Image Based Room Schedule Retrieval System

A Design Project Report

Presented to the School of Electrical and Computer Engineering of Cornell University

in Partial Fulfillment of the Requirements for the Degree of

Master of Engineering, Electrical and Computer Engineering

Submitted by

Shuai Yuan

MEng Field Advisor: Tsuhan Chen

Degree Date: May, 2012

Abstract

Master of Engineering Program

School of Electrical and Computer Engineering

Cornell University

Design Project Report

Project Title: Image based room schedule Retrieval System

Author: Shuai Yuan

Abstract:

My MEng project would provide a practical system that can allow students or faculties check the ECE courses' schedule simply by taking a picture using their IPhone or IPad. And they will get all the schedules for a particular room just after a second.

The whole project includes an App for IOS, and a website which can also check related information by selecting information from drop list. I am doing this project with Adarsh Kowdle, a Cornell PHD. At this stage this project only focuses on checking the schedule of ECE classrooms. But once this system is finished, it's not hard to expand the system to multiple usages by changing its database.

Content

1 Executive Summary

In summary, this Image based image schedule retrieval system was created for the iPhone, iPod Touch, and iPad.

A user-friendly web based room schedule retrieval system was implemented, which can allow users check the schedule by using the drop menu in the web page.

An application in IOS was developed, which allows users input by either using web input or camera input (the web page was embedded in this App)

A Tomcat Apache Web Server was set up on a server in AMP Lab, so that the server can be visited by Internet users.

The server was written in JAVA. A JAVA servlet was set to handle user's request and give back the relevance feedback.

A course schedule database was created in the server.

Aprise OCR was used to help recognize the photos sent by users.

A PHP script was used to receive the picture from user's I-devices.

A JAVA program was used to check and create the database. HTML Parser was used in this program.

The app was run successfully on the iPhone Simulator, iPad Simulator, iPhone,iPod Touch, and iPad.

2. Introduction

Consider a scenario, where one is standing outside a classroom and would like to get some information about the classes scheduled in that room, the availability of the room later in the day, etc. Consider another scenario where the student is outside a professor's office and would like to get some information about the professor's schedule. This project presents an image based data retrieval system to address this practical problem.

My MEng project will focus on the rooms and courses in ECE, which can be easily extended to the other buildings and courses in Cornell. The task of image based data retrieval system will be packaged into an application that can work across platforms and even using a web-browser. It will consist of two parts, 1) The Client and 2) The Server. I will focus on developing an application on the iOS platform as part of this project. On the client side, we first start with an image of the classroom number plate captured using a mobile device (I-device in this case). We perform character recognition to identify the room number. The room information along with the current date and time and then transmitted as a string to the server that will be located in the Ward Laboratory. On the server end, a script can be run to create the database by parsing all the information on the course roster page and generate a database for each room with relevant information about the courses, timing, professor, etc. The server will decode the picture sent from the mobile device and use it to retrieve the accurate course information for the queried room from the database. In addition the server will also retrieve the list of other courses scheduled in that room, so the student has a better idea about the course going on in that room.

Additional functionality can later be incorporated into the same framework so the user can query by day/time information, query by professor name to retrieve the professor's calendar, etc. The goal of the project is to achieve a fully functional application that would work on iOS devices, a web application that can work from any browser accessible via a laptop or other mobile devices.

3. Background

3.1 Iphone App Development

IOS

iOS (originally iPhone OS) is a mobile operating system developed and distributed by Apple Inc. Originally released in 2007 for the iPhone and iPod Touch, it has since been extended to support other Apple devices such as the iPad and Apple TV. Unlike Windows CE (Mobile and Phone) and Android, Apple does not license iOS for installation on non-Apple hardware. As of March 6, 2012, Apple's App Store contained more than 550,000 iOS applications, which have collectively been downloaded more than 25 billion times. It had a 16% share of the smartphone operating system units sold in the last quarter of 2010, behind both Google's Android and Nokia's Symbian.[In May 2010 in the United States, it accounted for 59% of mobile web data consumption (including use on both the iPod Touch and the iPad).

The user interface of iOS is based on the concept of direct manipulation, using multi-touch gestures. Interface control elements consist of sliders, switches, and buttons. The response to user input is immediate and provides a fluid interface. Interaction with the OS includes gestures such as swipe, tap, pinch, and reverse pinch, all of which have specific definitions within the context of the iOS operating system and its multi-touch interface. Internal accelerometers are used by some applications to respond to shaking the device (one common result is the undo command) or rotating it in three dimensions (one common result is switching from portrait to landscape mode).

iOS is derived from Mac OS X, with which it shares the Darwin foundation, and is therefore a Unix operating system.

In iOS, there are four abstraction layers: the Core OS layer, the Core Services layer, the Media layer, and the Cocoa Touch layer. The current version of the operating system (iOS 5.1.1) uses roughly 770 megabytes of the device's storage, varying for each model.

XCODE

Xcode is an Integrated Development Environment (IDE) containing a suite of software development tools developed by Apple for developing software for OS X and iOS. First released in 2003, the latest stable release is version 4.3.2 and is available via the Mac App Store free of charge for Mac OS X Lion users. Registered developers can download preview releases and previous versions of the suite through the Apple Developer website.

OBJECTIVE C

Objective-C is a reflective, object-oriented programming language that adds Smalltalk-style messaging to the C programming language.

Today, it is used primarily on Apple's Mac OS X and iOS: two environments derived from the OpenStep standard, though not compliant with it. Objective-C is the primary language used for Apple's Cocoa API, and it was originally the main language on NeXT's NeXTSTEP operating system. Generic Objective-C programs that do not use these libraries can also be compiled for any system supported by gcc or Clang.

3.2 Server

TOMCAT APACHE

Apache Tomcat (or simply Tomcat, formerly also Jakarta Tomcat) is an open source web server and servlet container developed by the Apache Software Foundation (ASF). Tomcat implements the Java Servlet and the JavaServer Pages (JSP) specifications from Oracle Corporation, and provides a "pure Java" HTTP web server environment for Java code to run.

Tomcat should not be confused with the Apache web server, which is a C implementation of an HTTP web server; these two web servers are not bundled together, although they are frequently used together as part of a server application stack. Apache Tomcat includes tools for configuration and management, but can also be configured by editing XML configuration files.

MyEclipse

MyEclipse is a commercially available Java EE and Ajax IDE created and maintained by the company Genuitec, a founding member of the Eclipse Foundation.

MyEclipse is built upon the Eclipse platform, and integrates both proprietary and open source solutions into the development environment.

MyEclipse has two primary versions (apart from the "Blue Edition and "MyEclipse Spring Edition referred to below): a professional and a standard edition. The standard edition adds database tools, a visual web designer, persistence tools, Spring tools, Struts and JSF tooling, and a number of other features to the basic Eclipse Java Developer profile. It competes with the Web Tools Project, which is a part of Eclipse itself, but MyEclipse is a separate project entirely and offers a different feature set.

MyEclipse has also been made available via Pulse (ALM), a provisioning tool that maintains Eclipse software profiles, including those that use MyEclipse. Additionally, MyEclipse is offering a customized version for IBM products, "MyEclipse Blue Edition", that adds specific support for Rational Software and WebSphere development. Currently, MyEclipse Blue Edition is available for Windows and Linux, though Mac is unsupported.

3.3 OCR

APRISE OCR SDK 4.0

About Asprise OCR SDK

Embedded a high performance OCR engine, Asprise OCR SDK is OCR software development kit that can be used with Java, .Net, Delphi, Visual Basic (VB), Borland C, etc. on multiple platforms . Windows, Linux, Mac OS, Solaris and AIX.

Features of Asprise OCR

An incomplete list of features offered by Asprise OCR:

. Highest Level of Accuracy

Asprise OCR can easily recognize difficult documents of poor image quality

. Excellent Format Retention

Text layouts on the input documents are preserved;

. High Speed

Asprise OCR uses optimized OCR engine to perform excellent recognition in very short time;

. Ease of Use

We strive to make the developer's life easier. Complex parameter configurations are removed from Asprise OCR SDK. You only have to supply the image document. Asprise OCR can intelligently determine the best setting internally.

. Barcode Recognition

Beside characters (letters and numbers), Asprise OCR can recognize almost every kind of bar code. You can choose to recognize barcode or characters or both. Currently, the following bar code formats are supported:

. CODE 128 (128b, 128C, 128raw)

. EAN 8 EAN 13

. UPC

. Code 3 of 9

. Code interleaved 2 of 5.

4. Statement of Problem

4.1 Workflow

Before all, I need to decide my work flow so that I can decide further movements.

4.2 Create the database

At this stage, we don't have an existing database for ECE schedule information. So the very first task is to create a database for all possible rooms' schedule. And then we can get whatever we want from this database.

4.3 Create a web application that can handle users' request

This webserver program will have two parts, one part is a user webpage that can allow user input the room number, the current time and the day they want to check. The other part is to analyze users' input and retrieve data from database by using a certain algorithm. Finally send all the schedule information back to the webpage.

4.4 IOS Application

The final output of this project is an IOS Application. So I need to implement an App that can embed the web page in. Also, this app should be able to allow users transmit pictures from I-device to server.

4.5 Figure out a way to send pictures from I-device to server.

I need to write a script that can ensure both of the picture in the album and the picture taken directly can be sent to server.

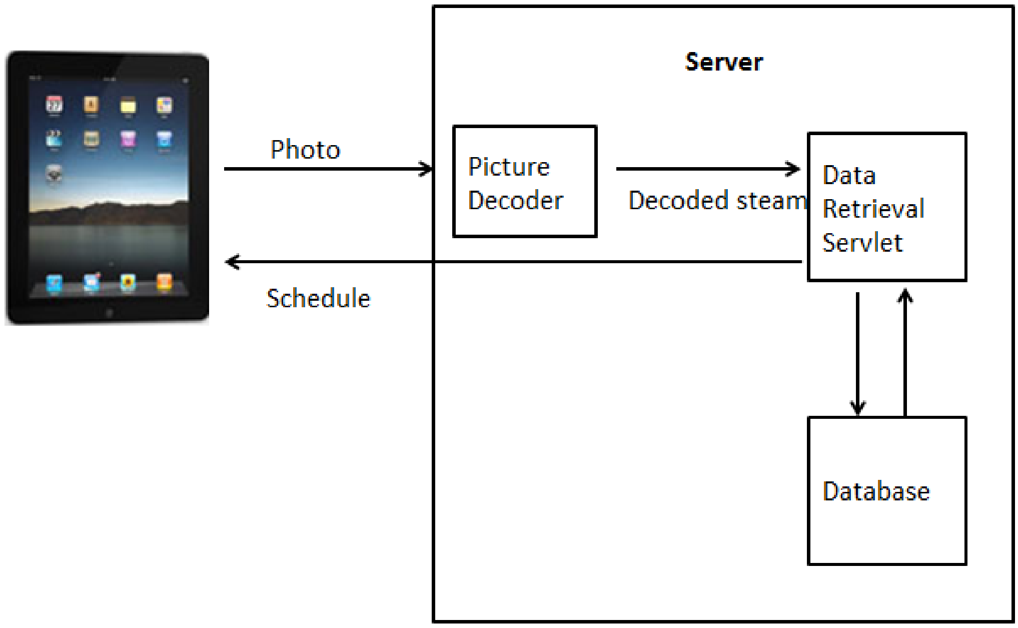
4.6 Figure out a way to decode the picture

This requires me to find out an efficient way to do OCR process. A steady OCR can enhance the whole stability of the system. So this problem is actually the key of this project.

5. Range of solutions

5.1 Workflow

After discussed with Adarsh Kowdle, we got the workflow below:



5.2 Database

The database needs to include all course schedules for the ECE classrooms. There should have an official database but as a student, I can't get the permit to access it. So I have to figure out a way to collect the data and create my own database.
The way I did it was to create the schedule database by parsing our ECE roster webpage (http://registrar.sas.cornell.edu/courses/roster/FA11/ECE/) using a java program. And pick out the info I need. Then create my database by using a certain algorithm.

5.3 Web Server

The function of this web app includes two parts: A web page that allows users input their queries and a back end program that used to retrieve the information from database as well as generate the feedback for users.
So my solution was to build the webpage by using JSP, which could be more flexible than the traditional html page. And at the back end I used Java servlet to deal with all kinds of queries.

5.4 IOS Application

I used Xcode as the IDE and Objective C as the language to implement this IOS Application. This App allows users to input by both using web or using album and camera, which satisfies my need perfectly.

5.5 Send pictures

I used a PHP script to send the picture. PHP script is set in the server and once a query happens, it will call this PHP script and this script will help transmit pictures .

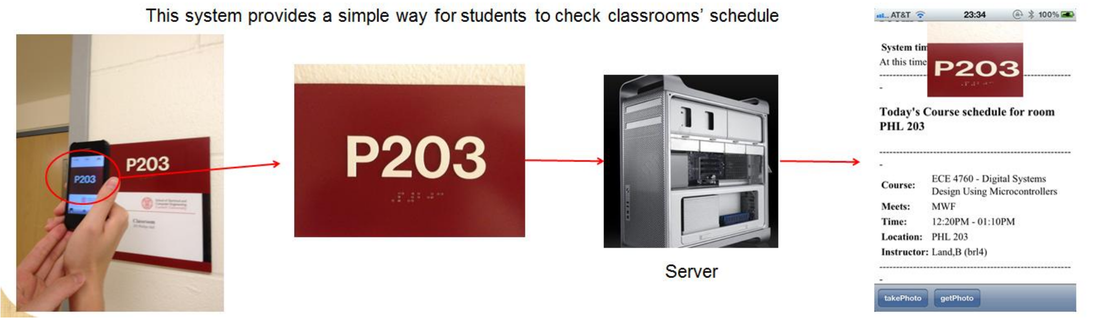
5.6 OCR

The OCR is the key of this application. After all kinds of evaluation, I chose ApriseOCR to perform this function. This OCR is a java jar file that can be used directly in the server side.

6. Design and Implementation

6.1 Overview

Below is the overview of this project. Basically, it allows users check a classroom's schedule by simply taking a picture of a number board for a classroom. The server will analyze user's request and then give the feedback to users.


This system provides a simple way for students to check classrooms' schedule
P203
P203
P203
Server
AT&T 23:34 100%
P203
takePhoto
getPhoto

System tim
At this time

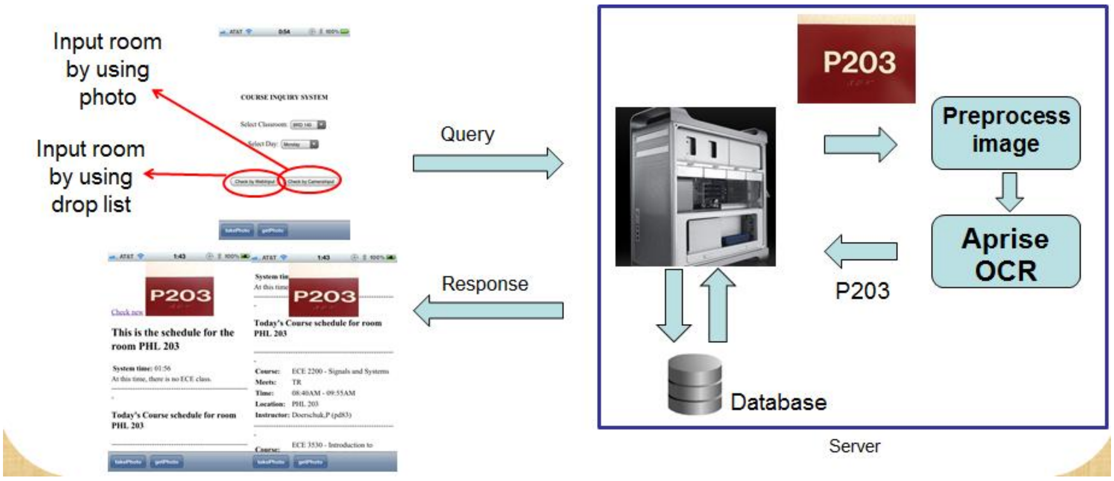
Today's Course schedule for room PHL 203

Course: ECE 4760 - Digital Systems Design Using Microcontrollers
Meets: MWF
Time: 12:20PM - 01:10PM
Location: PHL 203
Instructor: Land,B (brl4)

Figure 6.1 Procedure

6.2 Workflow

Below is the detailed work flow of my project. User can check room schedule by using the drop list on the web page, using their pictures in the album or taking a picture by their I-devices.



This is the schedule for the room PHL 203

System time: 01:56
At this time, there is no ECE class.

Today's Course schedule for room PHL 203

System tim
At this time

Today's Course schedule for room PHL 203

Course: ECE 2200 - Signals and Systems
Meets: TR
Time: 08:40AM - 09:55AM
Location: PHL 203
Instructor: Doerschuk,P (pd83)

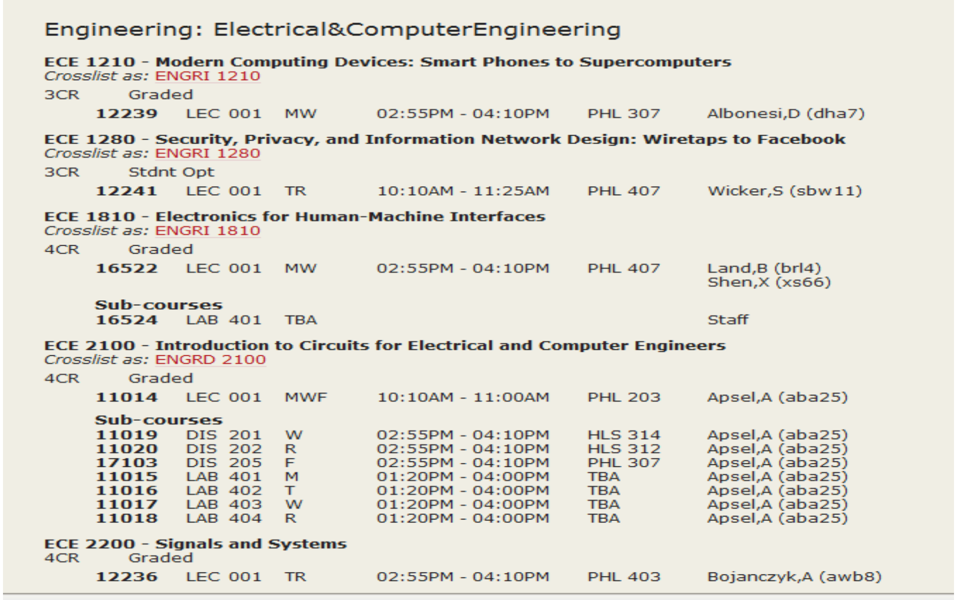
Course: ECE 3530 - Introduction to

Figure 6.2 Workflow

6.3 Database

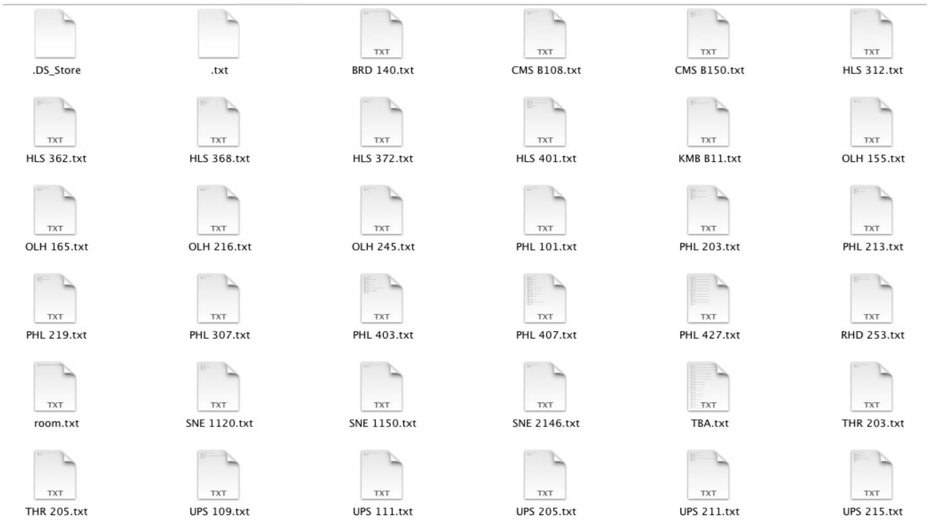
Database is created by using Datacontroller.java class

1. Once the server runs, it will firstly check whether there's an existing database. If not, the DataCreator() method in DataController.java is called.
2. After DataCreator() is called, semesterChecker() will be called to check the system time and decide which semester we are in. Then generates the url of the corresponding Cornell roster page.
3. Then htmlParser() is called to parse the roster page by using the url we get from step 2.
4. The room level database is created by using the information collected from step 3.



```
Engineering: Electrical&ComputerEngineering
ECE 1210 - Modern Computing Devices: Smart Phones to Supercomputers
Crosslist as: ENGRI 1210
3CR      Graded
      12239   LEC 001   MW       02:55PM - 04:10PM    PHL 307    Albonesi,D (dha7)
ECE 1280 - Security, Privacy, and Information Network Design: Wiretaps to Facebook
Crosslist as: ENGRI 1280
3CR      Stdnt Opt
      12241   LEC 001   TR       10:10AM - 11:25AM    PHL 407    Wicker,S (sbw11)
ECE 1810 - Electronics for Human-Machine Interfaces
Crosslist as: ENGRI 1810
4CR      Graded
      16522   LEC 001   MW       02:55PM - 04:10PM    PHL 407    Land,B (brl4)
                                                                 Shen,X (xs66)
      Sub-courses
      16524   LAB 401   TBA                                      Staff
ECE 2100 - Introduction to Circuits for Electrical and Computer Engineers
Crosslist as: ENGRD 2100
4CR      Graded
      11014   LEC 001   MWF      10:10AM - 11:00AM    PHL 203    Apsel,A (aba25)
      Sub-courses
      11019   DIS 201   W        02:55PM - 04:10PM    HLS 314    Apsel,A (aba25)
      11020   DIS 202   R        02:55PM - 04:10PM    HLS 312    Apsel,A (aba25)
      17103   DIS 205   F        02:55PM - 04:10PM    PHL 307    Apsel,A (aba25)
      11015   LAB 401   M        01:20PM - 04:00PM    TBA        Apsel,A (aba25)
      11016   LAB 402   T        01:20PM - 04:00PM    TBA        Apsel,A (aba25)
      11017   LAB 403   W        01:20PM - 04:00PM    TBA        Apsel,A (aba25)
      11018   LAB 404   R        01:20PM - 04:00PM    TBA        Apsel,A (aba25)
ECE 2200 - Signals and Systems
4CR      Graded
      12236   LEC 001   TR       02:55PM - 04:10PM    PHL 403    Bojanczyk,A (awb8)
```

Figure 6.3 Cornell roster page for ECE



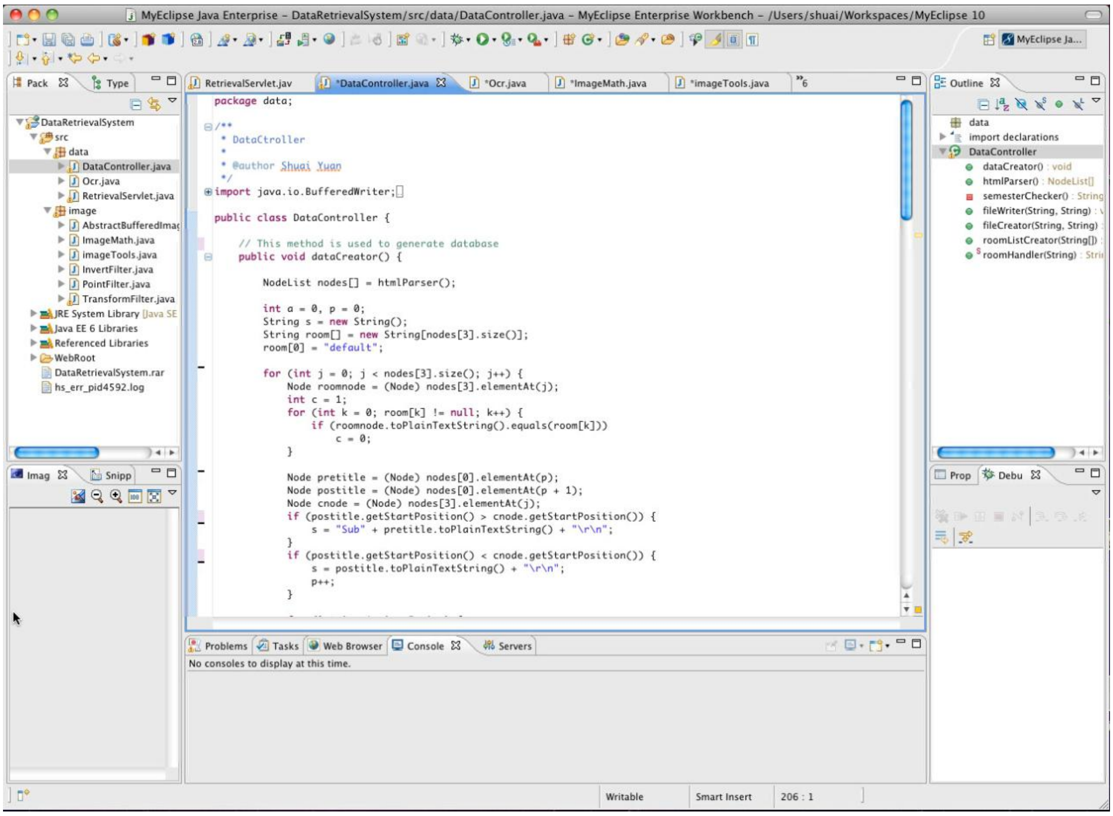
```
.DS_Store      .txt           BRD 140.txt    CMS B108.txt   CMS B150.txt   HLS 312.txt
HLS 362.txt    HLS 368.txt    HLS 372.txt    HLS 401.txt    KMB B11.txt    OLH 155.txt
OLH 165.txt    OLH 216.txt    OLH 245.txt    PHL 101.txt    PHL 203.txt    PHL 213.txt
PHL 219.txt    PHL 307.txt    PHL 403.txt    PHL 407.txt    PHL 427.txt    RHD 253.txt
room.txt       SNE 1120.txt   SNE 1150.txt   SNE 2146.txt   TBA.txt        THR 203.txt
THR 205.txt    UPS 109.txt    UPS 111.txt    UPS 205.txt    UPS 211.txt    UPS 215.txt
```

Figure 6.4 Database

6.4 Web Server

6.4.1 IDE

The IDE I used to write java code is Myeclipse. This IDE is designed especially for network development.



```
package data;

/**
 * DataCtroller
 *
 * @author Shuai Yuan
 */
import java.io.BufferedWriter;

public class DataController {

    // This method is used to generate database
    public void dataCreator() {

        NodeList nodes[] = htmlParser();

        int a = 0, p = 0;
        String s = new String();
        String room[] = new String[nodes[3].size()];
        room[0] = "default";

        for (int j = 0; j < nodes[3].size(); j++) {
            Node roomnode = (Node) nodes[3].elementAt(j);
            int c = 1;
            for (int k = 0; room[k] != null; k++) {
                if (roomnode.toPlainTextString().equals(room[k]))
                    c = 0;
            }

            Node pretitle = (Node) nodes[0].elementAt(p);
            Node postitle = (Node) nodes[0].elementAt(p + 1);
            Node cnode = (Node) nodes[3].elementAt(j);
            if (postitle.getStartPosition() > cnode.getStartPosition()) {
                s = "Sub" + pretitle.toPlainTextString() + "\r\n";
            }
            if (postitle.getStartPosition() < cnode.getStartPosition()) {
                s = postitle.toPlainTextString() + "\r\n";
                p++;
            }
```

```
No consoles to display at this time.
```

Figure 6.5 MyEclipse Interface

6.4.2 Server's working procedure

The server I used was Tomcat Apache 7.0.23. I deployed my java servlet and JSP pages onto this server. User will send their requests from the JSP page and once the request is submitted, the doPost() method in the servlet will handle the request.

The doPost() method will firstly check if the user uses the web input or uses a picture as a input. If the user uses the web input, then servlet will go check the database according to the user's input directly. Then retrieve the schedule from the database and generate a new html page that containing the feedback to the user.

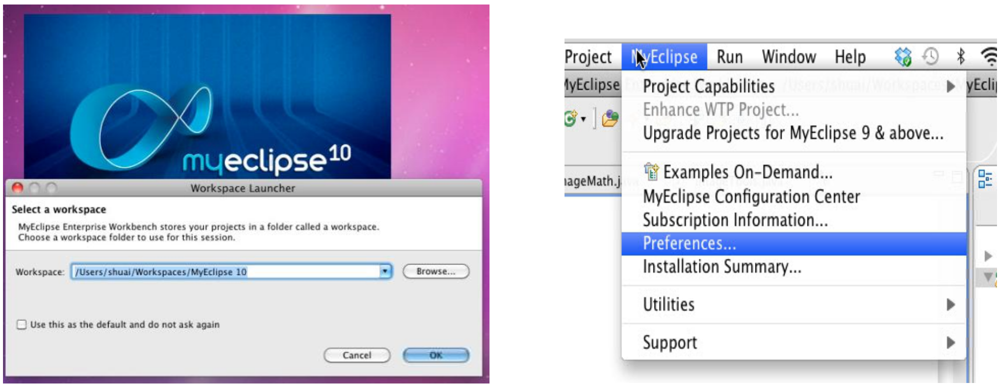
On the other hand, if the user is using a picture as input, the servlet will call a series of methods under image package to do preprocess for the picture and then call recPic() method to do OCR for the picture. Once it gets the recognized information, it will use it to check the database and the rest procedure is the same as the procedure I mentioned in the last paragraph.

6.4.3 Setup procedure

To setup, we need to use Myeclipse to deploy the project.

Step 1: Launch Myelipse, choose the workspace.

Step 2: Click on Myeclipse on the tool bar then choose Preferences.



Workspace Launcher

Select a workspace

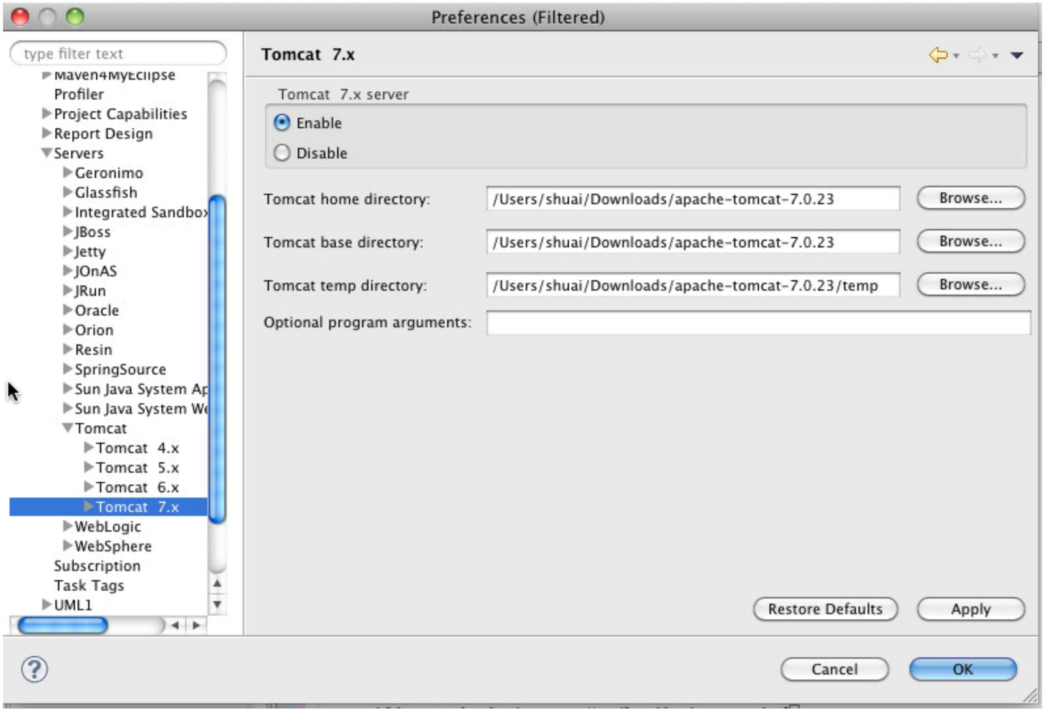
MyEclipse Enterprise Workbench stores your projects in a folder called a workspace.
Choose a workspace folder to use for this session.

Workspace: /Users/shuai/Workspaces/MyEclipse 10

□ Use this as the default and do not ask again

Figure 6.6 Step1,2

Step 3: Set the server configuration as below



Preferences (Filtered)

Tomcat 7.x

Tomcat 7.x server

● Enable

○ Disable

Tomcat home directory: /Users/shuai/Downloads/apache-tomcat-7.0.23

Tomcat base directory: /Users/shuai/Downloads/apache-tomcat-7.0.23

Tomcat temp directory: /Users/shuai/Downloads/apache-tomcat-7.0.23/temp

Optional program arguments:

Figure 6.7 Server configuration

Step 4 Deploy the project to the server. Step into Manage Deployments and then add the project to the tomcat apache server that we set in the step 3. And then start the server

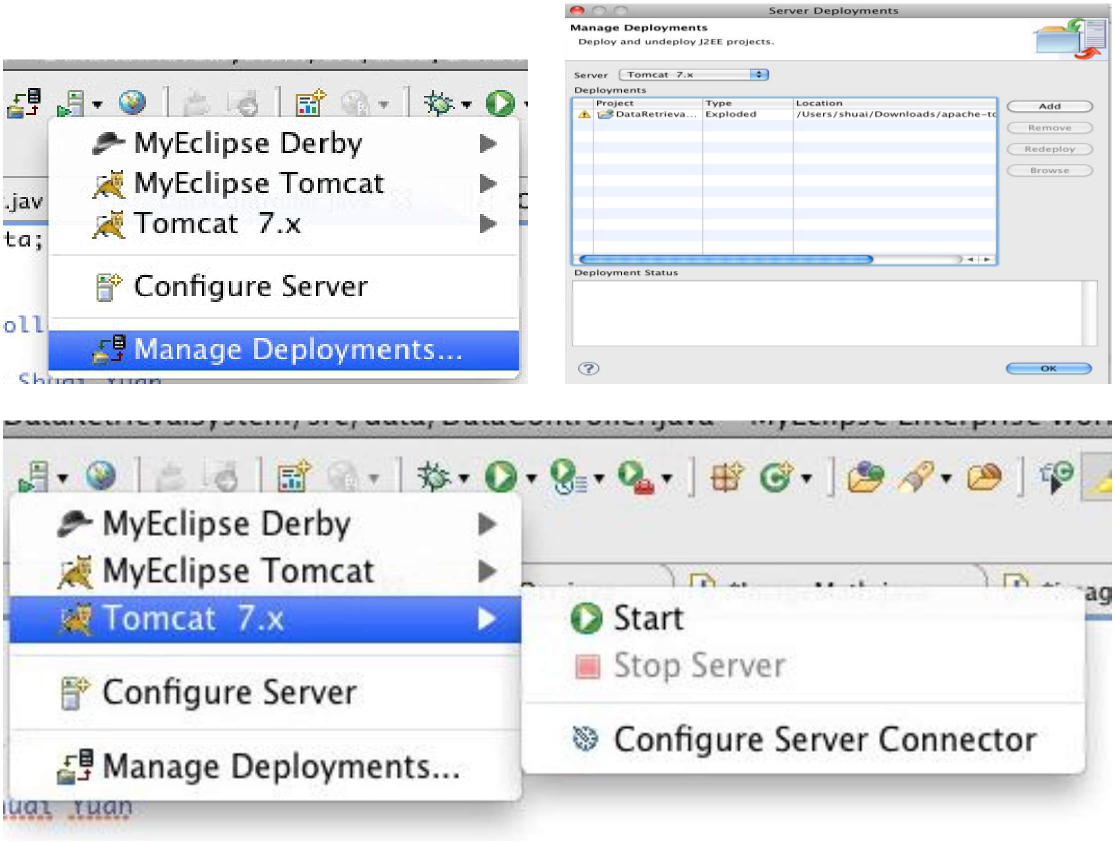
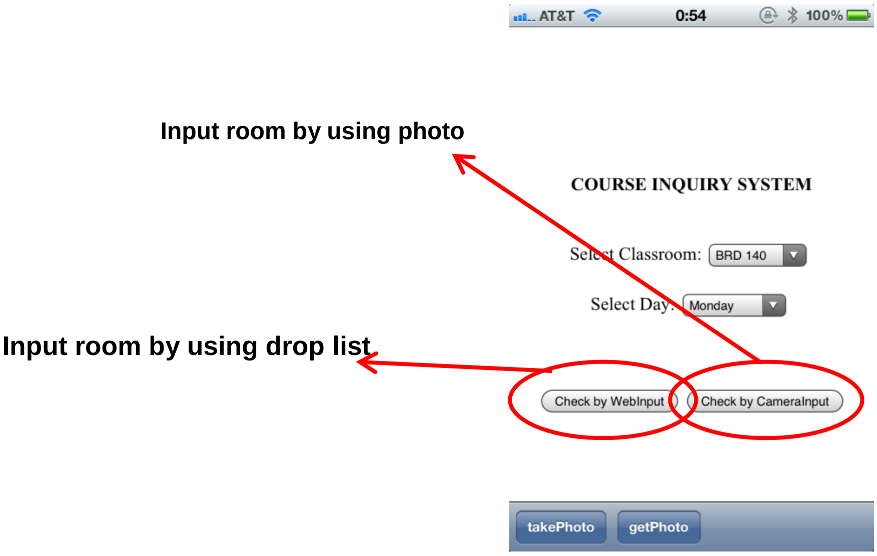


Figure 6.8

6.5 IOS Application

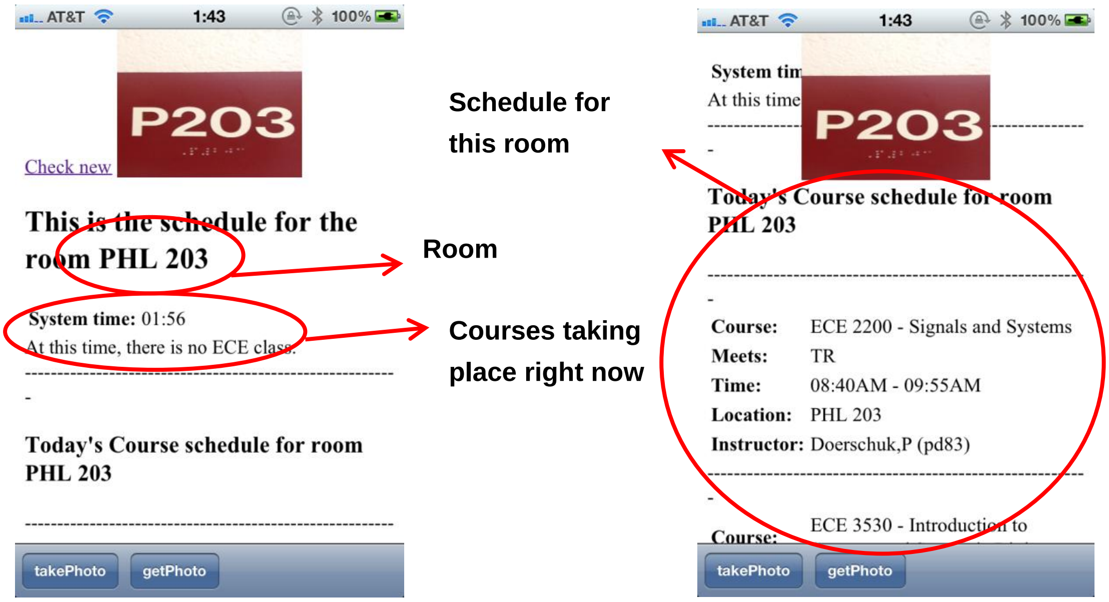
The application was implemented by using Xcode and the basic idea is to embed the web application into the App.

Main UI



Result UI

Result UI includes three parts, which are shown below



Check new

This is the schedule for the room PHL 203

System time: 01:56
At this time, there is no ECE class.

-

Today's Course schedule for room PHL 203

System tim
At this time

-

Today's Course schedule for room PHL 203

-

Course:	ECE 2200 - Signals and Systems
Meets:	TR
Time:	08:40AM - 09:55AM
Location:	PHL 203
Instructor:	Doerschuk,P (pd83)

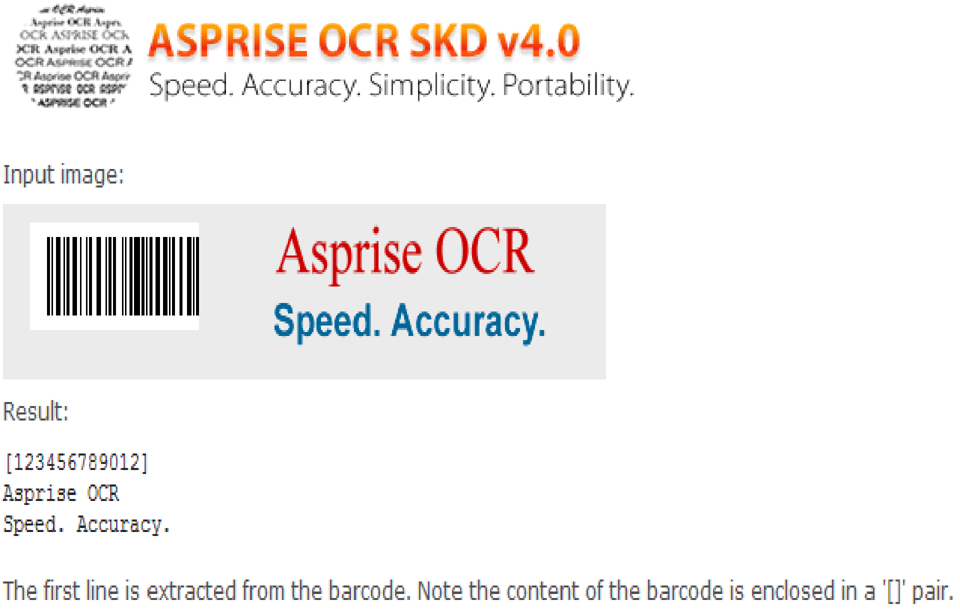
-

Course: ECE 3530 - Introduction to

Figure 6.9 App UI

6.6 OCR

As this project is based on image, so OCR is the key factor. After several trials, I choose to use ApriseOCR.



Input image:

Result:

```
[123456789012]
Asprise OCR
Speed. Accuracy.
```

The first line is extracted from the barcode. Note the content of the barcode is enclosed in a '[]' pair.

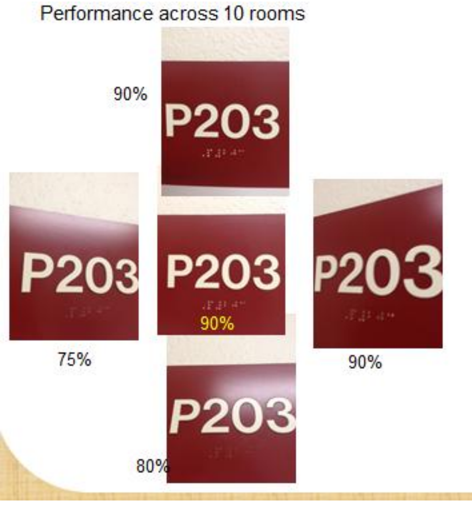
Figure 6.10 OCR demo

Before doing OCR, some preprocess are needed. The image package under the project was used for this purpose. For different buildings, we have different algorithms for preprocessing. There are two types of recognize results, one is the room number and the other one is blank (this happens when user sends an irrelevant picture.

7. Tests and Results

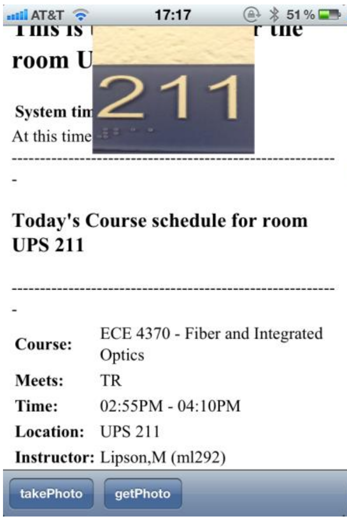
7.1 OCR Evaluation

I evaluated the performance of OCR by testing across 10 rooms and the average recognition rate is 85%. Below is the evaluation from different angles



7.2 Final program test/results

Generally speaking, this program works as expected. It can give a quick response after user sends the picture to server (within 5 seconds). And below are some results, which includes the good cases and fail cases.



This is t r the
room U

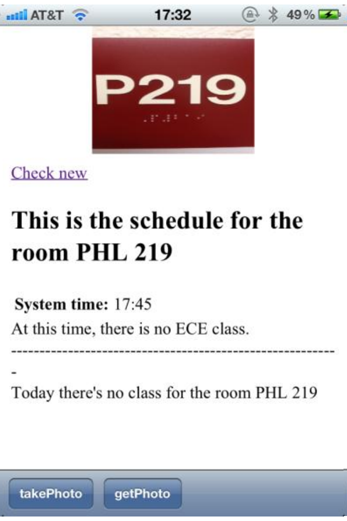
System tim
At this time

-

Today's Course schedule for room UPS 211

-

Course:	ECE 4370 - Fiber and Integrated Optics
Meets:	TR
Time:	02:55PM - 04:10PM
Location:	UPS 211
Instructor:	Lipson,M (ml292)

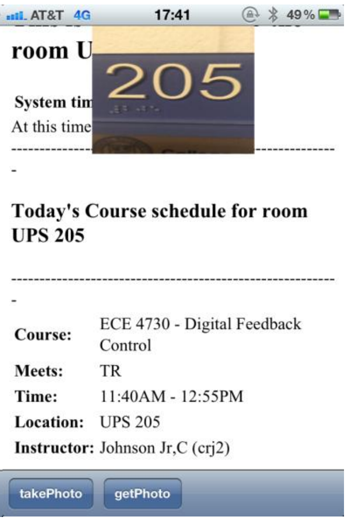


Check new

This is the schedule for the room PHL 219

System time: 17:45
At this time, there is no ECE class.

-
Today there's no class for the room PHL 219



room U

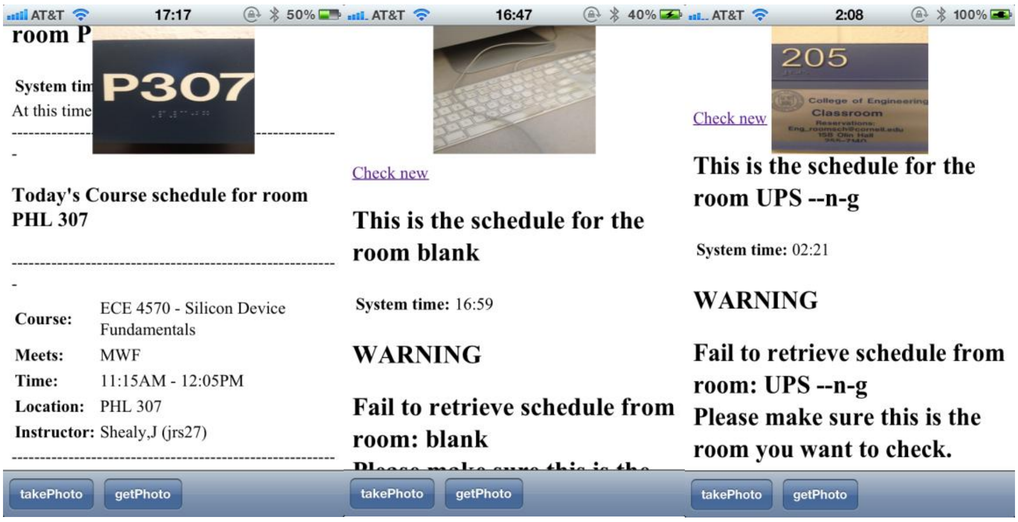
System tim
At this time

-

Today's Course schedule for room UPS 205

-

Course:	ECE 4730 - Digital Feedback Control
Meets:	TR
Time:	11:40AM - 12:55PM
Location:	UPS 205
Instructor:	Johnson Jr,C (crj2)



room P

System tim

At this time

Today's Course schedule for room PHL 307

Course:	ECE 4570 - Silicon Device Fundamentals
Meets:	MWF
Time:	11:15AM - 12:05PM
Location:	PHL 307
Instructor:	Shealy,J (jrs27)

Check new

This is the schedule for the room blank

System time: 16:59

WARNING

Fail to retrieve schedule from room: blank

Please make sure this is the

Check new

This is the schedule for the room UPS --n-g

System time: 02:21

WARNING

Fail to retrieve schedule from room: UPS --n-g

Please make sure this is the room you want to check.

8. Future work

1. **Improve the aesthetics of the app**
2. **Figure out a better way to recognize the different buildings**

   Now this project only focuses on two buildings, which are Phillips Hall and Upson Hall. What I did was to use different algorithms to decode the pictures from different buildings. This will relatively increase the complexity of the code. I hope in future I can figure out a better way to classify different picture from different buildings, for example, GPS.
3. **Extensions to the system**

   This project can be extended into some other query system just by changing the database. So based on this idea, we can add the following functions for this project

   1. Conference room reservations

   2. Professor information

9. Conclusion

To sum up, this project works as expected. The system can be used on Ipad, Iphone and Itouch. The App can help user retrieve the course schedule efficiently. On the other hand, user can also use the web application on all kinds of browsers. Both two types of inputs can perform well. And this system also provides a good expansibility that we can add many new functions to this system only by changing its database and some OCR preprocess algorithms. So this is a really practical project.

References

[1] Tomcat official site http://tomcat.apache.org/download-70.cgi

[2] Tomcat tutorial http://www.coreservlets.com/Apache-Tomcat-Tutorial

[3] Myeclipse http://en.wikipedia.org/wiki/Myeclipse

[4] Xcodehttp://en.wikipedia.org/wiki/Xcode

[5] IOS http://en.wikipedia.org/wiki/Ios

[6] Myeclipse official http://www.myeclipseide.com/module-htmlpages-display-pid-490.html

[7] Java servlethttp://www.oracle.com/technetwork/java/index-jsp-135475.html

[8] ApriseOCR http://asprise.com/product/ocr/index.php?lang=java

[9] ApriseOCR documentationhttp://asprise.com/product/ocr/doc/DevelopersGuide-Java.pdf