



PicAttendance

A Moodle plugin for attendance taking using face recognition

Ruan Ramos Monteiro Silva

Thiago Rosario Caetano

Supervisor: Siome Klein Goldenstein

Final year project for the
degree of Bachelor of Science in Computer Engineering

at the
Institute of Computing
University of Campinas

July 2015

“I don’t know how to stop it, there was never any intent to write a programming language [...] I have absolutely no idea how to write a programming language, I just kept adding the next logical step on the way.”

Rasmus Lerdorf, creator of PHP

Abstract

Institute of Computing
University of Campinas

Final year project - Computer Engineering

In this final year project for the degree of Bachelor of Science in Computer Engineering, we have made a plugin called PicAttendance (Picture Attendance) for the virtual learning environment Moodle, an open source platform widely used around the world in schools and universities. The plugin uses face detection and recognition algorithms from OpenCV to allow the teacher to take students attendance by simply taking a photo and uploading it to Moodle, greatly reducing the time spent by the teacher while still being accurate and cheap. By the end of the semester, we have managed to implement a complete and functional system.

Resumo

Instituto de Computação
Universidade Estadual de Campinas

Projeto Final de Graduação - Engenharia de Computação

Neste projeto final de graduação para o curso de Engenharia de Computação, nós desenvolvemos um plugin denominado PicAttendance (Picture Attendance, em inglês) para o ambiente virtual de aprendizagem Moodle, uma plataforma de software livre amplamente utilizada ao redor do mundo em escolas e universidades. O plugin faz uso de algoritmos de visão computacional da biblioteca de software OpenCV para detecção e reconhecimento de faces, tornando possível para o professor monitorar a frequência dos alunos apenas tirando fotos da turma na sala de aula e as enviando para o sistema Moodle. Desta forma, o tempo gasto pelo professor é reduzido e o controle da presença dos alunos permanece preciso e barato. Ao final do semestre, concluímos a implementação de um sistema completo e funcional.

Contents

Abstract	2
1 Introduction	5
1.1 Moodle	6
1.2 Face detection and recognition	7
1.3 Work division	7
2 Rationale	9
3 Goals	11
4 Development	13
4.1 Programming environment	13
4.2 Moodle specifications	13
4.2.1 Database and filesystem	13
4.2.2 Web pages	15
4.3 Computer vision algorithms	15
4.3.1 PHP-CPP	15
5 Results and use cases	17
5.1 A note on taking photos	17
5.2 Upload training photos (teacher)	18
5.3 Tag training photos (student)	21
5.4 Upload session photos (teacher)	21
5.5 Confirm session attendance (student)	28
5.6 Tag session photo (student)	30
5.7 Approve session tags (teacher)	35
5.8 Untag (student or professor)	38
6 Conclusions and further work	41
6.1 Achievements	41
6.2 Missing points	42
6.3 Ideas for further work	42
A Source code	44

Bibliography **45**

1 Introduction

This work is the final year project required by the University of Campinas for students in computer engineering to be done before graduation. According to the Institute of Computing, the final project is intended to be an activity that synthesizes all the different areas of computer science studied throughout the program. It is a way for students to integrate the knowledge acquired and put it in practice in a semester-long work.

Our project consists of a plugin module for Moodle, an open-source web platform used worldwide to assist teachers in their course management. More specifically, our plugin, called PicAttendance (Picture Attendance), was made to facilitate how teachers record the attendance of students, and the way we do it is by means of face detection and face recognition algorithms. The main idea is that the teacher takes photos of the class and uploads them to Moodle. Using the computer vision algorithms, the plugin is able to identify the students that attended class at the moment the photos were taken, making the attendance tracking process more efficient and less time-consuming for teachers.

A Moodle plugin is made of several components that are common to many web applications. It has some HTML pages that interacts with the user, runs on a web server that processes the requests and connects to a database to store its data. Apart from these general components, PicAttendance has a back end computer vision system that does all the face detection and face recognition processing. In order to integrate all these components, we had to recall concepts from several university courses. In addition to the introductory courses we had for programming, data structures and object-oriented programming, we had to use concepts learned in more advanced courses such as user interfaces, databases, software engineering, computer networks, distributed systems and machine learning, which makes our project appropriate from the perspective of the synthesis of knowledge. Moreover, the project was a great opportunity for us to learn more about web development, especially the front end part, in which neither of us had much experience and it is not a topic covered in any of the courses we took at the university.

In the remainder of this section, we will give some more background information regarding the Moodle platform and the computer vision algorithms used in our project,

followed by how we split the work between the two authors during the semester. Past this Introduction section, the rest of the thesis is structured in five more sections: Section 2, where we discuss the motivation behind this project and why it is relevant; Section 3, in which we describe how we envisioned our plugin; Section 4, in which we describe our whole project set up and the different programming tools needed; Section 5, where we show the final version of our project; Section 6, which gives the reader an idea of what we have learned, what we have managed to achieve in the time we had and some improvement suggestions as ideas for further work. This thesis ends with an appendix with URLs to the source code repositories of this work.

1.1 Moodle

Moodle is a virtual learning environment (VLE), which consists of a platform on the web designed to be a management system for educational institutions, such as schools, universities and even workplaces. It is free and open source, and its more than 56 million users around the world make Moodle the most widely used platform of its kind. The most recent statistics show that it has, currently, 46,603 registered sites in 214 countries. [1]

One of the main features that allowed Moodle to grow was its customization capability. The name *Moodle* is an acronym for *modular object-oriented dynamic learning environment*, which describes the system quite well. The platform is composed by different modules that can be added to the system core in the form of plugins based on what the administration decides to use. Thus Moodle is built mostly from the modules created by the open-source community that supports it, and the Attendance plugin [2], which is the source code from which our plugin was developed, is one of the existing modules in this network.

The University of Campinas has been using Moodle for about three years. Since 2015, Ensino Aberto, the internet-based system used at the University of Campinas to manage undergrad courses, started using Moodle as its VLE along with TelEduc, which was the original one and is still active. Moodle at University of Campinas seemed to be well accepted among students and teachers. The Institute of Computing alone has used Moodle for over twenty courses now. Given its worldwide reputation and local use in our university, Moodle seemed to us to be a good platform for our application, and that is why we decided to make a plugin for it.

1.2 Face detection and recognition

Within the context of computer vision, *face detection* is the problem of determining whether a given image contains human faces and, if it does, where. A very fast and effective approach to this problem is the Haar-like feature based cascade classifier first proposed by Viola and Jones [3] and improved by Lienhart [4]. The classifier is trained with many images of faces as positive examples and arbitrary images of the same size as negative examples. The word *cascade* means that the classifier is actually built of many simpler classifiers which are applied in sequence to every part of the image in search for a face.

Face recognition is the problem of, given a database of labeled faces and an unlabeled face, determining whether the unlabeled face belongs to one of the subjects in the database and, if it does, to whom. In a real-world application of face recognition, such as the present work, face detection is a necessary first step to face recognition, because the raw input images are usually arbitrary instead of controlled, standardized images of a single labeled face. There are several fundamentally different approaches to face recognition and the research in this area is very active to this day.

OpenCV (Open Source Computer Vision Library) [5] is an open-source, feature-rich, widely available and widely used software library for computer vision and machine learning written in C++. OpenCV has high-quality, ready-to-use implementations of more than 2500 algorithms, among which face detection and face recognition algorithms are an example. In this work, we chose to use OpenCV for its fast C++ implementation, permissive licence and active development. We used OpenCV's face detection and recognition as black boxes we built upon, instead of writing our own implementations. This resonates with the philosophy of this work being a synthesis and an integration of our computer science knowledge instead of a focused exercise in computer vision.

OpenCV currently offers three implementations of face recognition algorithms. Among them, we chose to use local binary pattern histograms (LBPH) [6] over the alternatives because it allows for a model to be updated with new images without a complete retraining. LBPH is also robust to changes in lighting (monotonic gray scale transformations).

1.3 Work division

One of the prerequisites for doing the final year project as a team was that we needed a clear division of the work. When we started planning the tasks for each of the authors, our initial idea of division was OpenCV-related, C++ programming for Caetano and

Moodle-related, PHP programming for Silva, and that is how we began. Caetano started learning about OpenCV, writing code for face detection and recognition, while Silva started working on the Moodle front, learning how a plugin is defined and its internal structure. After finishing writing the computer vision functions in C++, Caetano worked on writing a PHP interface for them so that they could be used by our Moodle plugin. After that part was finished, Caetano joined Silva on the work of the Moodle plugin, especially on coding some of the front end pages. Despite our work division, the whole project was a team effort, with frequent discussions about design and implementation. A rough list of the tasks and the author responsible for it is as follows.

1. Set up of the Linux virtual machine used in our project, including the installation of PHP, Apache, MySQL and Moodle - Ruan
2. Modelling and coding of the database tables - Ruan
3. Structuring and coding functions to access Moodle filesystem - Thiago
4. C++ functions for face detection and recognition and image processing using OpenCV - Thiago
5. PHP interface for face detection, recognition and cropping images - Thiago
6. Train view (both teacher and student sides) - Ruan
7. Session view (teacher side) - Ruan
8. Student profile view - Both
9. Attendance page (student side) - Ruan
10. Tagging page (student side) - Thiago
11. Approve view (teacher side) - Ruan
12. Writing thesis - Both

The responsible for the whole idea of designing and implementing this Moodle plugin for automated attendance taking was our supervisor, Siome Goldenstein. Our discussions with him were also of invaluable help to many design decisions of our application.

2 Rationale

The first question that deserves an answer is why should a teacher bother to take attendance at all. Despite the skepticism of students and some teachers alike, there is a significant body of research concerning class attendance and its relation to student performance. Ever since Romer [7] found quantitative evidence that “the difference in performance between a student who attends regularly and one who attends sporadically is about a full letter grade,” a substantial number of experiments has been performed and mostly reached the same conclusion. Marburger [8] went on to ask whether *mandatory* attendance *improved* student performance, and the answer was yes. More recently, Credé et al. [9] performed a thorough meta-analytic review of the available research. This review not only suggests a strong relationship between class attendance and college grades, but also states that class attendance appears to be a better predictor of college grades than any other known predictor, such as SAT grades, high school GPA, studying skills and amount of time spent studying. Moreover, the relationship is so strong that efforts to increase class attendance could have a large impact on average grades.

Another, pragmatic reason to care about class attendance is that often the teacher does not have a choice. In Brazil particularly, class attendance is a matter of federal law. The current law which regulates education in Brazil [10, 11] says that attendance is mandatory both for basic (primary and secondary) and higher education, except for distance courses. Specifically, students must attend at least 75 percent of all classes. As a consequence of the law, the undergraduate statute of University of Campinas [12] states that all courses must define a minimum attendance requirement of at least 75 percent.

Regardless of the reason, a teacher who chooses to take attendance will most likely be faced with a tradeoff among accuracy, cost and effort. Roll (register) calling is cheap and accurate for small classes but is prohibitively time-consuming for large university classes having more than 100 students. Passing an attendance sheet around is cheap and scales well with class size but is very vulnerable to fraud. Biometric attendance systems using fingerprint readers are accurate but costly. Our aim with PicAttendance

is to develop a solution that is free (as in freedom *and* as in free lunch), accurate and that minimizes the work performed by the teacher and the time spent with attendance during class.

3 Goals

The main goal of this project is to make the student attendance taking easy and less time-consuming for teachers. After we decided we wanted a Moodle plugin that automated this process using computer vision algorithms from OpenCV, we set some more specific goals for our project, described below.

- Create a training session

The face recognition method works by using previously uploaded photos that already have a corresponding set of students associated with the faces on them. Once we did some testing with the OpenCV module, we realized that if we only fed the system with the photos from previous sessions it would take around seven of them for the method to start working well, which could mean half the semester for some courses. So we decided to create a train page that would allow the teacher to upload more photos from class that are not meant for attendance, just for training the face recognition system. This way the plugin would be able to recognize students since the first session if properly trained, lessening the time spent by the instructor on it during the semester.

- Allow students to tag themselves on photos

When the teacher uploads a training photo, the students need the ability to tag themselves on it so that the system learns what their faces look like. Also, in case the face recognition system fails, the student has to tag himself on the class photo so that he confirms his attendance for that particular session. For both these cases a tagging page is necessary.

- Create an approval page

Giving to a student the ability to tag himself on photos for recording attendance means that the instructor needs to approve it, because the system could be easily deceived otherwise. So we designed an approval page, in which the instructor could see each tag created along with the previous face images of that same student. By

comparing them, the instructor is able to decide if the student actually attended the session or not. Maintaining the principle of making the instructor job easier, in case the system successfully recognizes a student, an approval is not necessary.

- Integrate OpenCV algorithms of face detection and recognition into PHP

In order to use the OpenCV library, which is written in C++, to build the face detection and recognition methods, we also needed a PHP extension for wrapping them so that they could be executed inside our Moodle plugin.

- Support small and large classes alike

The plugin needs to be efficient enough to work well with classes that have a large number of students and sessions.

- Allow image sharing between courses

If face images are shared among courses, then new courses added to the system will not need as much training.

- Easy to deploy and use

Achieving our goals will not matter if users do not know how to install the plugin or do not understand the different pages we created.

Our vision for PicAttendance is not merely a system that receives images and sends them to a face detection and recognition system. It is also a tagging system, that contains distinct pages for students and instructors with different kinds of interaction.

In the following sections we will describe in detail the development of the project and the results we achieved. In Section 6, we will analyze which goals were achieved and to what degree.

4 Development

In this section we will describe in more detail how the different components of PicAttendance were implemented, including programming tools, languages and design specifications.

4.1 Programming environment

Our set up for developing and testing the project was a Linux virtual machine provided by Koding [13]. The VM ran Ubuntu 14.04 and was hosted on Amazon. It was always on and allowed shared collaboration, which made for a great development environment. Regarding revision control, we used Git [14]. We also hosted the project on GitHub [15] to facilitate public access to our codebase and documentation. (See Appendix A.)

4.2 Moodle specifications

With the purpose of being able to deploy our application on the server of the University of Campinas in the future, we used Moodle 2.8, which is the same version in use there currently. The Moodle version is important because all the core APIs including the ones needed to structure the web pages (Form API), to communicate with the database (Data Manipulation API) and with the filesystem (File API) may change from version to version, although our application should not have any problems running on the latest version (2.9).

4.2.1 Database and filesystem

In order to be able to receive images and to generate data about them, such as which session each photo refers to, which students are present in the photo and so on, we had to create a database to store the data and a filesystem to save the images.

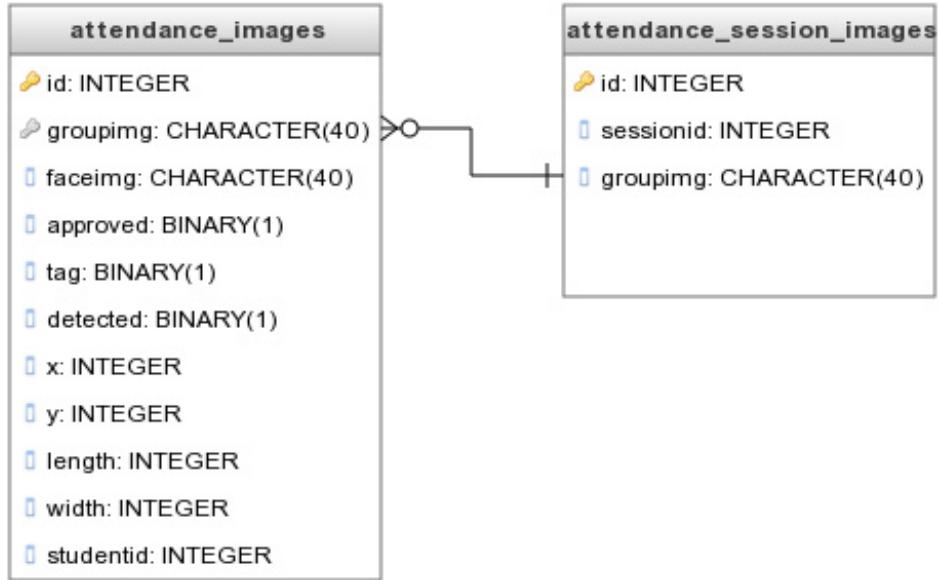


FIGURE 4.1: Database diagram

For the database, we used MySQL (version 5.5.43) [16]. We created two database tables (`attendance_images` and `attendance_session_images`), as shown in Figure 4.1. Table `attendance_images` associates each photo of the class taken by the instructor (field `groupimg`) with one of the faces detected in that image (`faceimg`), the coordinates of the rectangle that surrounds that face (`x`, `y`, `length` and `width`) and the id of the student recognized from it (`studentid`).

What is stored in the database are not the images themselves, but a hash code (SHA-1) of the image files. Given that we store a small number of images (an average course would hardly have more than 100 group photos) and how good SHA-1 is, we assumed that the probability of collisions would be insignificant and used the hash codes of the images as their names so that they could be uniquely identified. So our filesystem is structured in a very simple way: each course has a folder with all its group images and their corresponding face images, all named using their calculated hash code. The folders are inserted into Moodle's filesystem, and the way the image files are accessed is by using Moodle's File API.

Continuing with the description of the database tables, the last three fields of the main table (`approved`, `tag` and `detected`) are important flags that determine the state of the corresponding face image so that the approving mechanisms and the tagging page could be implemented. If `approved` = 1, then the student of that image has been recognized and its attendance has been recorded. If `tag` = 1, then the association between the image and the student id was made using the tagging system instead of the face recognition

method. If *detected* = 1, then the face coordinates were set by the face detection system instead of being manually marked by a student. The *detected* flag is particularly important to determine which face images are used in the training of the recognition system, because if a face could not be detected, then its features are probably not favorable to the recognition software, meaning that they would not help the system to identify the student in question.

The second table - `attendance_session_images` - associates each group image name with its corresponding session id. This table is essential to modify the attendance status of a student on a particular session. Also, both tables have an auto-incrementing integer field *id* as primary key, which is a prerequisite for every database table on moodle.

4.2.2 Web pages

The construction of the web pages were primarily made by using the *html_writer* class, that allows the user to output simple HTML tags, and Moodle's Form API. Besides a little bit of CSS, we also used JavaScript and jQuery (external library) to create the tagging page and the untag button. The final layout can be seen in Section 5.

4.3 Computer vision algorithms

As mentioned before, we chose the OpenCV library [5] to implement the face detection and recognition functions, mostly because of its high performance implementation in C++.

Our C++ code uses essentially four features from OpenCV: face detection (class *CascadeClassifier*), face recognition (class *FaceRecognizer*), cropping (class *Mat*, member function *Mat::operator()*) and encoding/decoding (functions *imencode* and *imdecode*).

When the student tags herself by drawing a square around her face, we needed a metric to match the square that was drawn to the squares corresponding to faces detected by the system. We used the intersection over union metric, which is widely used in computer vision.

4.3.1 PHP-CPP

After choosing C++ as the programming language for the vision methods, we needed a way to call C++ code from PHP. The mechanism offered by PHP for that is to create a *PHP extension*. PHP extensions are usually written in C. In order to write it in C++,

we used the PHP-CPP library [17], which allows the use of modern C++ and greatly simplifies the interface between the languages.

5 Results and use cases

We have successfully implemented a set of use cases which provide the functionality we envisioned for PicAttendance. Sections 5.2 through 5.8 present each of these use cases with a detailed, step-by-step guide and screenshots. This presentation serves three purposes:

1. To show the concrete results of our work in this project
2. To be a specification of the software requirements by means of use case descriptions
3. To document the implemented software

Each of these sections is titled like this: *Use case title (actor)*.

PicAttendance was developed by modifying the Attendance Moodle plugin. (See Section 4.) We only show here use cases that we have developed from scratch or substantially modified from Attendance. As such, some essential functionality (e.g. adding a new session) is not shown here.

Throughout this section, we assume that Moodle and PicAttendance are properly installed and running and that a course is created. (See Appendix A.) Most use cases start from the PicAttendance page within the course. In order to reach this page, go to the course page and click *Attendance*. (We intend to change this to *PicAttendance* in the future for clarity.)

Screenshots show the mouse cursor pointing to the element of interest. All faces in photos other than Silva's (one of the authors) have been pixelated.

5.1 A note on taking photos

For best results when taking photos of the class, teachers should ask students to look directly at the camera and arrange themselves in such a way that the face of every

student is entirely visible in the photo. Asking the students to take off their glasses and hats for a moment and maintaining a neutral expression is also helpful.

PicAttendance can handle multiple photos of a single session (class). This feature is supposed to be used whenever a single photo of everyone would be too difficult to take or the resolution of each face in a single photo of everyone would be too low.

5.2 Upload training photos (teacher)

Like any other machine learning technique, face recognition requires *training* to work. Training a face recognition system means feeding it with a set of photos together with a corresponding set of correct *labels* telling whose face each photo depicts. PicAttendance is programmed to use the photos of each session (class) and the corresponding tags made by students as training data for the following classes. This means that its recognition performance is likely to increase along the semester, but also that it will probably perform very badly in the first few classes. To address this issue, PicAttendance allows the teacher to upload photos that are meant just for training and not for attendance. This can be done at any point during the course, but it is of course more effective in the beginning, preferably in the very first class.

Do not use the same photo for training and attendance.

1. At some point during class, take group photos of the students. (See Section 5.1.)
You may take as many training photos as you want (even none at all), but if this is the first class and you want the recognition to be up and running now, you should take five or so. The photos should be slightly different from each other. To achieve that, you may walk around a little and take them from different angles. The students should always be looking directly at the camera.
2. After class, go to the PicAttendance page on Moodle. Select the *Train* tab if it is not already selected. (See Figure 5.1.)
3. Click *Choose File*. (See Figure 5.2.)
4. In the dialog that appears, browse to the photo and select it. (See Figure 5.3.)
(This dialog is different depending on your operating system.)
5. Click *Upload Image* and wait until the page reloads. (See Figure 5.4.) You may repeat these steps to upload as many training pictures as necessary.
6. Instruct the students to tag themselves on training photos on Moodle. (See Section 5.3.)

The screenshot shows a Moodle-based attendance management interface. The browser title is "test_small: Attendance". The URL in the address bar is "rrms.koding.io/moodle/mod/attendance/train.php?id=65". The top navigation bar includes links for "Perfil padrão", "Ruan Silva", and a user menu. The main title is "Test course: S". Below it, the breadcrumb trail shows: Home > Courses > Miscellaneous > test_small > General > Attendance > TRAIN. On the right, there is a button labeled "Update this Attendance". The left sidebar, titled "NAVIGATION", lists "My home", "Site pages", "My profile", "Current course" (with "test_small" expanded), "General" (with "Small files", "Forum", and "Attendance" listed), and "Courses". The "Attendance" link under "General" is highlighted. The main content area is titled "Attendance for the course :: Test course: S" and contains tabs for "Sessions", "Add", "Report", "Export", "Settings", "Train" (which is selected and highlighted in blue), and "Approve". A file input field "PicAttendance:" with the placeholder "choose File" and a "No file chosen" message is present, along with a "Upload Image" button. At the bottom of the page, an "ADMINISTRATION" section is visible with a link to "Attendance administration".

FIGURE 5.1: Upload training photos (teacher) (1 of 4)

This screenshot is identical to Figure 5.1, showing the Moodle Attendance page for "Test course: S". The browser title is "test_small: Attendance" and the URL is "rrms.koding.io/moodle/mod/attendance/train.php?id=65". The top navigation bar, main title, breadcrumb trail, and right-side "Update this Attendance" button are all the same. The left sidebar and main content area also remain identical, featuring the "Train" tab as the active selection. The "Attendance" link in the sidebar is still highlighted.

FIGURE 5.2: Upload training photos (teacher) (2 of 4)

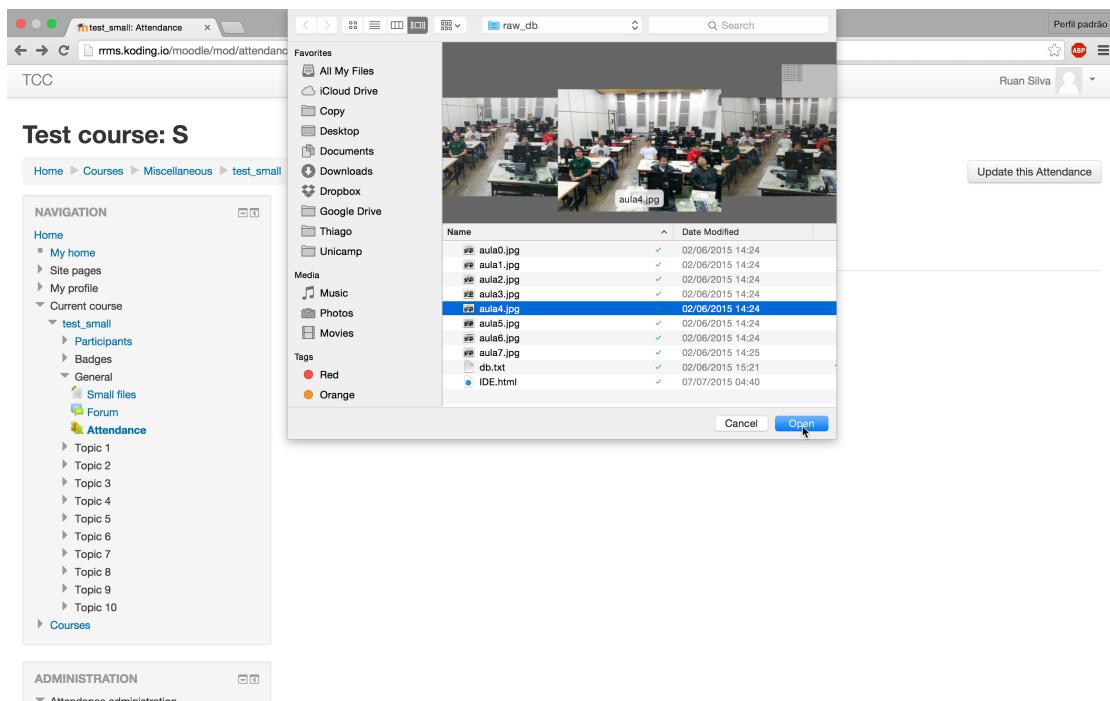


FIGURE 5.3: Upload training photos (teacher) (3 of 4)

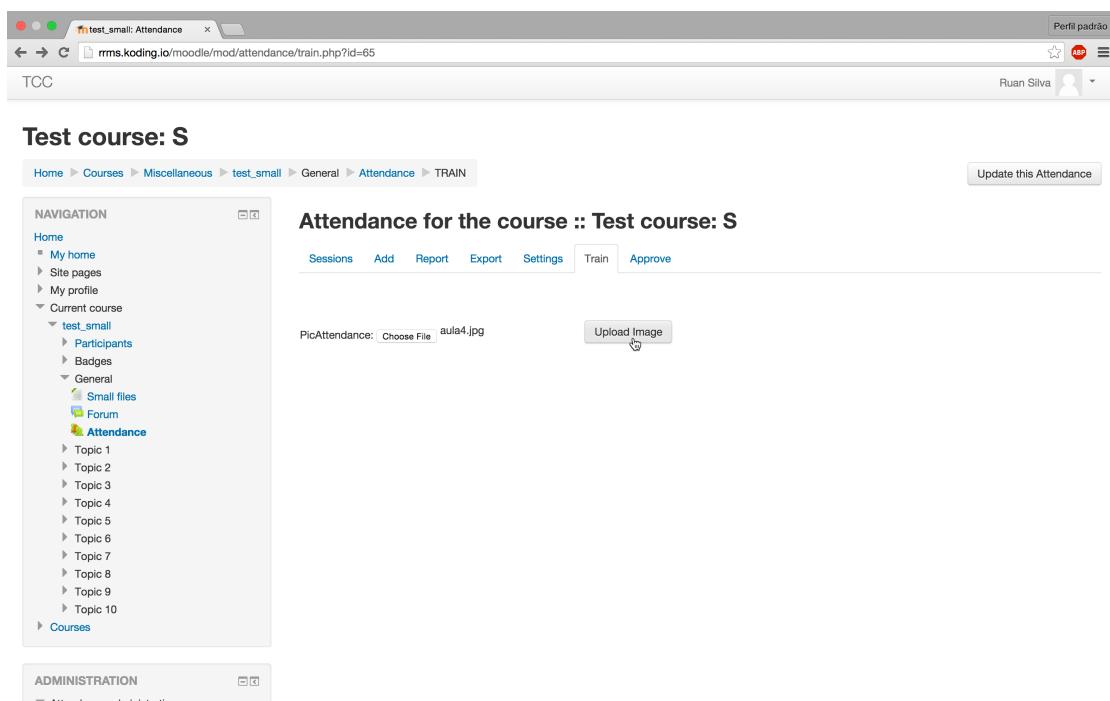


FIGURE 5.4: Upload training photos (teacher) (4 of 4)

The screenshot shows a Moodle attendance page for a course named 'test_small'. The student profile for 'Ruan Ruan' is displayed, showing sessions completed: 0 (Present: 0, Absent: 0), an attendance grade of 0.00, and an attendance percent of 0.00. The table below lists 8 regular class sessions from May 15 to July 15, all marked as 'Common' type. The last session on July 15 has a 'Submit attendance' link next to it.

#	Type	Date	Time	Description	Status	Remarks
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session		?
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session		?
3	Common	29.05.15 (Fri)	20:55	Regular class session		?
4	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session	Submit attendance	
5	Common	15.06.15 (Mon)	16:25 - 17:25	Regular class session		?
6	Common	17.06.15 (Wed)	16:25 - 17:25	Regular class session		?
7	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session	Submit attendance	
8	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Submit attendance	
9	PicAttendance: Training					

FIGURE 5.5: Tag training photos (student) (1 of 6)

5.3 Tag training photos (student)

1. Go to the PicAttendance page on Moodle. Select the *This course* tab if it is not already selected. (See Figure 5.5.)
2. Click *PicAttendance: Training*. (See Figure 5.6.)
3. Check all photos that show your face. (See Figure 5.7.)
4. Click *Save changes*. (See Figure 5.8.)
5. You will be redirected back to the PicAttendance page, and all your training photos will be displayed. (See Figure 5.9.) If you made a mistake, you can always untag a photo. (See Section 5.8.)

Each face photo may only be tagged by one student. If you or another student clicks *PicAttendance: Training* after you have tagged your photos, your photos will not be there anymore. (See Figure 5.10.)

5.4 Upload session photos (teacher)

1. At some point during class, take group photos of the students. (See Section 5.1.)

The screenshot shows a Moodle attendance report for a student named Ruan Ruan. The URL is rms.koding.io/moodle/mod/attendance/view.php?id=65. The page title is "Test course: S". The navigation bar includes links for Home, My courses, Miscellaneous, test_small, General, Attendance, and Attendance report.

NAVIGATION:

- Home
- My home
- Site pages
- My profile
- Current course
- test_small
- Participants
- Badges
- General
- Small files
- Forum
- Attendance
- Topic 1
- Topic 2
- Topic 3
- Topic 4
- Topic 5
- Topic 6
- Topic 7
- Topic 8
- Topic 9
- Topic 10
- My courses

ADMINISTRATION:

- Course administration

Ruan Ruan:

Sessions completed: 0
Present: 0
Absent: 0
Attendance grade: 0 / 0
Attendance percent: 0.00%

Attendance Record:

#	Type	Date	Time	Description	Status	Remarks
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session	?	
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session	?	
3	Common	29.05.15 (Fri)	20:55	Regular class session	?	
4	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session	Submit attendance	
5	Common	15.06.15 (Mon)	16:25 - 17:25	Regular class session	?	
6	Common	17.06.15 (Wed)	16:25 - 17:25	Regular class session	?	
7	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session	Submit attendance	
8	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Submit attendance	
9	PicAttendance: Training					

FIGURE 5.6: Tag training photos (student) (2 of 6)

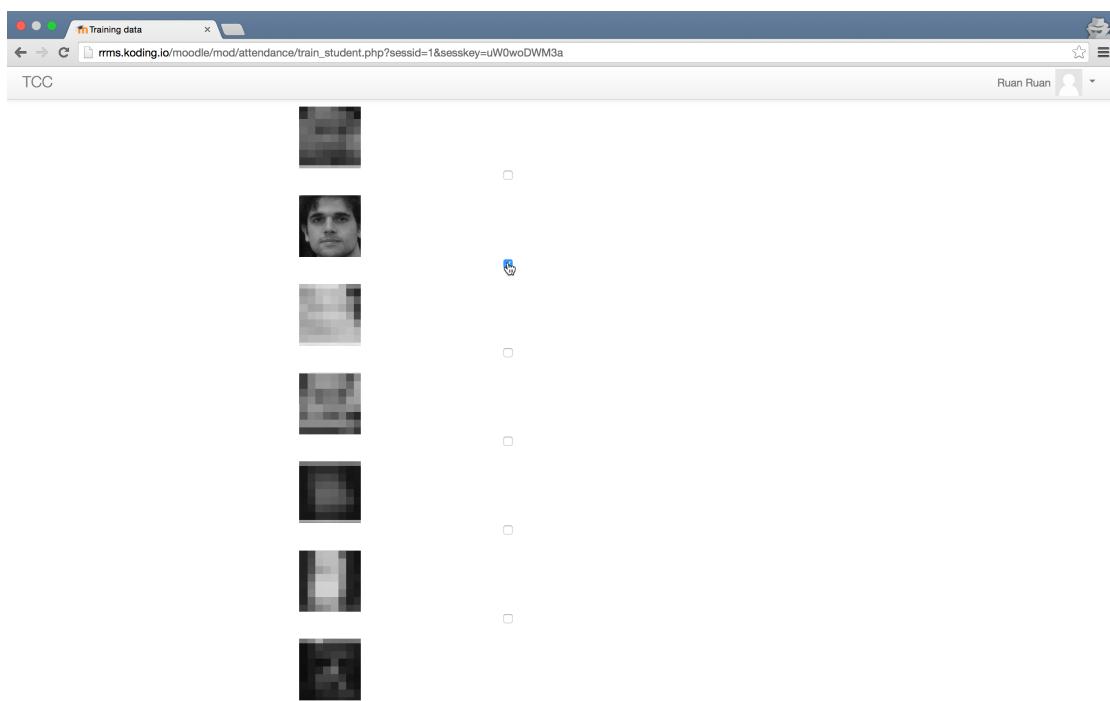


FIGURE 5.7: Tag training photos (student) (3 of 6)

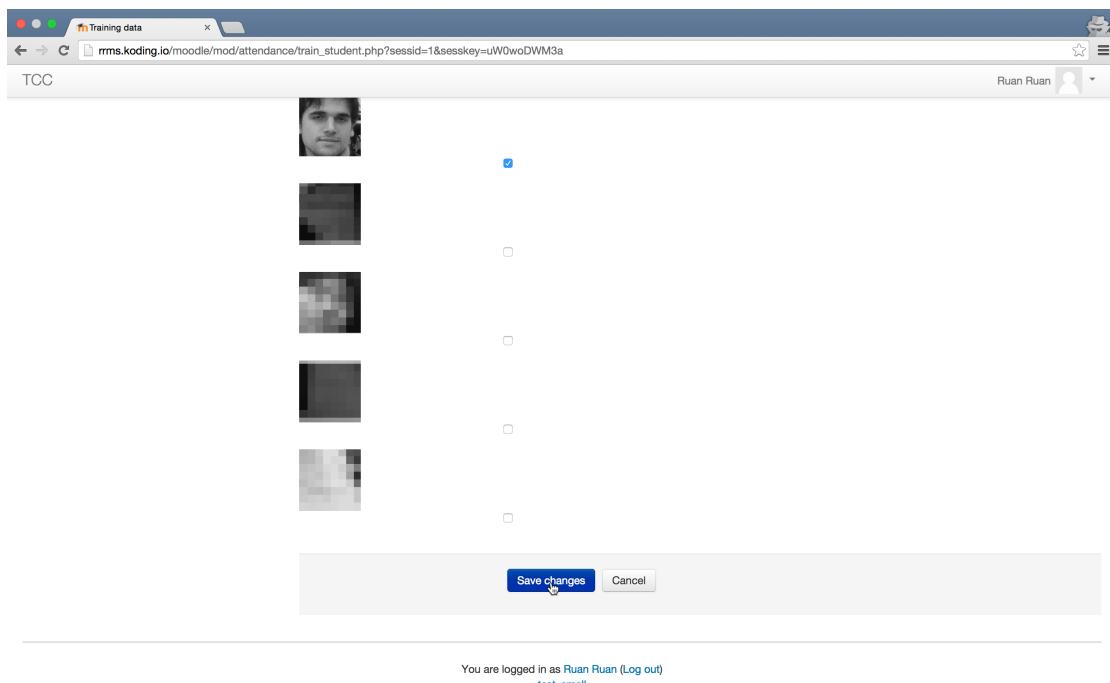


FIGURE 5.8: Tag training photos (student) (4 of 6)

#	Type	Date	Time	Description	Status	Remarks
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session		?
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session		?
3	Common	29.05.15 (Fri)	20:55	Regular class session		?
4	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session		Submit attendance
5	Common	15.06.15 (Mon)	16:25 - 17:25	Regular class session		?

FIGURE 5.9: Tag training photos (student) (5 of 6)

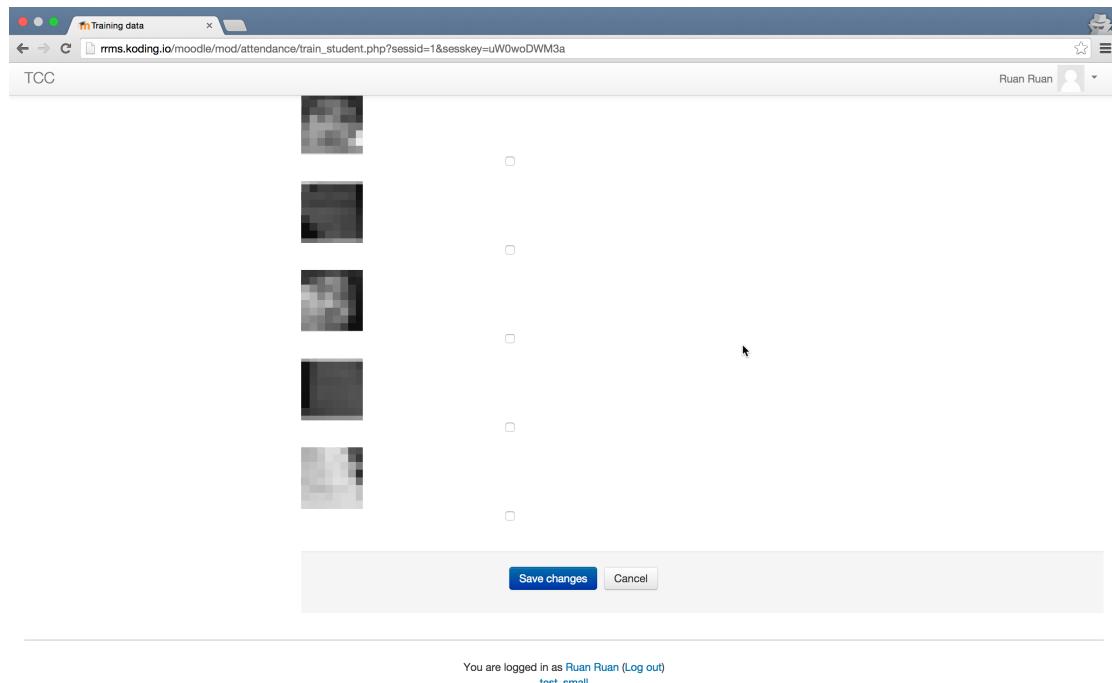


FIGURE 5.10: Tag training photos (student) (6 of 6)

2. After class, go to the PicAttendance page on Moodle. Select the *Sessions* tab if it is not already selected. (See Figure 5.11.)
3. Find the session corresponding to the class whose pictures you want to upload and click the green icon to the right. (See Figure 5.12.) You may have to browse among the available sessions in order to find it. (We assume you have already created the session corresponding to this class. If you have not, you must do this before you can upload photos by selecting the *Add* tab. Adding sessions is not documented here because it has not been changed from the Attendance plugin.)
4. Click *Choose File*. (See Figure 5.13.)
5. In the dialog that appears, browse to the photo and select it. (See Figure 5.14.) (This dialog is different depending on your operating system.)
6. Click *Upload Image*. (See Figure 5.15.)
7. After the upload is complete, the photo will be displayed in the session page. (See Figure 5.16.) You may repeat these steps to upload as many pictures of the session as necessary.
8. Instruct the students to confirm their attendance on Moodle. (See Section 5.5.)

The screenshot shows the Moodle Attendance page for the course 'Test course: S'. The navigation bar includes links for Home, Courses, Miscellaneous, test_small, General, Attendance, and Attendance. A top banner says 'Perfil padrão' and shows a user profile for 'Ruan Silva'. The main content area is titled 'Attendance for the course :: Test course: S' and displays a table of sessions for May. The table columns are #, Type, Date, Time, Description, and Actions. There are three sessions listed:

#	Type	Date	Time	Description	Actions
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session	
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session	
3	Common	29.05.15 (Fri)	20:55	Regular class session	

Below the table, there is a note: 'Hidden sessions: 0' and buttons for 'Choose...' and 'OK'.

FIGURE 5.11: Upload session photos (teacher) (1 of 6)

This screenshot is identical to Figure 5.11, showing the Moodle Attendance page for 'Test course: S'. The table of sessions is the same. However, the 'Date' column for the first session (27.05.15) now has a tooltip 'Change attendance' displayed above it.

FIGURE 5.12: Upload session photos (teacher) (2 of 6)

The screenshot shows a Moodle attendance page for a course named "Test course: S". The page title is "Attendance for the course :: Test course: S". The navigation bar includes links for Sessions, Add, Report, Export, Settings, Train, and Approve. On the left, there's a sidebar with a navigation menu and an administration section. The main content area displays a list of students with their first names and surnames. A file upload dialog is overlaid on the page, showing a thumbnail of a classroom photo and file details: "aula5.jpg", "JPEG - 2,1 MB", "Created: 02/06/2015 14:24", "Modified: 02/06/2015 14:24", "Last opened: 02/06/2015 14:24", and "Dimensions: 3264 x 2448". There are buttons for "Cancel" and "Open".

FIGURE 5.13: Upload session photos (teacher) (3 of 6)

This screenshot is identical to Figure 5.13, showing the same Moodle attendance page and file upload dialog. The list of students and the file details for "aula5.jpg" are the same.

FIGURE 5.14: Upload session photos (teacher) (4 of 6)

The screenshot shows a Moodle attendance page for a course named 'Test course: S'. The URL is rrms.koding.io/moodle/mod/attendance/take.php?sessionid=1&groupype=0&id=65&page=1. The page title is 'Attendance for the course :: Test course: S'. The navigation bar includes links for Home, Courses, Miscellaneous, test_small, General, Attendance, and Attendance. A 'Update this Attendance' button is visible.

NAVIGATION:

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- Courses

ADMINISTRATION:

- Attendance administration
 - Edit settings
 - Locally assigned roles
 - Permissions
 - Check permissions
 - Filters
 - Logs
 - Backup
 - Restore
- Course administration

Attendance Session Details:

27 May 2015 16:25 - 17:25
Regular class session

Attendance Table:

#	First name / Surname	P	A	Remarks
1	Hanna Hoffmann	<input type="radio"/>	<input type="radio"/>	
2	Tereza Horáková	<input type="radio"/>	<input type="radio"/>	
3	Ethan Johnson	<input checked="" type="radio"/>	<input type="radio"/>	
4	Jacob Johnson	<input checked="" type="radio"/>	<input type="radio"/>	
5	Michael Johnson	<input type="radio"/>	<input type="radio"/>	
6	William Johnson	<input type="radio"/>	<input type="radio"/>	

File Upload Options:

Page 1 of 4 ►
View mode PicAttendance:
Sorted list

FIGURE 5.15: Upload session photos (teacher) (5 of 6)

The screenshot shows a Moodle attendance page for a course named 'Test course: S'. The URL is rrms.koding.io/moodle/mod/attendance/take.php?id=65&sessionid=7&groupype=0. The page title is 'Attendance for the course :: Test course: S'. The navigation bar includes links for Home, Courses, Miscellaneous, test_small, General, Attendance, and Attendance. A 'Update this Attendance' button is visible.

NAVIGATION:

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
 - Courses

ADMINISTRATION:

- Attendance administration
 - Edit settings
 - Locally assigned roles
 - Permissions
 - Check permissions
 - Filters
 - Logs
 - Backup
 - Restore
- Course administration

Attendance Session Details:

6 July 2015 15:05 - 17:05
Regular class session

Attendance Table:

#	First name / Surname	P	A	Remarks
1	Hanna Hoffmann	<input type="radio"/>	<input checked="" type="radio"/>	
2	Tereza Horáková	<input type="radio"/>	<input type="radio"/>	
3	Paulo Oliveira	<input type="radio"/>	<input type="radio"/>	

File Upload Options:

Page 1 of 4 ►
View mode PicAttendance:
Sorted list

Classroom Photo:

FIGURE 5.16: Upload session photos (teacher) (6 of 6)

The screenshot shows a Moodle course named 'test_small' with a sub-page titled 'Attendance'. The URL is rrms.koding.io/moodle/mod/attendance/view.php?id=65. The page title is 'Test course: S'. The navigation bar includes 'Home', 'My courses', 'Miscellaneous', 'test_small', 'General', 'Attendance', and 'Attendance report'. The 'Attendance' tab is selected. The main content area is titled 'Ruan Ruan' and displays a message: 'These are the photos we think it's you. Click on one to untag yourself.' Below this are seven small profile pictures of a man. Statistics show 'Sessions completed: 0', 'Present: 0', 'Absent: 0', and 'Attendance grade: 0 / 0'. An 'Attendance percent: 0.00' is also shown. A table lists five sessions:

#	Type	Date	Time	Description	Status	Remarks
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session	?	
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session	?	
3	Common	29.05.15 (Fri)	20:55	Regular class session	?	
4	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session	Submit attendance	
5	Common	15.06.15 (Mon)	16:25 - 17:25	Regular class session	?	

The 'ADMINISTRATION' section at the bottom has a 'Course administration' link.

FIGURE 5.17: Confirm session attendance (student) (1 of 4)

5.5 Confirm session attendance (student)

1. Go to the PicAttendance page on Moodle. Select the *This course* tab if it is not already selected. (See Figure 5.17.)
2. Find the session at which you want to confirm your attendance. Click *Submit attendance*. (See Figure 5.18.)
3. If PicAttendance recognized a face as being yours, you should see something like Figure 5.19. If the photo really shows your face, click *Confirm*. (If the photo does not show your face, click *Change photo*. If the page you see is different from Figure 5.19, no face was recognized as being yours. Either way, see Section 5.6.)
4. If the face recognized by the system is correct and you click *Confirm*, your attendance is immediately recorded without the need for teacher approval. In the PicAttendance page, your *Present* counter increases by one and the photo of your face for this session is displayed among the other training and approved session photos. (See Figure 5.20.)

The screenshot shows a Moodle attendance page for a course named 'test_small'. On the left, a navigation sidebar lists 'My profile', 'test_small' (selected), 'Participants', 'Badges', 'General', 'Small files', 'Forum', and 'Attendance' (selected). Below that is a 'My courses' section. A 'ADMINISTRATION' section includes 'Course administration' and 'My profile settings'. The main content area displays a message: 'These are the photos we think it's you. Click on one to untag yourself.' followed by a row of eight small profile pictures. Below this, session statistics are shown: 'Sessions completed: 0', 'Present: 0', 'Absent: 0', 'Attendance grade: 0 / 0', and 'Attendance percent: 0.00'. A table lists session details:

#	Type	Date	Time	Description	Status	Remarks
1	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session	?	
2	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session	?	
3	Common	29.05.15 (Fri)	20:55	Regular class session	?	
4	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session	Submit attendance	
5	Common	15.06.15 (Mon)	16:25 - 17:25	Regular class session	?	
6	Common	17.06.15 (Wed)	16:25 - 17:25	Regular class session	?	
7	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session	Submit attendance	
8	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Submit attendance	
9	PicAttendance: Training					

You are logged in as Ruan Ruan (Log out) test_small

FIGURE 5.18: Confirm session attendance (student) (2 of 4)

The screenshot shows a Moodle attendance page for a course named 'test_small'. The navigation sidebar is identical to Figure 5.18. The main content area shows a large thumbnail of a student's face with the caption 'Regular class session'. Below the photo is a 'Change photo' link. A confirmation dialog box is overlaid on the page with two buttons: 'Confirm' (highlighted in blue) and 'Cancel'.

FIGURE 5.19: Confirm session attendance (student) (3 of 4)

#	Type	Date	Time	Description	Status	Remarks
1	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Present	Self-recorded
2	Common	27.05.15 (Wed)	16:25 - 17:25	Regular class session		?
3	Common	28.05.15 (Thu)	20:15 - 21:15	Regular class session		?
4	Common	29.05.15 (Fri)	20:55	Regular class session		?
5	Common	10.06.15 (Wed)	21:25 - 22:25	Regular class session		

FIGURE 5.20: Confirm session attendance (student) (4 of 4)

5.6 Tag session photo (student)

PicAttendance will do its best to recognize student faces. When recognition works, all they have to do is the use case described in Section 5.5, and the teacher has absolutely no additional work. When recognition fails, however, PicAttendance lets students tag themselves.

Tagging is accessible in three situations:

- When no face is recognized as being the student's
- When the student clicks *Change photo*
- When the student untags an already approved photo and then clicks *Submit attendance* on the corresponding session (see Section 5.8)

Regardless of the situation, the procedure for tagging is the same. Figure 5.21 shows the tagging page.

1. On the tagging page, scroll vertically and horizontally until you find your face in the photo. (See Figure 5.22.)
2. Click and drag to draw a rectangle around your face. Be as precise as possible. You can try as many times as you want. (See Figure 5.23.)

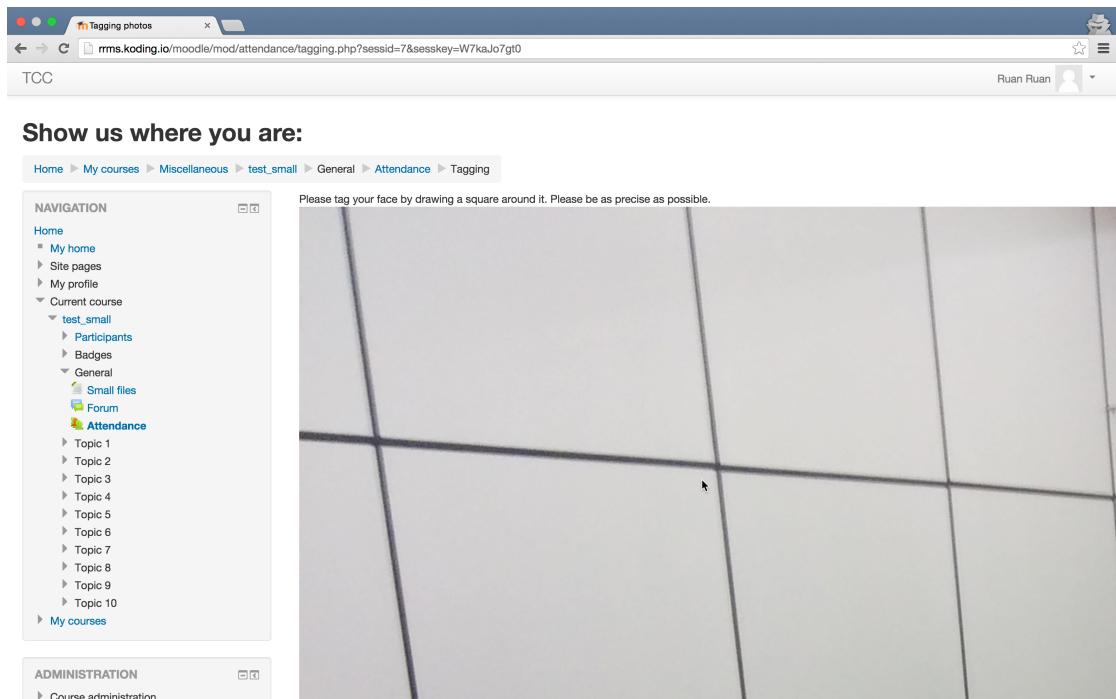


FIGURE 5.21: Tag session photo (student) (1 of 8)

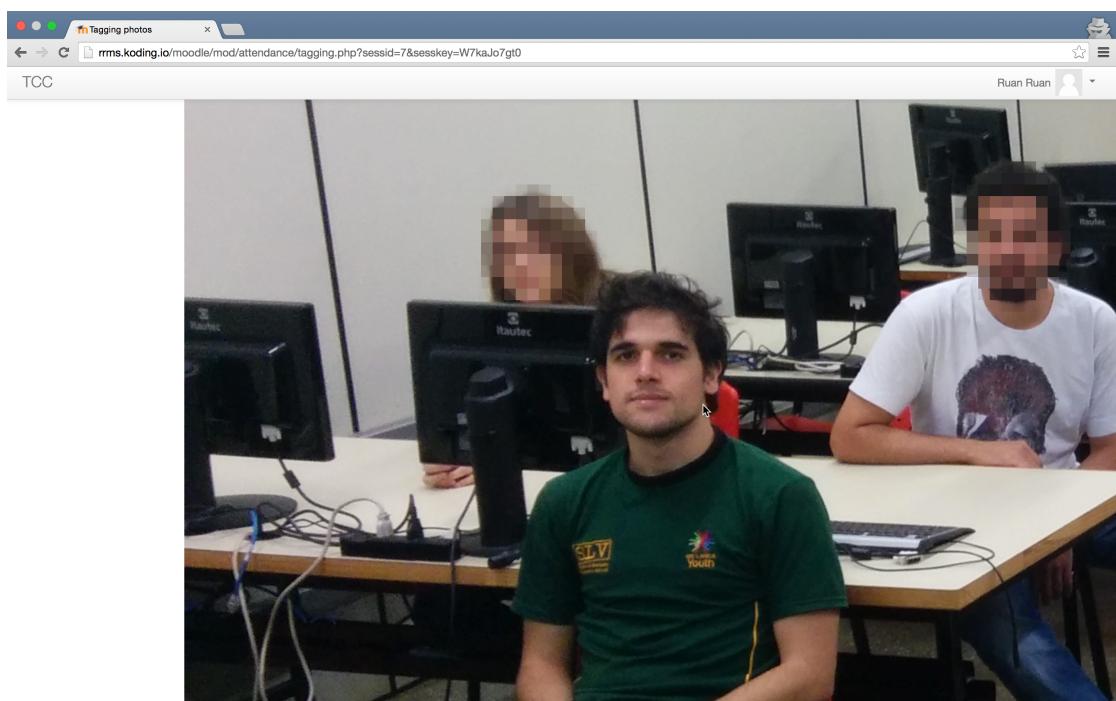


FIGURE 5.22: Tag session photo (student) (2 of 8)

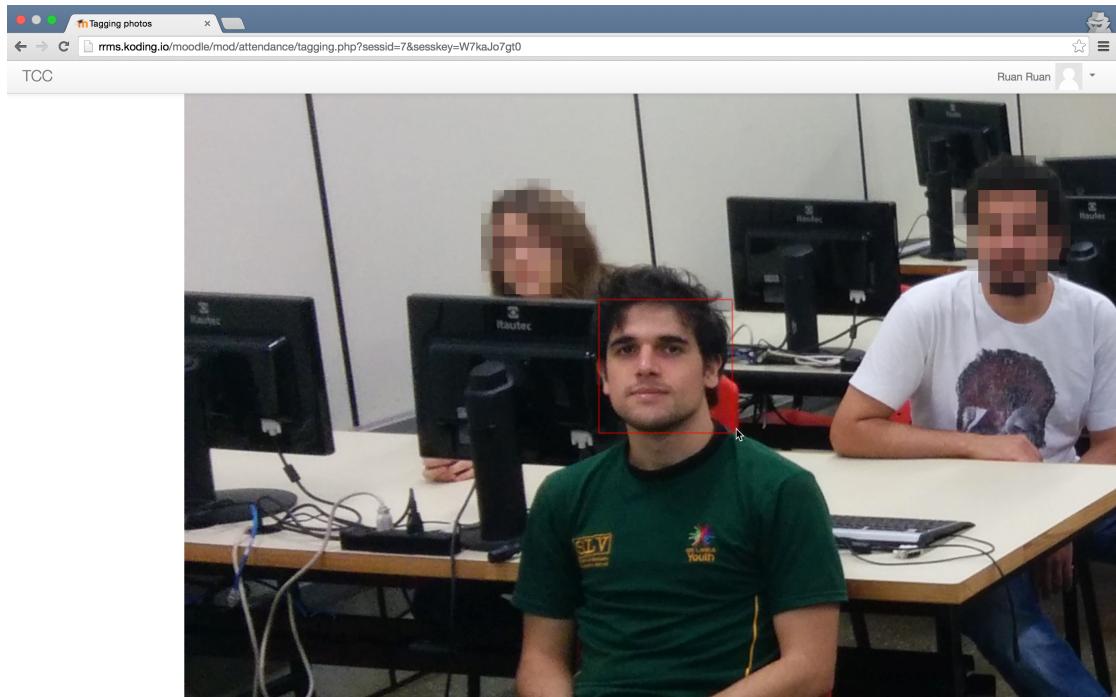


FIGURE 5.23: Tag session photo (student) (3 of 8)

3. Scroll to very bottom of the page and click *Submit Tagging*. (See Figure 5.24.)
4. If the system is able to detect a face that approximately matches the square you have drawn, you will see something like Figure 5.25. (If you do not, skip to next step.) If the face detected really is yours, click *Yes, this is me!*, otherwise click *No, use the square tag I've drawn!*. You may also ignore these two buttons and tag yourself again in the large photo below.
5. You will see something like Figure 5.26. As indicated, your attendance will only be confirmed after the teacher's approval.

The major difference between this use case and that of Section 5.5 is that tagging allows the student to tag herself anywhere in the photo, even if she did not attend. To avoid fraud, a tagging must be approved by the teacher. (See Section 5.7.) While the teacher's approval is pending, the *Present* counter in the student's PicAttendance page will not be incremented and the corresponding face photo will not be added to the other training and approved face photos. (Compare Figures 5.27 and 5.20.)

If the teacher uploads more than one photo for the session, tagging works much the same way, except for additional links for navigating among the session photos. (See Figure 5.28.)

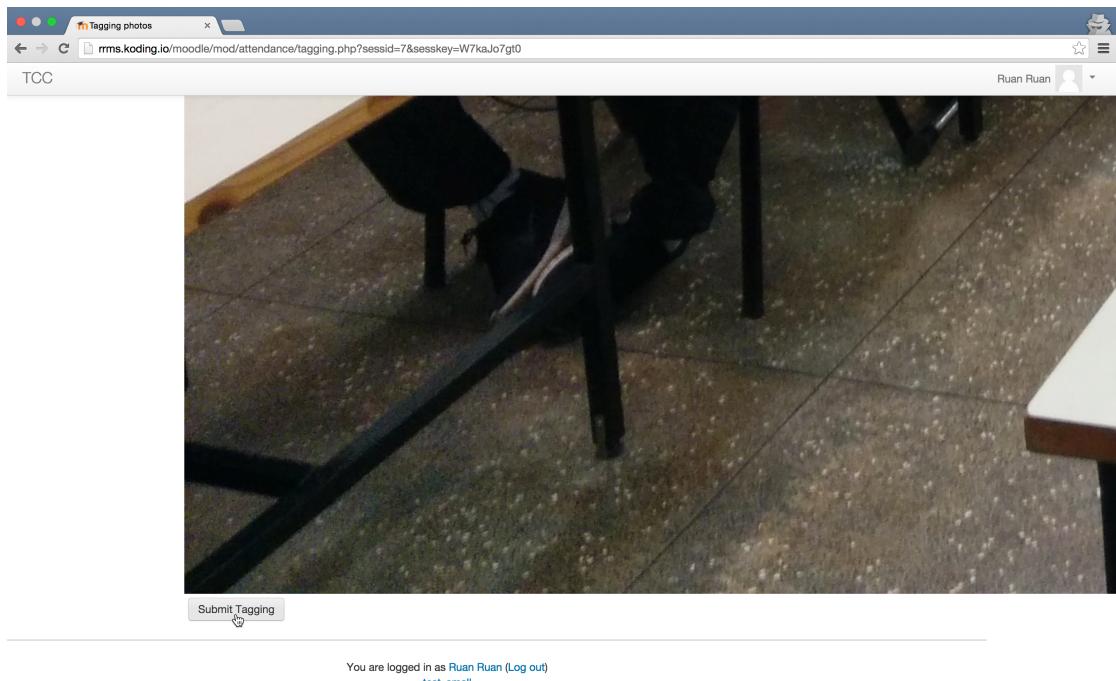


FIGURE 5.24: Tag session photo (student) (4 of 8)

A screenshot of a Moodle page titled "Tagging photos". The URL is "rrms.koding.io/moodle/mod/attendance/tagging.php?sessid=7&sesskey=W7kaJo7gt0&imghash=6151221be353d1f0973ad6d463b623d9bce08739". The page displays a detected face with the text "Show us where you are:". Below the face are two buttons: "Yes, this is me!" and "No, use the square tag I've drawn!". A message states: "Awesome! We have found a previously detected face that approximately matches your drawing. Is this your face? If it isn't, don't worry! We can also use the square you've drawn for your tagging instead." To the right of the face is a large, blank white area with a thin black border, likely a placeholder for a drawing.

FIGURE 5.25: Tag session photo (student) (5 of 8)

The screenshot shows a Moodle attendance page for a course named 'test_small'. The URL is rrms.koding.io/moodle/mod/attendance/attendance.php?sessid=7&sesskey=W7kaJ07gt0&id=65. The page title is 'Test course: S'.

NAVIGATION:

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- My courses

ADMINISTRATION:

- Course administration

Attendance:

6 July 2015 15:05

Regular class session

 Change photo

Teacher approval pending.

FIGURE 5.26: Tag session photo (student) (6 of 8)

The screenshot shows a Moodle attendance report page for a course named 'test_small'. The URL is rrms.koding.io/moodle/mod/attendance/view.php?id=65. The page title is 'Test course: S'.

NAVIGATION:

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- My courses

ADMINISTRATION:

- Course administration

Attendance report:

This course All courses

Ruan Ruan

These are the photos we think it's you. Click on one to untag yourself.



Sessions completed: 0
Present: 0
Absent: 0

Attendance grade: 0 / 0

Attendance percent: 0.00

July

#	Type	Date	Time	Description	Status	Remarks
1	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session	Submit attendance	
2	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Submit attendance	
3	PicAttendance: Training					

FIGURE 5.27: Tag session photo (student) (7 of 8)

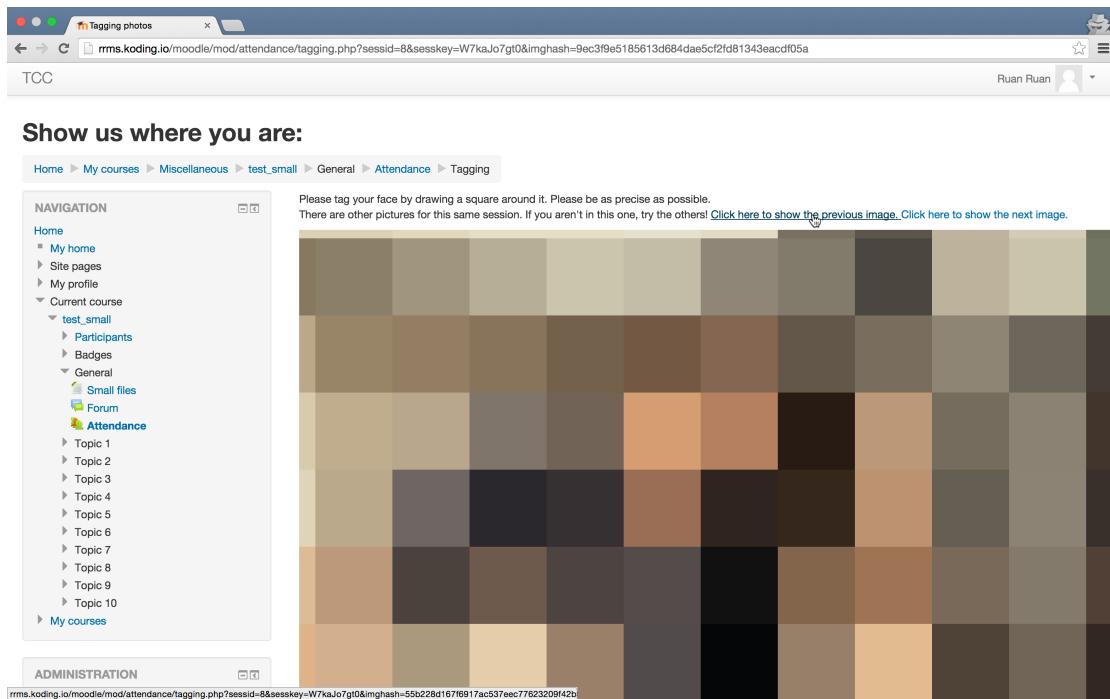


FIGURE 5.28: Tag session photo (student) (8 of 8)

5.7 Approve session tags (teacher)

1. Go to the PicAttendance page on Moodle. Select the *Approve* tab if it is not already selected. (See Figure 5.29.)
2. A list of taggings to be approved will be shown. Each entry in this list has the following structure: On the first line of the entry are all face photos already approved for that student; on the second line is the new face photo that was tagged and is waiting for approval. This structure helps you decide if the tagging is valid by comparing previous photos of a student with the new one. Mark *Yes* for taggings that you want to approve, and *No* for taggings you do not want to approve. You do not have to approve or disapprove all entries at once; you may leave some of them unmarked for now. (See Figure 5.30.)
3. Click *Save changes*. (See Figure 5.31.)
4. The taggings that you have approved or disapproved will not appear anymore. (See Figure 5.32.)

After the teacher approves the tagging of a photo, the corresponding student gets the attendance for the session and the photo is displayed among the approved session photos in the student's PicAttendance page.

Test course: S

Home > Courses > Miscellaneous > test_small > General > Attendance > Approve

NAVIGATION

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- Courses

ADMINISTRATION

- Attendance administration
 - Edit settings
 - Locally assigned roles
 - Permissions
 - Check permissions

Attendance for the course :: Test course: S

Sessions Add Report Export Settings Train Approve

Update this Attendance

Save changes

FIGURE 5.29: Approve session tags (teacher) (1 of 4)

Test course: S

Home > Courses > Miscellaneous > test_small > General > Attendance > Approve

NAVIGATION

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- Courses

ADMINISTRATION

- Attendance administration
 - Edit settings
 - Locally assigned roles
 - Permissions
 - Check permissions

Attendance for the course :: Test course: S

Sessions Add Report Export Settings Train Approve

Save changes

FIGURE 5.30: Approve session tags (teacher) (2 of 4)

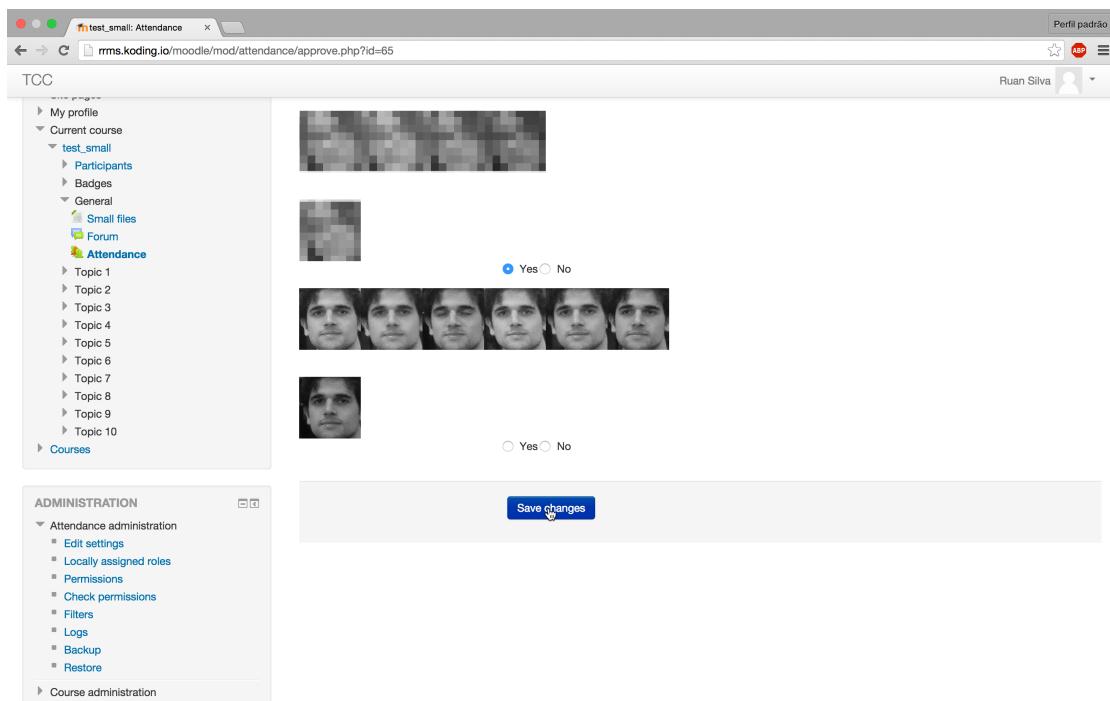


FIGURE 5.31: Approve session tags (teacher) (3 of 4)

This screenshot shows a Moodle attendance approval interface for a specific course named 'Test course: S'. The URL is rms.koding.io/moodle/mod/attendance/approve.php?id=65. The left sidebar shows a navigation tree with 'test_small' selected under 'Current course'. The main area displays a grid of student photos. Below each photo is a radio button labeled 'Yes' or 'No'. A 'Save changes' button is located at the bottom right of the grid. A 'Update this Attendance' button is visible in the top right corner.

FIGURE 5.32: Approve session tags (teacher) (4 of 4)

The screenshot shows a Moodle attendance report for a course named 'test_small'. The left sidebar lists course navigation items like 'Participants', 'Badges', 'General', 'Attendance', and 'PicAttendance: Training'. The main content area displays a student profile for 'Ruan Ruan' with a placeholder photo. Below it, a section titled 'These are the photos we think it's you. Click on one to untag yourself.' shows eight small portrait photos of a man. Underneath these are session statistics: 'Sessions completed: 1', 'Present: 1', 'Absent: 0', 'Attendance grade: 1 / 1', and 'Attendance percent: 100.00'. A calendar shows 'July'. Below the calendar is a table of attendance records:

#	Type	Date	Time	Description	Status	Remarks
1	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Present	Self-recorded
2	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session		Submit attendance
3	PicAttendance: Training					

The bottom navigation bar includes links for 'All', 'All past', 'Months', 'Weeks' (which is selected), and 'Day'.

FIGURE 5.33: Untag (student or professor) (1 of 4)

5.8 Untag (student or professor)

If a student confirms her attendance (Section 5.5) by mistake (i.e., when the student *Confirms* a recognized face that is actually someone else's), or when the teacher accidentally approves a tagging that is wrong, this mistake can be reverted by *untagging*. Training face photos tagged incorrectly may also be untagged.

1. Go to the *student's* PicAttendance page on Moodle. Teachers can go to this page by first going to a session page (see Figures 5.13 through 5.16) and then clicking the student's name. Select the *This course* tab if it is not already selected. (See Figure 5.33.)
2. Click the photo you would like to untag. (See Figure 5.34.)
3. In the dialog that appears, click *Untag*. (See Figure 5.35.)
4. The photo will be untagged and not displayed among the training and approved session photos anymore. If the photo is a training photo, it will be available for tagging again when students click on *PicAttendance: Training*. If the photo is a session photo, the student's attendance for that session will be revoked. Should the student tag herself again after that, the tagging must be approved by the teacher, even if the untagged photo was previously recognized by the system automatically.

Test course: S

Home > My courses > Miscellaneous > test_small > General > Attendance > Attendance report

NAVIGATION

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- My courses

ADMINISTRATION

- Course administration

Ruan Ruan

These are the photos we think it's you. Click on one to untag yourself.

Sessions completed: 1
Present: 1
Absent: 0
Attendance grade: 1 / 1
Attendance percent: 100.00

#	Type	Date	Time	Description	Status	Remarks
1	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Present	Self-recorded
2	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session		Submit attendance
3	PicAttendance: Training					

FIGURE 5.34: Untag (student or professor) (2 of 4)

Test course: S

Home > My courses > Miscellaneous > test_small > General > Attendance > Attendance report

NAVIGATION

- Home
- My home
- Site pages
- My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- My courses

ADMINISTRATION

- Course administration

Ruan Ruan

These are the photos we think it's you. Click on one to untag yourself.

Sessions completed: 1
Present: 1
Absent: 0
Attendance grade: 1 / 1
Attendance percent: 100.00

Are you sure you want to untag this photo? This will also revoke the attendance.

Untag

#	Type	Date	Time	Description	Status	Remarks
1	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Present	Self-recorded
2	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session		Submit attendance
3	PicAttendance: Training					

FIGURE 5.35: Untag (student or professor) (3 of 4)

The screenshot shows a Moodle attendance report for a course named "test_small". The URL is rrms.koding.io/moodle/mod/attendance/view.php?id=65. The page title is "Test course: S". The navigation bar includes links for Home, My courses, Miscellaneous, test_small, General, Attendance, and Attendance report.

NAVIGATION:

- Home
 - My home
 - Site pages
 - My profile
- Current course
 - test_small
 - Participants
 - Badges
 - General
 - Small files
 - Forum
 - Attendance
 - Topic 1
 - Topic 2
 - Topic 3
 - Topic 4
 - Topic 5
 - Topic 6
 - Topic 7
 - Topic 8
 - Topic 9
 - Topic 10
- My courses

ADMINISTRATION:

- Course administration

Attendance Report for Ruan Ruan:

Ruan Ruan

These are the photos we think it's you. Click on one to untag yourself.

Sessions completed: 0

Present: 0
Absent: 0

Attendance grade: 0 / 0

Attendance percent: 0.00

July

All All past Months Weeks Day

#	Type	Date	Time	Description	Status	Remarks
1	Common	4.07.15 (Sat)	17:00 - 19:00	Regular class session	Submit attendance	
2	Common	6.07.15 (Mon)	15:05 - 17:05	Regular class session	Submit attendance	
3	PicAttendance	Training				

FIGURE 5.36: Untag (student or professor) (4 of 4)

6 Conclusions and further work

This project has been, most of all, a learning process on how to put different components of software together to build an interesting application, which one could argue is the core of a software engineer’s work. We have spent many hours reading through documentation and testing in order to understand how to put all the concepts we had learned in practice and working together. Starting from understanding the Moodle architecture and the OpenCV modules separately; going through many design and implementation discussions on which programming language to use, which fields to put on a database table, how a particular page had to be shown to the user and so on; up to integrating the two major components together: the Moodle block and the computer vision block. It was a team effort, with each part having its own work and with our supervisor giving feature tips and guidance. We passed by many essential steps that any software engineering team in the industry has to go through in order to build real world web applications.

After presenting our results and having our initial goals in mind, we will now discuss what we have managed to achieve in the time we had and what we have not achieved, with some new ideas we had along the process that we will add as suggestions for further work.

6.1 Achievements

By the end of the semester we had a complete, functional system. Our latest PicAttendance version had the face detection and recognition algorithms successfully integrated with Moodle using the PHP-CPP library. We also created a fully operational tagging system with the corresponding approval mechanisms, which completes the vision we had for this application. In the Appendix we added a reference to our GitHub repository that contains a quick deploying guide.

6.2 Missing points

Here follows the points we did not manage to finish but that were part of our original proposal.

1. *Shared filesystem between courses*

Different courses do not have access to each others files in the current implementation. That is a desirable feature because a new course could use the face images that an old course already has on a student, which would imply less training for the new course.

2. *Real world testing*

Deploying our application in a real world setting would give us a better assessment of the plugin's overall performance and robustness, such as its behavior when used in small and large classes (which was one of our original goals). It would also allow us to establish a good threshold value for matching a square drawn on the tagging page to the squares corresponding to faces detected by the vision system.

3. *Make the plugin available to the Moodle community*

For that we need better packaging and following of Moodle guide styles and good practices. For example, instead of using hard-coded strings for the user interface, use Moodle's language packs to separate user interface string from code.

6.3 Ideas for further work

Besides implementing the points we mentioned in the last section, we had some ideas during the semester that did not fit in the scope of our final project.

1. *Cache for the face recognition training*

As it is now the plugin is executing the training process for the face recognition method every time a new class photo is uploaded. Caching some of the training information could generate great performance improvements.

2. *Multiple face recognition implementations*

There is a large variety of methods in computer vision for face recognition, and we only used one in our plugin. If the plugin had more implementations of algorithms available, the instructor could choose which implementation he wanted to use or

even a combination of methods. It would allow the instructor to customize the application as he wished in order to try to generate the best results for his class.

3. Integration with mobile devices

A valuable feature for teachers would be photo upload via mobile devices. They would be able to take and upload photos on the fly during class.

A Source code

PicAttendance's source code is available at

<https://github.com/ruanslv/PicAttendance> and

<https://github.com/thiagocaetano/PicAttendance-PHP-Extension>.

The repositories contain deploying instructions. This software may be distributed under the terms of the GNU General Public License. [18]

Bibliography

- [1] Moodle Pty Ltd. Moodle stats. URL <https://moodle.net/stats/>.
- [2] Dmitry Pupinin, Artem Andreev, Dan Marsden, et al. Attendance Moodle module. URL https://moodle.org/plugins/view/mod_attendance.
- [3] Paul Viola and Michael Jones. Rapid object detection using a boosted cascade of simple features. In *Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on*, volume 1, pages I–511. IEEE, 2001.
- [4] Rainer Lienhart and Jochen Maydt. An extended set of haar-like features for rapid object detection. In *Image Processing. 2002. Proceedings. 2002 International Conference on*, volume 1, pages I–900. IEEE, 2002.
- [5] Itseez. OpenCV. URL <http://opencv.org/>.
- [6] Timo Ahonen, Abdenour Hadid, and Matti Pietikäinen. Face recognition with local binary patterns. In *Computer vision-eccv 2004*, pages 469–481. Springer, 2004.
- [7] David Romer. Do students go to class? should they? *The Journal of Economic Perspectives*, pages 167–174, 1993.
- [8] Daniel R Marburger. Does mandatory attendance improve student performance? *The Journal of Economic Education*, 37(2):148–155, 2006.
- [9] Marcus Credé, Sylvia G Roch, and Urszula M Kiesczynka. Class attendance in college a meta-analytic review of the relationship of class attendance with grades and student characteristics. *Review of Educational Research*, 80(2):272–295, 2010.
- [10] Brazilian law. Lei de diretrizes e bases da educação nacional, . URL <http://portal.mec.gov.br/arquivos/pdf/lbd.pdf>. In Portuguese.
- [11] Brazilian law. Parecer homologado, . URL http://portal.mec.gov.br/cne/arquivos/pdf/2002/pces282_02.pdf. In Portuguese.

- [12] University of Campinas. Regimento geral de graduação. URL <http://www.dac.unicamp.br/portal/grad/regimento/>. In Portuguese.
- [13] Koding, Inc. Koding. URL <https://koding.com/>.
- [14] Software Freedom Conservancy. Git. URL <https://git-scm.com/>.
- [15] GitHub, Inc. GitHub. URL <https://github.com/>.
- [16] Oracle Corporation. MySQL. URL <https://www.mysql.com/>.
- [17] Copernica. PHP-CPP. URL <http://www.php-cpp.com/>.
- [18] Gnu general public license, version 3. <http://www.gnu.org/licenses/gpl.html>, June 2007. Last retrieved 2015-07-18.