

# **Recording Class Attendance in a Student Registration System**

Sanhu Li, Mao Zheng  
Department of Computer Science  
University of Wisconsin-La Crosse  
La Crosse WI, 54601  
[mzheng@uwlax.edu](mailto:mzheng@uwlax.edu)

Hao Fan  
School of Information Management  
Wuhan University  
Wuhan, Hubei, 430079  
[hfan@whu.edu.cn](mailto:hfan@whu.edu.cn)

## **Abstract**

Recording classroom attendance by hand for large classes can be a time consuming task. We have developed a mobile application and registration system that allows collection of attendance information. The information is collected with Android devices and transferred and stored in a web-based student registration system. Both the Android application and the registration system are part of this project. The registration system serves as a test-bed for the recorded data and can be used to analyze the data.

There are two main challenges when recording class attendance. First, it is important to be certain the student physically comes to the classroom. Second, the application needs to verify the recorded data is for the correct student. The current solution provided in this project is to let the instructor generate a random QR code during class time, in the classroom. The student will use his/her mobile device to scan the QR code and submit the scanned data along with his/her student ID to the registration system during the recording time period. Only students who are enrolled in the class are able to submit the scanned data and this data can only be recorded once during a valid time period. The instructor can view the recorded class attendance information and each student can verify that his/her attendance was recorded correctly.

There are two major platforms in the mobile device community: iOS and Android. This project chose Android development mainly for the reason of its openness. In addition, all the tools in the Android development are free and no special hardware is required.

The web-based student registration system uses Bootstrap framework as the theme. The front-end are static html files using AngularJs together with jQuery and Ajax to provide the dynamic pages. In the back end, the system uses JSP, Spring-mvc framework and MySQL database. Connections are managed by the Hibernate.

# 1 Introduction

Recording classroom attendance by hand for large classes can be a time consuming task. We have developed a mobile application and registration system that allows collection of attendance information. The information is collected with Android devices and transferred and stored in a web-based student registration system. Both the Android application and the registration system are part of this project. The registration system serves as a test-bed for the recorded data and can be used to analyze the data.

There are two main challenges when recording class attendance. First, it is important to be certain the student physically comes to the classroom. Second, the application needs to verify the recorded data is for the correct student. The current solution provided in this project is to let the instructor generate a random QR code during class time in the classroom. The student will use his/her mobile device to scan the QR code and submit the scanned data along with his/her student ID to the registration system during the recording time period. Only students who are enrolled in the class are able to submit the scanned data and this data can only be recorded once during the valid time period. The instructor can view the recorded class attendance information and each student can verify that his/her attendance was recorded correctly.

## 2 The Web-based Student Registration System

A big component of this project is the web-based student registration system. It serves as the test bed: 1) To verify student's class registration records: only students which are registered for the current course are able to submit their class attendance. 2) To ensure the recording time for the class attendance is valid. For example: the course CS741 in Spring 2015 is offered Monday Wednesday and Friday from 11:00 am to noon. If the class attendance recording time is on Tuesday at 11:00 am, the system should report an error message.

Most of the validations discussed above are addressed in the database design. Below is the E-R diagram for the student registration system. The system uses MySQL for the database. The student\_record table is used to keep all the class attendance records for a student. The student\_ID needs to be existed in the student\_course table that is used to store the student's course registrations. Similarly, every class attendance record has a date and time that can be used to check if it is within the valid course time.

The student registration system has three types of users: Student, Instructor and Registrar. The use case diagram is shown in Figure 2.

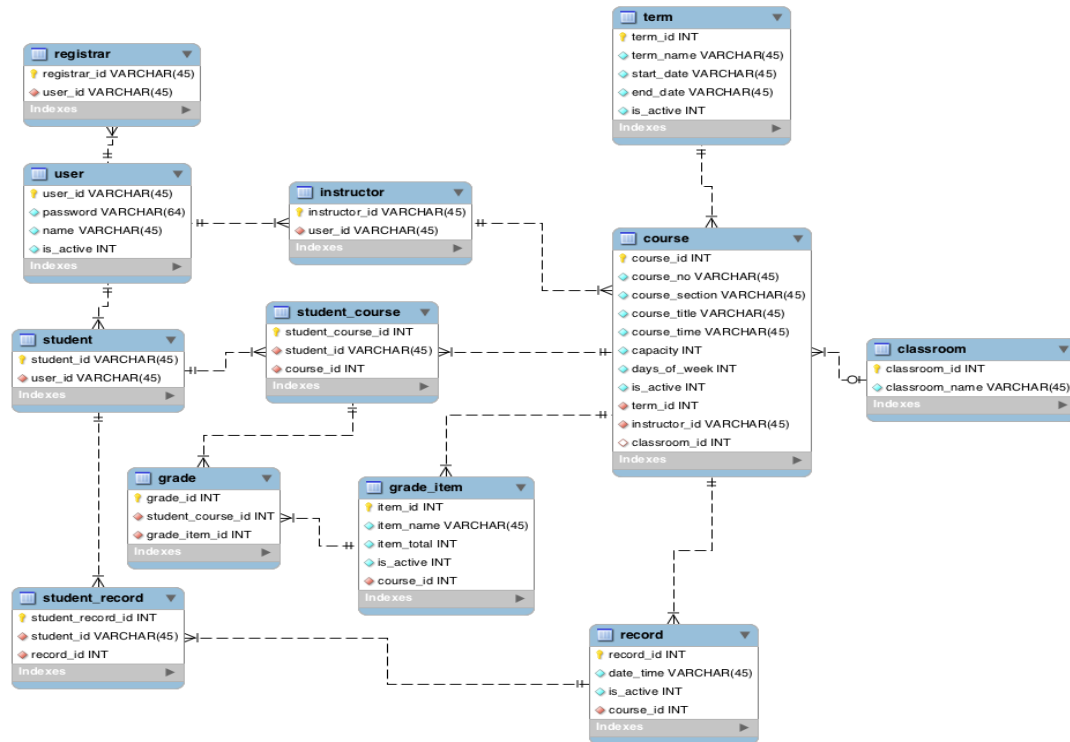


Figure 1 E-R Diagram for the Student Registration System

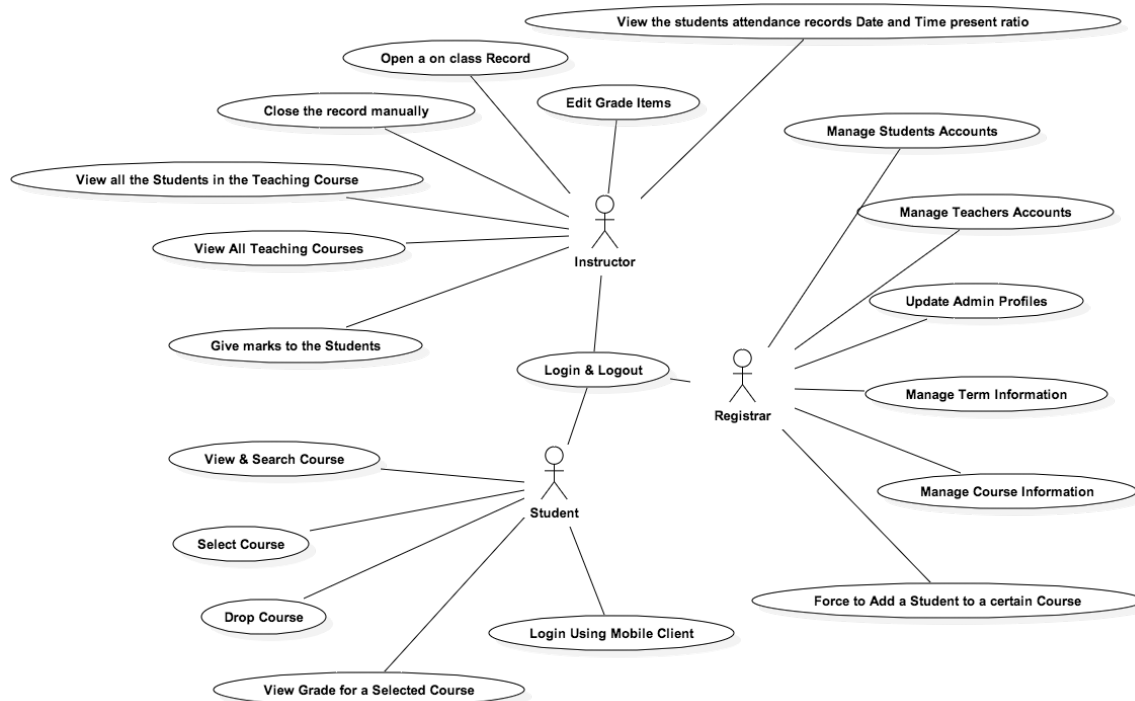


Figure 2 Use Case Diagram for the Student Registration System

### 3 Generating QR Code

The system uses the Z-xing API [3] to generate the QR code based on the record ID. However the record ID will be encrypted using AES first. The key which is used for encryption and decryption by AES is a random generated string.

The Source code for generating the QR code is shown below.

```
" alt="qr code"/>
```

Figure 3 Source Code Segment for Generating the QR Code.

The client side of this project is a mobile application developed in the Android environment. There are two major platforms in the mobile device community: iOS and Android. This project chose Android development mainly for the reason of its openness. In addition, all the tools in the Android development are free and no special hardware is required.

The student can use his/her mobile device to scan the QR code and submit the scanned the image along with his/her ID and the stored key in the app, to the server. The server will check if the student ID, record ID, recording time and class time are valid and consistent.

Figure 4 is the screen shot of the student scanning the QR code using his/her mobile device. Figure 5 is the test data of the recorded class attendance along with the date and time.

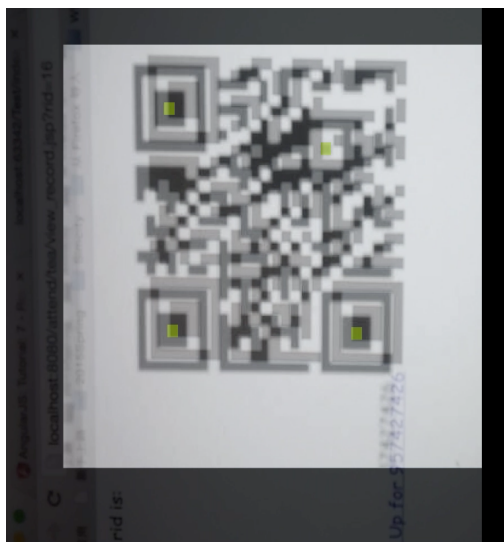


Figure 4 Scanning the QR code

Record Id	Record Date
4	12/03/2014
6	12/03/2014
7	12/03/2014
10	02/02/2015
11	02/02/2015
12	02/02/2015
13	02/15/2015
16	03/18/2015 03:29:31

Figure 5 Recorded Class Attendance

## 4 Spring MVC Framework

This web-based student registration system uses a Spring-mvc framework and the MySQL database on the back end. Connections are managed by the Hibernate.

The Interceptor class in Spring-mvc framework helps the validation. As we described in the previous section, there are three types of users in the system. Each has his/her own access. Instead of having to validate the user's role on every page, we can define the access based on the user's role in the Interceptor class. The code segment is shown below:

```
<mvc:interceptors>
  <mvc:interceptor>
    <mvc:mapping path="/student/**"/>
      <bean class="org.lsh.security.PrivilegeInterceptor">
        <constructor-arg index="0">
          <set>
            <value>Student</value>
          </set>
        </constructor-arg>
      </bean>
    </mvc:interceptor>
    <mvc:interceptor>
      <mvc:mapping path="/instructor/**"/>
        <bean class="org.lsh.security.PrivilegeInterceptor">
          <constructor-arg index="0">
            <set>
              <value>Instructor</value>
            </set>
          </constructor-arg>
        </bean>
      </mvc:interceptor>
      <mvc:interceptor>
        <mvc:mapping path="/officer/**"/>
          <bean class="org.lsh.security.PrivilegeInterceptor">
            <constructor-arg index="0">
              <set>
                <value>Officer</value>
              </set>
            </constructor-arg>
          </bean>
        </mvc:interceptor>
      </mvc:interceptors>
```

Figure 6 The Interceptor Class

We are using AngularJs in the static html files, together with jQuery and Ajax to provide the dynamic pages on the front-end. It supports a clear and better Model-View-Controller(MVC) structure. If the data is changed, the view will be automatically changed. The controller only needs to manipulate the data triggered by the user event. Below is the code segment example for writing the sidebar using angular.js. It will be directed into individual static html files that use Ajax to communicate with the server to get the data.

```

<script src="http://ajax.googleapis.com/ajax/libs/angularjs/1.2.26/angular.min.js"></script>
<script>
var sidebar = {
  student: [
    {text: 'Add Student', url: 'addStudent.html'},
    {text: 'Disable Student', url: 'disableStudent.html'},
    {text: 'Update Student', url: 'updateStudent.html'},
    {text: 'Search Student', url: 'searchStudent.html'}
  ],
  instructor: [
    {text: 'Add Instructor', url: 'addInstructor.html'},
    {text: 'Disable Instructor', url: 'disableInstructor.html'},
    {text: 'Update Instructor', url: 'updateInstructor.html'},
    {text: 'Search Instructor', url: 'searchInstructor.html'}
  ],
  registrar: [
    {text: 'Add Registrar', url: 'addRegistrar.html'},
    {text: 'Disable Registrar', url: 'disableRegistrar.html'},
    {text: 'Update Registrar', url: 'updateRegistrar.html'},
    {text: 'Search Registrar', url: 'searchRegistrar.html'}
  ]
};

var app = angular.module('app', []);
app.controller('nav-controller', function() {
  this.tab = 1;
  this.name = 'OfficerName';
  this.setTab = function (tabNum) {
    this.tab = tabNum;
  };
  this.isActive = function (check) {
    return this.tab == check;
  };
});

```

Figure 7 Angular.js Code Segment Example

## 5 Conclusions

The future work of the current project is aiming to research some indoor wifi-position methods. This is a better way to ensure the student physically comes to the classroom. We need to add the functions of submitting current wifi environment from the client to the server and checking if it is from one of the classroom's wifi hotspots.

## References:

- [1] Computer Science 2008, An Interim Revision of CS 2001(<http://www.acm.org/education/curricula/ComputerScience2008.pdf>)
- [2] Android Developer's Guide. <http://developer.android.com/guide/index.html>
- [3] Z-xing API <http://zxing.github.io/zxing/apidocs/>
- [4] Abelson, W.F., Collins, C., Sen, R. Unlocking Android – A Developer's Guide. Manning Pub. April 2009.
- [5] Victor Matos, Rebecca Grasser, Building Applications for the Android OS Mobile Platform: A Primer and Course Materials, Journal of Computing Sciences in Colleges, Volume 26 Issue 1, pp: 23-29, October 2010

- [6] Derek Riley, Using Mobile Phone Programming to teach Java and Advanced Programming to Computer Scientist, ACM Special Interest Group on Computer Science Education SIGSCE 2012, pp:541-546. Feb. 29-March 3, 2012.
- [7] B. N. Schilit, N. Adams, and R. Want. Context-aware Computing Applications. In Proceedings of IEEE Workshop on Mobile Computing Systems and Applications, pp. 85-90, Santa Cruz, CA, Dec. 1994. IEEE Computer Society Press.
- [8] A. K. Dey and G. D. Abowd, "Towards a Better Understanding of Context and Context-awareness," in CHI 2000 Workshop on the What, Who, Where, When and How of Context-awareness, 2000.
- [9] R. Lowe, P. Mandl, M. Weber. "Context Director: A Context-aware Service for Mobile Context-aware Computing Applications by the Example of Google Android", Tenth Annual IEEE International Conference on Pervasive Computing and Communications Workshops (PERCOM Workshops 2012), Lugano, Switzerland, March 2012.