CIS 651 – Mobile Application Programming

FINDIT

Project Report

By,

Yadav Murthy (SUID - 990783888)

Mona Ramesh (SUID - 264581836)

Table of Contents

1. INTRODUCTION

FindIt is an iOS application which is a perfect companion for a person who is new to a place. It is a guide to the neighborhood. As the name suggests its allows users to find nearby places based on various categories. The categories include restaurants, coffee shops, banks, libraries, hospitals, pharmacies, supermarkets, bars etc. User registration and login is not required to use this app. Users can just download the app and make use of it. It provides a short video which lists top places based on each category. It also displays the current weather information based on user's current location which will help the user to plan accordingly. The app also provides route to the destination selected by the user. Text to Speech has been implemented with an aim to help the blind community use the app as well.

2. UTILITIY

- Allows the user to search for a destination category-wise.
- Provides suggestions for each category through a short video.
- Provides current weather information.
- Provides the user with route and directions by using Apple Maps.

When a person is new to a place, he/she is unaware about the nearest restaurants, coffee shops, banks, hospitals etc. At such situations, a mobile application which lists all the nearest places based on various categories will be of great help. FindIt exactly delivers the same. This application doesn't require user registration. Instead, they can just download the application from the app store and make the best use of it. It is very simple and user friendly.

3. MODEL VIEW CONTROLLER (MVC)

Model View Controller is a software architectural pattern for implementing user interface on computers. It divides a given application into three interconnected parts in order to separate internal representations of information from the ways that information is presented to and accepted from the user.

Components:

Model – Model layer holds data and knows nothing about user interface. Usually it represents the real things in the world of the user.

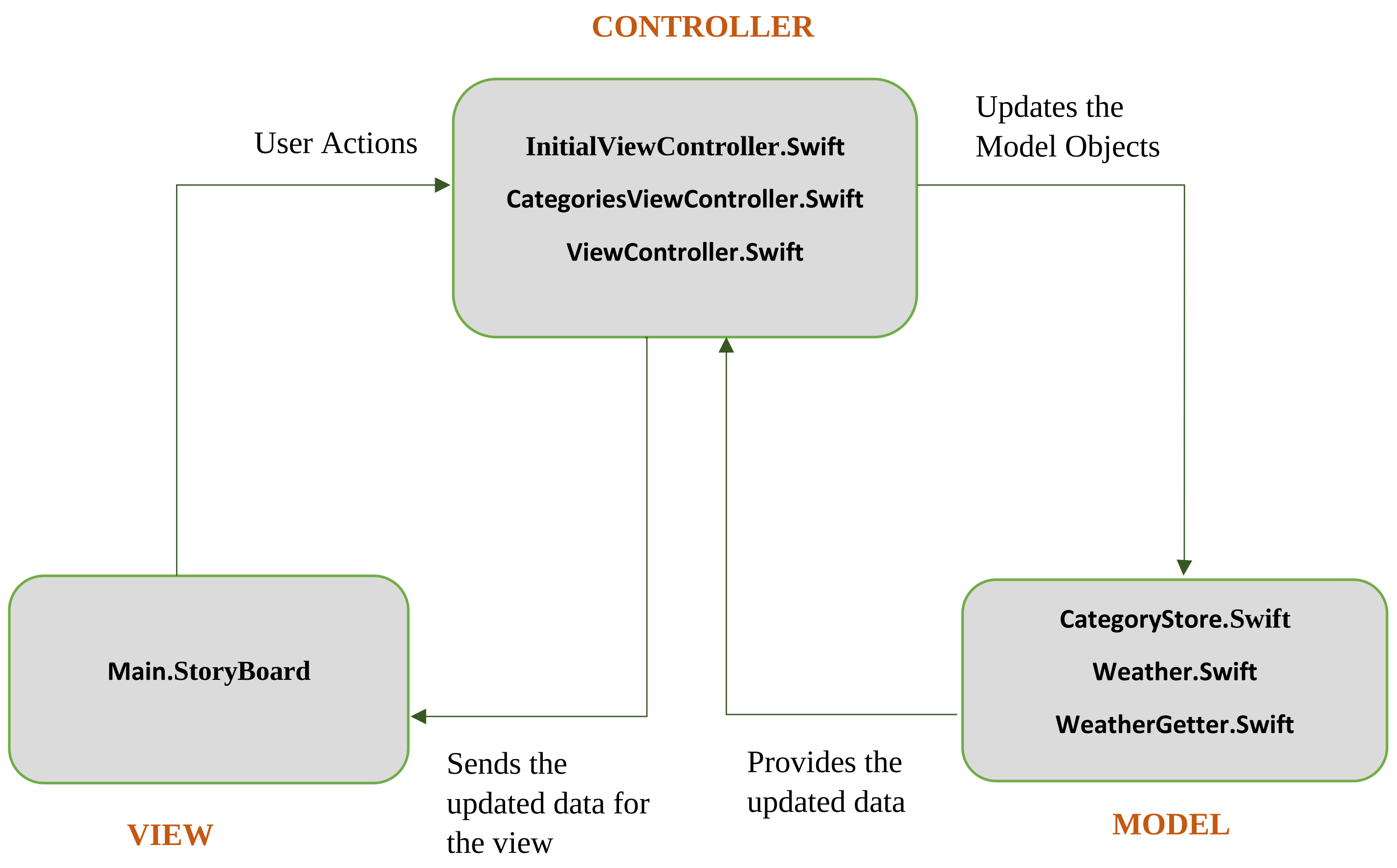
View – View layer contains objects that are visible to the user. E.g. Buttons, Text Field etc.

Controller – The controller layer is where the application is managed. Controller objects or controllers are the managers of the application and make sure both the model and the views are synchronized throughout the application.

Interactions:

1. View accepts user input and passes it to the controller
2. Controller receives the input from the view, makes the necessary processing, applies logic on the data and asks the model to update the data objects correspondingly.
3. Model stores, maintains and updates the data and sends the updated data value to controller.
4. Controller passes the updated data to the view and view displays the data as output to users.

FindIt MVC Architecture


CONTROLLER
InitialViewController.Swift
CategoriesViewController.Swift
ViewController.Swift
User Actions
Updates the Model Objects
Main.StoryBoard
VIEW
Sends the updated data for the view
Provides the updated data
CategoryStore.Swift
Weather.Swift
WeatherGetter.Swift
MODEL

4. TECHNICAL DETAILS

Some of the technical details and Application Programming Interfaces (API) used in the development of the app is described below:

4.1 Google Places API

- The Google Places API Web Service allows you to query for place information on a variety of categories, such as: establishments, prominent points of interest, geographic locations, and more.
- You can search for places either by proximity or a text string. A Place Search returns a list of places along with summary information about each place.
- Our base URL is
  https://maps.googleapis.com/maps/api/place/nearbysearch/json?
- A Nearby Search lets you search for places within a specified area
- We will append required parameters to the base URL which includes
  - **API key** – application's API key
  - **Location** – The latitude/longitude around which to retrieve place information. This must be specified as *latitude*, *longitude*.
  - **Radius** – Defines the distance (in meters) within which to return place results. The maximum allowed radius is 50,000 meters
- We will append optional parameters to the base URL which will narrow down the search criteria based on the category
  - **Type** – Restricts the results to places matching the specified type. Only one type may be specified. In our case, we will pass the category which the user selects
  - **Keyword -** A term to be matched against all content that Google has indexed for this place. We will again pass the category as a keyword.

```
let urlString
= "https://maps.googleapis.com/maps/api/place/nearbysearch/json?location=
\(lat),\(long)&radius=1000&type=\(category)&keyword=\(category)&key=
AIzaSyD0862g64lX3ElcCQ7-yK-XKShAVtgWLxU"
```

4.2 Open Weather Map API

- OpenWeatherMap API is used to access current weather data for any location on Earth including over 200,000 cities.
- Current weather is frequently updated based on global models and data from more than 40,000 weather stations.
- Data is available in JSON, XML, or HTML format. In our app, we will receive the data in JSON format and parse it.
- Our Base URL is as follows
  http://samples.openweathermap.org/data/2.5/weather?
- In our case, we will extract weather information based on geographic coordinates - latitude and longitude.
- Hence we should append latitude and longitude as parameters in our base URL along with API key.

```
let weatherRequestURL
= NSURL(string: "\(openWeatherMapBaseURL)?APPID=\(openWeatherMap
APIKey)&lat=\(lat)&lon=\(long)")!
```

4.3 UITableViewController

UITableView:

- A UITableView displays a single column of data with a variable number of rows.
- A UITableView is a viewobject which needs the following
  - **View controller** - to handle its appearance on the screen.
  - **Data source** – to ask number of rows and data for that rows to be displayed. Without data source, a table view is just an empty container
  - **Delegate** – to inform other objects of events.
- An instance of UITableViewController fulfills all the above 3 roles.
- In our app, we have used table view to display the various categories for users to select.
- We have also included corresponding image for each category in each table view cell.

4.4 UINavigationController

- A UINavigationController maintains an array of view controllers presenting related information in a stack. When a UIViewController is on top of the stack, its view is visible.
- In our application, on clicking table view's cell, navigation controller is invoked which launches the next view controller for further processing.

4.5 UIStoryboardSegue

- A segue prepares for and performs a visual transition between two view controllers in our app's storyboard file.
- The starting point of a segue can be a button, table row, or gesture recognizer that initiates the segue.
- The endpoint of a segue is the view controller that we want to display.
- We have used segue in our application to pass data from one view controller to next view controller when navigating.
- The segue will get prepared by the following function:

```
override func prepareForSegue(segue: UIStoryboardSegue, sender: AnyObject?)
```

In our application, table view's text field is the point where the navigation and segue starts. On clicking any table view cell, the cell's information is carried to next view controller for further processing.

4.6 MKMapItem

- The MKMapItem class encapsulates information about a specific point on a map. This information includes the map location and any other data that might be relevant, such as the name of a business at that location. Apps use this class to share map-related data with the Maps app.

- In our application, once we fetch all the locations we store it in MKMapItem instance. We will launch the Apple Maps application by passing all the locations via MKMapItem array in the following manner:

- mapItems.append(mapItem)
  where mapItem = MKMapItem
  mapItems = [MKMapItem]()

4.7 UIAlertController

The UIAlertController displays an alert message to user. In addition to displaying the alerts certain actions can also be handled through the alertcontroller. We add actions using the addAction()

```
func showSimpleAlert(title title: String, message: String) {
    let alert = UIAlertController(
        title: title,
        message: message,
        preferredStyle: .Alert
    )
    let okAction = UIAlertAction(
        title: "OK",
        style:  .Default,
        handler: nil
    )
    alert.addAction(okAction)
    presentViewController(
        alert,
        animated: true,
        completion: nil
    )
  }
```

4.8 Animations

- Animations provide fluid visual transitions between different states of the user interface.
- In iOS, animations are used extensively to reposition views, change their size, remove them from view hierarchies, and hide them.

- Animations are used to convey feedback to the user or to implement interesting visual effects.

- We have included simple animations in displaying weather information. The weather information labels fly in from left to right and vice versa.

```
        UIView.animateWithDuration(0.6, delay: 0, options: [.CurveEaseOut],
animations: {
         self.view.layoutIfNeeded()
         }, completion: nil)
```

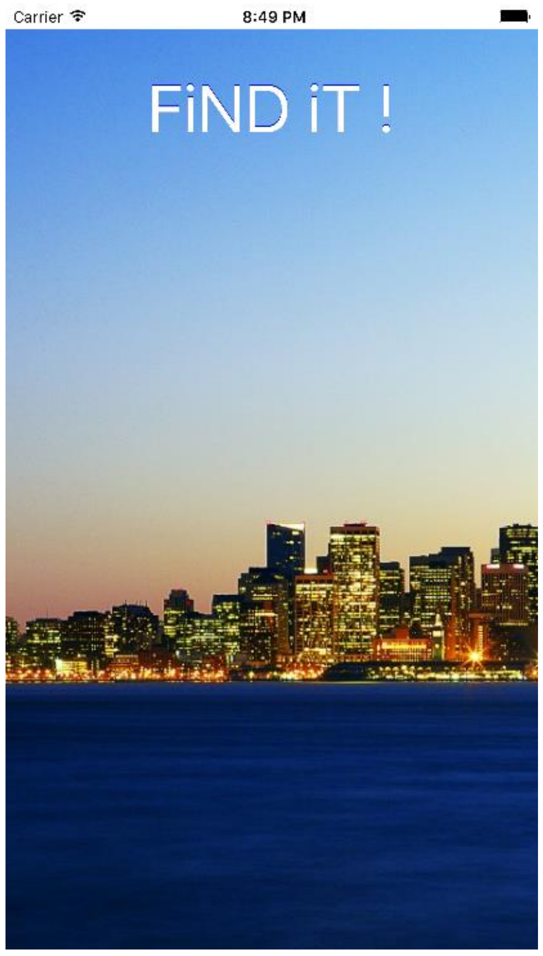
4.9 Text to Speech

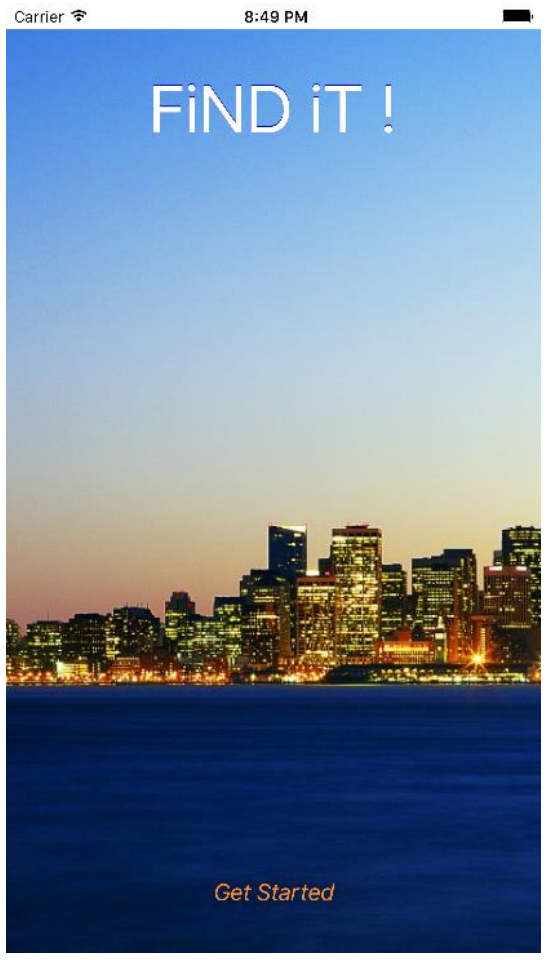
- A text-to-speech (TTS) system converts normal language text into speech.
- Speech synthesis is the artificial production of human speech.
- In our application, we have implemented the AVFoundation Framework for text to speech functionality.

```
          let synth = AVSpeechSynthesizer()
          synth.delegate = self;
          var myUtterance = AVSpeechUtterance(string: "")
          let speak = "Welcome to Find it. A guide to your neighborhood. Tap on
    get started."
          myUtterance = AVSpeechUtterance(string: speak)
          myUtterance.rate = 0.5
          synth.speakUtterance(myUtterance)
```

5. SCREENSHOTS AND APPLICATION FLOW

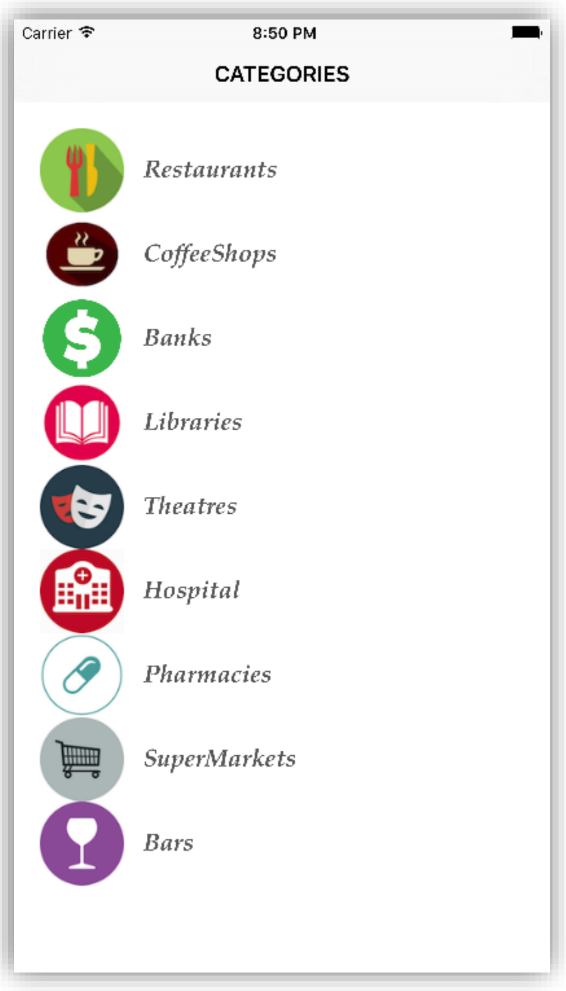
Screenshots:



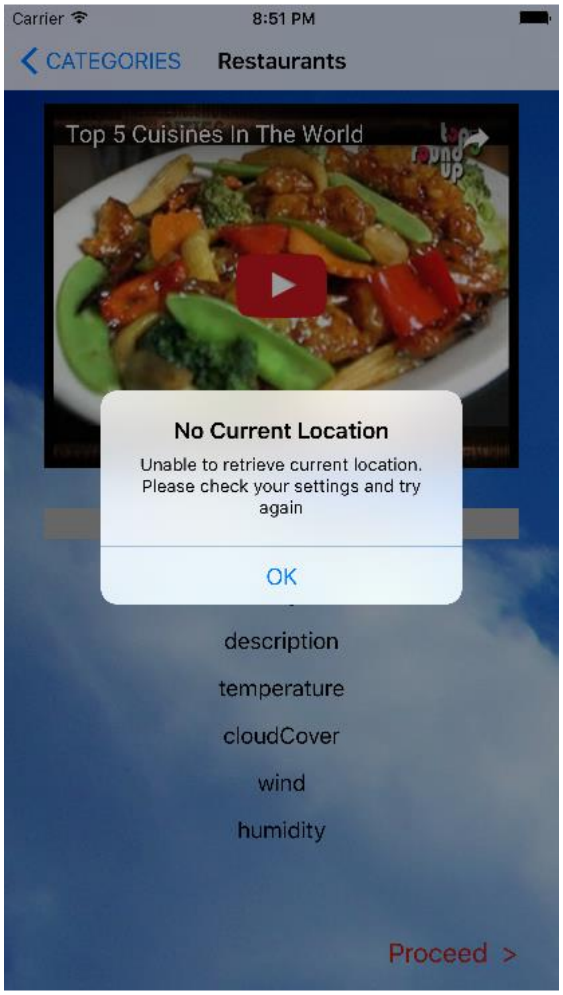


Screen 1: App launch screen
Welcome text to speech.

Screen 2: On completion of text to speech, “Get Started” button is displayed.

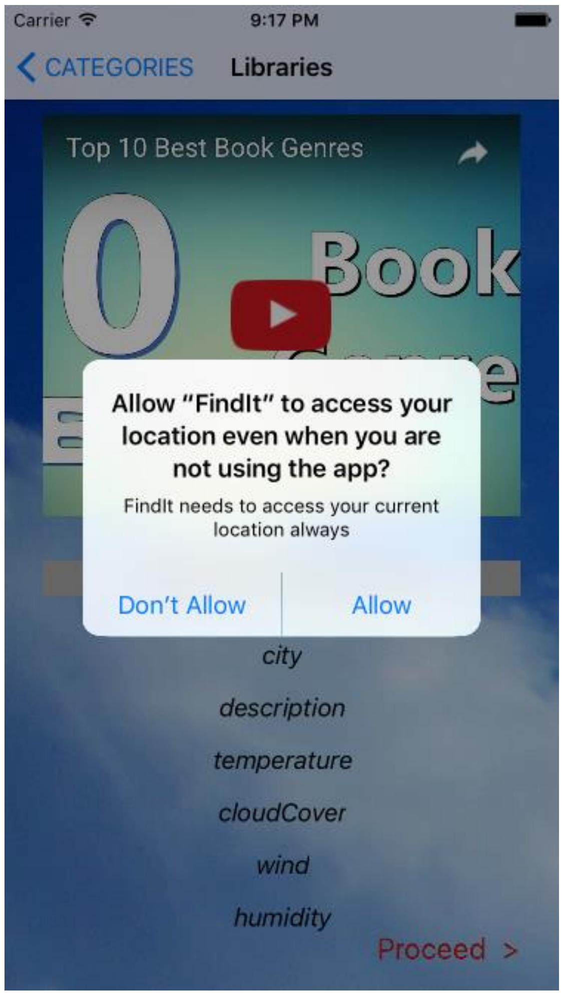


Screen 3: On clicking "Get Started" button, the category list is displayed.



No Current Location

Unable to retrieve current location. Please check your settings and try again

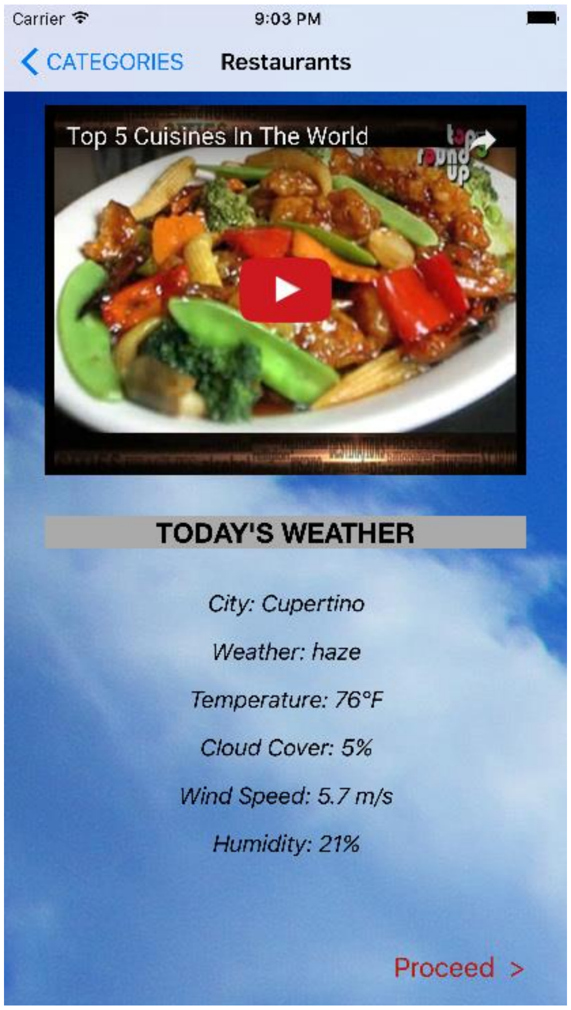


Allow "FindIt" to access your location even when you are not using the app?

FindIt needs to access your current location always

Screen 4: When location services are OFF, an alert is displayed.

Screen 5: The app requests for permission to access the current location.



City: Cupertino

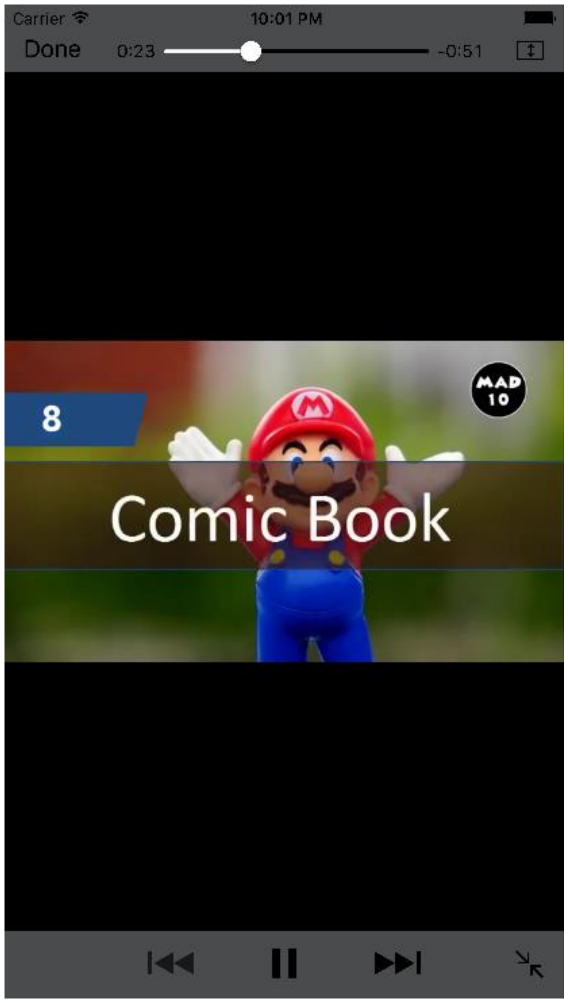
Weather: haze

Temperature: 76°F

Cloud Cover: 5%

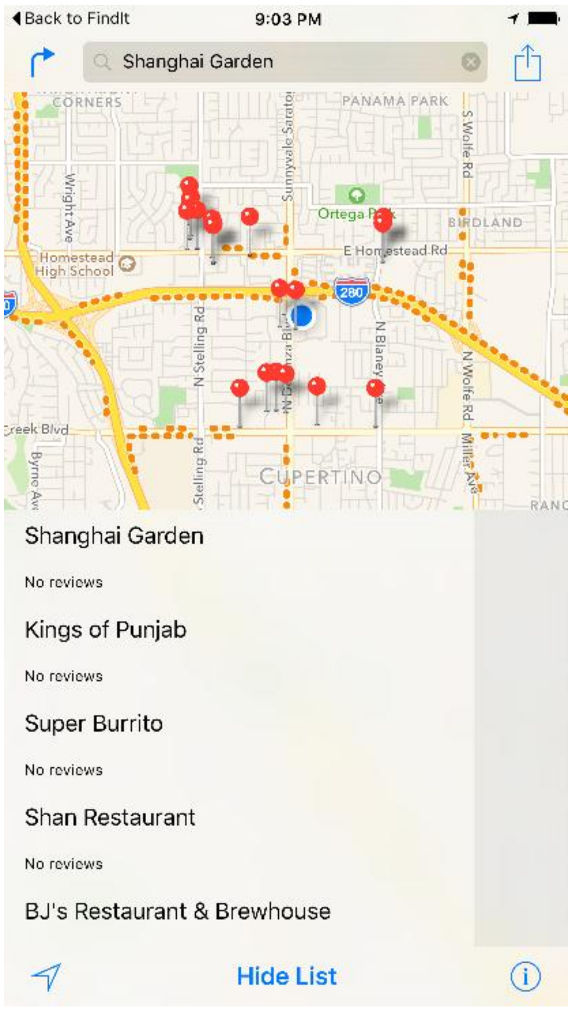
Wind Speed: 5.7 m/s

Humidity: 21%



Screen 6: When location services are ON, Weather information is displayed.

Screen 7: On clicking the video, the YouTube video would be loaded and played.



Shanghai Garden

No reviews

Kings of Punjab

No reviews

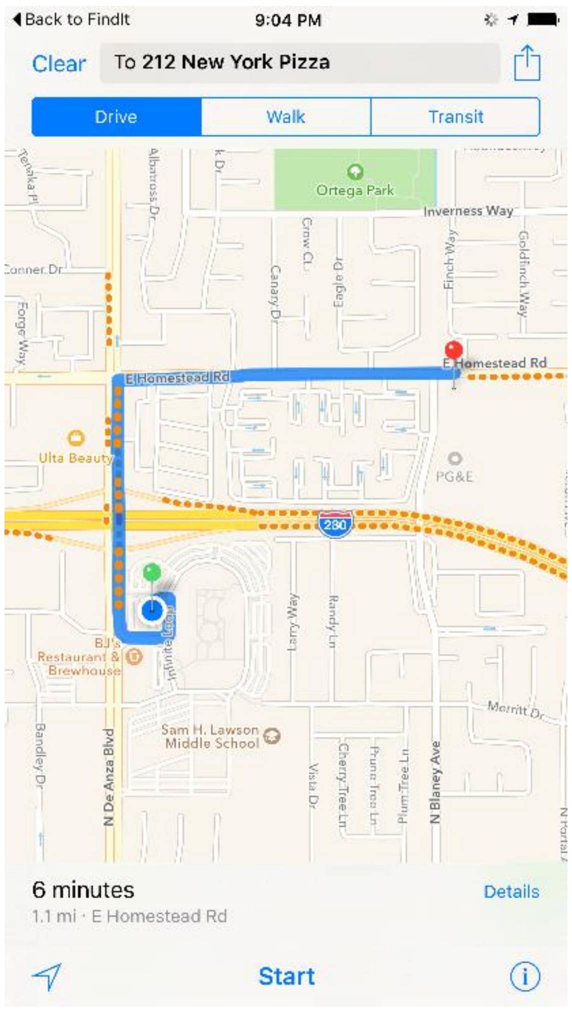
Super Burrito

No reviews

Shan Restaurant

No reviews

BJ's Restaurant & Brewhouse



6 minutes

1.1 mi · E Homestead Rd

Screen 8: On clicking the "Proceed > " button, a list of nearby places will be displayed as a list view and shown on Map.

Screen 9: On selecting a place, directions are shown. The user can select either Drive, Walk, or Transit options.

6. SUMMARY

- The main purpose of the application is to help and guide a person who is new to a place in finding the nearest places.
- Allows searching of nearby places based on various categories.
- The current weather information is also provided to the user.
- They can also watch a short informational video based on the category selected.
- Once the user selects a place, he can also seek direction in 3 ways
  - Walk
  - Drive
  - Transit

7. FUTURE WORK

- Include more categories to search effectively and efficiently.
- Registration and Login facility can be added so that the user can rate the places in the application, store his favorites and review them.
- Enable or disable text to speech button. It gives the user more flexibility.
- Enable notifications.

8. REFERENCES

1. https://developer.apple.com/reference/
2. https://developers.google.com/places/
3. https://openweathermap.org/api
4. https://developer.apple.com/reference/avfoundation/avspeechsynthesizer
5. https://developer.apple.com/reference/mapkit/mkmapitem