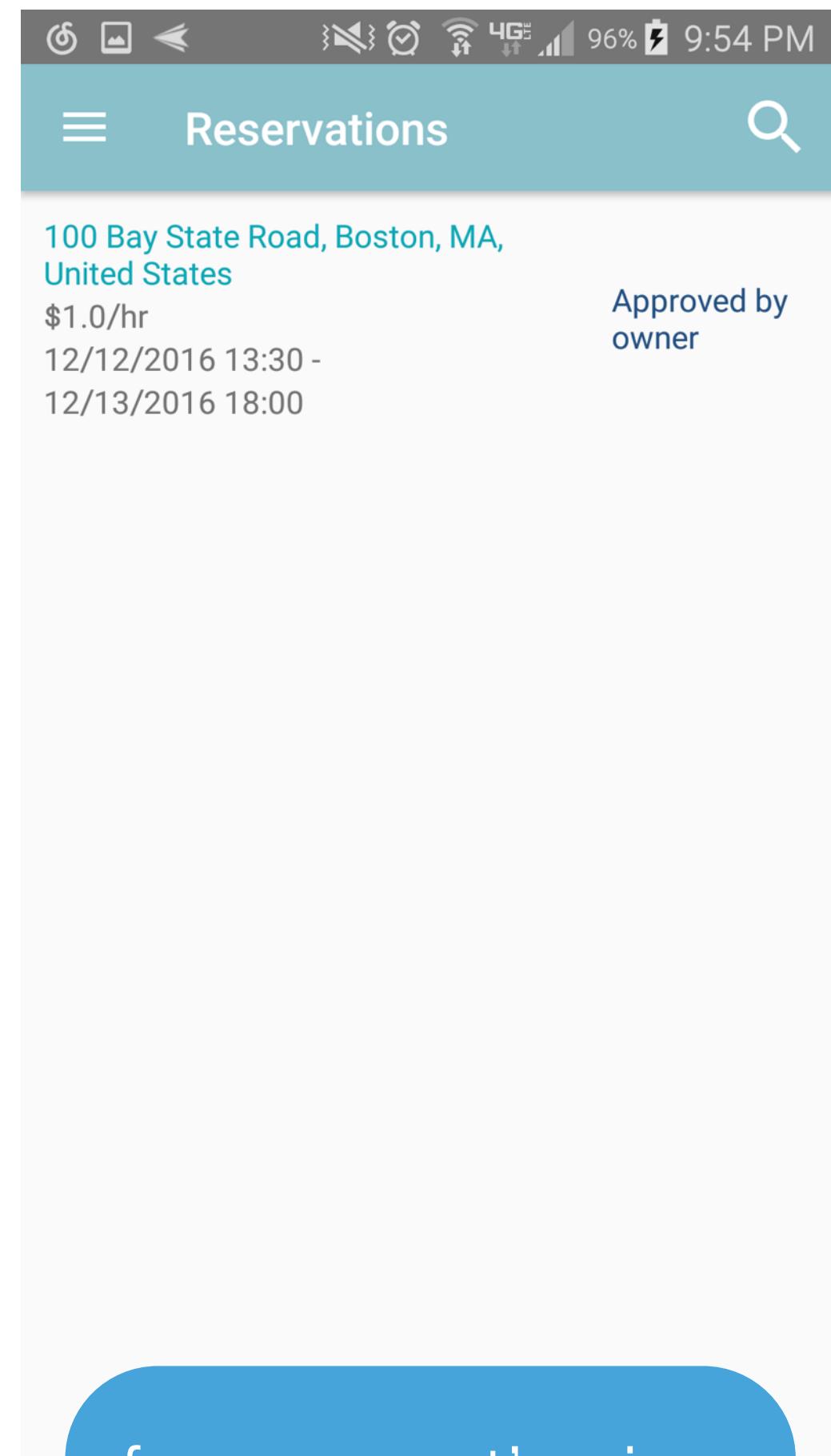
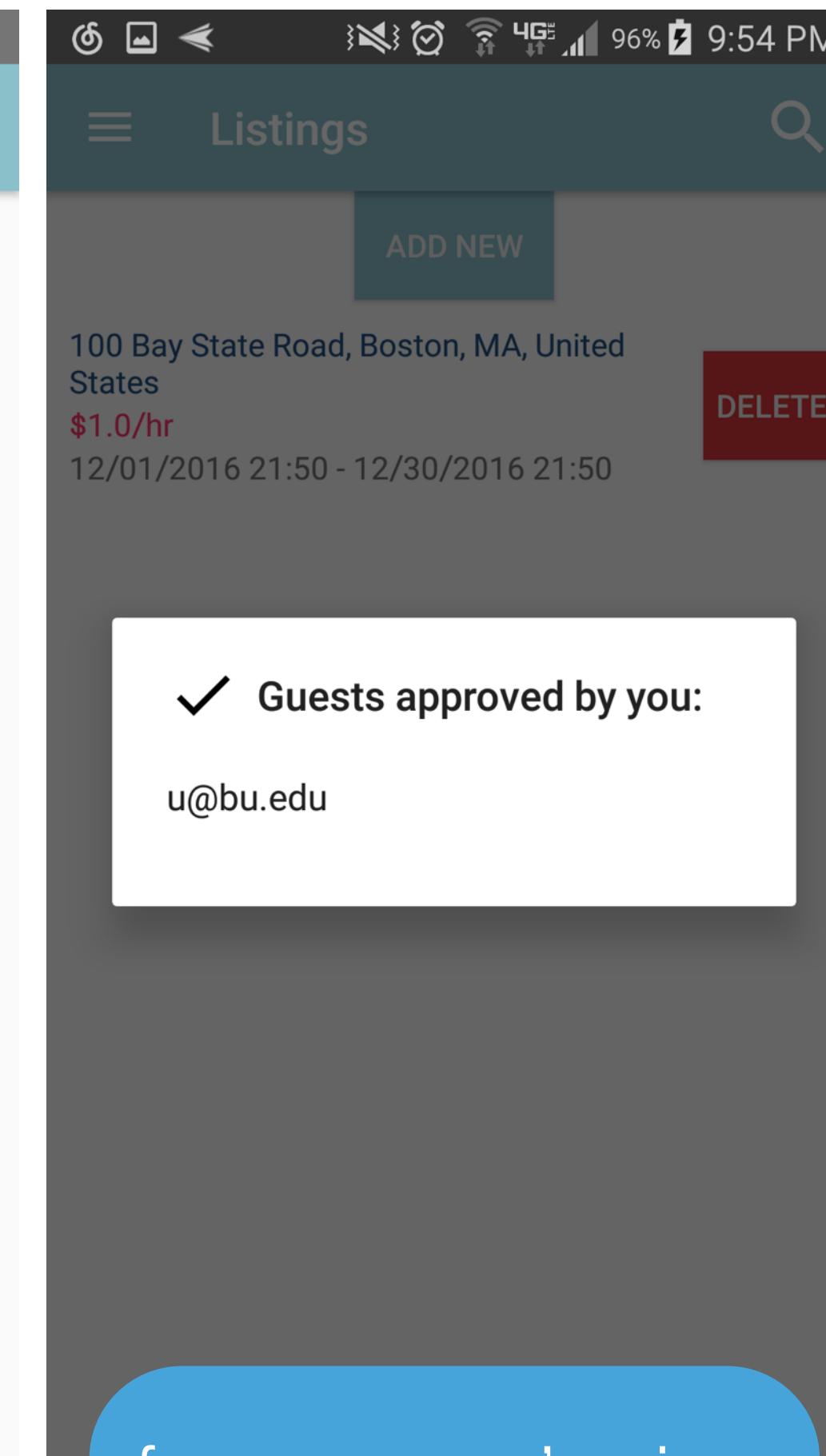
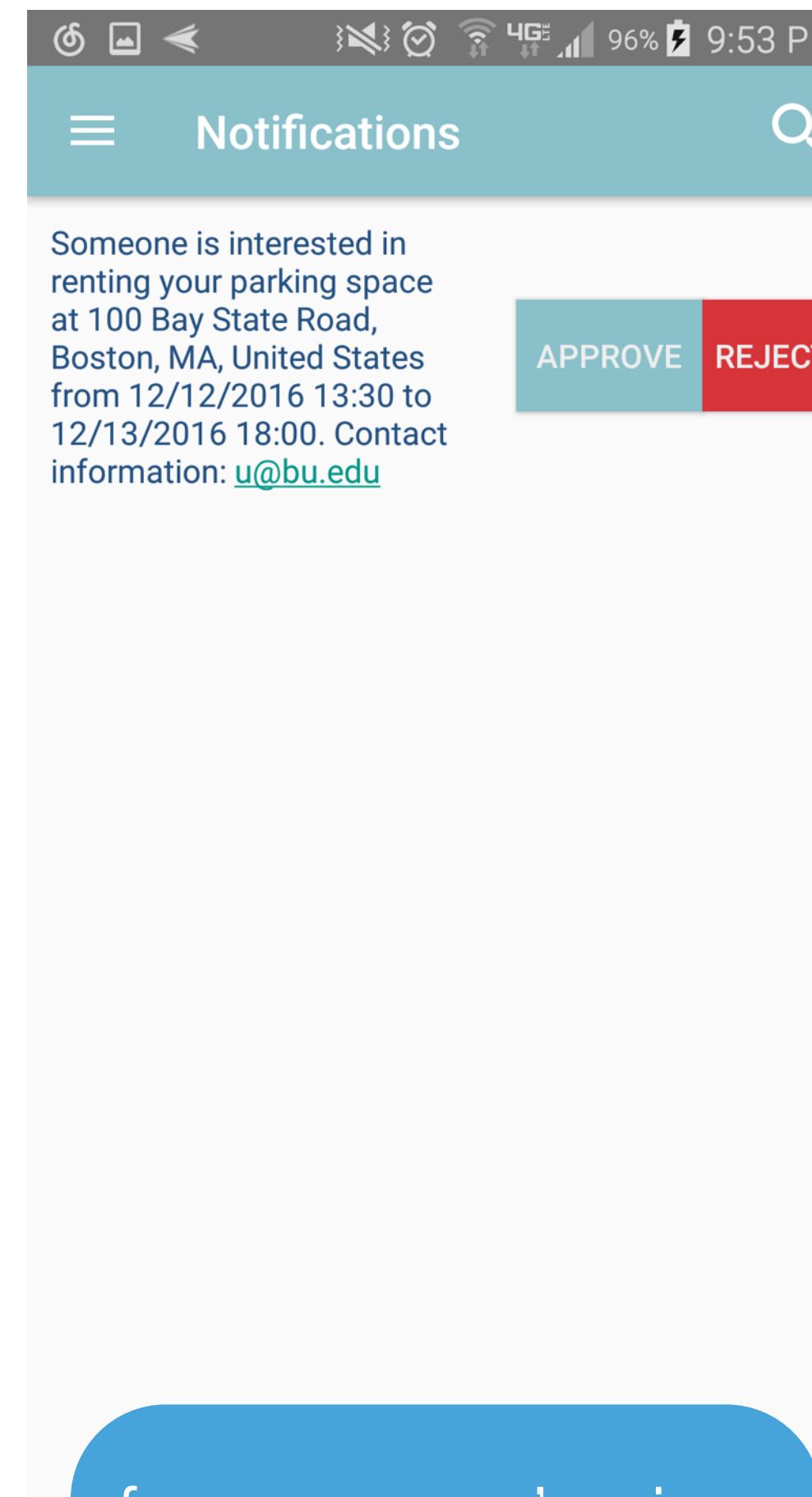
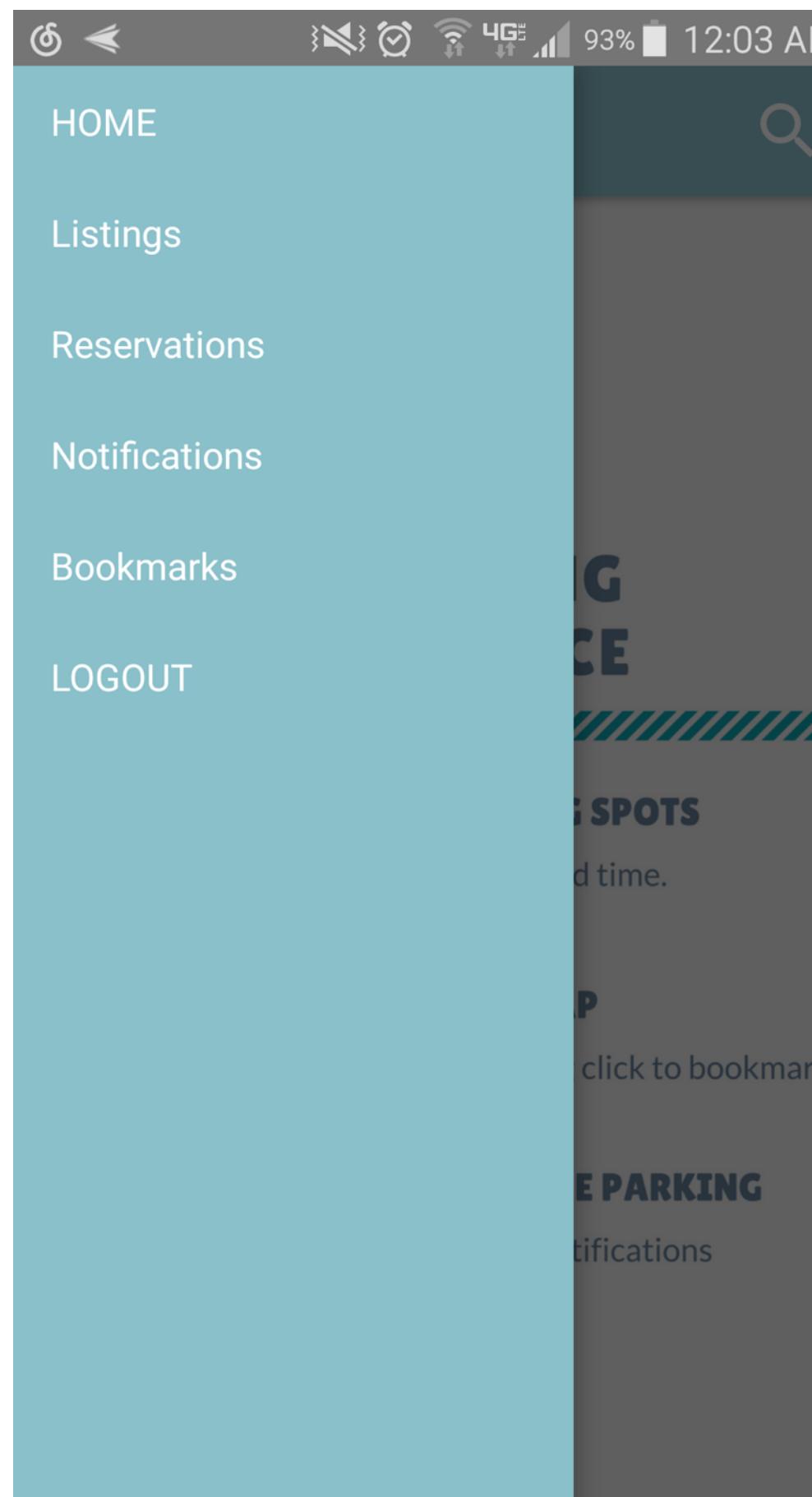
BOOKMARK



# SCREEN FLOW

— — — — —

## NOTIFICATION



from owner's view

from owner's view

from guest's view

# SCREEN FLOW

## NAVIGATION

The image shows a sequence of four mobile screen captures illustrating a navigation flow:

- Screen 1: Profile Overview**  
The top navigation bar includes icons for battery, signal, and time (12:03 AM). The main menu on the left lists: HOME, Listings, Reservations, Notifications, Bookmarks, and LOGOUT. A large dark sidebar on the right contains text: "G SPOTS", "click to bookmark", and "PARKING".
- Screen 2: Reservations**  
The top navigation bar shows time (9:54 PM). The title is "Reservations". A listing is shown: "100 Bay State Road, Boston, MA, United States" (Approved by owner), "\$1.0/hr", and the date range "12/12/2016 13:30 - 12/13/2016 18:00".
- Screen 3: Bookmarks**  
The top navigation bar shows time (9:52 PM). The title is "Bookmarks". A listing is shown: "100 Bay State Road, Boston, MA, United States" (Approved by owner), "\$1.0/hr", and the date range "12/12/2016 13:30 - 12/13/2016 18:00".
- Screen 4: Google Maps**  
The top navigation bar shows time (10:04 PM). The title is "Bookmarks". A listing is shown: "100 Bay State Road, Boston, MA, United States" (Approved by owner), "\$1.0/hr", and the date range "12/12/2016 13:30 - 12/13/2016 18:00". Below the listing is a "REQUEST" button (light blue) and a "DELETE" button (red). A purple arrow points from the address in the Reservations screen to the address in the Bookmarks screen. Another purple arrow points from the address in the Bookmarks screen to the Google Maps icon.

**click on address**

**Google**

# ARCHITECTURE

## Data Models

The data repository for this app is **Amazon DynamoDB**

I created two tables, one for users and one for listings.

primary key

	username	bookmarks	email	listings	notifications	password	reservations
	user1	[]	u@bu.edu	[{"S": "afb..."}]	[]	1234	[{"L": [{"..."}]}
	user2	[]	u2@bu.edu	[]	[]	1234	[]

matching

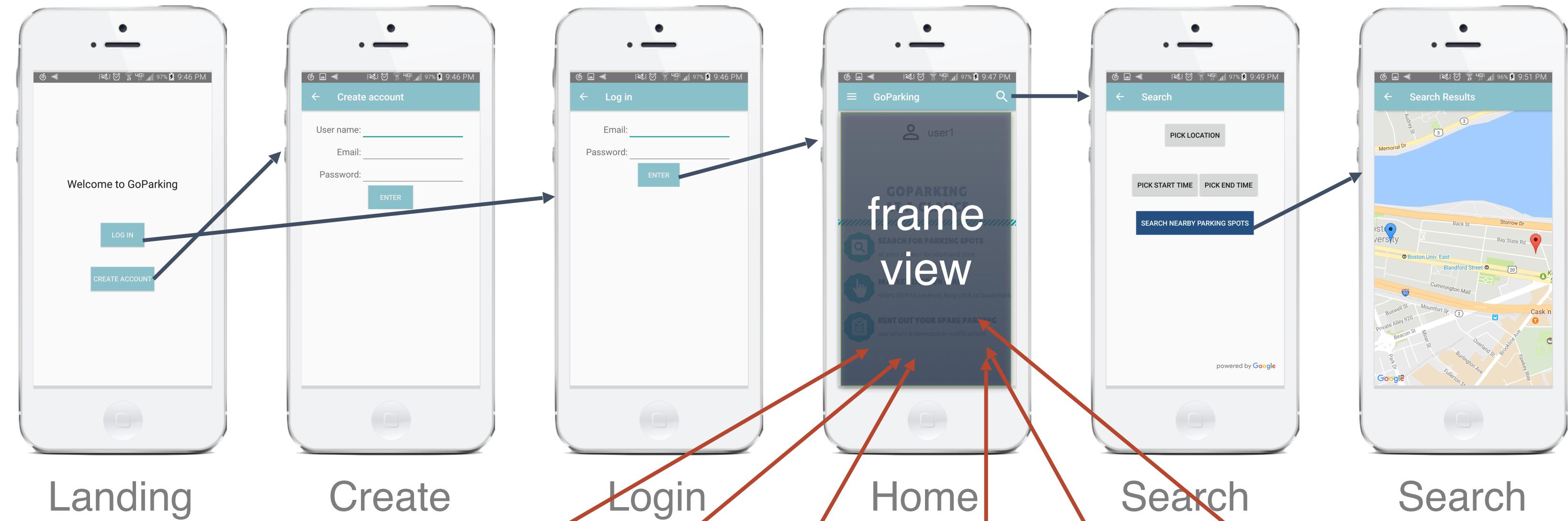
primary key

	id	address	approved	availableStarting	availableUntil	latitude	longitude	numSpots	owner	price
	afb2d0aa-c	100 Bay Stat...	[{"S": "u@b..."}]	12/01/2016 21:50	12/30/2016 21:50	42.3497666	-71.0979066	10	user1	1

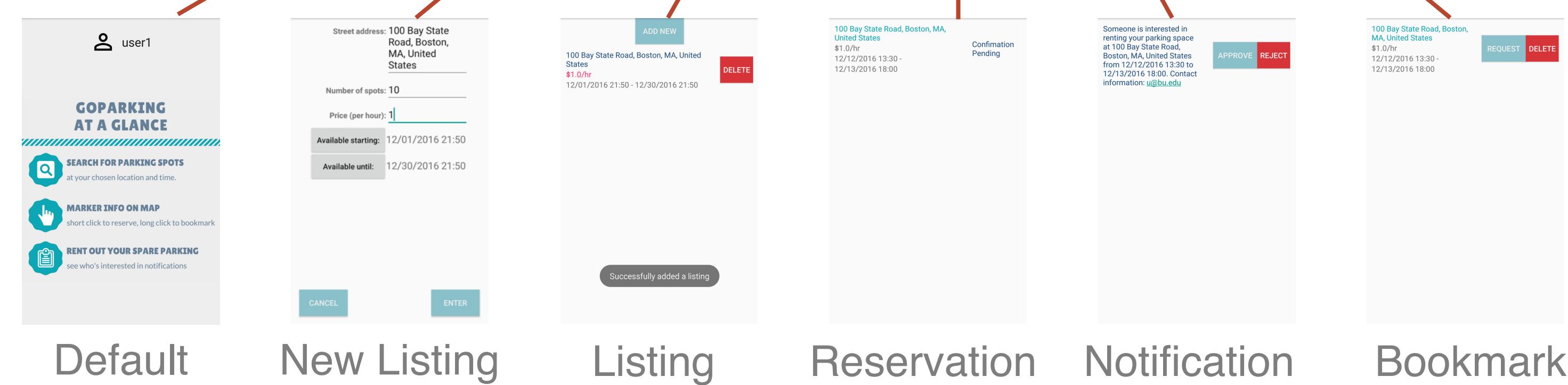
# ARCHITECTURE

## Activities & Fragments

### Activities:



### Fragments:



# ARCHITECTURE

APIs & UI libraries

**Google Maps Geocoding API:** converts street addresses into coordinates (latitude and longitude), also helps to check if an address is valid as the API returns “error” when querying an invalid address

**Google Places API:** provides a UI dialog that displays an interactive map for user to select a location in the search activity, and provides autocomplete service when user starts typing in an address when creating a new listing.

Google Maps Android API: displays map with customizable markers in the search result activity

DatePickerDialog, TimePickerDialog: allows user to quickly select year, month, day, hour, and minute

AlertDialog: displays a title, message, and yes/no buttons for user feedback

# PROBLEMS & SOLUTIONS



**The most difficult part is planning out the app.**

- UI decisions
- Data flow
- Have to reason from different user perspectives
- Some use cases are intermingled

One specific problem: Maintaining user identity when transitioning between activities (e.g. home activity and search activity)

Solution: Adding the username data, which uniquely identifies a user, into intent, and using startActivityForResult and setResult to pass around the username between activities

**But many things also become easier with practice.**

- Populating list view with custom array adapters
- Attaching OnClickListeners to buttons
- Writing database queries
- Providing feedback to user via Toast
- Figuring out what exactly this app should do...

# DEMO

QUESTIONS 

