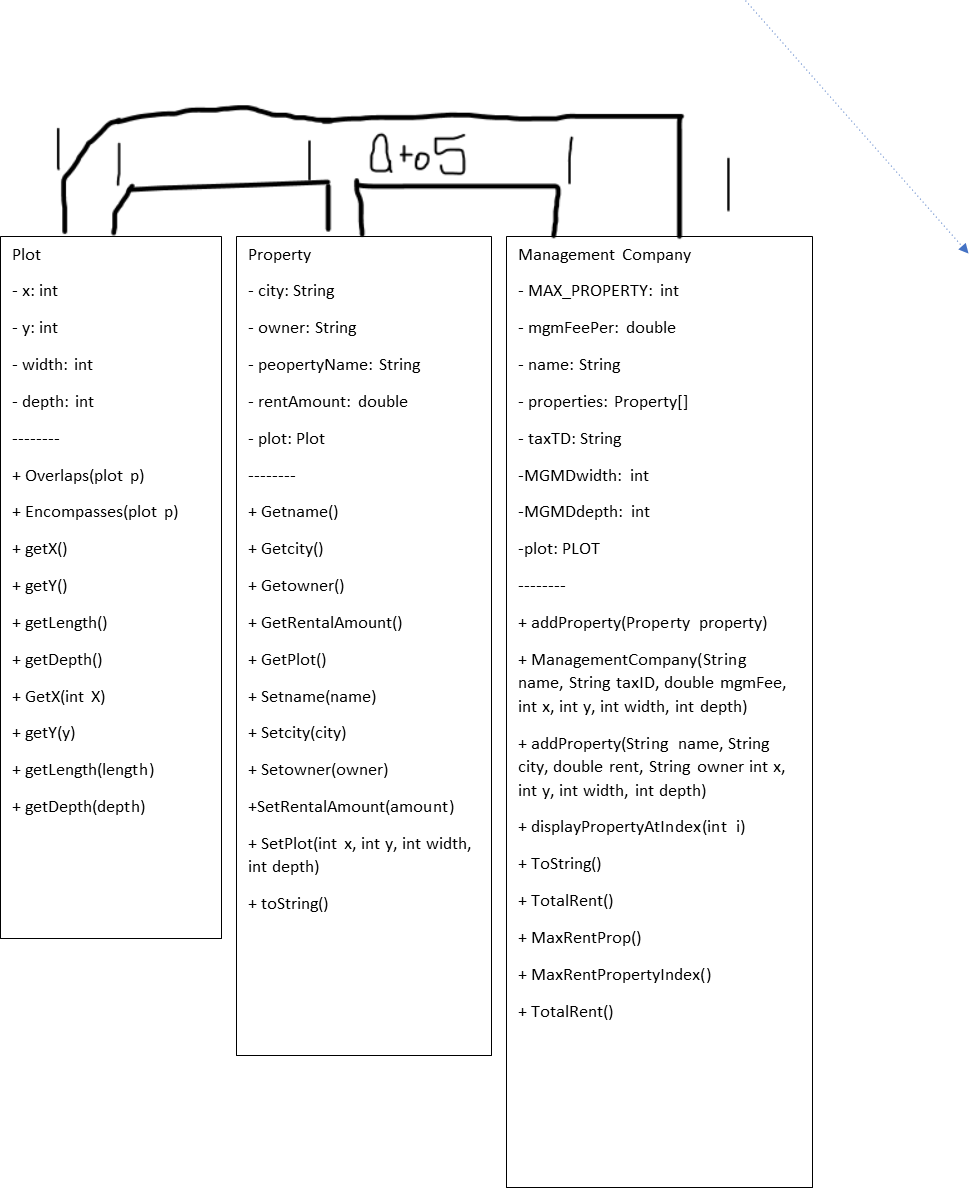
My Turn it in.

UML Class Diagram:



Pseudo-Code:

Class Property {

//Instance variables:

String name, city, owner; Double rental\_Amount; Plot plot. //Assume they are private

//Constructor

Property(){

This.Name = “”;

This.City = “”;

This.Owner = “”;

This.rental\_Amount = 0;

This.plot = new Plot(); //Assume default constructor

}

Property(String propertyName, String city, double rentAmount, String owner){

This.Name = propertyName;

This.City = city;

This.Owner = owner;

This.rental\_Amount = rentAmount;

This.plot = new Plot(); //Assume default constructor

}

Property(String propertyName, String city, double rentAmount, String owner, int x, int y, int width, int depth){

This.Name = propertyName;

This.City = city;

This.Owner = owner;

This.rental\_Amount = rentAmount;

This.plot = new Plot(x, y, width, depth);

}

Property(Property p){

//Copy constructor, however you do a copy constructor, use Clone method?- }

//Getters, assume public

Getname(){

Return this.name;

}

Getcity(){

Return this.city;

}

Getowner(){

Return this.owner;

}

GetRentalAmount(){

Return this.rental\_Amount;

}

GetPlot(){

Return this.Plot;

}

//Setters

Setname(name){

This.name = name

}

Setcity(city){

This.city = city

}

Setowner(owner){

This.owner = owner

}

SetRentalAmount(amount){

This.rental\_Amount = amount

}

SetPlot(int x, int y, int width, int depth){

This.plot.everything = parameter.everthing //All primitive data

}

//Override

Public String toString(){

Return “Property Name: ” + this.getName() +  
 “Located in ” + this.getCity() +  
 “Belonging to: “ + this.getOwner() +  
 “Rent Amount: “ + this.getRental\_Amount();

}

}

Class Plot{

//Instance variables:

Int x, y, length, depth //Assume they are private

//Constructors

Plot(Plot p){

//Copy constructor, however you do a copy constructor, use Clone method?-

}

Plot(){

This.x = 0;

This,y = 0;

This.width = 1;

This.depth = 1;

}

Plot(int x, int y, int width, int depth){

This,x = x;

This.y = y

This.width = width;

This.depth = depth;

}

//methods

Overlaps(plot p){

~~//Use a for loop (two for loops for a 2D array?) to determine whether or not the plots of land overlap in a 2D array, use an if statement for each element of the 2D array to check if a space is aready “occupied” when placing the second plot.~~

//Create a 2D array of booleans from grabbing the biggest x + width, and the biggest y + depth;

//Find biggest x + width

TwoD\_array map = new TwoD\_array(biggest x + width, and the biggest y + depth);

//Use a for loop to parse through plot p on the 2D array

For(int I = x; I < (biggest x + width; I++)){

For(int j = y; I < (biggest y + depth; I++){

map[i][j] = true;

}

}

//Use this

For(){

For(){

if(map[i][j] == true){

return false;

//Otherwise it keeps going.

}

}

}

}

Encompasses(plot p){

//Checks to see if plot of land is inside of another one.

//Use the for loop structure in the overlap method, but instead of checking to see if all the “blocks” are outside, we’re checking to see if they’re all INSIDE.

Boolean Answer = true;

If(p.getX + p.getWidth <= this.getX + this.getWidth){

}

Else{ answer = false; }

If(p.getY + p.length <= this.getY + this.getLength){

}

Else{ answer = false; }

If(p.getX >= this.getX){

}

Else{ answer = false; }

If(p.getY >= this.getY)

}

Else{ answer = false; }

Return answer

}

//Getters

GetX(){return this.X;}, getY() {return this.Y ;}, getLength() {return this.length ;}, getDepth() {return depth;}

//Setters

GetX(int X){this.X = X;}, getY(y) {this.Y = y;}, getLength(length) {this.length = length;}, getDepth(depth) {this.depth = depth;}

}

Class Management Company{

//Instance Variables

Final private MAX\_PROPERTY = 5;

Private Double mgmFeePer;

Private String name;

Private property[] properties;

Private String taxID;

Private int MGMT\_WIDTH = 10;

Private int MGMT\_DEPTH =10;

Private plot plot;

ManagementCompany(){

MgmFeePer = 0;

Name = “”;

TaxID = “”;

this.plot = new plot(0, 0 , 0, 0)  
//Create an empty array of properties.

}

ManagementCompany(String name, String taxID, Double mgmFee){

This.MgmFeePer = mgmFee;

This.Name = name;

This.TaxID = taxID;

this.plot = new plot(0, 0 , 0, 0)  
//Create an empty array of properties.

}

ManagementCompany(String name, String taxID, double mgmFee, int x, int y, int width, int depth){

This.name = name;

This.taxID = taxTD;

This.mgmFee = mgmFee;  
this.x = x;

This.y = y;

This.width = width;

This.depth = depth;

//Create an empty array of properties.

}

ManagementCompany(Managementcompany othercomapny){

This.name = name;

This.taxID = taxTD;

This.mgmFee = mgmFee;  
this.x = x;

This.y = y;

This.width = width;

This.depth = depth;

//Copy array of properties.

SomeArray[] = new array;

For(int I = 0; I < othercompany.propertyarray.getlength; I++){

Somearray[i] = new property(othercompany.propertyArray[i].getProperty());

}

}

Int addProperty(Property property){

//instantiate integer as a very large number  
 //Have a for loop find the smallest “empty” element.

//store int value in integer if an empty element is found

//store property at element [smallest “empty” element]

//return value

}

Int addProperty(String name,String city,double rent,String owner ){

//instantiate integer as a very large number  
 //Have a for loop find the smallest “empty” element.

//store int value in integer if an empty element is found

//plot created\_Plot = new Plot();

//Instantiate property(name, string, city, rent, owner, created\_Plot)

//store property at element [smallest “empty” element]

//return value

}

Int [addProperty](https://word-edit.officeapps.live.com/we/ManagementCompany.html#addProperty(String,%20String,%20double,%20String,%20int,%20int,%20int,%20int))( String name, String city, double rent, String owner int x, int y, int width, int depth){

//Used this code in other add properties.

Int place = -1

//new plot(x,y,width,depth);

//property newProperty = new property(city,owner,rent,name);

If(check for null property object){

//True? Return -2  
}

If(this.plot.encompasses(newProperty.getPlot()) == false){

Return –3;

}

For(int I = 0; I < this.properties.length; I++){

If(element is empty){

Place = I;

}

Else If(newProperty.getPlot().overlaps(this.properties[i])==true){

Return –4;

}

}

If(place != -1){ this.properties[place] = newProperty}

Return place;

//Do they want me to place it anyway?

}

Display propertyatIndex(int i){

//return one long string with the property’s information by using a combination of getters, this, “ “, and +;

}

GetMAX\_PROPERTY(){

Return MAX\_PROPERTY;

}

//Override

ToString(){

//return one long string with all the properties information by using a combination of getters, this, “ “, and +;

}

MaxRentProp(){

//Use a for loop and an instantiated value to compare with all other elements with an if statement to see which one has the highest rent, make sure to check for null spaces, store the highest rent in the instantiated value. Return instantiated value;

}

MaxRentPropertyIndex(){

//Use a for loop and an instantiated value to compare with all other elements with an if statement to see which one has the highest rent, make sure to check for null spaces, store element of the property with the highest rent in the instantiated value. Return instantiated value;

}

TotalRent() {

Return this.properties[0] + this.properties[1] + this.properties[2] + this.properties[3] + this.properties[4];

}

//Instiate double or int;

//Use a for loop to parse through the properties array, making sure not to try and grab any null elements, add up all the rents values to the instantiated variable.

Return (instantiated variable);

}

}

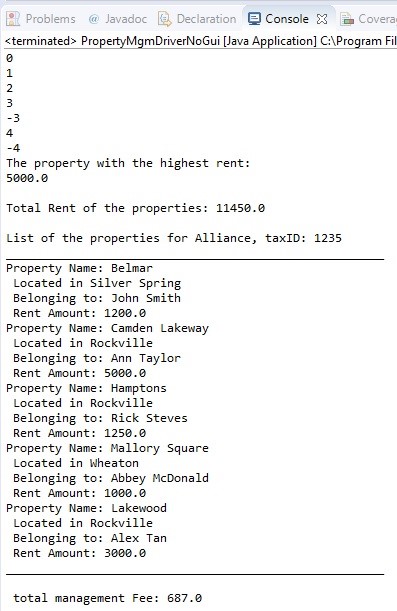
//End of Pseudo-Code

Test-Plan:

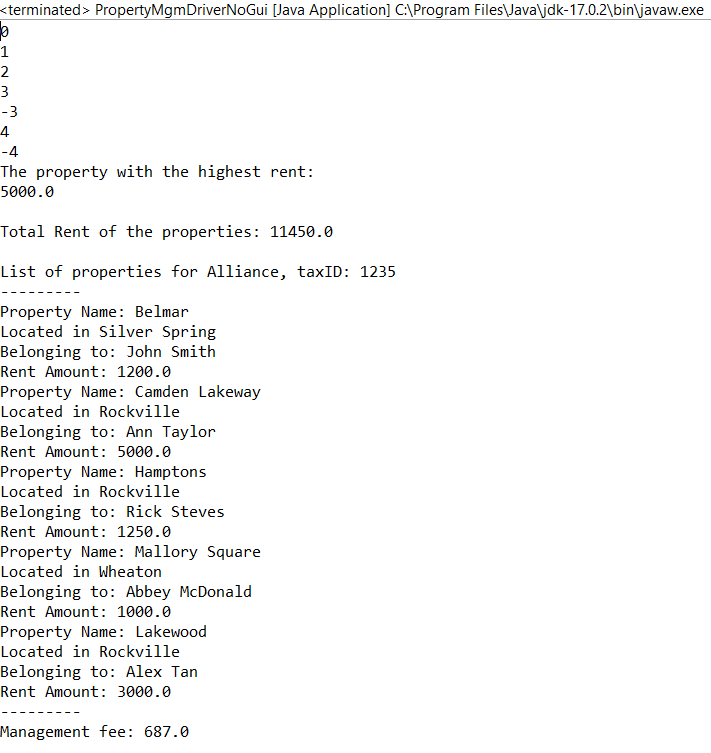
|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case #** | **Input** |  | **Expected Output** | **Actual Output** | **Did the test pass?** |
| 1 | (No GUI test table) |  | (What was in the example screenshot) | 0  1  2  3  -3  4  -4  The property with the highest rent:  5000.0    Total Rent of the properties: 11450.0  (To String Method) | Y |
| 2 | (GUI example) (Placing a house.) |  | “Your first house was placed” | (House was placed.) | Y |
| 3 | (GUI example) (Trying to place a house that overlaps another.) |  | “Overlaps another property” | (House was not placed because it overlapped) | Y |
| 4 | (GUI example) (Trying to place a house when the properties array is full..) |  | (House can’t be added because it exceeds the max value of the array.) | (The house couldn’t be added because the Array was full) | Y |

Test Table Screenshots:

(Original)



(Mine)



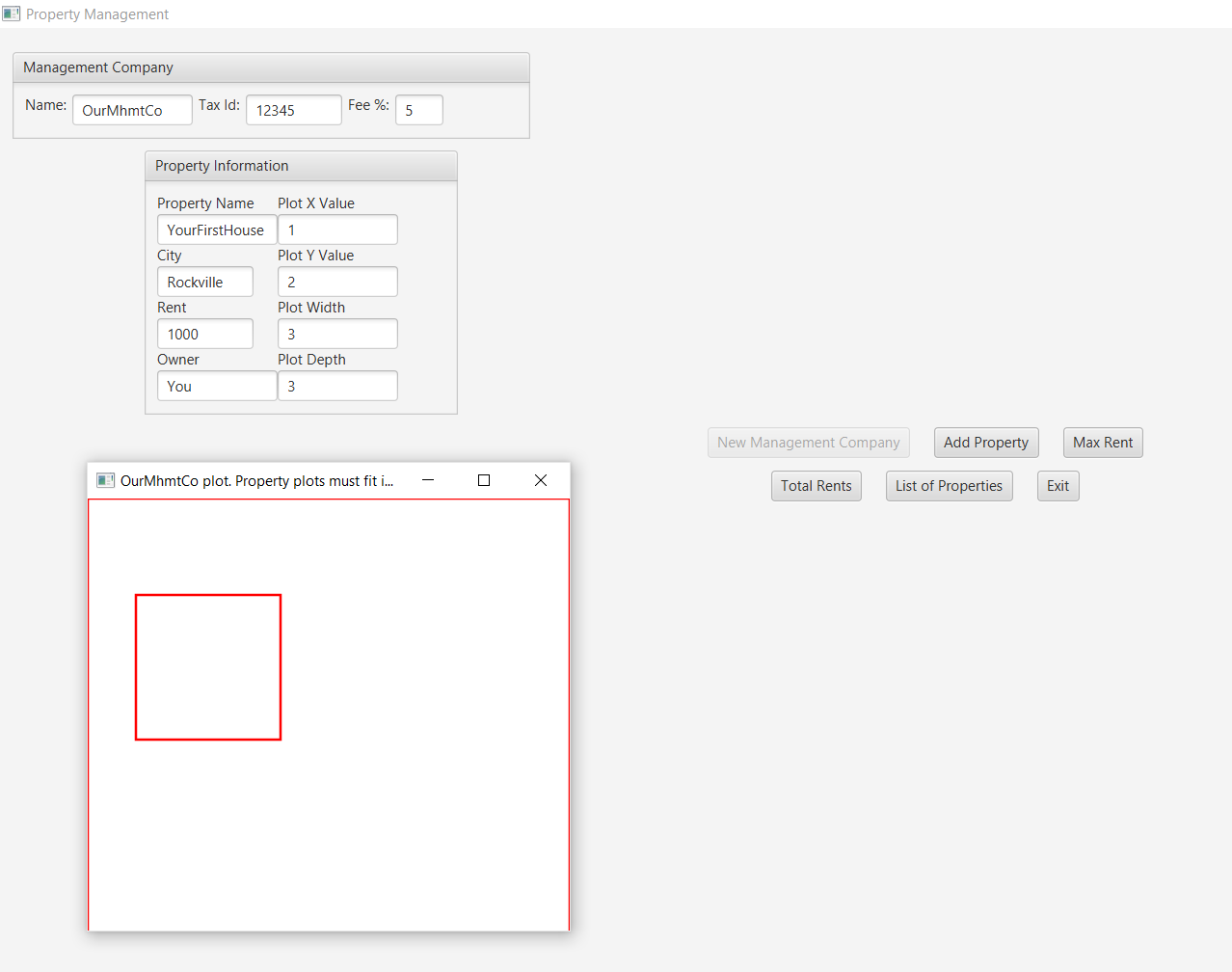
(Test Case 2, GUI)

(example)

A screenshot of a cell phone

Description automatically generated

(Mine)

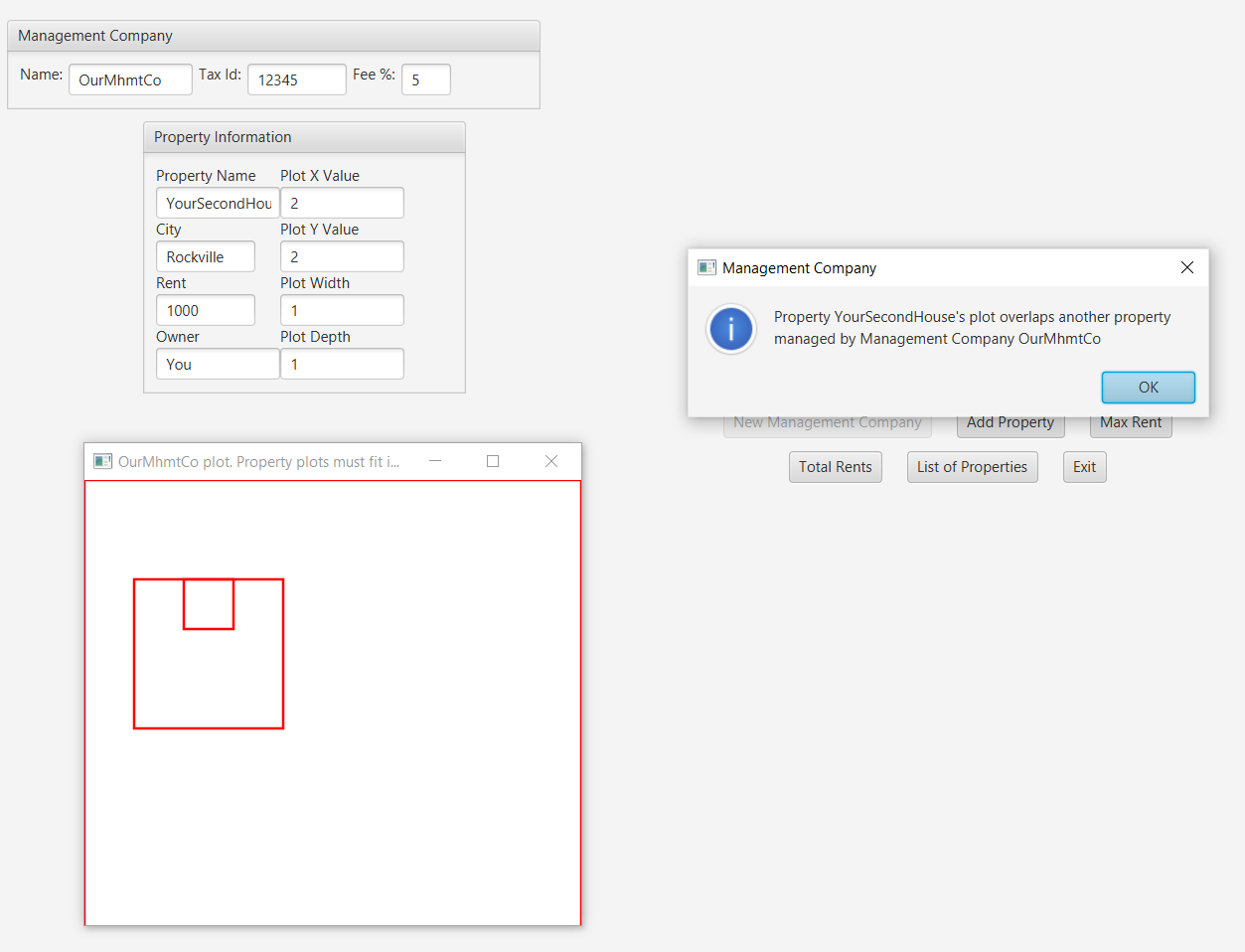


(Third Test):

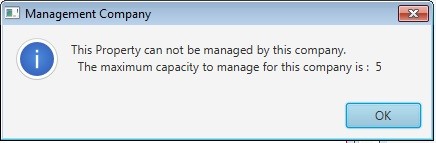
(Example)

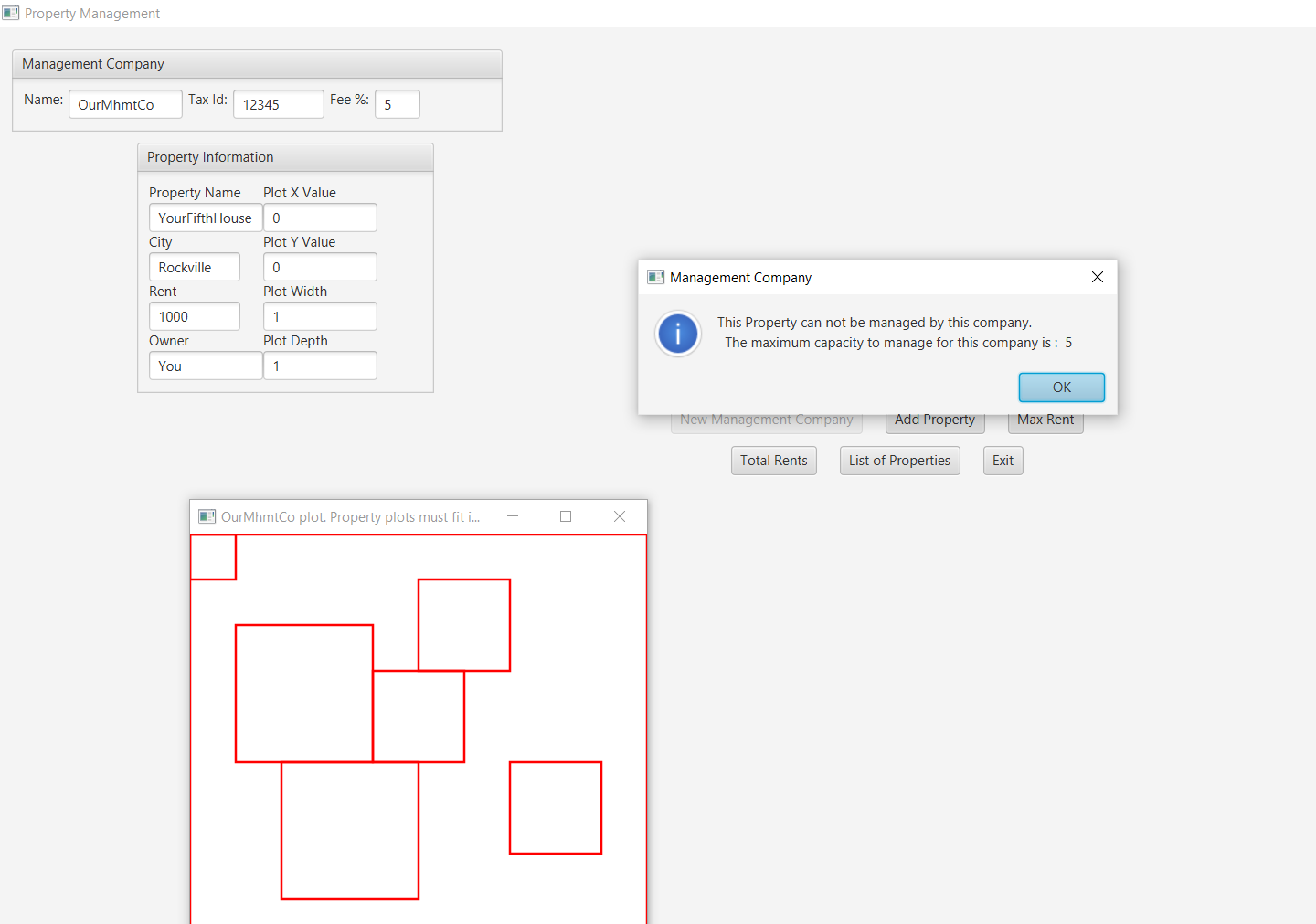
A screenshot of a social media post

Description automatically generated

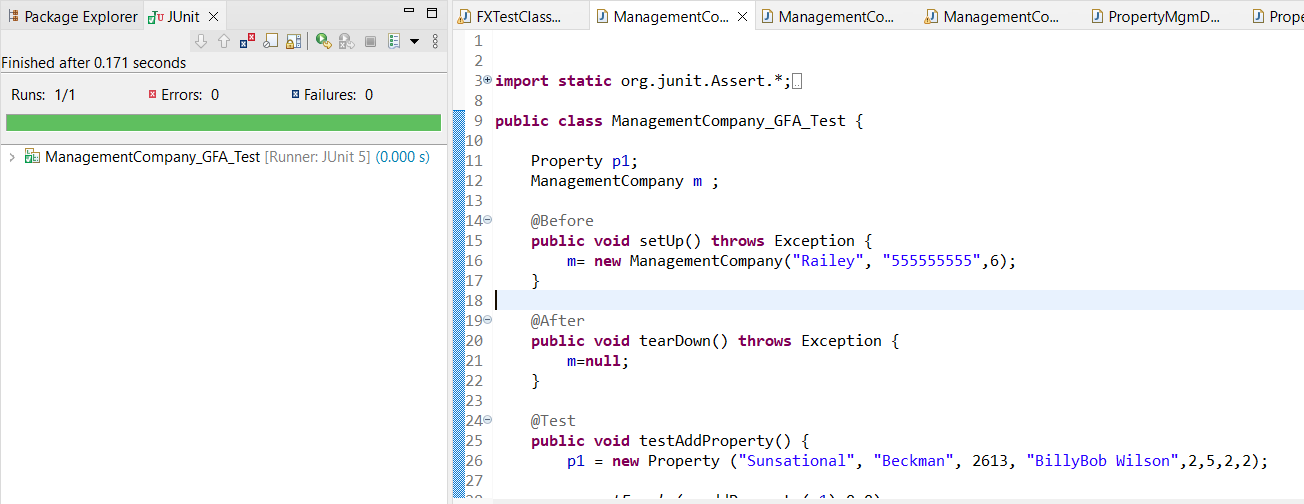


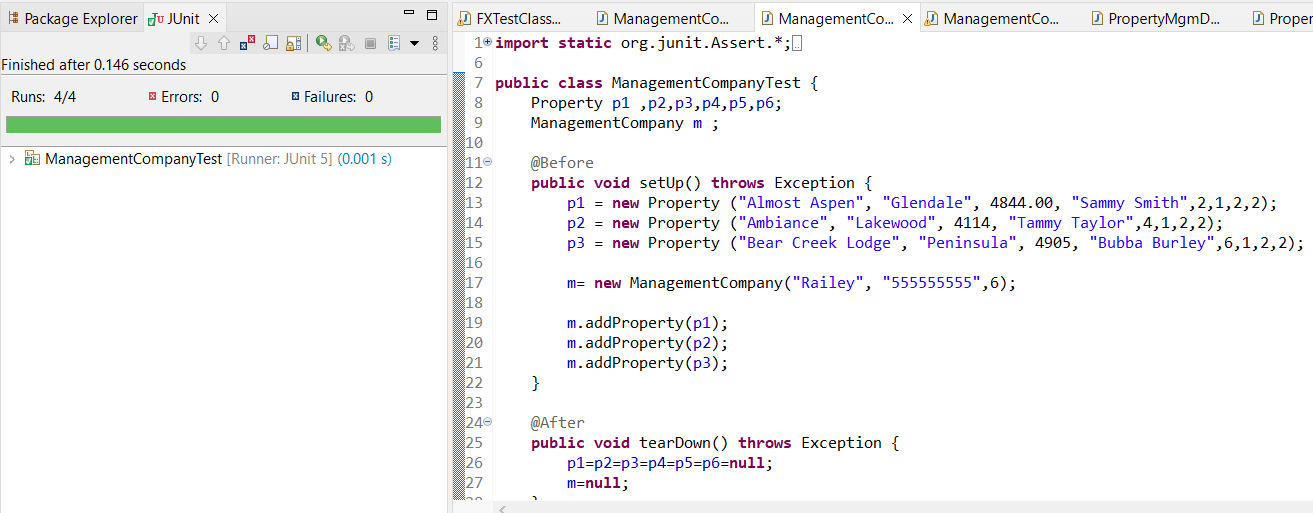
(Fourth)



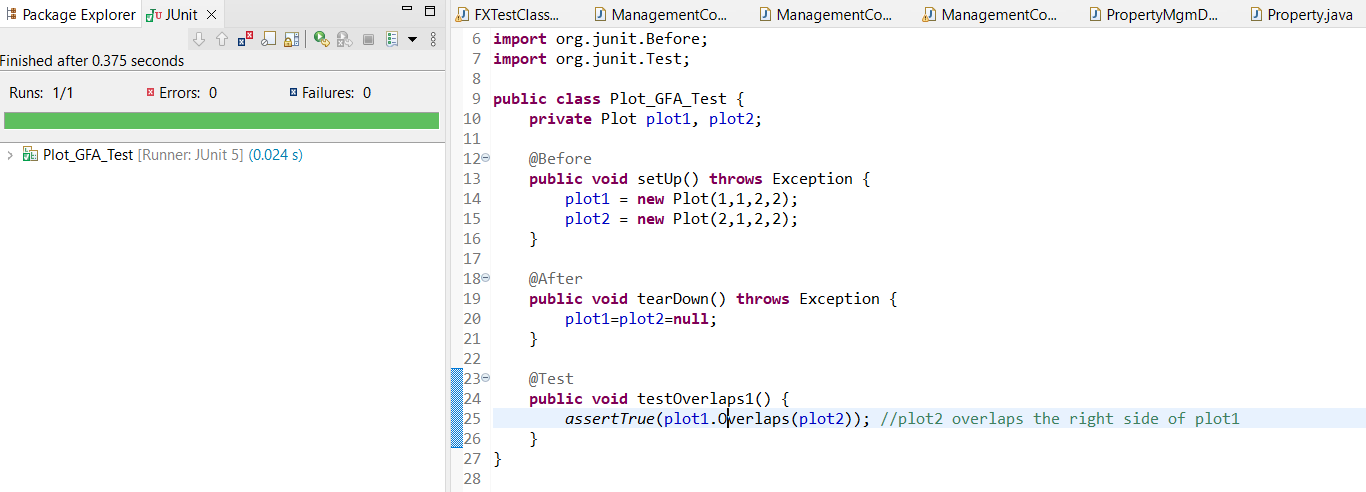


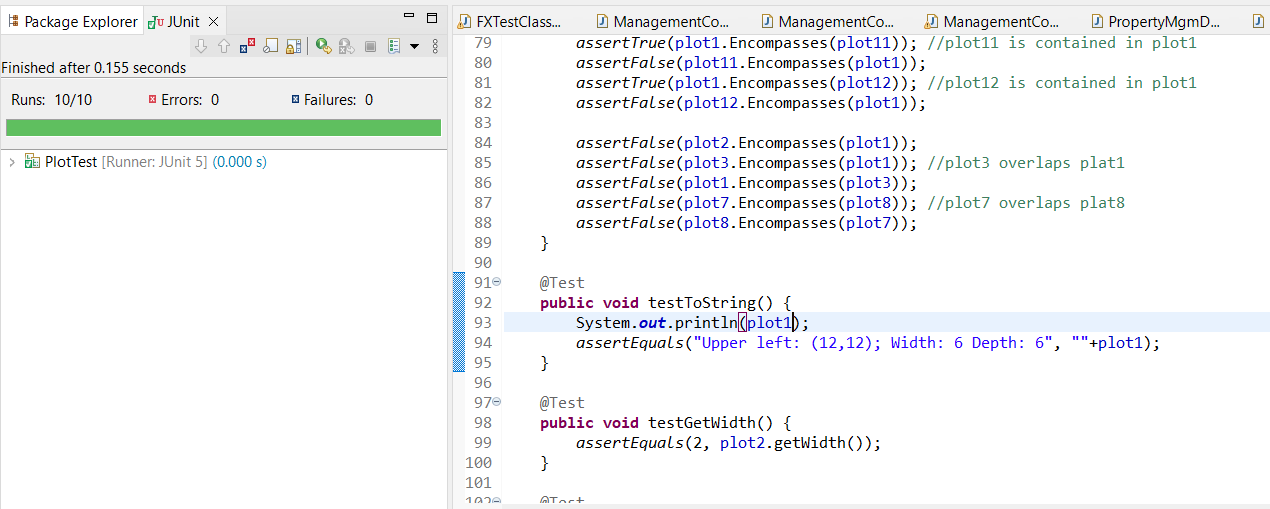
J-Unit:











Git-Hub:

Lessons Learned:

Learning Experience: highlight your lessons learned and learning experience from working on this project.

What have you learned? I learned how to how to solve a really interesting problem, kinda like a puzzle really, I had to come up with a lot of really clever work arounds in order to solve the unique problems involving checking if two pieces of land were overlapping eachother or not.

What did you struggle with? As per usual I struggled with getting this work done on time. Lately my life has been pretty hectic with lots of personal drama, and while I try to not let it slow me down, sometimes it’s just really hard to get up and do work.

What will you do differently on your next project?

Next time I’ll try starting as soon as I can for what it’s worth, but I don’t know how much of a difference that will make if I’m being honest.

Include what parts of the project you were successful at, and what parts (if any) you were not successful at.

For whatever reason, I cannot get the Javadocs to generate, like at all. I think it’s my java version?- I don’t know how that affects anything, but I think that’s what the complier error refered to. I also can’t get the images to load up on the JavaFX class, even though I included the folder in the src folder... I guess I just simply, really don’t know how to get it working other then putting the folder in there.

Check-List:

Assignment 4 Check List (include Yes/No or N/A for each item)

|  |  |  |  |
| --- | --- | --- | --- |
| **#** |  | **Y/N or N/A** | **Comments** |
|  | **Assignment files:** |  |  |
|  | * FirstInitialLastName\_ Assignment 4\_Moss.zip |  |  |
|  | * FirstInitialLastName\_Assignment4\_Complete.zip |  |  |
|  | **Program compiles** | **Y** |  |
|  | **Program runs with desired outputs related to a Test Plan** | **Y** |  |
|  | **Documentation file:** |  |  |
|  | * Comprehensive Test Plan | **Y** |  |
|  | * Screenshots for each Junit Test | **Y** |  |
|  | * Screenshots for each Test case listed in the Test Plan | **Y** |  |
|  | * Screenshots of your GitHub account with submitted Assignment# (if required) |  |  |
|  | * UML Diagram | **Y** |  |
|  | * Algorithms/Pseudocode | **Y** |  |
|  | * Flowchart (if required) | **n/a** |  |
|  | * Lessons Learned | **y** |  |
|  | * Checklist is completed and included in the Documentation | Y |  |