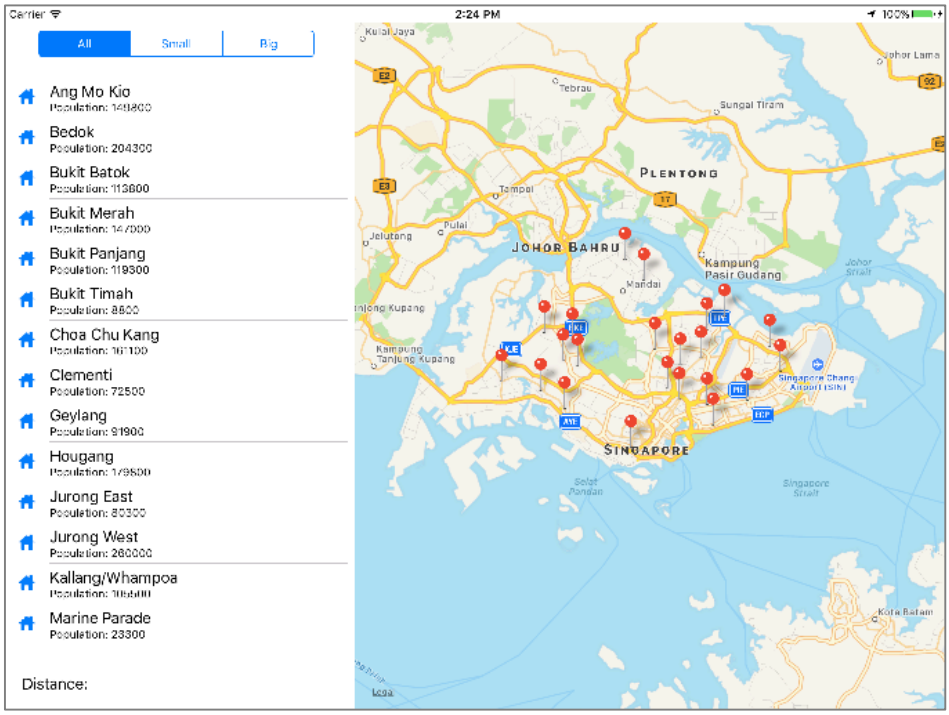
Before you start

Download and unzip the XCode project template from Blackboard. The file name is:

PracticalTestRevision.zip

Background

Your customer wants to create an application that allows their users to view all estates spread around Singapore.

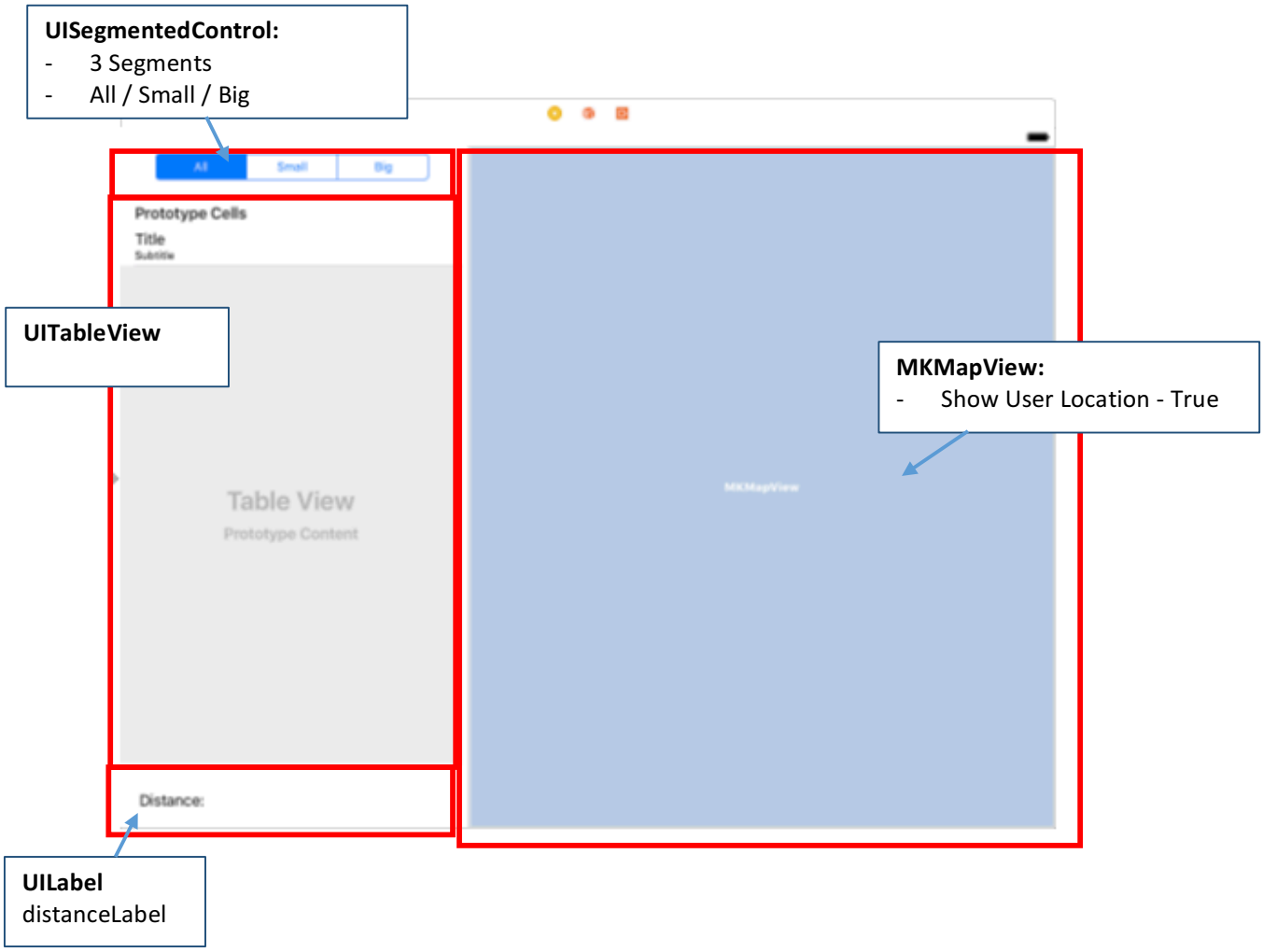


These are some of the features of this application:

1. A Table View on the left showing the list of estates, along with a subtitle indicating the population.
2. A segmented control at the top of the Table View to allow users to highlight different estates in the Table View by its population size.
3. A map of estates represented by pins at their geo-coordinates.
4. A label showing the distance of the estate from the user's current GPS position.

Question 1

a) Design the following View Controller in the Storyboard using the iPad Pro 9.7 inch layout. You do **NOT** need to implement any auto-layout.



NOTE:
*The Table View must consist of a single Table View Cell with a **Subtitle** style.*

b) Create a new View Controller class of any suitable name with the following IBOutlets:

a. tableView – UITableView
b. mapView – MKMapView

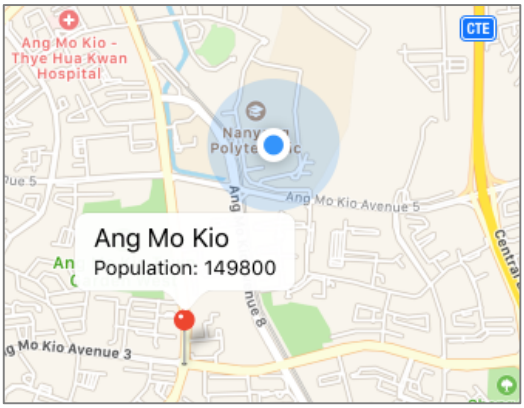
c. sizeSegment – UISegmentedControl
d. distanceLabel – UILabel

Connect the View Controller and the corresponding IBOutlets to your Storyboard.

c) Request permissions from the user to always use his/her location using a new instance of the CLLocationManager object.

NOTE: The Privacy – Location Usage Descriptions in the Info.plist has already been set for you.

d) Run your app in your iPad Air simulator. Your map should be able to display the user's location on the map after you select one of the five pre-defined GPX locations. Fix any issues that arise.



NOTE: You can configure the map to show the Satellite view if the default Standard view doesn't show up.

Question 2

a) Create a new Estate object with the following fields:

a.	name	String	(Swift **non-optional**)
b.	population	Int	(Swift **non-optional**)
c.	latitude	Double	(Swift **non-optional**)
d.	longitude	Double	(Swift **non-optional**)

b) In your View Controller, declare a new array of Estate objects with a suitable variable name.

c) In your View Controller's viewDidLoad function, send a **HTTP POST** request to the server at:

http://crowd.sit.nyp.edu.sg/itp312_2017s1/estate/list

The JSON result format is as follows:

```
[
	{
		"name" : "Ang Mo Kio",
		"pop" : 149800,
		"latitude" : 1.369115,
		"longitude" : 103.845436
	},
	{
		"name" : "Bedok",
		"pop" : 204300,
		"latitude" : 1.323604,
		"longitude" : 103.927338
	},
	…
]
```

d) When the download from the server is successful, create a list of Estate objects by extracting all the data from the JSON result and inserting them into the array you created in part (b).

e) Once complete, ensure that the tableView is reloaded.

f) Run your app in your iPad Air simulator and fix any issues that arise.

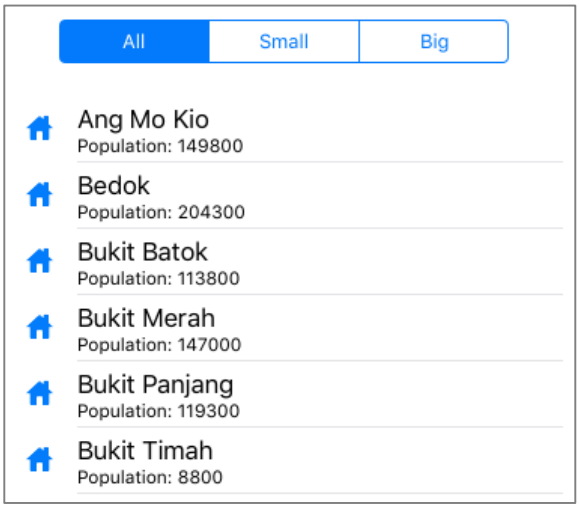
NOTE:

i. Up to this point, you will not be able to see any data appear in the Table View, even though you are able to download the data to your iPad simulator.

ii. The HTTP.swift and SwiftyJSON.swift is already included in the project for you.

Question 3

a) Connect the necessary delegates, and implement all the necessary delegate functions, and set any necessary labels and image views to show your data in the following format:



NOTE:
Each estate must be tagged with the "house" icon found in the Assets.xcassets file.

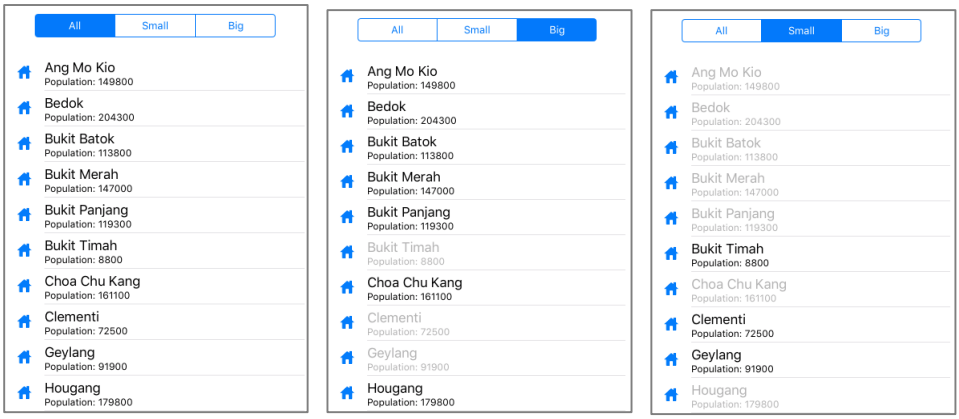
b) Run your app in your iPad Air simulator and fix any issues that arise.

Question 4

a) Modify your View Controller to set the colour of the estate's name and population in the tableView depending on what the user taps on the size segment. (UISegmentedControl).

Size selected	Colour of the TableView Cell's Text
All	All Black
Small	If estates's population < 100000, Black If estates's population >= 100000, Gray
Large	If estates's population < 100000, Gray If estates's population >= 100000, Black

You should be able to see the following when the different sizes are selected:



NOTE:
A black colour's components are (red: 0.0, green: 0.0, blue:0.0, alpha: 1.0)
A gray colour's components are (red: 0.7, green: 0.7, blue:0.7, alpha: 1.0)

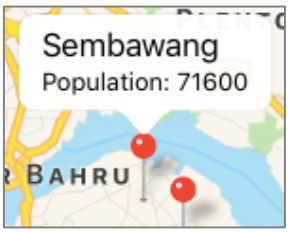
b) Run your app in your iPad Air simulator and fix any issues that arise.

Question 5

a) Create a Swift function called **showMapPins** that takes in no parameters.

In order to show pins on the map, in the **showMapPins** function, write code to create map annotations from the list of **all** estates and add it to your mapView, regardless of what was selected in the size segment.

The pin and its callout should look like the following:

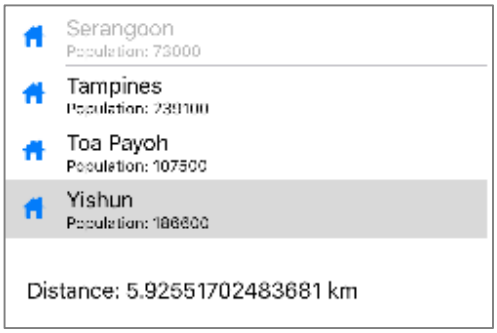


NOTE: The MapAnnotation class has already be written for you

b) Modify your View Controller's viewDidLoad function to show the map pins of all estates by calling **showMapPins** at an appropriate line in your code.

c) Connect the necessary delegates, and implement all necessary codes so that your app:
   i. receives location updates from iOS every **5 meters**.
   ii. receives the user's current GPS location and saves it.

d) Implement the didSelectRowAt for your tableView so that when the user selects an estate, the distance (in kilometers) from the user's current GPS location and the estate's location is computed and displayed in the distance label, as show in the example below:



As an example, if your user is currently in Nanyang Polytechnic, then the correct distances from some of the estates are:

Distance from Nanyang Polytechnic	
Ang Mo Kio	1.140 km
Bedok	10.603 km
Bukit Batok	9.774 km
Bukit Merah	11.065 km

Distance from Pasir Ris MRT	
Ang Mo Kio	11.572 km
Bedok	5.927 km
Bukit Batok	20.717 km
Bukit Merah	17.177 km

NOTE: You do NOT need to round the distance to 3 decimal points.

e) Run your app in your iPad Air simulator and fix any issues that arise.