FINAL TECHNICAL REPORT

eTendance

Abstract

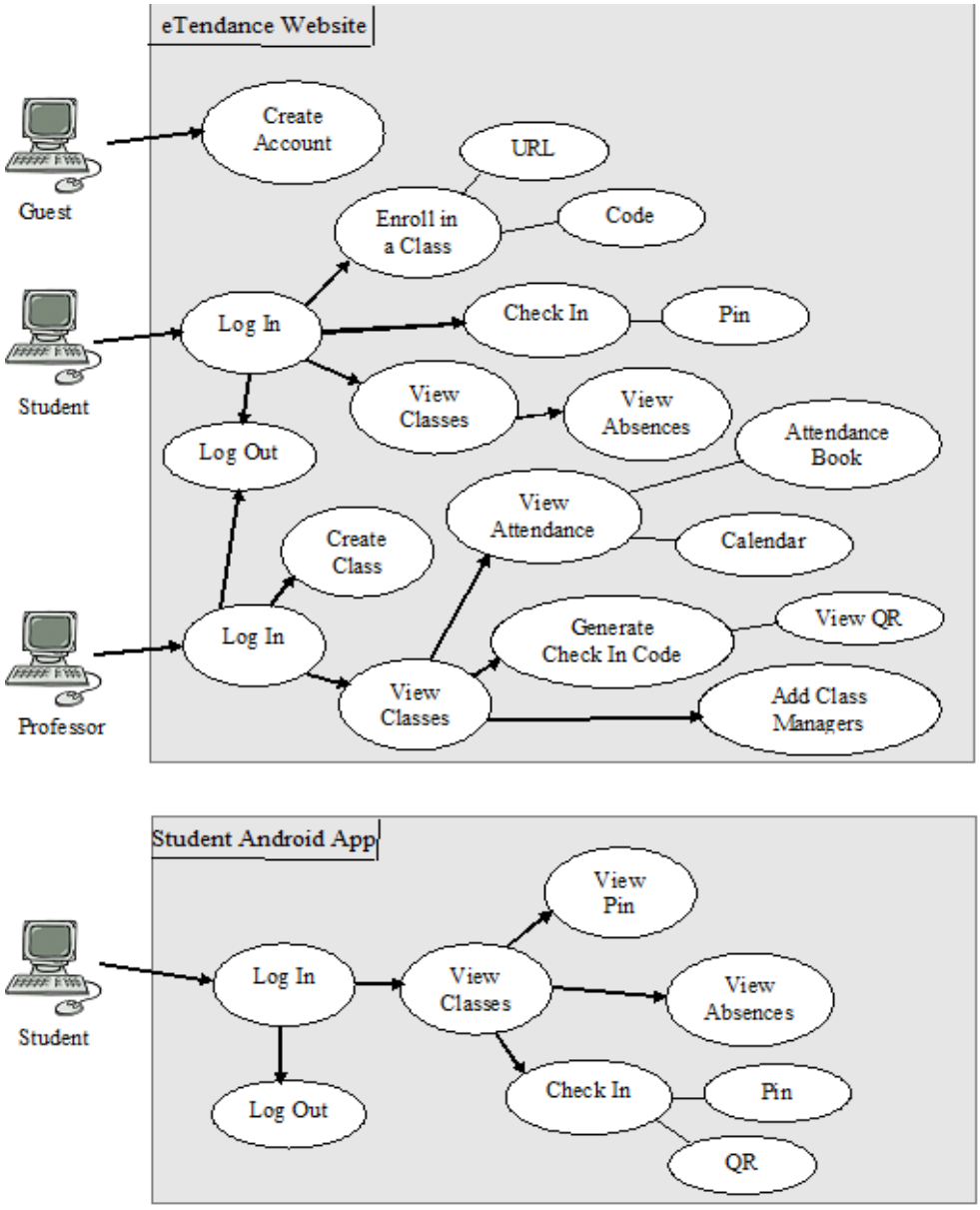
Final technical document for Fall 2013 DB/GUI collaborative project.

Table of Contents

Introduction

Our team calls itself Student Software Solutions and consists of Alex Baird, Bre'Shard Busby, Taylor Ellington, Jiaxin Li, Christopher Pellizzi, and Courtney Redlinger. Our product is eTendance, an online attendance management tool. This product is for professors to easily take, view, and edit attendance. Students can also use eTendance to track their absences for classes.

Use Case Diagram



Software Features

eTendance is an online attendance manager designed for professors and students. Professors can create classes and generate unique check in codes for each day. Students can quickly check into a class using their laptop or smartphone. Attendance is immediately available, so professors and students can easily keep track of absences

- Registration-
    - Users can create accounts as either a student or a professor.
        - Accounts track:
            - First name
            - Last name
            - Email
            - Username
            - Password
            - Status(Professor or Student)

- Login -
    - Users can use their username and password to log in to the product and access other features based on their status as either a professor or student.

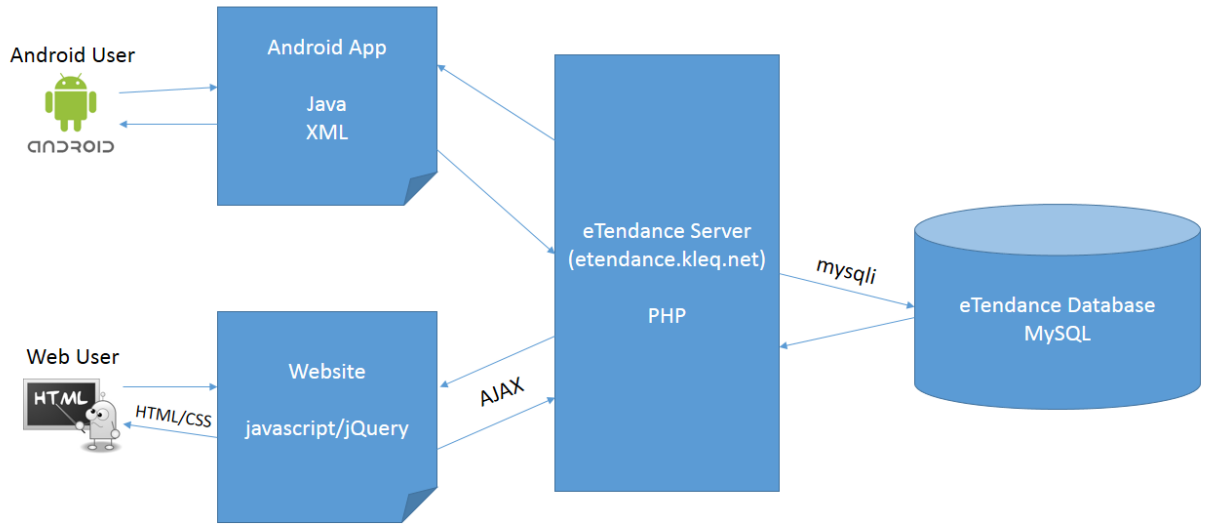
- Professor-
    - Professors can create classes.
    - Professors can generate enrollment codes for their classes.
    - Professors can view a list of the classes they own.
    - Professors can view a roster of students enrolled in their class.
    - Professors can remove students from their class.
    - Professors can take attendance.
        - Professor indicates day on which class takes place.
        - Professor is provided with a PIN or QR code to be given to students.
        - Professor can open attendance for that day.
        - Students can use their functionality to check into an open class.
        - Professor can close attendance for that day.
    - View attendance in classes the professor owns.
        - Calendar view shows a month by month calendar that highlights class days and lists the number of absences for that day.
        - Attendance book view shows a list of all students in the class with each class day and indicates if the student was present or absent.
    - Professor can manually edit attendance.

- Professors can add and remove class managers to and from their class.
    - Class managers have access to all professor functionality for the class they were added to, except the ability to add or remove class managers.

- Student-
    - Students can, by using enrollment codes provided by a professor, enroll in classes.
    - Students can view a list of the classes they are enrolled in.
    - Students can view the dates they were absent in each class.
    - Students can, by using the a PIN number or QR code (Android App only) check in to class, and be recorded as present.
    - Students can, by using the Android App, see if attendance is open or closed for that class.

Software Architecture

Diagram



Technologies/Libraries

Database

- MySQL
- mysqli
- phpmyadmin

Server

- PHP
- phpqrcode – QR generating library in PHP
- Calendar PHP library

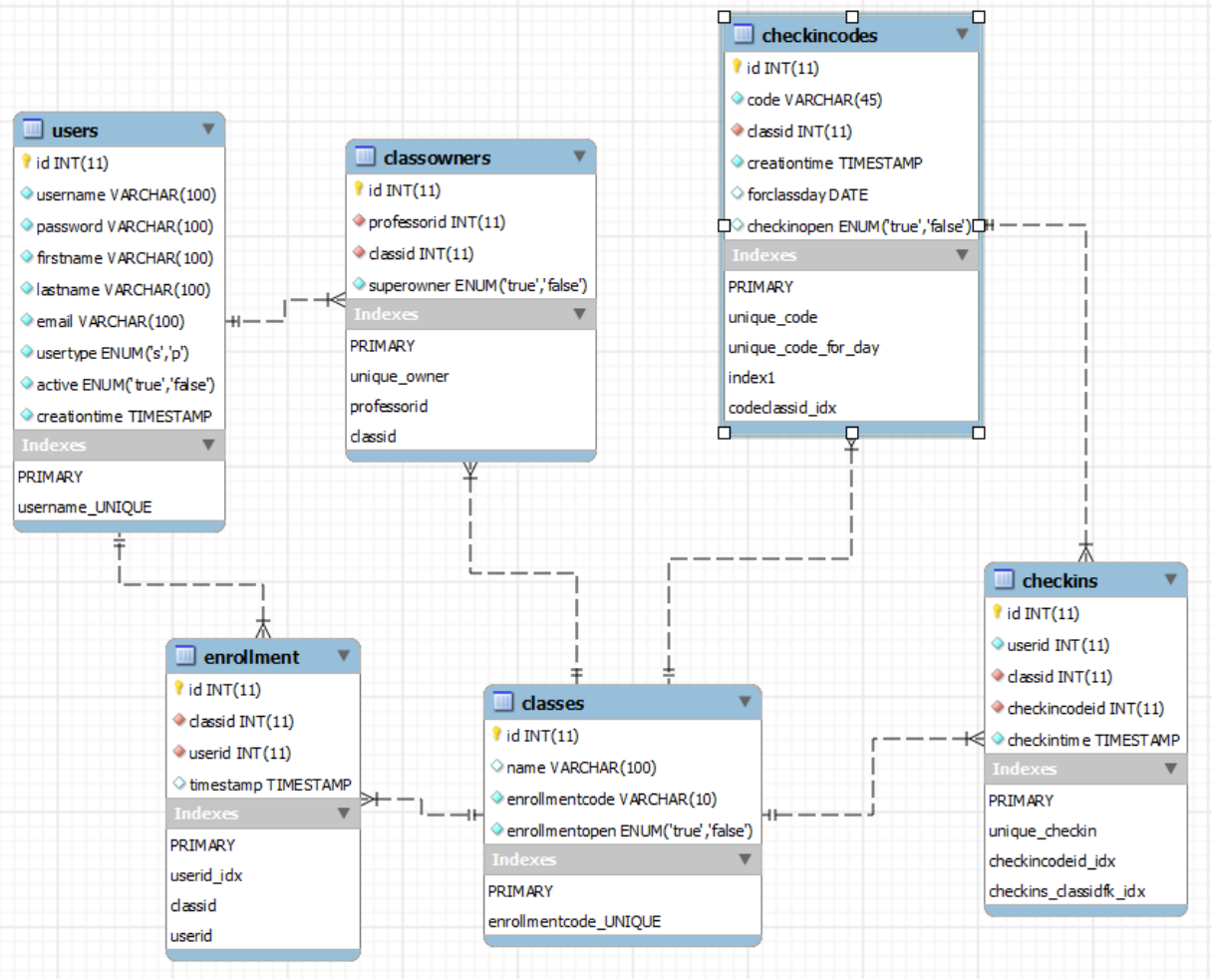
Website

- javascript/JQuery
- AJAX
- HTML/CSS

Android App

- Java/XML
- ZXing – QR/Barcode scanner library

Database Model



For additional information, please see Appendix A: Data Dictionary.

User Interface

Web

By going to etendance.kleq.net, users can access our website (Figure 1).

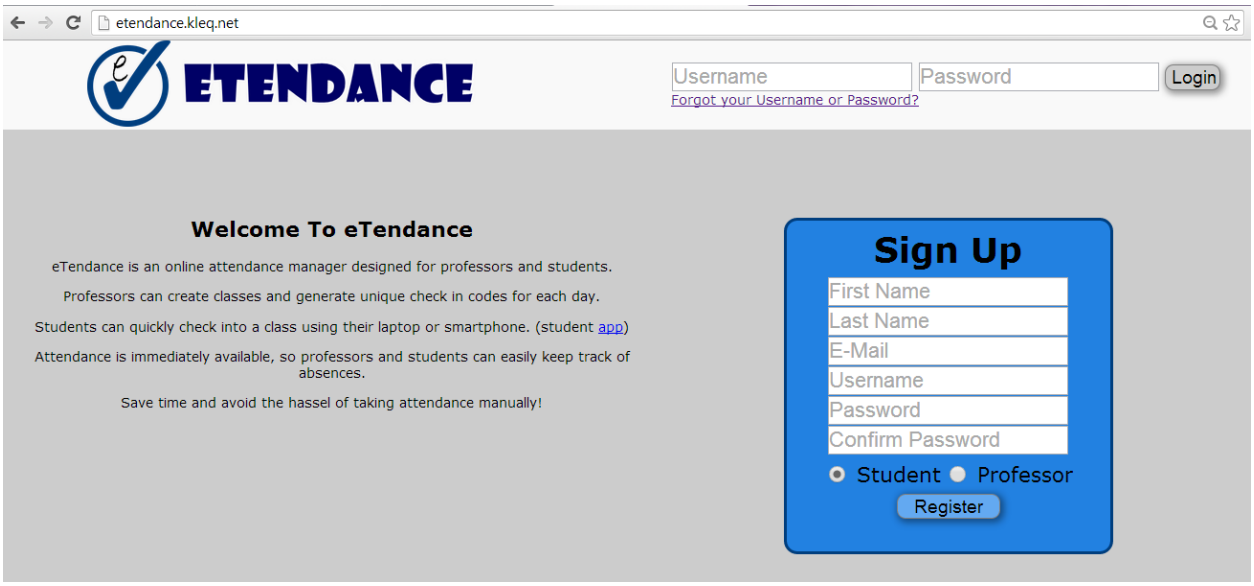


Figure 1: eTendance Home Page

Logging into a professor account takes you to the Professor Dashboard (Figure 2).

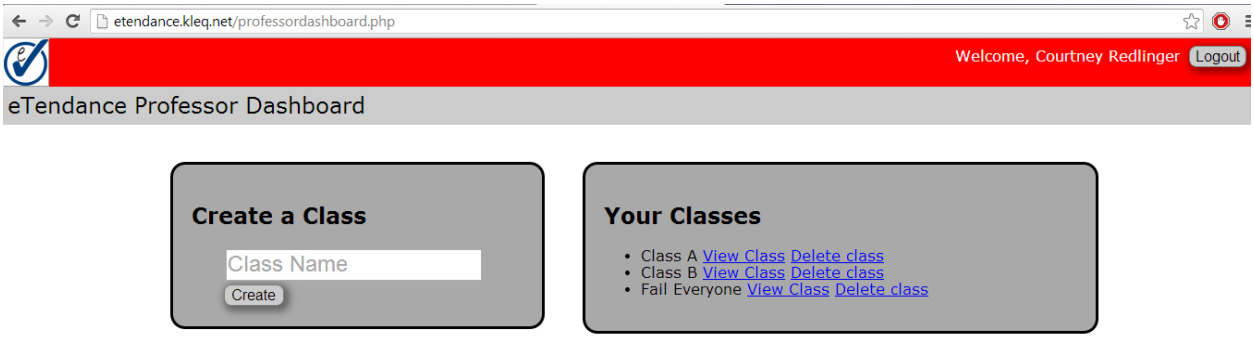


Figure 2: Professor Dashboard

By clicking view class, professors have several options. The first is to generate a Check In code for that day or any in the future (Figure 3).

Figure 3: Generated Check In Code/QR

Professors can also view attendance for their classes. For this, they have two options: Attendance Book View (Figure 4) and Calendar View (Figure 5). Attendance Book View shows all students and all sessions of the class and allows professors to edit attendance manually. Calendar View allows professors to select a day (days a code was generated are highlighted) and view the absences for that day.

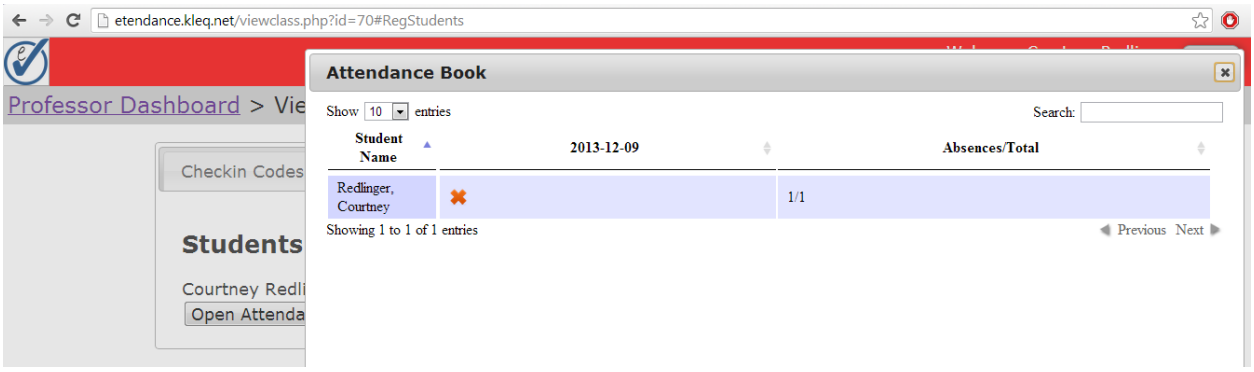


Figure 4: Attendance Book View

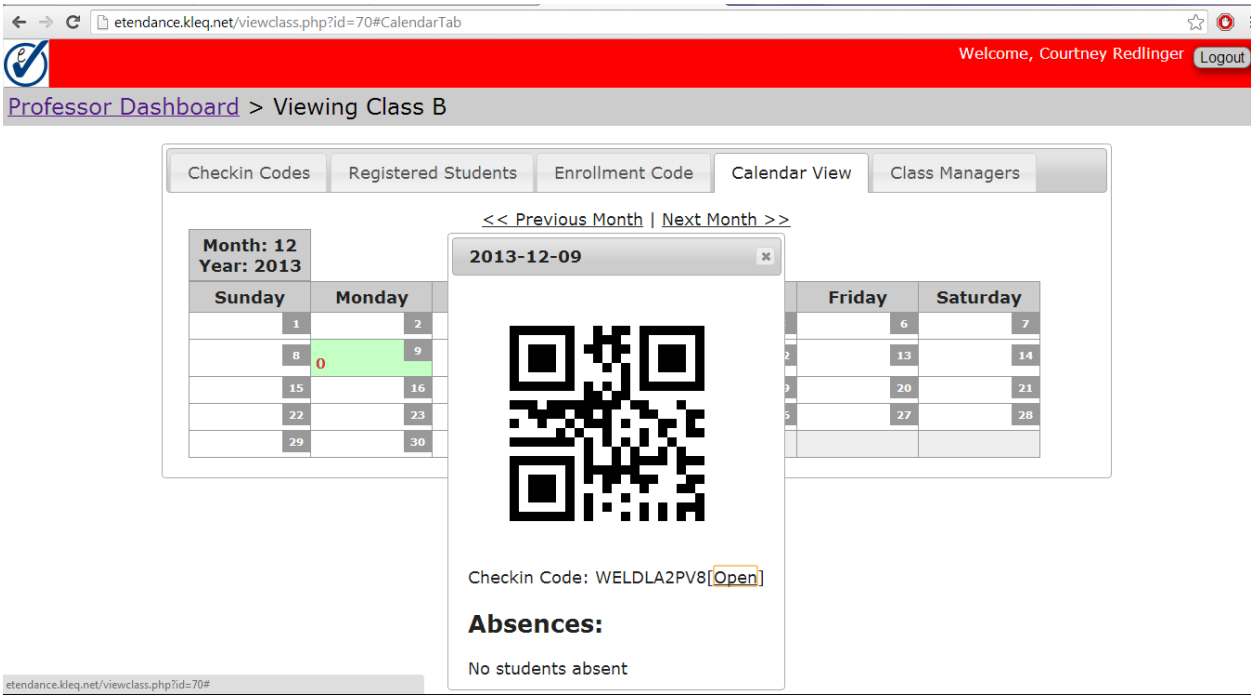


Figure 5: Calendar View

Professors also have the option to add class managers to each course to accommodate teaching assistants or other professors (Figure 6). A class manager must be someone with a professor account on eTendance, as they are then granted all professor options for that class (with the exception of the Class Managers tab).

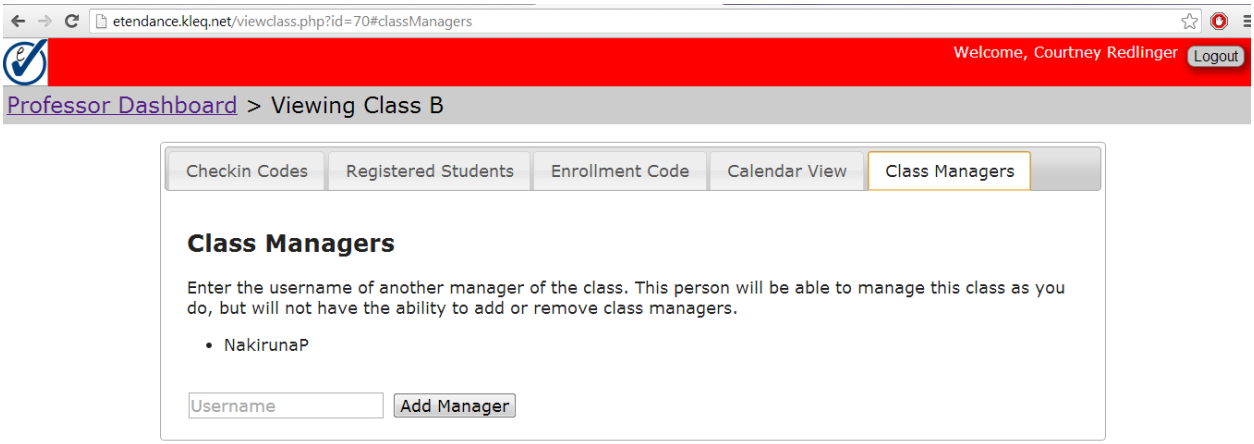


Figure 6: Class Managers

Finally, professors can view the enrollment code (Figure 7), which they distribute to their students. The enrollment code comes in two forms: a simple code and a link with the code embedded. The code and link can be regenerated if needed.

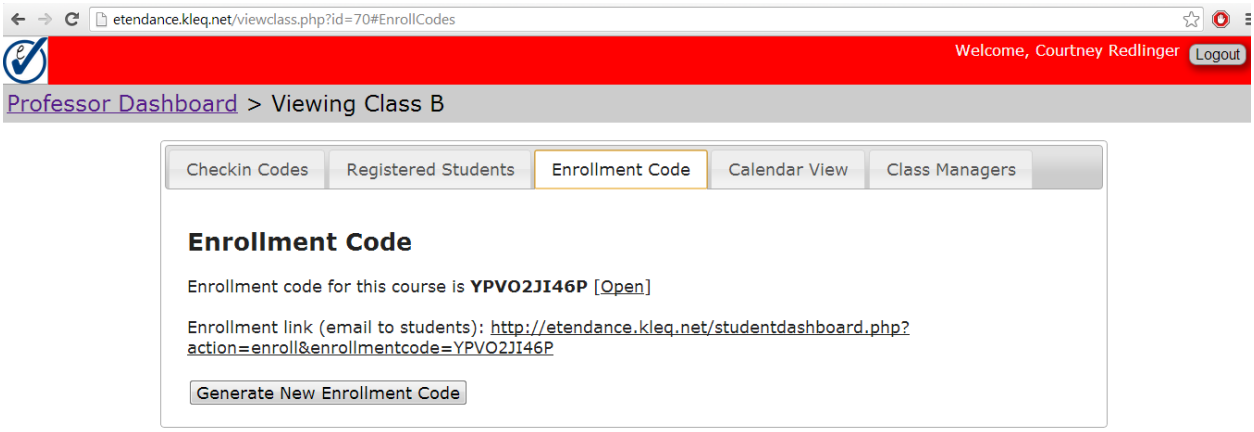


Figure 7: Enrollment Code and Link

Once a student receives an enrollment link, clicking it will take them to the eTendance home page, where they will be prompted to either log in or create an account (Figure 8 p1). Once they do so, they will be enrolled in the course (Figure 8 p 2).

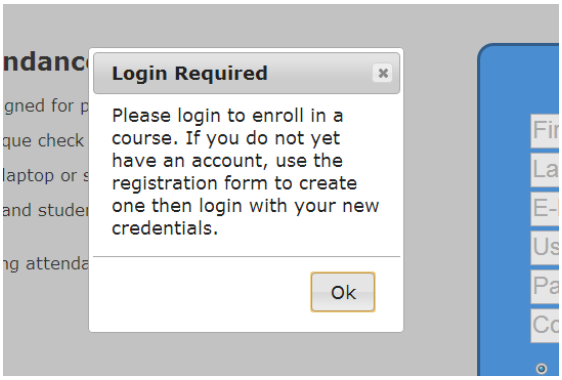


Figure 9: p1 Prompt at Home Page

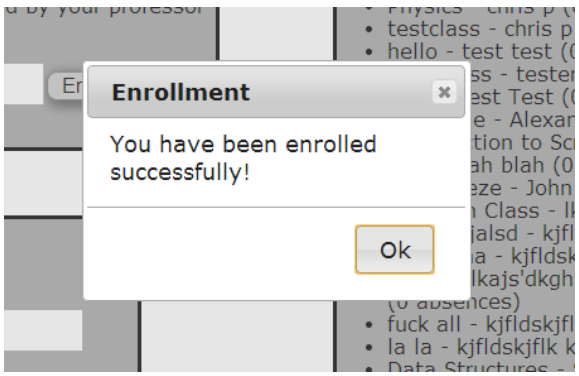


Figure 8: p2 Successfully Enrolled

Alternatively, students can also manually enter the code into the corresponding form on the Student Dashboard (Figure 10), which is reached when a user logs into a student account. This page is also where students can enter a Check In code given by a professor.

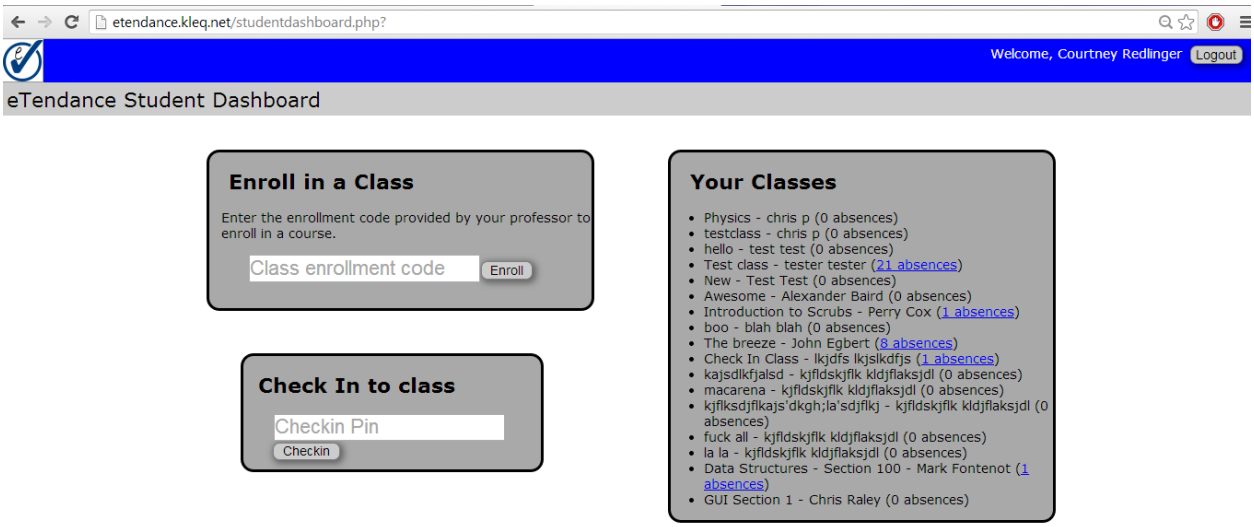


Figure 10: Student Dashboard

Students also have the ability to view their absences in any class (Figure 11) by clicking the absences link next to the desired class. If students have no absences, the link will not be active.

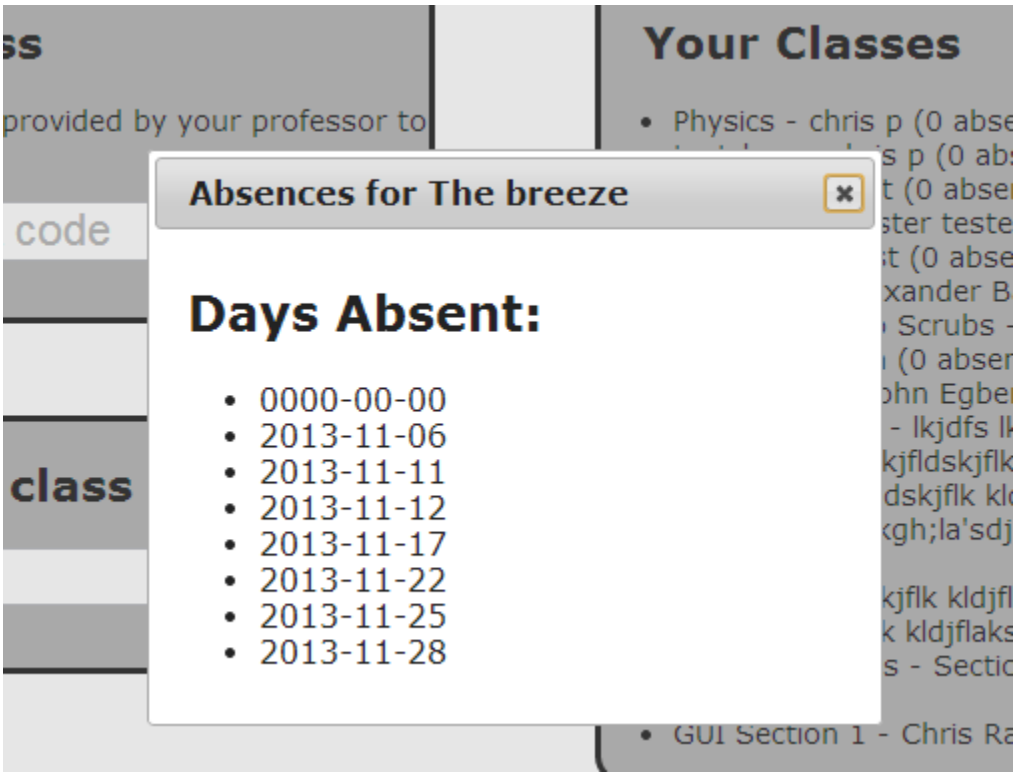


Figure 11: Student View of Absences

Android App

The eTendance android application is available for student use only. Students must already have an account created to use the app, as they will be prompted to log in first (Figure 12).

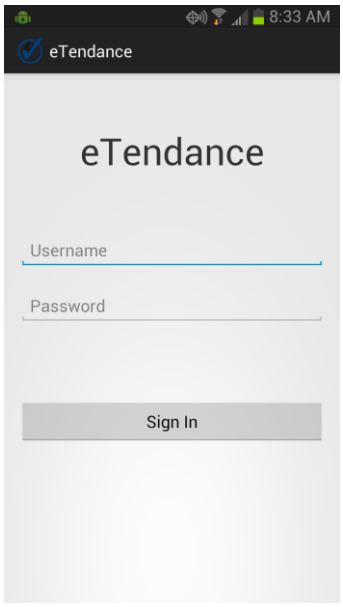


Figure 12: Android App Sign In Screen

Once a student signs into their account, they will be brought to the Class View Screen (Figure 13).

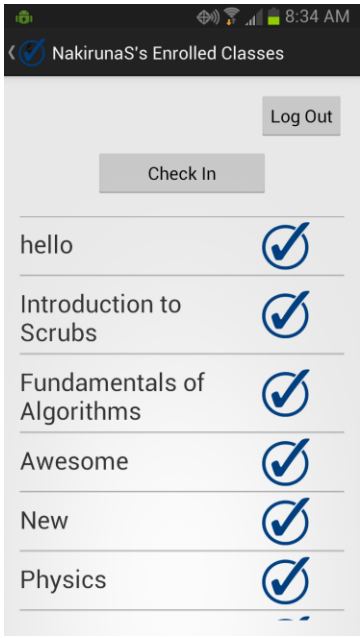


Figure 13: Class View Screen

From here, students can either choose a class to view their absences (Figure 14), or they can move to the Check In Screen (Figure 15). On the Check In Screen, students can either enter the attendance pin or scan the QR code provided by the professor. The QR option uses the default scanner on the student's phone. If the student does not have one, they are directed to the ZXing scanner (free) in the Play Store.

Figure 14: View Absences for a Class

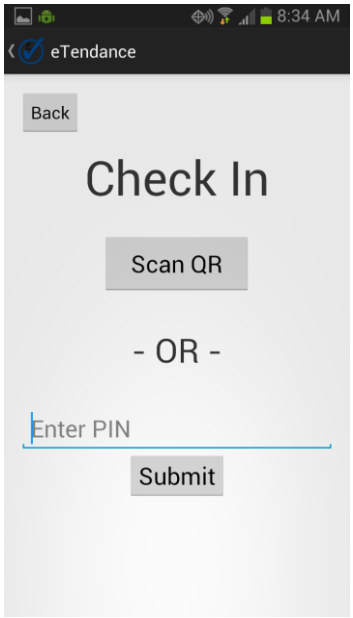


Figure 15: Check In Screen

Testing

During the development process, there were a few major technical issues. First, our original database set up had several foreign key issues and had to be redone. Later, the javascript calendar library we originally chose did not integrate well into our website, so it was swapped out with the PHP calendar library. Finally, the ZXing QR/barcode scanner library was difficult to work with, requiring a long build process and having integration issues.

Our test team, the creators of KeyChat, were helpful in finding the minor bugs and inconveniences in our software. They did not report any major bugs, but we didn't find any either, apart from the difficulties mentioned above.

We, in turn tested KeyChat, an online instant messenger. Their team was very good about quickly fixing any issues that were reported and improving their user experience. Unfortunately, we had issues testing their application, as it was developed for an iPhone and no one in the eTendance group had one.

Team Reflection

Initially Student Software Solutions had lofty goals of implementing a replacement for access.smu.edu in a semester, however, the team rapidly realized that was not an attainable goal. Instead, the team decided, they would create an attendance tracking tool for professors and students to use in the classroom. As time would show, this goal was not only attainable, but would prove to be a fantastic learning experience for each member of the team.

Student Software Solutions was made up of students with differing levels of experience, some were new to the technologies used in this project, while others had used them outside of academia. Every member of the team met with challenges, regardless of skill level and experience. For the team members new to these technologies and tools, other members of the team proved to be an invaluable resource. Many issues were overcome during team meet ups. Team members learned the finer points of tools such as PHP, CSS, SQL, and HTML. More experienced members of the team helped others build new skillsets by teaching debugging, syntactical quirks, and idiosyncrasies of each language. For members of the team with more experience, there were some difficulties with learning new tools and incorporating them into the product.

For example, one of the most important features of the android app was the ability to read a QR code. To accomplish this the team turned to a third party library called ZXing, which appeared to be the friendliest of the QR libraries to incorporate. However ZXing was not a drop in

solution, instead it had to be built using a series of other tools. However this set of other tools intrinsically had a steep learning curve that the time frame of the iteration could not accommodate. A solution was eventually found, in the form of locating a pre-built version of the library and adding it to the app.

If Student Software Solutions were to pass along any advice, it would be this: err on the side of over-communicating. More discussion will lead to fewer misunderstandings, better group direction, and a more open group environment that promotes collaboration.

Future Features

In coming versions of eTendance, there are several features we would like to implement. First, we would like to integrate a timer into the system that professors can set to automatically close check in at a certain time. This could also be used to add tardies into our system as well. We also hope to allow for excused absences with the ability make notes with the reasons. Another feature we would like to implement is suggested grade calculation based on the ratio of unexcused absences to class sessions. Finally, we would like to give professors the abilities to select regular class day(s) for automatic check in generation and to upload a list of email addresses to automatically send out the enrollment code. Overall, we would like to continue improving this product by giving it more features that make it more convenient for professors to use.

Appendix A: Data Dictionary

For use with the Database Model.

users – Records of every educator and student in the system

- id – INT(11): A unique number to identify each user, this table's primary key
- username – VARCHAR(100): A user created, unique identifier used for authentication purposes.
- password – VARCHAR(100): A user created pass-phrase used in conjunction with username for login to the service.
- firstname – VARCHAR(100): The user's first name, entered by the user at sign up.
- lastname – VARCHAR(100): The user's last name, entered by the user at sign up.
- email – VARCHAR(100): The user's email address, entered by the user at sign up.
- usertype – ENUM('s', 'p'): The users designation, either student (s) or professor (p), controls what features the user has access to.
- active – ENUM ('true','false'): the user's status, for an unimplemented feature.
- creationtime – TIMESTAMP: the date and time the user joined the service.

classowners – Mapping of professors to classes

- id – INT(11): Unique number identifying relationship, this table's primary key.
- professorid – INT(11): a foreign key, referencing the users.id primary key.
- classid – INT(11): a foreign key, referencing the classes.id primary key.
- superowner – ENUM('true','false'): Designates the if the professor in relation is the true owner of the class versus a class manager.

checkincodes – The list of checkin PINs or codes, what class it is for, and other relevant data.

- id – INT(11): Unique number identifying each checkincode relation, this table's primary key.

code – VARCHAR(45): A randomly generated series of characters used by students to check in to classes when attendance is being taken, can be displayed raw or encoded in a QR code. Each code is unique and only one can be created for each day.

classid – INT(11): A foreign key, referencing the classes.id primary key.

creationtime – TIMESTAMP: the date and time the code was generated by a professor.

forclassday – DATE: The class day the code is for.

checkinopen – ENUM('true','false'): Denotes if the attendance for that day is open, or if the professor has closed it.

enrollment – Mapping of classes to the students enrolled in them.

id – INT(11): A unique number identifying each enrollment relation, this table's primary key/

classid – INT(11): A foreign key, references class.id.

userid – INT(11): A foreign key, references user.id.

timestamp – TIMESTAMP: the date and time the student enrolled in the class.

classes – The listing of all classes being offered by all professors. Ownership of classes is established in classowners, and students enrollment is tracked in enrollment.

id – INT(11): This table's primary key, uniquely identifies each class.

name – VARCHAR(100): The class's name, provided by the professor.

enrollmentcode – VARCHAR(10): The code used by students to enroll in this class.

enrollmentopen – ENUM('true','false'): Status flag to determine if students can enroll in the class or if enrollment has been closed by the professor.

checkins – Mapping of class, user, and check in code. This table represents attendance by students to the classes they are enrolled in.

id – INT(11): This table's primary key, uniquely identifies the relation.

userid – INT(11): A foreign key, relates to users.id.

classid – INT(11): A foreign key, relates to classes.id.

checkincodeid – INT(11): a foreign key, relates to checkincodes.id.

checkintime – TIMESTAMP: the time and date the student checked in to class using the checkincode.