

Web and Mobile App for Event System

Mini Project Features Report

1st Dec, 2015

MENG Lingnan

15432548

Introduction

In this Event Management System, there are three types of registered users (**admin, staff and member**). Each role of these three user groups has some similar functions and different authorities.

In this report, I will introduce the features in three roles respectively.

Data Structure

In web part, we have two models named `User` and `WebNews` (representing events) associated with each other. User can register the events. Event can be registered by

```
[ { "username": "Sally",
  "password": "123456",
  "role": "admin",
  "id": 1,
  "createdAt": "2015-11-25T04:35:12.225Z",
  "updatedAt": "2015-11-29T03:02:24.274Z"},

{
  "username": "Martin",
  "password": "123456",
  "role": "staff",
  "dept": "Computer Science",
  "id": 2,
  "createdAt": "2015-11-25T04:35:12.226Z",
  "updatedAt": "2015-11-28T12:53:04.182Z"},

{
  "username": "Cindy",
  "password": "123456",
  "role": "student",
  "dept": "Computer Science",
  "id": 3,
  "createdAt": "2015-11-25T04:35:12.227Z",
  "updatedAt": "2015-11-25T09:29:58.875Z"}]
```

```
[ {
  "name": "Singing Contest",
  "shortDes": "一年一度既HKBU Singing Contest又到喇！身為浸大人既你點可能錯過！？無論你係咩系、Bachelor",
  "fullDes": "一年一度既HKBU Singing Contest又到喇！身為浸大人既你點可能錯過！？無論你係咩系、Bachelor",
  "image": "https://scontent-hkg3-1.xx.fbcdn.net/hphotos-xfa1/v/t1.0-9/537765_488319141230614_203",
  "venue": "AC Hall",
  "organizer": "Music Society",
  "date": "2015-12-12",
  "startTime": "19:30",
  "endTime": "23:30",
  "quota": "10",
  "highlighted": "checked",
  "id": 1,
  "createdAt": "2015-11-25T04:35:12.231Z",
  "updatedAt": "2015-11-25T04:35:12.231Z"
} ]
```

Log in

Before users log in the system, the menu will be like above. If userInfo in session is null, then `login` button will show up and any function needing log in will not show up in the menu to prevent errors.

```
<ul class="nav navbar-nav navbar-right">
  <li class="dropdown">
    <% if(session.userInfo!=null){ %>
      <a href="#" class="dropdown-toggle" data-toggle="dropdown"
          role="button" aria-haspopup="true" aria-expanded="false">
        Hello, <%=session.userInfo.username%> <span class="caret"></span></a>
      <ul class="dropdown-menu">
        <li><a href="/../user/showRegister/">Registered Events</a></li>
        <li role="separator" class="divider"></li>
        <li><a href="/../user/logout">logout</a></li>
      </ul>
    </li>
    <% }else { %>
      <li><a href="/../user/login">Login</a></li>
    <% } %>
  </ul>
```

If the users make mistakes when log in the system, there will be alerts like below.

Login

Username does not exist.

Password

Password is wrong.

Sign in

No Account? Register here!

View and Register Event

When you click on one event, you will come to the detail page of the event. In this page, you can view the event and register it. The 'Register' button will be different subject to the quota. If you have not logged in, this button will redirect you to the login page. If you have logged in and remaining quota is more than 0, clicking on this button will register the event for you.

The event will be marked with a star if it is highlighted, or will be marked with an empty star.

★Singing Contest



一年一度既HKBU Singing Contest又到喇！身為浸大人既你點可能錯過！？無論你係咩系、Bachelor/AD Promgramme，我地都歡迎你報名！身邊有唱得之人就梗係要推薦佢黎喇！主辦單位：音樂學會 報名日期：4/2-8/2 (12-6pm) 報名地點：Main Podium 詳情可閱 Poster 或去到報名Counter查詢

Organizer: Music Society

Date: 2015-12-12

Venue: AC Hall

Start Time: 19:30

Quota: 10

End Time: 23:30

Remain Quota: 5

Register

If the remaining quota is equal to 0, you cannot register it any more. The button will become disabled and show message **Full** on it.

★Culture and Politics in European Cinema



Michael Balcon and Twentieth Century British Cinema: Art, Industry and National Culture. Dr. Mark Hampton.

Organizer: Government and International Studies

Date: 2015-11-11

Start Time: 19:00

Venue: SWT501

End Time: 21:00

Quota: 5

Full

Remain Quota: 0

If you have registered this event, then the button will become a [UnReg](#) button. You can cancel the registration on this event by clicking on it.

★Workshop on Design for Sustainability



Aspects to focus on Design green partnership agreement between Kyoto and Hong Kong Ex. Low-carbon society, waste management, etc. Discover cultural differences Forcibly ideate green solutions using the cultural differences Extend the ideas by combining participants' research topics.

Organizer: Music Society

Date: 2015-12-12

Venue: AC Hall

Start Time: 19:30

Quota: 10

End Time: 23:30

Remain Quota: 5

[UnReg it](#)

The html code is shown below. The link of button is determined in this page subject to the value of `reg`.

```

<% if(reg=="UnReg"){%
<a href="/../User/removeRegister?pid=<%=WebNews.id%>" class="btn btn-default navbar-btn">UnReg it</a>
<% } else if(WebNews.cquota<=0) { %>
<a disabled class="btn btn-default navbar-btn">Full</a>
<%} else {%
<a href="/../user/addRegister/?pid=<%=WebNews.id%>" class="btn btn-default navbar-btn">Register
</a>
<%}%

```

The value of `reg` should be set in `UserController.js`. It is a flag recording the status of User and Events.

```

view: function (req, res) {
    WebNews.findOne(req.params.id).exec( function(err, mod) {
        if (mod != null){
            var reg="";
            if(req.session.userInfo==null){
                WebNews.findOne(req.params.id).populateAll().exec(function (err, m) {
                    mod.cquota = mod.quota - m.registeredBy.length;
                    return res.view('WebNews/view', {'WebNews': mod, 'reg':'Reg'});
                });
            } else {
                WebNews.findOne(req.params.id).populateAll({id:req.session.userInfo.id})
                    .exec( function (err, model) {
                        WebNews.findOne(req.params.id).populateAll().exec(function (err, m) {
                            model.cquota = model.quota - m.registeredBy.length;
                            if(model.registeredBy.length!=0)
                                reg = "UnReg";
                            else reg = "Reg";
                            return res.view('WebNews/view', {'WebNews': model, 'reg':reg});
                        });
                });
            }
        } else
            return res.send("No such Event");
    });
}

```

Administrator

In this system, there is only one administrator with the rights to create, update and delete any events there. Administrator can also view all the registered members of an event.

Sally is administrator in this system. After Sally logging in the system, the login button will change into a dropdown menu containing `Registered Events` and `Logout` function.

In the `admin` function of the administrator, Sally can edit or delete all the events of any department. But for staff, they only can edit and delete the events belonging to their department.

```

admin: function(req, res) {
  if(req.session.userInfo.role=="admin"){
    WebNews.find().sort('highlighted desc').exec( function(err, WebNews) {
      return res.view('WebNews/admin', {'WebNews': WebNews});
    });
  }
  else{
    WebNews.find().sort('highlighted desc')
    .where({organizer: req.session.userInfo.dept})
    .exec( function(err, WebNews) {
      return res.view('WebNews/admin', {'WebNews': WebNews});
    });
  }
}

```

Staff

After the staff, for example, Martin, the teacher of Computer Sciende department logging in the system, he will see a menu like below.

Staff as well as administrator can create a new event. Staff can create events belonging to his department, while administrator can create events under all department, which is similar to Edit function.

```

<div class="form-group">
  <label class="control-label col-sm-2">Organizer: </label>
  <div class="col-sm-8">
    <select class="form-control" name="WebNews[organizer]">
      <% if(req.session.userInfo.role!="admin"){%>
        <option><%=req.session.userInfo.dept%></option>
      <%} else{%
        <option>Computer Science</option>
        <option>Music Society</option>
        <option>Government and International Studies</option>
        <option>Academic of Film</option>
      <%} %>
    </select>
  </div>
</div>

```

Student

After the student, for example, Cindy, a student of Computer Sciende department logging in the system, she will see a menu like below.

Before display the menu, the system will have a judgement what role the user is. If no login information or the user is student, the menu will only shows [Home](#) and [Search](#) button in navbar. If the user is not student which means he is administrator or staff, then the menu will add a [create](#) button and an [admin](#) button (for administrator) or

an `Edit` button (for staff).

```
<div class="collapse navbar-collapse" id="Home">
  <ul class="nav navbar-nav">
    <li><a href="/..../WebNews/search">Search</a></li>
    <% if(session.userInfo!=null){%
        if(session.userInfo.role != "student"){%>
      <li><a href="/..../WebNews/create">Create</a></li>
        <% if(session.userInfo.role == "admin"){%>
      <li><a href="/..../WebNews/admin">Admin</a></li>
        <% }else {%
      <li><a href="/..../WebNews/admin">Edit</a></li>
        <%}} } %>
    </ul>
</div>
```

Mobile App

The Mobile App use json provided in web as the database. A new model named `RegList` should be created to get the user's registered list.

```
exports.definition = {
  config: {
    "columns": {
      "id": "INTEGER PRIMARY KEY",
      "name": "text",
      "venue": "text",
      "date": "text",
      "startTime": "text",
      "endTime": "text",
      "organizer": "text",
      "quota": "text",
      "shortDes": "text",
      "fullDes": "text",
      "image": "text",
      "createdAt": "text",
      "updatedAt": "text"
    },
    "URL": "http://localhost:1337/user/showR",
    "debug": 1, //debug mode enabled
    "adapter" : {
      "type" : "sqlrest",
      "collection_name" : "regList",
      "idAttribute" : "id"
    }
  }
}
```

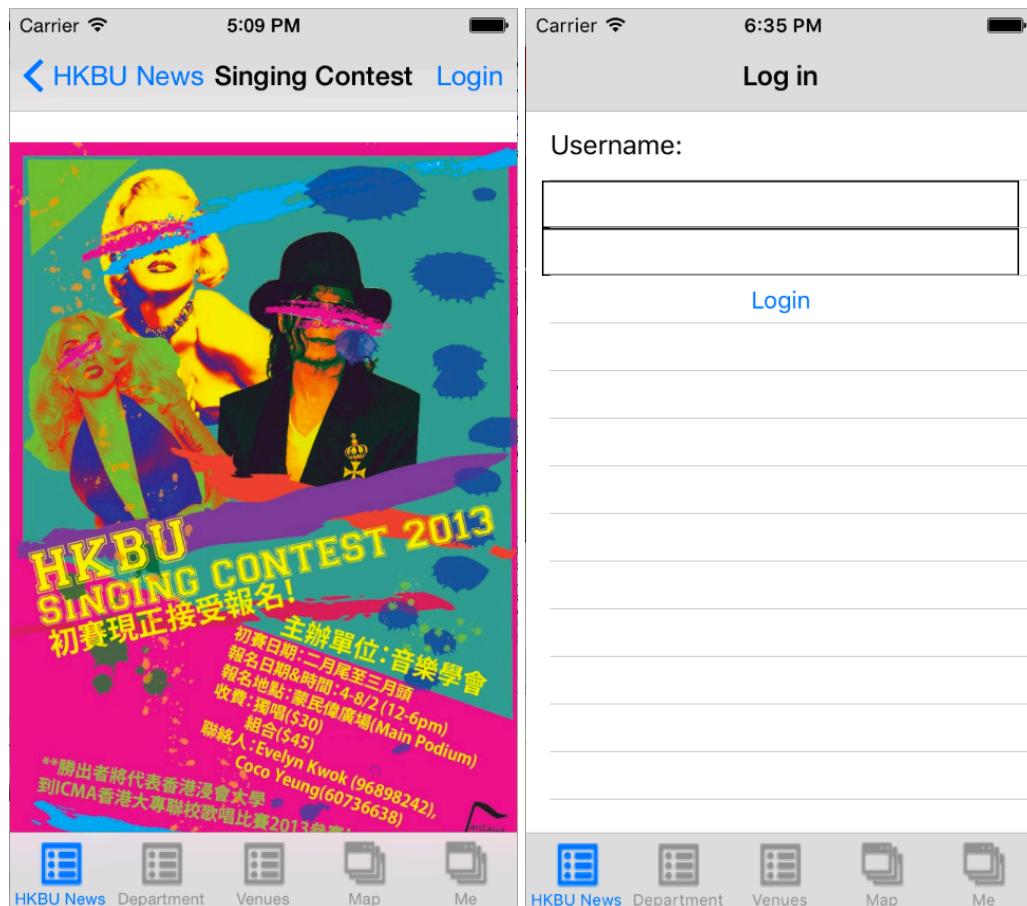
The `showR` function is create in web part. It will return all the registered events of the user.

```

showR: function( req, res ) {
    User.findOne(req.session.userInfo.id).populateAll().exec( function (err, model) {
        return res.json(model.register);
    })
},

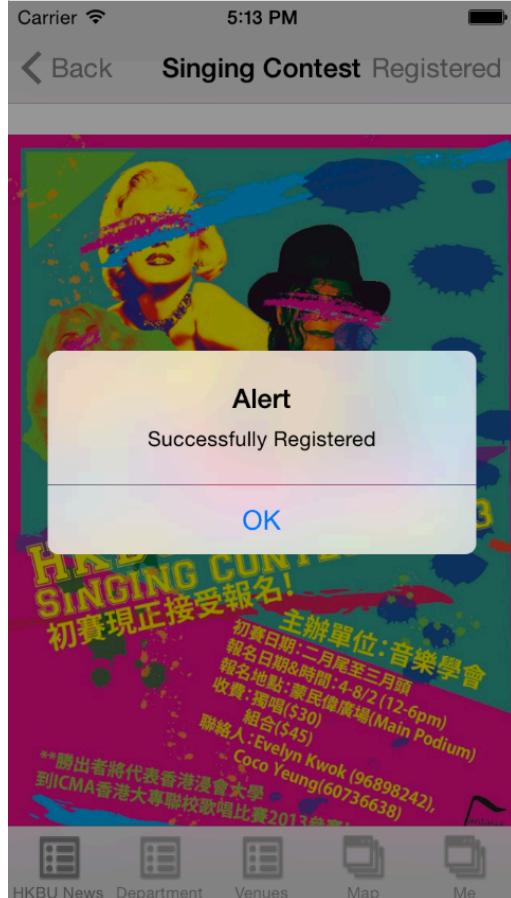
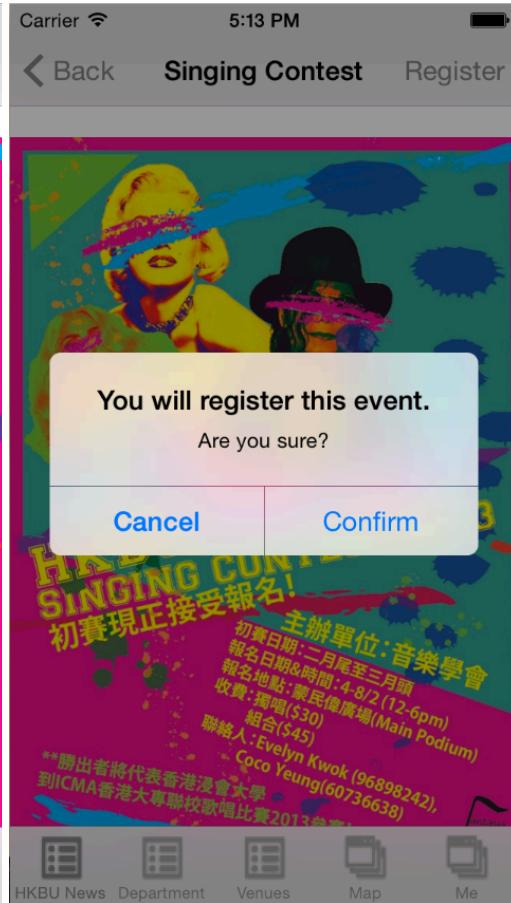
```

Open the mobile app, and user will see the events list page. User can view the short descriptions of all the events on this page. Tap one event, and user can view the full information of that event.



If users have not logged in the system, the right button of the navbar will show `login`. Tap the `login`, then users will come to the login page to input their username and password.

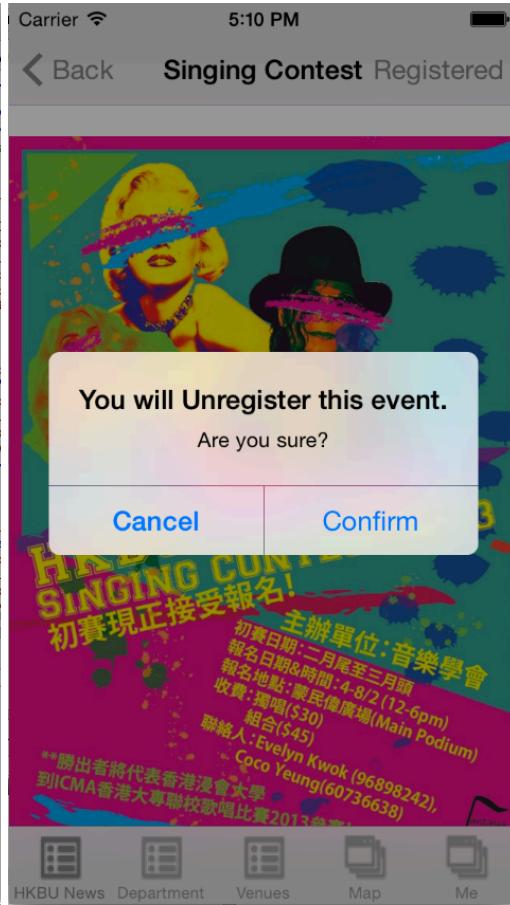
```
function login(e) {
    var username = $.username.value;
    var password = $.password.value;
    var thisuser = Alloy.Collections.UserList.where({username:username,password:password});
    console.log(JSON.stringify(thisuser));
    if(JSON.stringify(thisuser)!=="[]"){
        var userid = thisuser[0].get("id");//at(0).get("uid");
        Alloy.Globals.meid = userid;
        var xhr = Ti.Network.createHTTPClient();
        xhr.onload = function(e){};
        xhr.open('POST', 'http://localhost:1337/user/login');
        xhr.send({
            "username":username,
            "password":password
        });
    }
    if(Alloy.Globals.meid!=0) {
        var eventListController = Alloy.createController('UserInfo');
        if(Alloy.Globals.tabgroup.getActiveTab()==Alloy.Globals.logTab){
            Alloy.Globals.tabgroup.getActiveTab().open(eventListController.getView());
        }else {
            $.win.close();
            Alloy.Globals.eventDetailWin.close();
            var eventListController = Alloy.createController('eventDetail',{
                name: Alloy.Globals.logename
            });
            Alloy.Globals.tabgroup.getActiveTab().open(eventListController.getView());
        }
    }
}
```



The following code is to show the confirm box.

```
function Register(e) {
    if($.regButton.title=="Login"){
        Alloy.Globals.logename=name;
        var eventListController=Alloy.createController('Login');
        Alloy.Globals.tabgroup.getActiveTab().open(eventListController.getView());
    }else if($.regButton.title=="Register"){
        $.regit.title="You will register this event.";
        $.regit.message="Are you sure?";
        $.regit.show();
    }else if($.regButton.title=="Registered"){
        $.regit.title="You will Unregister this event.";
        $.regit.message="Are you sure?";
        $.regit.show();
    }
}
```

After logging in the system, if user have not registered the event, the right button of the navbar will show **Register**. User can tap the **Register** to confirm the registration on the event.



The following code shows how to realize the `Reg` and `UnReg` function in js.

```

function confirm(e){
    if($.regButton.title=="Register"){
        var xhr = Ti.Network.createHTTPClient();
        xhr.onload = function(e){};
        xhr.open('POST','http://localhost:1337/user/addRegister/?pid='+eid);
        xhr.send();

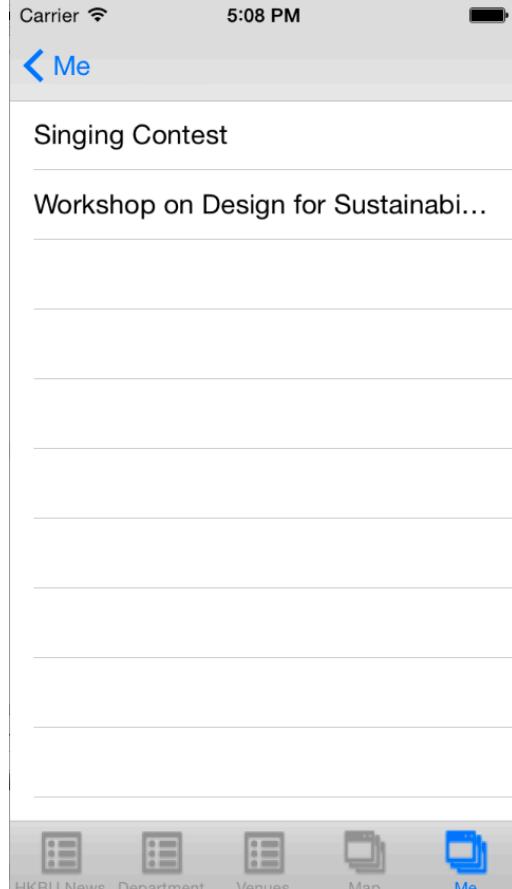
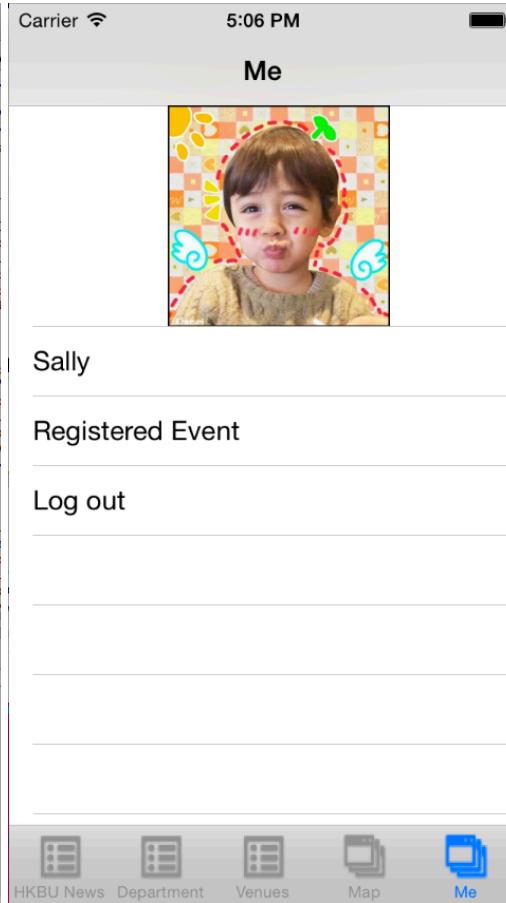
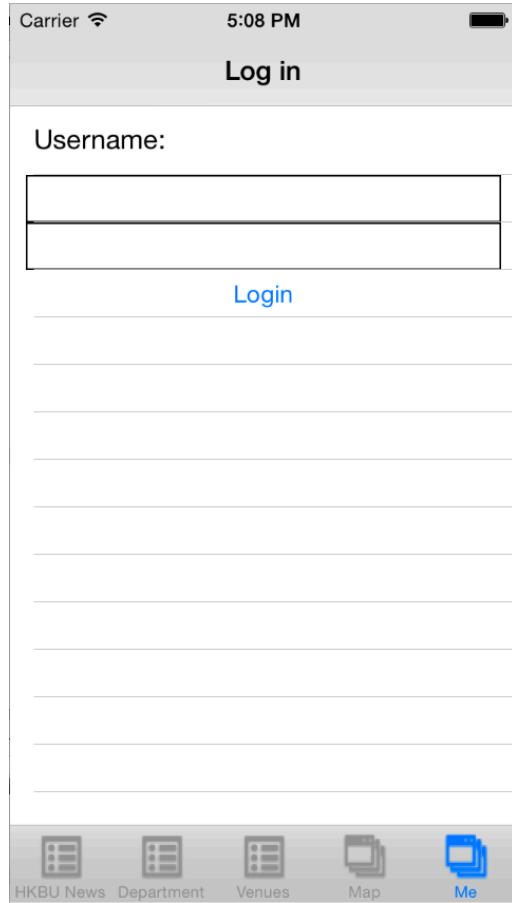
        var model = Alloy.createModel('regList',{id:eid,uid:Alloy.Globals.meid});
        Alloy.Collections.regList.add(model);
        model.save();
        $.regButton.title="Registered";
        alert("Successfully Registered");
    }else{
        var xhr = Ti.Network.createHTTPClient();
        xhr.onload = function(e){};
        xhr.open('POST','http://localhost:1337/user/removeRegister/?pid='+eid);
        xhr.send();

        var model = Alloy.Collections.regList.where({id:eid})[0];
        var rest = Alloy.Collections.regList.where({id:eid});
        console.log("before"+rest.length);
        Alloy.Collections.regList.remove(model);
        rest = Alloy.Collections.regList.where({id:eid});

        model.destroy();
        $.regButton.title="Register";
        alert("You have Unregisterer it");
    }
}

```

After logging in the system, if user have registered the event, the right button of the navbar will show **Registered**. User can tap the **Registered** to cancel the registration on the event.



User also can login in **Tab 5**. In this tab, user can check his registered event and the detail of the events.

```
if(Alloy.Globals.meid==0||Alloy.Globals.meid==null){
    var eventListController = Alloy.createController('Login');
    $.logTab.open(eventListController.getView());
}
$.logTab.addEventListener('focus',function(e) {
    if(Alloy.Globals.meid==0||Alloy.Globals.meid==null){
        var eventListController = Alloy.createController('Login');
        $.logTab.open(eventListController.getView());
    }else {
        var eventListController = Alloy.createController('UserInfo');
        Alloy.Globals.tabgroup.getActiveTab().open(eventListController.getView());
    }
});
```

The code above is to judge whether the user has logged in or not. `Alloy.Globals.meid` saves the user id after login.