COMP7480 Project Report

16431669 SUN Xiaomeng

Project Description

This project is to implement a system for online house renting. The project contains two parts, the web system and the mobile application for iPhone. In the web system, I use html+css+js as framework and for the javascript part, I use sails.js as the main library to implement the whole web system. In the mobile app, I use the Appcelerator Titanium framework to develop the whole iOS application. Both the two parts use the MVC framework. In the model part, I choose ORM for mapping. In the view part, I use Angular.js and Bootstrap for better user interface and in the controller part, I use express.js for data binding.

Web System

Functions:

1. User login/logout
2. User registration
3. CRUD operations for houses
4. Declare/undeclare interest for houses,
5. Check interested users for the user's houses
6. Admin operations.

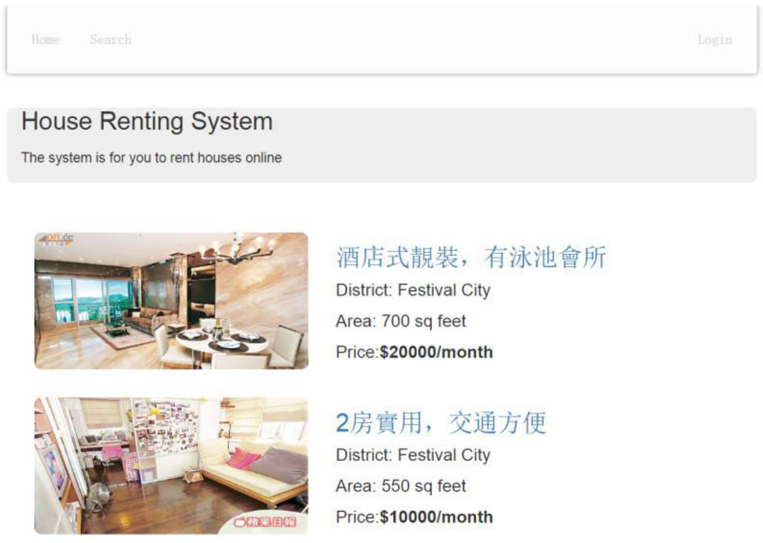
Bonuses:

Most of my project is similar to the given example, but there are some differences and some bonuses.

1. User Interface

(1) The homepage is different from the example. I choose to show one house at

one column rather than two houses in the example.



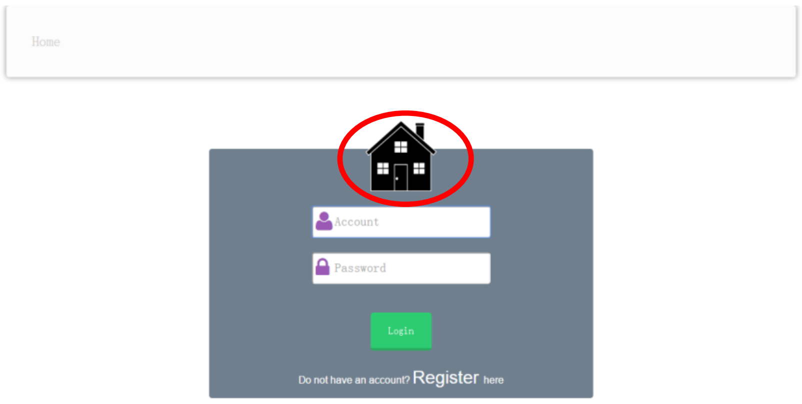
House Renting System

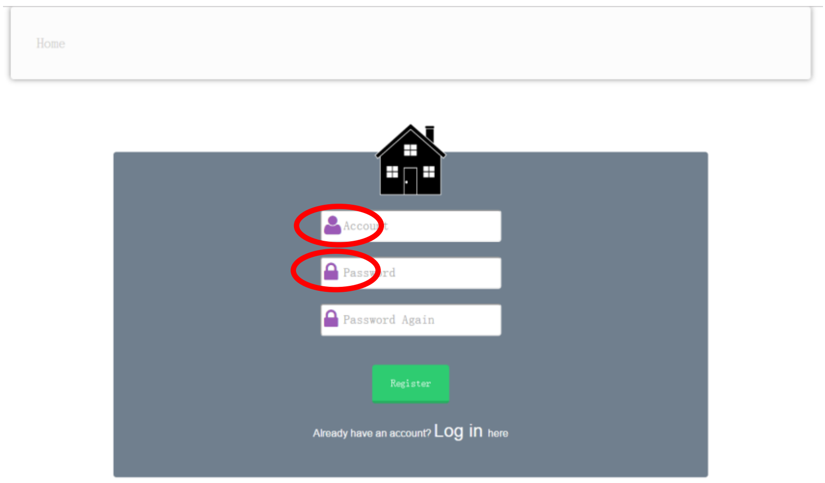
The system is for you to rent houses online

酒店式靚裝，有泳池會所
District: Festival City
Area: 700 sq feet
Price:**$20000/month**

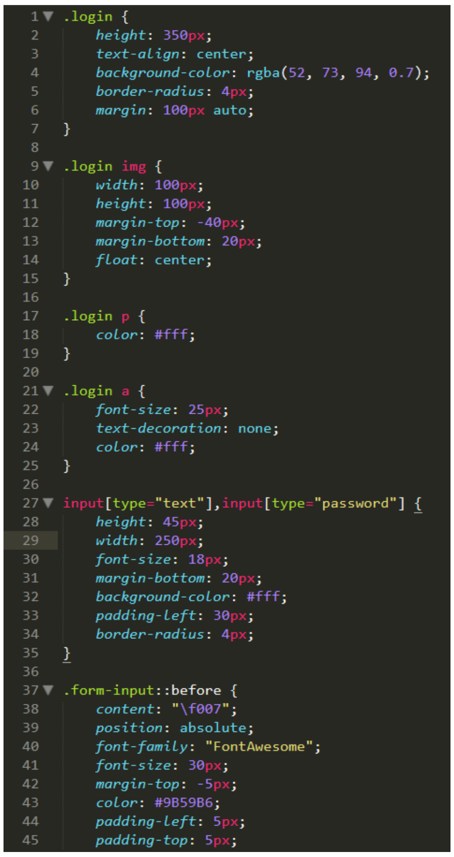
2房實用，交通方便
District: Festival City
Area: 550 sq feet
Price:**$10000/month**

(2) The login and register page is totally developed by myself so they look very different. I use some icons in these pages.

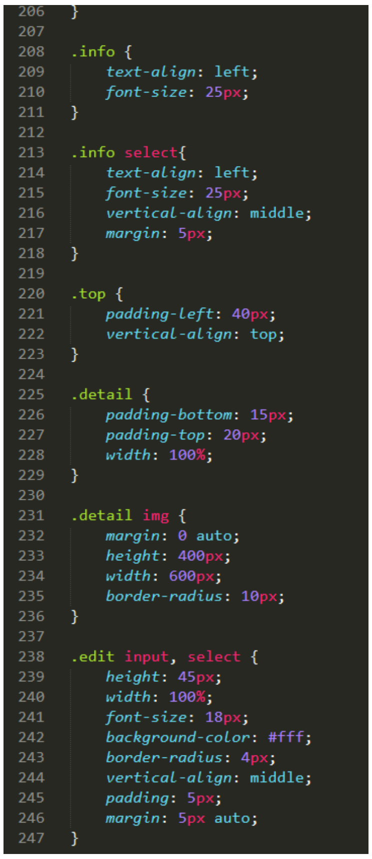




To achieve the effect above, I write a css file with many css statements. Also, in the whole web system, I write many css statements to make the interface look better. Here is the part of the css file.

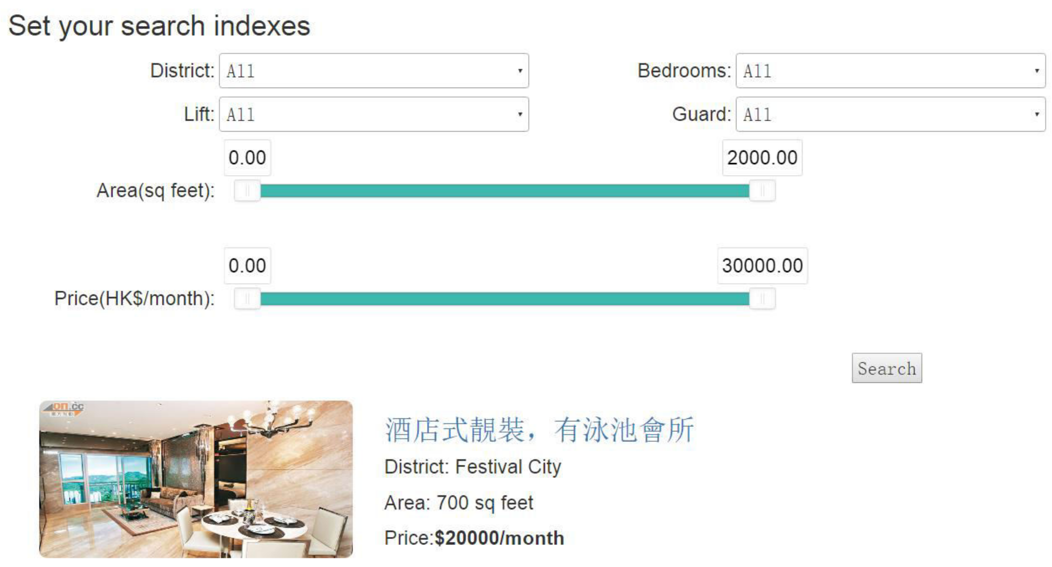


```css
.login {
    height: 350px;
    text-align: center;
    background-color: rgba(52, 73, 94, 0.7);
    border-radius: 4px;
    margin: 100px auto;
}

.login img {
    width: 100px;
    height: 100px;
    margin-top: -40px;
    margin-bottom: 20px;
    float: center;
}

.login p {
    color: #fff;
}

.login a {
    font-size: 25px;
    text-decoration: none;
    color: #fff;
}

input[type="text"],input[type="password"] {
    height: 45px;
    width: 250px;
    font-size: 18px;
    margin-bottom: 20px;
    background-color: #fff;
    padding-left: 30px;
    border-radius: 4px;
}

.form-input::before {
    content: "\f007";
    position: absolute;
    font-family: "FontAwesome";
    font-size: 30px;
    margin-top: -5px;
    color: #9B59B6;
    padding-left: 5px;
    padding-top: 5px;
```



```css
}

.info {
    text-align: left;
    font-size: 25px;
}

.info select{
    text-align: left;
    font-size: 25px;
    vertical-align: middle;
    margin: 5px;
}

.top {
    padding-left: 40px;
    vertical-align: top;
}

.detail {
    padding-bottom: 15px;
    padding-top: 20px;
    width: 100%;
}

.detail img {
    margin: 0 auto;
    height: 400px;
    width: 600px;
    border-radius: 10px;
}

.edit input, select {
    height: 45px;
    width: 100%;
    font-size: 18px;
    background-color: #fff;
    border-radius: 4px;
    vertical-align: middle;
    padding: 5px;
    margin: 5px auto;
}
```

(3) In the search page, the layout of the whole page is different.



Set your search indexes

District: All　　Bedrooms: All

Lift: All　　Guard: All

Area(sq feet): 0.00 – 2000.00

Price(HK$/month): 0.00 – 30000.00

酒店式靚裝，有泳池會所

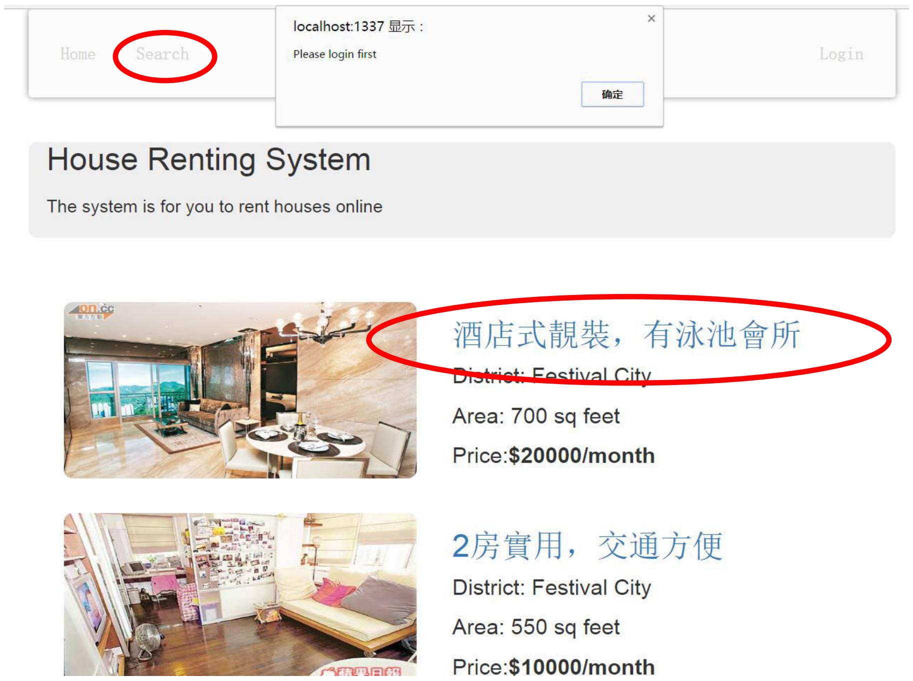
District: Festival City

Area: 700 sq feet

Price:**$20000/month**

2. Logic

In my web system, I design that the visitors can only browse the houses in the homepage. If the user wants to search for more houses, he should login first. It is different from the example in which the visitor is required to login only when he wants to declare interest for the house.



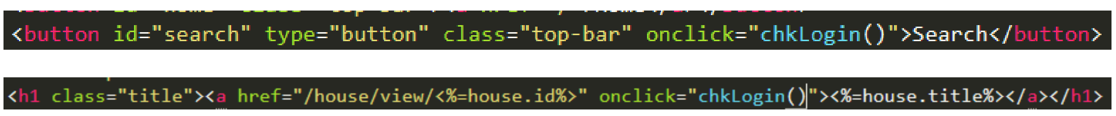
House Renting System

The system is for you to rent houses online

酒店式靚裝，有泳池會所
District: Festival City
Area: 700 sq feet
Price:**$20000/month**

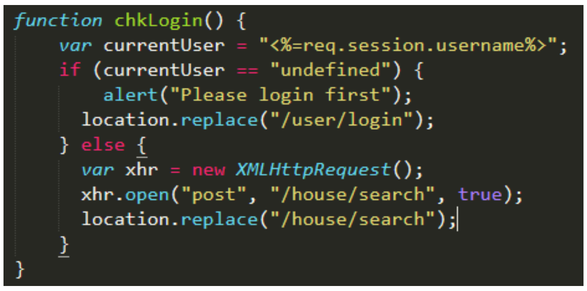
2房實用，交通方便
District: Festival City
Area: 550 sq feet
Price:**$10000/month**

The code to implement it is as follows:



```
<button id="search" type="button" class="top-bar" onclick="chkLogin()">Search</button>
```

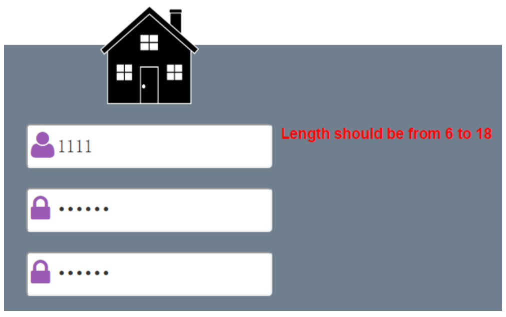
```
<h1 class="title"><a href="/house/view/<%=house.id%>" onclick="chkLogin()"><%=house.title%></a></h1>
```



```
function chkLogin() {
    var currentUser = "<%=req.session.username%>";
    if (currentUser == "undefined") {
        alert("Please login first");
      location.replace("/user/login");
    } else {
      var xhr = new XMLHttpRequest();
      xhr.open("post", "/house/search", true);
      location.replace("/house/search");
    }
}
```

3. Functionality

(1) In the register page, I add the function of user and password validation.

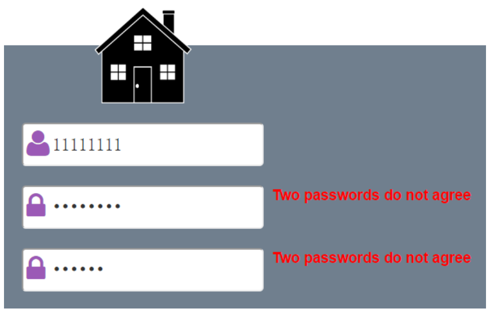


1111

••••••

••••••

Length should be from 6 to 18



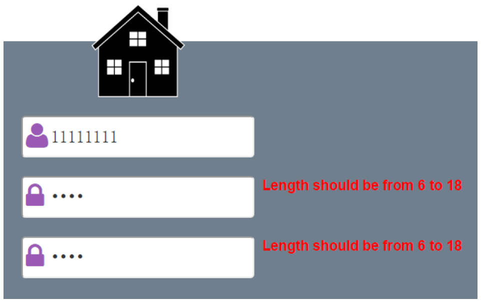
11111111

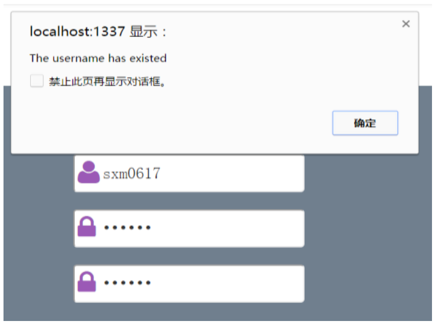
••••••••

••••••

Two passwords do not agree

Two passwords do not agree

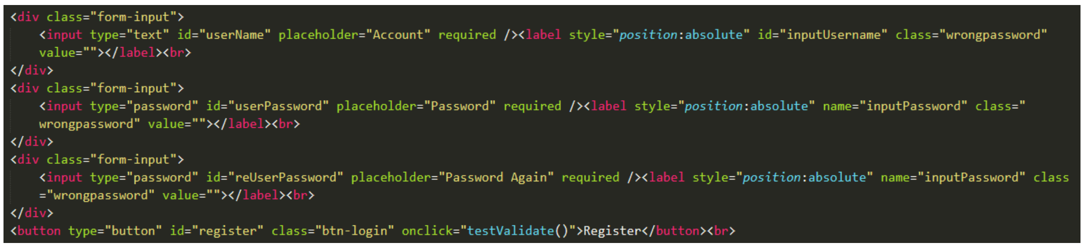


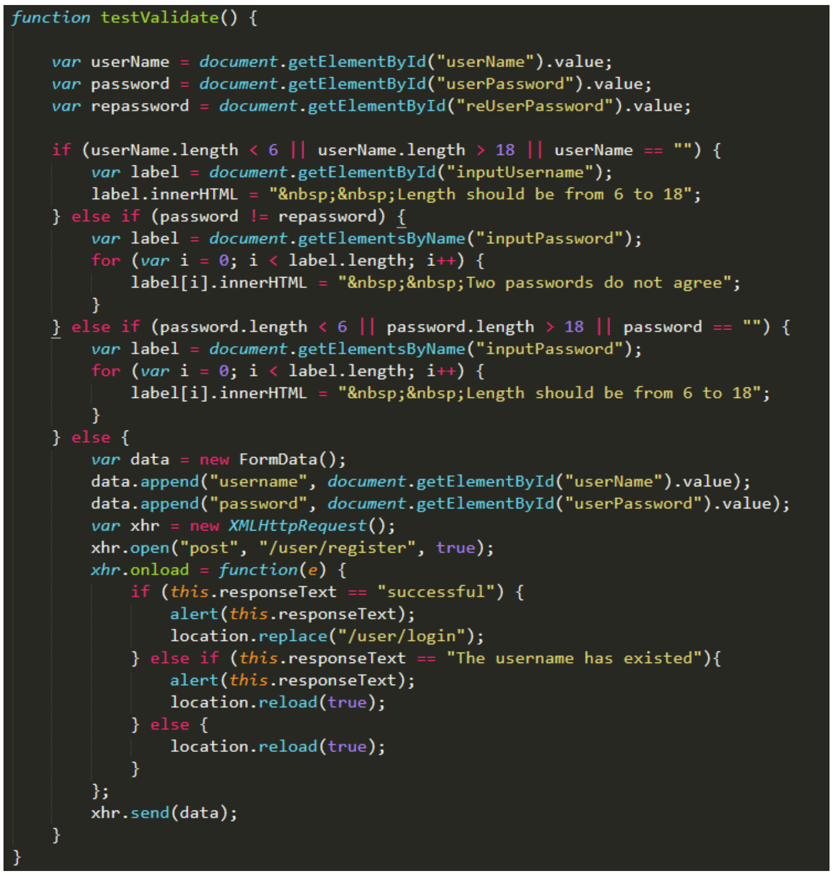

localhost:1337 显示：

The username has existed

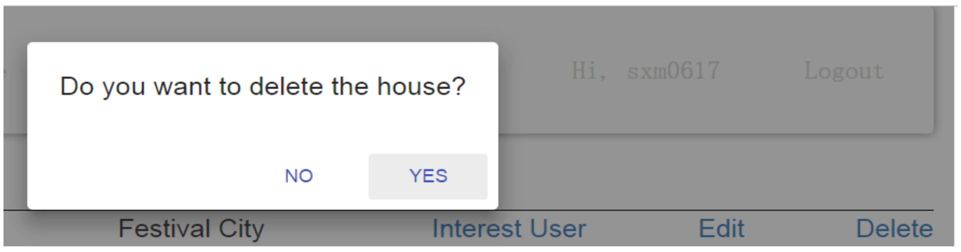
禁止此页再显示对话框。

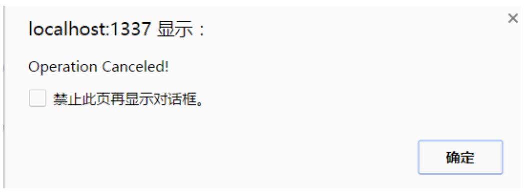
The code to achieve this is as follows:


```
<div class="form-input">
    <input type="text" id="userName" placeholder="Account" required /><label style="position:absolute" id="inputUsername" class="wrongpassword"
    value=""></label><br>
</div>
<div class="form-input">
    <input type="password" id="userPassword" placeholder="Password" required /><label style="position:absolute" name="inputPassword" class="
    wrongpassword" value=""></label><br>
</div>
<div class="form-input">
    <input type="password" id="reUserPassword" placeholder="Password Again" required /><label style="position:absolute" name="inputPassword" class
    ="wrongpassword" value=""></label><br>
</div>
<button type="button" id="register" class="btn-login" onclick="testValidate()">Register</button><br>
```


```
function testValidate() {

    var userName = document.getElementById("userName").value;
    var password = document.getElementById("userPassword").value;
    var repassword = document.getElementById("reUserPassword").value;

    if (userName.length < 6 || userName.length > 18 || userName == "") {
        var label = document.getElementById("inputUsername");
        label.innerHTML = "&nbsp;&nbsp;Length should be from 6 to 18";
    } else if (password != repassword) {
        var label = document.getElementsByName("inputPassword");
        for (var i = 0; i < label.length; i++) {
            label[i].innerHTML = "&nbsp;&nbsp;Two passwords do not agree";
        }
    } else if (password.length < 6 || password.length > 18 || password == "") {
        var label = document.getElementsByName("inputPassword");
        for (var i = 0; i < label.length; i++) {
            label[i].innerHTML = "&nbsp;&nbsp;Length should be from 6 to 18";
        }
    } else {
        var data = new FormData();
        data.append("username", document.getElementById("userName").value);
        data.append("password", document.getElementById("userPassword").value);
        var xhr = new XMLHttpRequest();
        xhr.open("post", "/user/register", true);
        xhr.onload = function(e) {
            if (this.responseText == "successful") {
                alert(this.responseText);
                location.replace("/user/login");
            } else if (this.responseText == "The username has existed"){
                alert(this.responseText);
                location.reload(true);
            } else {
                location.reload(true);
            }
        };
        xhr.send(data);
    }
}
```

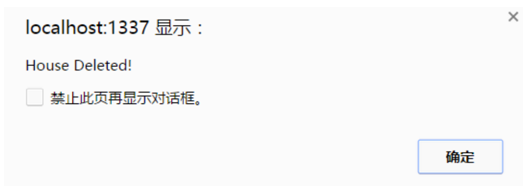
(2) For some operations, I add some confirm windows by Angular.js.


Do you want to delete the house?


localhost:1337 显示：

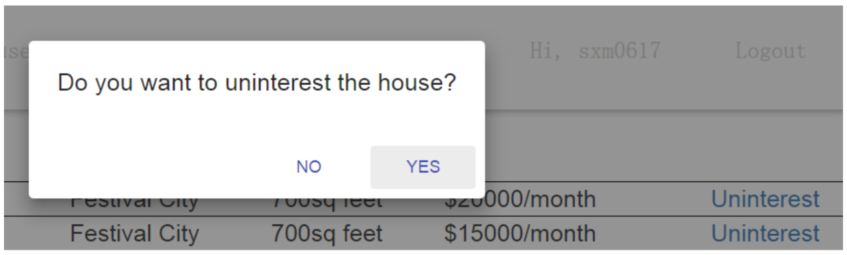
Operation Canceled!

禁止此页再显示对话框。


localhost:1337 显示：

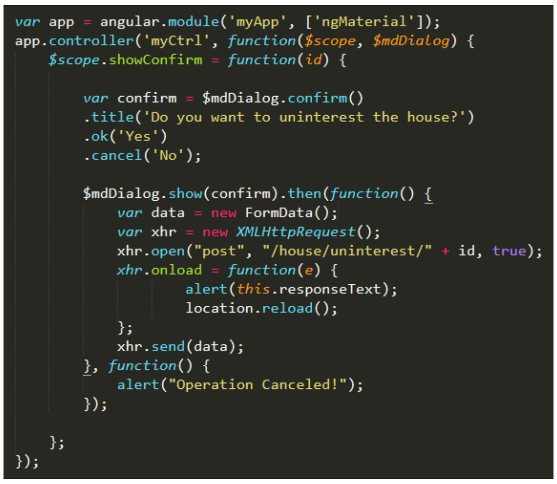
House Deleted!

禁止此页再显示对话框。

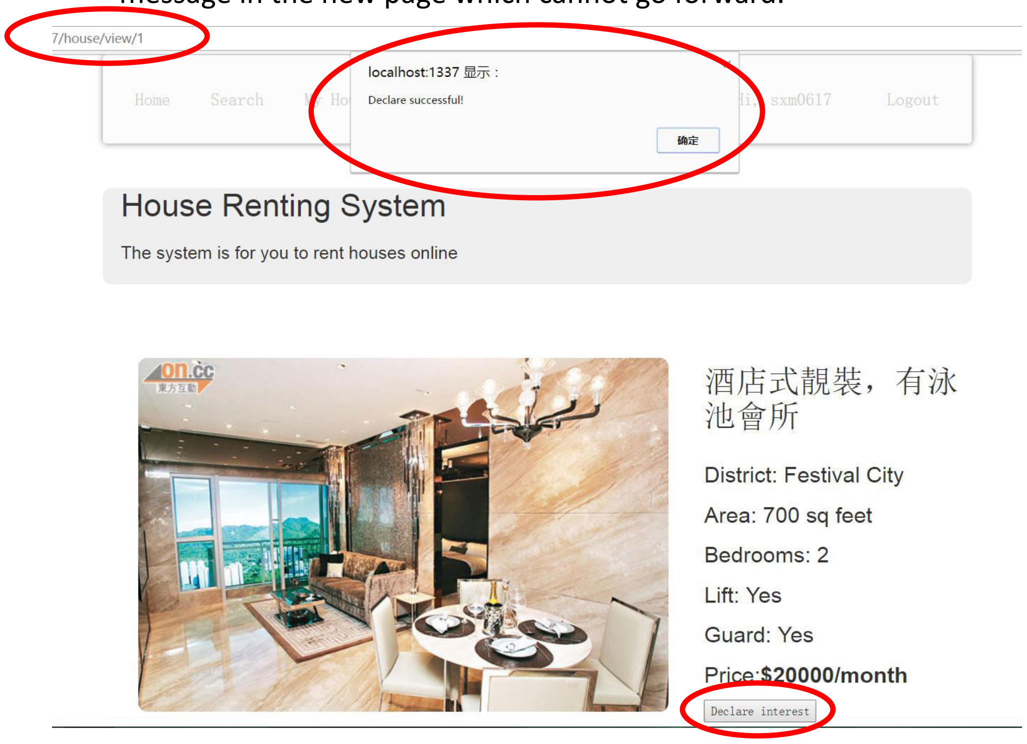

Do you want to uninterest the house?

Festival City	700sq feet	$20000/month	Uninterest
Festival City	700sq feet	$15000/month	Uninterest

The code is the following:


```
var app = angular.module('myApp', ['ngMaterial']);
app.controller('myCtrl', function($scope, $mdDialog) {
    $scope.showConfirm = function(id) {

        var confirm = $mdDialog.confirm()
        .title('Do you want to uninterest the house?')
        .ok('Yes')
        .cancel('No');

        $mdDialog.show(confirm).then(function() {
            var data = new FormData();
            var xhr = new XMLHttpRequest();
            xhr.open("post", "/house/uninterest/" + id, true);
            xhr.onload = function(e) {
                    alert(this.responseText);
                    location.reload();
            };
            xhr.send(data);
        }, function() {
            alert("Operation Canceled!");
        });

    };
});
```

(3) I add some and alert boxes and page jumping instead of just showing a message in the new page which cannot go forward.



localhost:1337 显示：

Declare successful!

House Renting System

The system is for you to rent houses online

酒店式靚裝，有泳池會所

District: Festival City

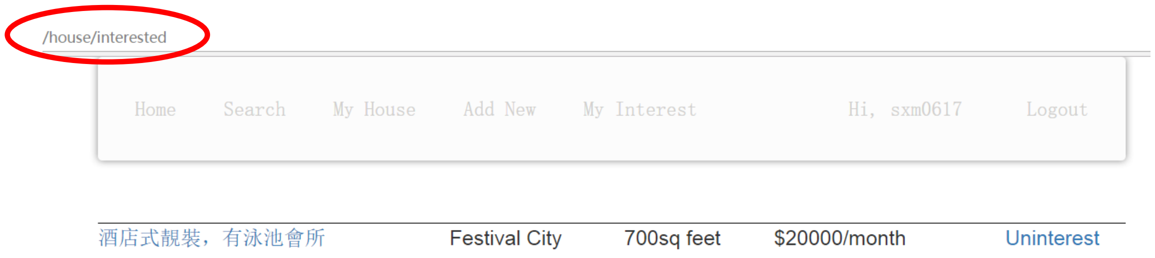
Area: 700 sq feet

Bedrooms: 2

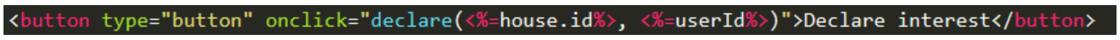
Lift: Yes

Guard: Yes

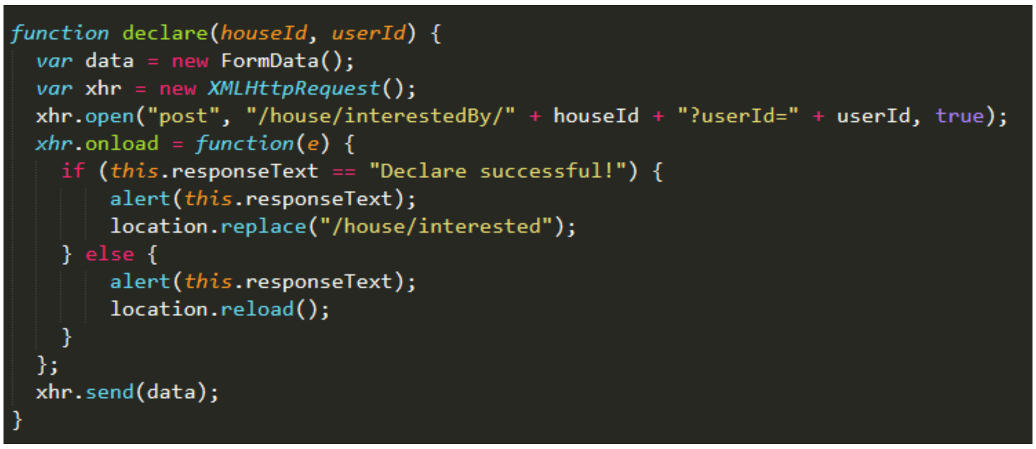
Price:$20000/month



The code is designed like this:



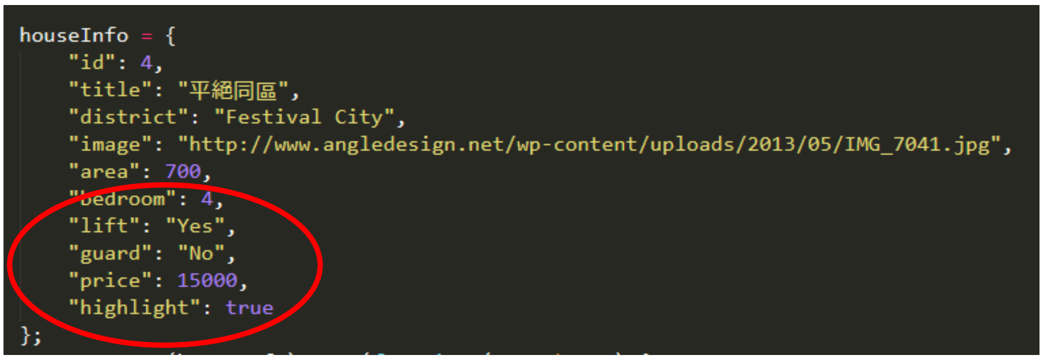
```
<button type="button" onclick="declare(<%=house.id%>, <%=userId%>)">Declare interest</button>
```



```
function declare(houseId, userId) {
  var data = new FormData();
  var xhr = new XMLHttpRequest();
  xhr.open("post", "/house/interestedBy/" + houseId + "?userId=" + userId, true);
  xhr.onload = function(e) {
    if (this.responseText == "Declare successful!") {
        alert(this.responseText);
        location.replace("/house/interested");
    } else {
        alert(this.responseText);
        location.reload();
    }
  };
  xhr.send(data);
}
```

4. Others

To fulfill the content, I add more attributes for house information and I choose to show only some of them on the homepage.



```
houseInfo = {
    "id": 4,
    "title": "平絕同區",
    "district": "Festival City",
    "image": "http://www.angledesign.net/wp-content/uploads/2013/05/IMG_7041.jpg",
    "area": 700,
    "bedroom": 4,
    "lift": "Yes",
    "guard": "No",
    "price": 15000,
    "highlight": true
};
```

Bugs:

1. The pagination function sometimes has problems on the search page.
2. In the edit page, some of the information which are in the selection form are not the exact value stored in the model.

Mobile App

Functions:

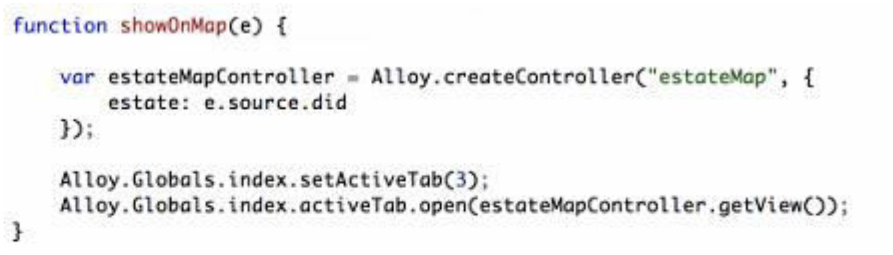
1. User login/logout
2. House browse
3. House filter
4. Find house on the map
5. Declare/Undeclare interest

Bonuses:

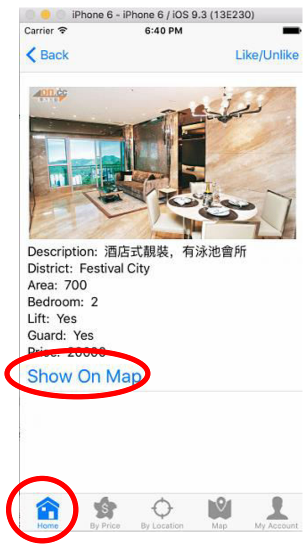
Most of my project is similar to the given example, but there are some differences and some bonuses.

1. Tab jumping

   I add some jump function between tabs so that it is easy to see which tab implements which functions. For example, once it is required to see the house on the map, it will jump to tab 4.



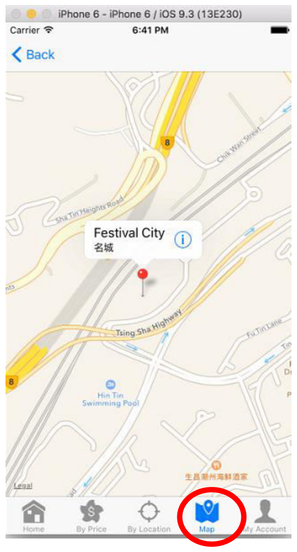
```
function showOnMap(e) {

    var estateMapController = Alloy.createController("estateMap", {
        estate: e.source.did
    });

    Alloy.Globals.index.setActiveTab(3);
    Alloy.Globals.index.activeTab.open(estateMapController.getView());
}
```



iPhone 6 - iPhone 6 / iOS 9.3 (13E230)
Carrier 6:40 PM
Back Like/Unlike

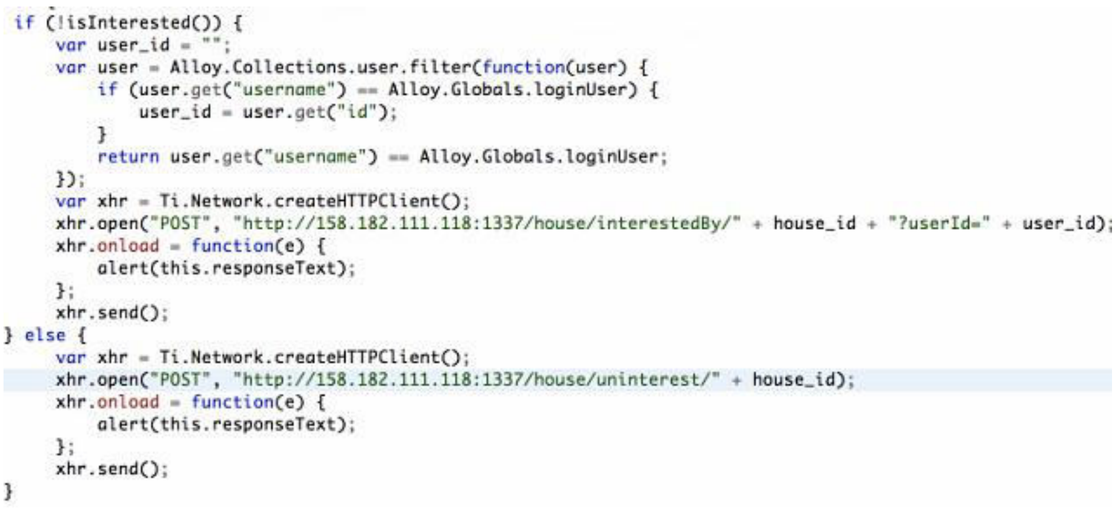
Description: 酒店式靚裝，有泳池會所
District: Festival City
Area: 700
Bedroom: 2
Lift: Yes
Guard: Yes
Price: 20000
Show On Map

Home By Price By Location Map My Account



2. Like/Unlike function

   Instead of making two functions, I combine them into one function and every time the user click this button, the system will check whether the user has ever liked it before. If not, the user will be considered liking the house, otherwise the user will be considered not liking the house.



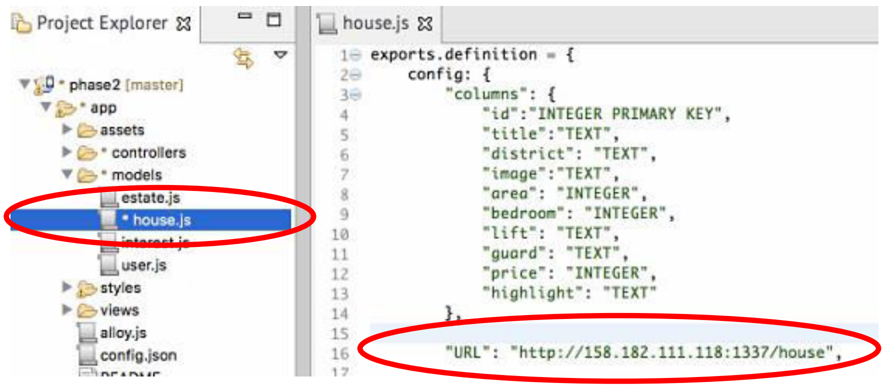
```
if (!isInterested()) {
    var user_id = "";
    var user = Alloy.Collections.user.filter(function(user) {
        if (user.get("username") == Alloy.Globals.loginUser) {
            user_id = user.get("id");
        }
        return user.get("username") == Alloy.Globals.loginUser;
    });
    var xhr = Ti.Network.createHTTPClient();
    xhr.open("POST", "http://158.182.111.118:1337/house/interestedBy/" + house_id + "?userId=" + user_id);
    xhr.onload = function(e) {
        alert(this.responseText);
    };
    xhr.send();
} else {
    var xhr = Ti.Network.createHTTPClient();
    xhr.open("POST", "http://158.182.111.118:1337/house/uninterest/" + house_id);
    xhr.onload = function(e) {
        alert(this.responseText);
    };
    xhr.send();
}
```

Bugs:

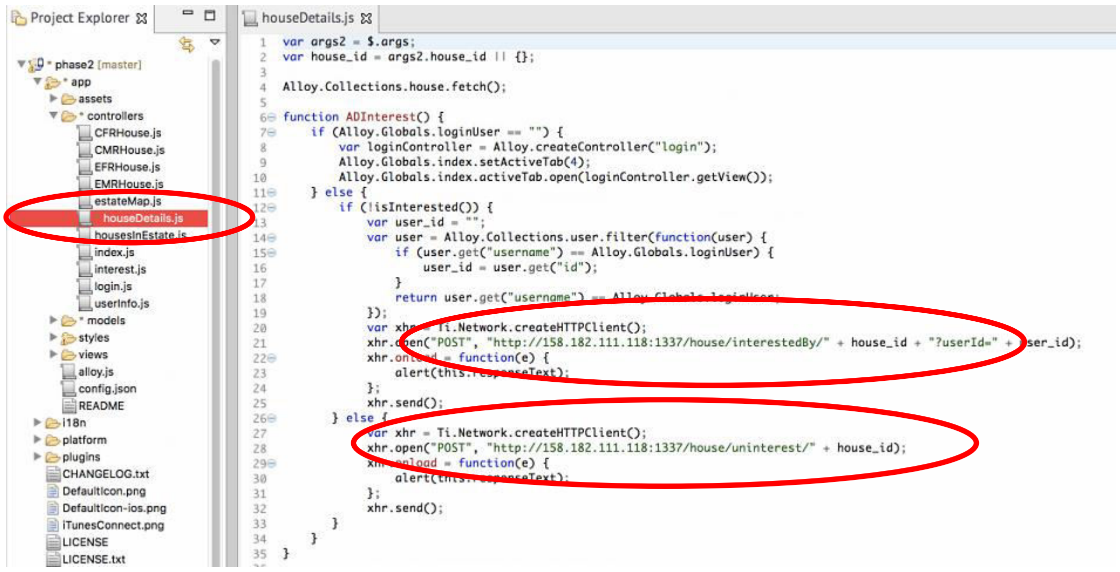
The mobile app cannot update the information in real time, only to restart the whole app. For example, if one user delete one house, the app will still show the house.

Other Information

To run the project on your computer, the IP address should be changed to your own computer's IP address. There are the places using IP address:



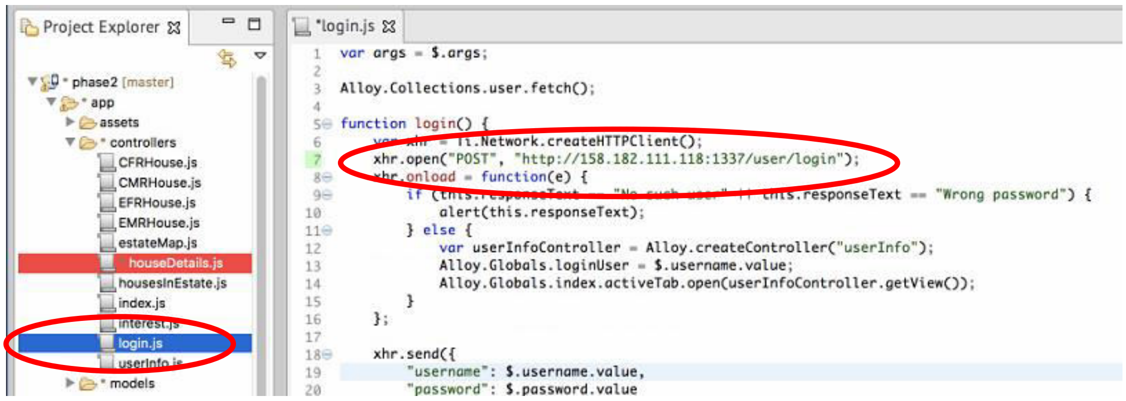
```
exports.definition = {
    config: {
        "columns": {
            "id":"INTEGER PRIMARY KEY",
            "title":"TEXT",
            "district": "TEXT",
            "image":"TEXT",
            "area": "INTEGER",
            "bedroom": "INTEGER",
            "lift": "TEXT",
            "guard": "TEXT",
            "price": "INTEGER",
            "highlight": "TEXT"
        },

        "URL": "http://158.182.111.118:1337/house",
```



```
Project Explorer
▼ phase2 [master]
  ▼ app
    ▶ assets
    ▼ controllers
        CFRHouse.js
        CMRHouse.js
        EFRHouse.js
        EMRHouse.js
        estateMap.js
        houseDetails.js
        housesInEstate.js
        index.js
        interest.js
        login.js
        userInfo.js
    ▶ models
    ▶ styles
    ▶ views
      alloy.js
      config.json
      README
  ▶ i18n
  ▶ platform
  ▶ plugins
    CHANGELOG.txt
    DefaultIcon.png
    DefaultIcon-ios.png
    iTunesConnect.png
    LICENSE
    LICENSE.txt
```

houseDetails.js

```
var args2 = $.args;
var house_id = args2.house_id || {};

Alloy.Collections.house.fetch();

function ADInterest() {
    if (Alloy.Globals.loginUser == "") {
        var loginController = Alloy.createController("login");
        Alloy.Globals.index.setActiveTab(4);
        Alloy.Globals.index.activeTab.open(loginController.getView());
    } else {
        if (!isInterested()) {
            var user_id = "";
            var user = Alloy.Collections.user.filter(function(user) {
                if (user.get("username") == Alloy.Globals.loginUser) {
                    user_id = user.get("id");
                }
                return user.get("username") == Alloy.Globals.loginUser;
            });
            var xhr = Ti.Network.createHTTPClient();
            xhr.open("POST", "http://158.182.111.118:1337/house/interestedBy/" + house_id + "?userId=" + user_id);
            xhr.onload = function(e) {
                alert(this.responseText);
            };
            xhr.send();
        } else {
            var xhr = Ti.Network.createHTTPClient();
            xhr.open("POST", "http://158.182.111.118:1337/house/uninterest/" + house_id);
            xhr.onload = function(e) {
                alert(this.responseText);
            };
            xhr.send();
        }
    }
}
```



```
Project Explorer
▼ phase2 [master]
  ▼ app
    ▶ assets
    ▼ controllers
        CFRHouse.js
        CMRHouse.js
        EFRHouse.js
        EMRHouse.js
        estateMap.js
        houseDetails.js
        housesInEstate.js
        index.js
        interest.js
        login.js
        userInfo.js
    ▶ models
```

*login.js

```
var args = $.args;

Alloy.Collections.user.fetch();

function login() {
    var xhr = Ti.Network.createHTTPClient();
    xhr.open("POST", "http://158.182.111.118:1337/user/login");
    xhr.onload = function(e) {
        if (this.responseText == "No such user" || this.responseText == "Wrong password") {
            alert(this.responseText);
        } else {
            var userInfoController = Alloy.createController("userInfo");
            Alloy.Globals.loginUser = $.username.value;
            Alloy.Globals.index.activeTab.open(userInfoController.getView());
        }
    };

    xhr.send({
        "username": $.username.value,
        "password": $.password.value
```