Requirements

Use Case: User searches a city/county in the location picker

Use Case: User selects a property to further evaluate property details

Use Case: User bookmarks a property

```
#Class is designed to read in a string and use a search algorithm to find available locations
#within the specified area
Public Class LocationPicker {
        Private String input
        Private String finalLocation
        Private Location loc



        #Location Picker Constructor
        Public LocationPicker(String input) {
                This.input = input
                finalLocation = “”
                inputReader(input)
        }

        #Read line of input and then call search algorithm
        Public void inputReader(String in) {
                Scanner scan = new Scanner()
                String line = scan.nextLine(in)

                finalLocation = search(line)
        }

        #Search using a min heap algorithm
        Public List search(String in) {
                List list = loc.getPropertyList()
                #sort list of properties based on what user searches
                List result = insertionSort(list, in)

                #Display Locations within a specified radius
                Return result




        }

        Public List insertionSort(List arr, String in)
        {
                for (int i = 1; i < arr.length; i++)
                {
                        int valueToSort = arr[i]
                        int j
                        for ( j = i; j > 0 && arr[j - 1] > valueToSort; j--) {
                                arr[j] = arr[j - 1]
                        }
                        arr[j] = valueToSort
                }
                Return arr


        #Gets the found final location
        Public String getFinalLocation() {
                Return finalLocation
        }

}

#Location class stores list of locations that are within searched area
Public Class Location {
        Private List propertyList
        Private int numberOfProperties

        #Location Constructor
        Public Location() {
                propertyList = new List()
                numberOfProperties = 0
        }

        #Insert Location to a list if it is within searched area
        Public void insert() {
                Property prop = new Property(name)
                list.add(prop)



        }

        #Clear or remove all locations before next search is made
        Public void clear() {
                propertyList = new List()
                numberOfProperties = 0
        }

        #Returns list of properties
        Public List getPropertyList() {
                Return propertyList
        }


}

#Property class gets details from each available property
Public Class Property {
        Private String name
        Private String address
        Private int dimensionX
        Private int dimensionY
        Private double cost
        Private boolean connectivity

        #Property Constructor, stores information for a property
        Public Property(name, address, dimX, dimY, cost) {
                This.name = name
                This.address = address
                This.dimensionX = dimX
                This.dimensionY = dimY
                This.cost = cost

        }

        #Gets name of property
        Public String getName() {
                Return name
        }

        #Gets address of property
        Public String getAddress() {
                Return address
        }

        #Gets dimensions of property
        Public int getDimensions() {
                Return dimensionX * dimensionY
        }
        #Gets cost of property
        Public double getCost() {
                Return cost
        }

        #Returns true if property has internet access
        Public boolean getConnectivity() {
                Return connectivity
        }
}
```

UI Description

Main Location picker home screen: The main function that this page displays is the search bar in which a user can use to search for potential locations. This page of the application also contains a link to “bookmarked properties”, so that the user can easily access any properties that they have saved.

Location Picker after search: After the user searches for a location he or she is prompted with an outline of the searched area with properties highlighted within that location. Next to that outline are some of the properties. The user can choose to scroll through those properties or click directly on the map to view the exact spots that they are looking for.