Attendance App Analysis Document

Mutually Linked (Ian Koeppe & Larry Singleton)

|  |  |
| --- | --- |
| [1 PURPOSE](#_v8oss2ko63y0) | 2 |
| [2 AUDIENCE](#_e8uzvpw3yuvv) | 2 |
| [3 INTRODUCTION](#_wgtboj3evag6) | 2 |
| [3.1 Purpose of the System](#_epowca4ifdpj) | 2 |
| [3.2 Scope of the System](#_r7yg55vaxu25) | 3 |
| [3.3 Objectives and Success Criteria of the Project](#_qba9u2m5qbtn) | 4 |
| [3.4 Definitions, Acronyms, and Abbreviations](#_vvpdv9lmw5kv) | 4 |
| [3.5 References](#_spoap3tcp156) | 4 |
| [3.6 Overview](#_5n1pyoxzhv9) | 5 |
| [4 CURRENT SYSTEM](#_qh6q3xda94e) | 5 |
| [5 PROPOSED SYSTEM](#_vc8hts4e2dji) | 6 |
| [5.1 Overview](#_7teqkz4cbi9s) | 6 |
| [5.2 Functional Requirements](#_gahdhwu3pxk) | 6 |
| [5.2.1 Functional Requirements of the Web Portal](#_vx8gqwxuuhpn) | 6 |
| [5.2.2 Functional Requirements of the Mobile Application](#_tvsipmi5tj1n) | 6 |
| [5.3 Non-Functional Requirements](#_8piw718chrg7) | 7 |
| [5.3.1 Usability](#_mbqoqrh0tuvv) | 7 |
| [5.3.2 Reliability](#_bgg8rvpv31hb) | 7 |
| [5.3.3 Performance](#_g81zmbcz81ow) | 8 |
| [5.3.4 Supportability](#_qp2nokxwhj4t) | 8 |
| [5.3.5 Implementation](#_2h2u8fu74sgd) | 8 |
| [5.3.6 Interface](#_s7aw5qqq4poe) | 8 |
| [5.3.7 Packaging](#_kklq6zqtknu9) | 8 |
| [5.3.8 Legal](#_s97hhp1nak2u) | 8 |
| [5.4 System Models](#_ozajp8l0sj19) | 9 |
| [5.4.1 User Stories](#_gh1m1941d5lu) | 9 |
| [5.4.1.1 Instructor Stories](#_hvafriykrob9) | 9 |
| [5.4.1.2 Student Stories](#_p59fd8z0oqbu) | 9 |
| [5.4.2 Scenarios](#_plrtt9esziea) | 10 |
| [5.4.2.1 Instructor Scenarios](#_xz8jyiyzepus) | 10 |
| [5.4.2.2 Student Scenarios](#_dx4x4730m5jb) | 11 |
| [5.4.3 Use Case Model](#_k4m1b9472o5v) | 13 |
| [5.4.4 Object Model](#_twr0ctpdg4wj) | 14 |
| [5.4.5 Dynamic Model](#_cwsyvptwt3y) | 14 |
| [5.4.6 User Interface/Navigational Paths & Screen Mockups](#_olxqwfey64qh) | 15 |
| [5.4.6.1 Web Admin Portal](#_imzwerp5z3t4) | 15 |
| [5.4.6.1.1 Login Screen](#_nsi62nzevjvt) | 15 |
| [5.4.6.1.2 Welcome Page](#_ykff8eg1q639) | 16 |
| [5.4.6.1.3 Classes Page](#_jd3nc5yysxn4) | 17 |
| [5.4.6.1.4 Class Page](#_fcjj7j3f60pg) | 18 |
| [5.4.6.1.5 Disputes](#_69gh398kw5ha) | 19 |
| [5.4.6.1.6 Reports](#_tl6mwukov4dk) | 20 |
| [5.4.6.2 Mobile Application](#_pzoq8rmyszw1) | 21 |
| [5.4.6.2.1 Welcome Page](#_9blvy5upx0vq) | 21 |
| [5.4.6.2.2 Class Page](#_9vdwteh7xio3) | 22 |
| [5.4.6.2.3 Dispute Page](#_c55qkxl0ihn1) | 23 |
| [5.4.6.2.4 Attendance Page](#_nvvolzu3blq5) | 24 |
| [6 GLOSSARY](#_cdec9iehuzvr) | 24 |

# 1 PURPOSE

Educators in a classroom setting who are tasked with collecting attendance records are in need of a streamlined process to decrease the amount time expended on the task, and therefore increase instruction time. A classroom attendance application is proposed to meet the requirements of both the educator and the student. This application is intended to automate the manual entry of recording attendance in a number of ways, including but not limited to, an online authenticated web portal and a mobile device application (android/ios). As mobile devices already possess location information, it is ideal for purposes of ensuring a student is in a given location at the beginning of a class period and we propose using this information to perform attendance in the background with minimal effort by students, and no unnecessary involvement of the professor. Authentication is proposed using UNO ID credentials and possibly adjunct biometrics such as fingerprints to limit the number of students attempting to circumvent the attendance process.

# 2 AUDIENCE

Our application has two principal users, professors and students. Students will initially use our system via a mobile application, whereas professors will administer the application via website portal. For our educator audience, we want to ensure it is simple to initially set up their classes at the start of the semester since the point is to reduce time spent performing attendance activities, not just displace it. Similarly, for students the application must be seamless as they will not want to be burdened by a new mandatory process.

# 3 INTRODUCTION

## 3.1 Purpose of the System

The system’s purpose is to address pain points which exist for teachers and educators to track the attendance of students on a day-to-day, and even class-to-class basis. These pain points include, but are not limited to:

* Disruption at the beginning of class for roll call.
* Timely processes involving the management of attendance sign up sheets.
  + Remembering to bring sheets to class.
  + Passing the sheets around the classroom while lecturing which can be a distraction to students.
  + Keeping track of the sheets throughout the semester.
  + Ultimately entering the attendance into another system later on which requires scanning through each sheet looking for names which are oftentimes in an arbitrary order.

Our proposed system will address each of these pitfalls through a streamlined process which essentially parallelizes the work of attendance tracking to each student. It does this by requiring each student to be accountable for marking their own attendance which not only limits the total time necessary to track attendance, but removes much of the burden from the teachers. The system will also report data in such a way as to limit needless calculations when a teacher wishes to incorporate attendance into students’ grades.

## 3.2 Scope of the System

The scope of this project covers the creation of both a native mobile app (iOS & Android) and admin web portal. The following two paragraphs discuss the subscope of each of these subsystems, the mobile app and website, respectively.

The mobile app’s scope contains the functionality necessary for students to mark attendance, view their past attendance for the current semester, and dispute any missed or tardy attendances for which a student feels he or she should receive credit. The mobile application will also support biometrics and UNO login information as verification of identity when marking attendance. It is not within scope for the mobile app to support teachers as a means of creating classes and accessing attendance information. This must be done through the web application. Additionally, it is within scope for the mobile app to send push notifications to students when it is time to check in for class to reduce the likelihood of student’s simply forgetting.

The scope of the web portal is to allow teachers to create, edit, delete classes, add and remove students from these classes, and monitor the attendance of students throughout the semester. Additionally, teachers should be able to resolve any disputes student’s raise. It is not a required function of the web application at this time to allow students an additional way of checking in to class. This must be done via the mobile application.

We would also like to note additional functionality outside the scope of this project. We are not trying to address students who may try to circumvent the attendance process by “spoofing” their location or by some other means of marking attendance while actually missing class. Means of avoiding current attendance tracking processes already exist. A student could have a friend sign his or her name to an attendance sheet without actually being present. It is not our goal to address these problems as they are largely unpreventable. However, we hope through the use of UNO verification and/or fingerprint scanners to mitigate this problem to some degree.

Another extension of the project, but outside the current scope, is automatic attendance submission without student interaction. This would require the app to run in the background on a student’s phone and monitor when it is time to check in for class. At that time it could check the student’s location and automatically mark their attendance. This would be ideal as it requires no student intervention and addresses issues of students forgetting to check in. Unfortunately, an app which can monitor a student’s movement at any time raises additional privacy concerns, and for some mobile phone makers, such as Apple, apps are limited in what can be performed in the background. Therefore, at this time, we will require the student to manually submit their attendance.

## 3.3 Objectives and Success Criteria of the Project

In order to consider the project a success, it must provide the following items:

* both an iOS and Android version of the app to support the majority of students.
* a web portal for teachers to create and edit classes as well as monitor student attendance.
* interface with the UNO authentication system to allow ease of verifying students’ identities.
* be able to support a large university, such as UNO, which has hundreds of courses a semester.
* be able to support, at scale, the simultaneous check in of hundreds of students.
* retain 99.9% uptime during semester sessions.

## 3.4 Definitions, Acronyms, and Abbreviations

* UNO - University of Nebraska at Omaha
* RAD - Requirements Analysis Document
* CAA - Classroom Attendance Application
* API - Application Program Interface
* OS - Operating System
* NUID - Nebraska University Identification
* IE - Internet Explorer

## 3.5 References

The following online attendance applications have been found to be in use. Some are freely available to teachers, some provide an evaluation period, others provide for significantly wider scope than is intended for this effort.

* <https://www.myattendancetracker.com/>
* <http://www.engineerica.com/accuclass>
* <https://www.rediker.com/solutions/student-information-system/school-attendance-software>

## 3.6 Overview

The concept of a classroom attendance application is a simple, yet important idea. The ability to reliably record and keep track of student activities is of paramount importance in younger audiences as in elementary school systems. As the maturity level of students increase, along with their level of education, a tracking system needs to adjust accordingly. Many instructors at the University level encourage attendance of lectures by offering a small percentage of overall grade as reward for making it to class. Other instructors have taken it a step further, not only providing such positive reinforcements, but also injecting negative feedback if too many classes are missed. The act of recording attendance should be the responsibility of the student, yet provide tracking and analytic abilities for the instructor.

# 4 CURRENT SYSTEM

Attendance tracking has been around for a decades, maybe even centuries, with the earliest solutions comprised of paper-based record keeping. In the 21st century, surrounded by technology, there are still some courses where professors continue to use roll call, or sign up sheets, as their go-to attendance process. Not only do these manual procedures take a chunk out of already limited lecture time, but they are susceptible to error. For example, given a professor passes around an attendance sheet, he or she must also collect it, keep track of it, and at some point, use it in grade calculations. If at some point it is misplaced, there is no fallback. If, when manually perusing the sign up sheet to determine if a student is there for a given day, the professor glosses over the student’s name, that student does not get credit. Educators require a simple solution to this simple process which does not require so much manual entry, does not impose upon class time, and is less susceptible to error.

Consider also the various software solutions which already exist for tracking attendance (see [3.5 References](#_spoap3tcp156)). Even the majority of these systems require duplicate entry of student records and are not tightly integrated into the technical systems provided by the educational institution. In many cases, they still mandate a professor perform roll call or pass around a sign-up sheet, which is then entered into the attendance tracking system. In our modern culture, we can be confident that almost every college student has a smart device. Since it is to the benefit of the student to attend each lecture, we should use these devices, transfer the responsibility of attendance to individual students, and take away some of the burden from educators.

# 5 PROPOSED SYSTEM

## 5.1 Overview

Our proposed system consists of a web application and mobile app. The mobile app will serve as the student’s way of marking their own attendance. Student’s will also be able to use it to view their attendance records and dispute any tardies or absences. The web application will give teachers insight into the attendance of the student currently enrolled in their classes and allow them to use this information for analysis or grading purposes.

## 5.2 Functional Requirements

### 5.2.1 Functional Requirements of the Web Portal

* The web application must require user authentication.
* The website’s authentication must only allow professors to sign in.
* The welcome page should provide one-click-away navigation to the professor’s classes, disputes, and profile settings.
* The classes page should display the professor’s active semester courses, and only the authenticated professor’s courses, as well as support adding and removing students from the classes.
* The class should be configurable to allow a check in window of X minutes before class starts and Y minutes after class has begun.
* The reports page should provide the number of absences and tardiness for quick computation of attendance percentages to use in grading.
* The disputes page should list all the active disputes for the students enrolled in the professor’s classes and allow the professor to accept or reject the dispute.
* Accepted disputes should update the student’s attendance record for the disputed day to show present.
* Rejected disputes should keep the student’s attendance record for the disputed day unchanged, but mark it as disputed to prevent future disputes of the same record.

### 5.2.2 Functional Requirements of the Mobile Application

* The mobile application must require student authentication via UNO credentials, or biometric fingerprint scan.
* The mobile application’s authentication must only allow student’s to sign in.
* The welcome page should show student the active semester’s courses in which he or she is enrolled.
* The welcome page must also contain a button which the student can use to check in. This button should become active or disabled based on whether the student has is currently in a check in window for one of his or her enrolled courses.
* The welcome page should only load the student’s active courses at login, not each time the view becomes active, as these should not change within the course of a student’s session.
* A message above the check in button should display what course the student can currently check in for, and if it is not an active check in window, it should display the amount of time until the next check in window and for which class that would be.
* The class view should display all the attendance records for each day of the course which has passed. The records should contain the date and status. Valid statuses are: present, tardy, absent, and pending review.
* The class view should update the attendance records’ statuses each time it is loaded to ensure the most recent data is available.
* Only attendance records with a status of tardy or absent should provide the student with a button to the dispute page.
* When an attendance record is under dispute, or the dispute has been rejected, it should no longer show the button to the dispute page.
* The dispute page should provide a single textbox which allows the student to type a freeform text justification for why he or she was late or missed class.

## 5.3 Non-Functional Requirements

### 5.3.1 Usability

* The administrative web portal will support all modern browsers including Chrome, FireFox, Safari, and IE on both desktops and mobile devices.
* The mobile application must support both iOS (iPhones) and Android devices.
* Any action within the system which takes longer than 3 seconds should display a loading gif or message to the user to wait.
* Any function of the application should be reachable in less than 3 clicks.
* A student should be able to mark attendance in less than 10 seconds.
* The web portal should be responsive design to assist use in mobile browsers.

### 5.3.2 Reliability

* Data on student attendance should be retained for the last year.
* Nightly backups should be made of the system’s data after the final class has completed.
* Production bug patches should be deployable within 10 minutes of verifying the fix.
* Uptime must be 99.9% from the first active check in window of the day to the end of the last check in window of the day throughout each academic semester.

### 5.3.3 Performance

* Given each class could contain over one hundred students, the system could potentially have thousands of students marking attendance around the same time window. Therefore, the backend portion of the system must be able to support this level of traffic.
* The system should be capable of handling normal, intermittent usage by students throughout the day who are simply checking their attendance records.

### 5.3.4 Supportability

* System administrators should be able to access and update the production database for debugging and resolution of issues.
* Mean time to bug detection should be less than 1 hour.

### 5.3.5 Implementation

* The system must integrate with UNOs authentication.
* The system must support fingerprint scans as a quicker authentication alternative after the user has logged in at least once with their UNO credentials.
* The system must integrate with UNOs enrollment database to automatically provide teachers with a list of their courses for the semester, and students with the courses in which they are enrolled.

### 5.3.6 Interface

* The interface must provide a consistent look throughout the web portal and mobile device.
* Actionable interface items should be clearly identifiable, thus not appearing as static elements on the page.

### 5.3.7 Packaging

* Student mobile app must be meet all requirements for submission to the Google and Apple App Stores.
* The mobile apps must be free to uno students.

### 5.3.8 Legal

* By logging into the mobile application, students must be made aware that they are agreeing to allow the attendance application to retrieve their enrollment information. A disclosure should provide the use and limits of information obtained.
* Both the Android and iOS apps must request access to a student’s location so as to obtain consent.

## 5.4 System Models

### 5.4.1 User Stories

User stories are broken out into two groups for clarity. First instructor stories are presented, followed by student stories.

#### 5.4.1.1 Instructor Stories

The following user stories are related to the professor perspective.

As a UNO **Instructor**...

1. I want to be able to manage all my class rosters for a semester in order to track attendance.
2. I want students to check in securely so their personal information is protected during the check in process.
3. I want students to check in with their UNO id / network id, so the check in process is as easy as possible.
4. I want students who are registered for my class to be allowed to check in, so I know who is present.
5. I want students who are **not** registered for my class, to **not** be allowed to check in, so the integrity of the attendance data is protected.
6. I want to be able to view the attendance records of classes, so I can obtain reporting and apply grading.
7. I want to control the check in window, so students are only allowed to check in for the class within a provided time interval.
8. I want to allow early check in, so students have enough time before class starts. For example, 15 minutes early.
9. I want to allow limited late check in, so students arriving within the late window have an opportunity to record their attendance. For example, 15 minutes late.
10. I want to allow late check in, so students arriving after the late time, are listed as tardy.
11. I want to be able to approve/reject an appeal, so attendance is recorded correctly.

#### 5.4.1.2 Student Stories

The following user stories are from the student perspective.

As a UNO **Student**...

1. I want to be able to see a report of my attendance records, so I can verify the records are correct.
2. I want my attendance to be marked, so I get credit for being in class.
3. I want to record my attendance using a mobile device, so I can get credit for being in class.
4. I want to be able to appeal a tardiness/absence, so I can get credit for being in class.
5. I want to be alerted when my attendance record is marked tardy or absent, so I know when a negative activity occurs on my account.
6. I want to know when I have successfully recorded my attendance, so that I have instant verification of a successful attendance record. (ie: thank you message).
7. I want to see my name displayed when I record attendance, so that I am sure my attendance was recorded properly.
8. I want to be able to select options for attendance notification, so that I can optionally choose to receive an email or text message upon successfully recording attendance.

### 5.4.2 Scenarios

As with user stories, the scenarios are also separated into the two categories of instructor and student.

#### 5.4.2.1 Instructor Scenarios

**Feature**: Class Attendance Application - Instructor Perspective

As a UNO Instructor

In order to track and manage class attendance

I want to have a class attendance application to allow students to electronically register.

This system allows check in during a predefined window prior and just after schedule class times

**Background**:

Given I am a UNO Instructor

When I am instructing a class

Then I can manage class attendance electronically

**Scenario**: Manage class roster

Given I am a UNO Instructor

When I am instructing a class

Then I can manage the class roster

And Enter students that have registered for the class

**Scenario**: Secure access to the system

Given I am a UNO Instructor

When I log into the system

Then I want my credentials to be secure

**Scenario**: Student secure access to the system

Given I am a UNO Instructor

When Students access the system

Then I want their credentials to be secure

**Scenario**: Register attendance

Given I am a UNO Instructor

When A student logs into the system

Then The student is allowed to register attendance for the current class

**Scenario**: Disallow Registration of attendance (negative scenario)

Given I am a UNO Instructor

When A student logs into the system

But The student is not registered in the class

Then The student is not allowed to register attendance for the current class

**Scenario**: Report on attendance

Given I am a UNO Instructor

When I request an attendance report

Then I want to see all students records of attendance

**Scenario**: Control check in window

Given I am a UNO Instructor

When I log into the system

Then I want to be able to set start and end times for each class

And Only allow attendance to be logged during that window

**Scenario**: Late Check in

Given I am a UNO Instructor

When A student checks in after the window expires

Then The student is marked tardy

**Scenario**: Manage appeals

Given I am a UNO Instructor

When A student submits an appeal request message

Then I can review the message

And Determine if the message warrants dismissal of the tardy mark

#### 5.4.2.2 Student Scenarios

**Feature**: Class Attendance Application - Student Perspective

As a UNO Student

In order to receive credit for attendance

I want to have a class attendance application to electronically register my attendance

This system allows check in during a predefined window prior and just after schedule class times

**Background**:

Given I am a UNO Student

When I show up for class

Then I can electronically record my attendance

**Scenario**: Verify attendance records

Given I am a UNO Student

When I log into the system

And Navigate to the reporting section

Then I can view my attendance records

And Validate they are correct

**Scenario**: Check in and record attendance

Given I am a UNO Student

When I log into the system

Then I can record my attendance

**Scenario**: Mobile Tools

Given I am a UNO Student

When I want to check into a class using a mobile device

Then I download the app

And I log in to record my attendance

**Scenario**: Late for class

Given I am a UNO Student

When I arrive late for class

Then I want to be able to check in

And Submit a message to the instructor with a reason for being late

**Scenario**: Notification of Negative Action

Given I am a UNO Student

When I am absent/late

Then I want to be notified via email of the negative action

**Scenario**: Notification of Positive Action

Given I am a UNO Student

When I successfully check in to a class

Then I receive a visual notification of a successful positive action (Thank you message)

**Scenario**: Verify login

Given I am a UNO Student

When I log into the system

Then I see my name to verify I have logged in correctly

**Scenario**: Notification Options - SMS / Email

Given I am a UNO Student

When I log into the system

And Navigate to the options section

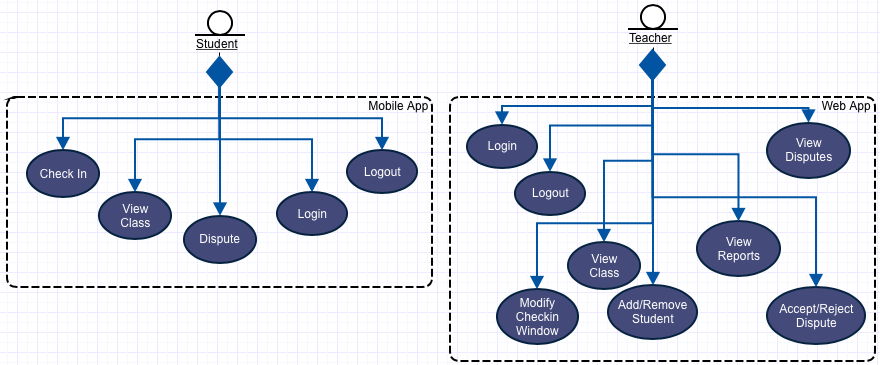
Then I can specify which form of notification I receive

When There is an action on my account

### 5.4.3 Use Case Model

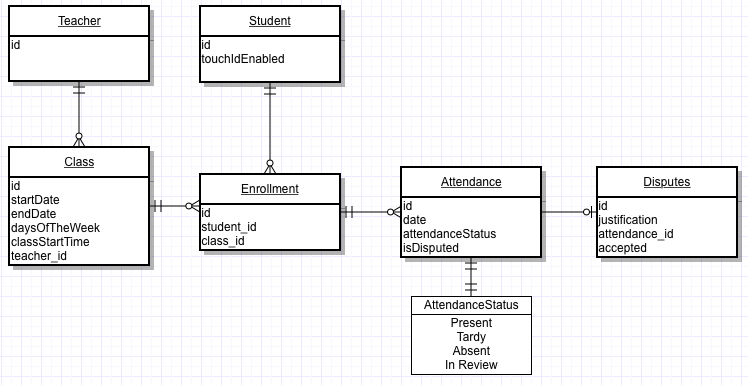
While both the mobile app and web portal fall within the boundaries of the attendance tracking system, we have split the two into separate use case models to show their distinctions and emphasize that the users of the mobile app are limited to students, and the only member interacting with the web portal is the instructor.

In both diagrams the actors must authenticate and are provided login and logout actions. Students are provided read-only functionality in the forms of viewing classes and attendance records. They can also check in for class via the simple check in button, and dispute an attendance record from the class view page. Instructors are provided additional functionality to include viewing and maintaining their classes as well as viewing attendance reports, and accepting or rejecting student disputes.

[[1]](#footnote-0)

### 5.4.4 Object Model

The following object diagram represents the major entities at work in the proposed system. Teachers will have 0 or more classes which they will teach each semester. Students, will have an enrollment for each class they are scheduled to take. In some cases students are not enrolled in any courses so we allow the 0 or many relationship on students to enrollment. For each enrollment, which signifies one student is signed up for one class, we will have 0 or many attendance records which track each date the class meets and whether the student was present, tardy, or absent. When a student disputes an undesirable attendance record, it creates a dispute record which we limit to 0 or 1 as students cannot dispute the same attendance record multiple times.



### 5.4.5 Dynamic Model

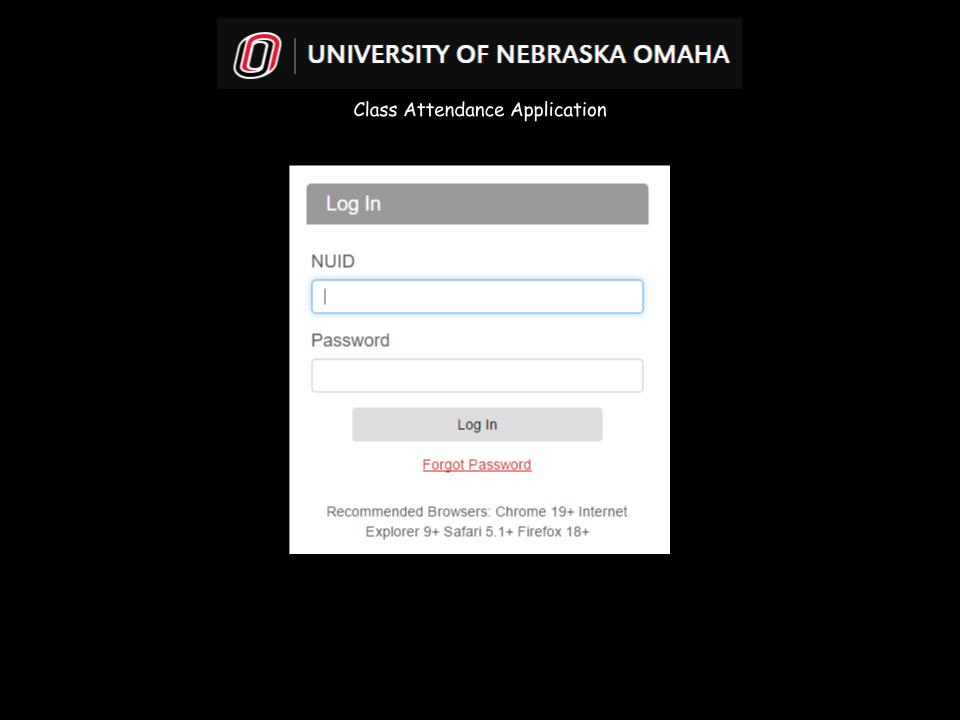
### 5.4.6 User Interface/Navigational Paths & Screen Mockups

#### 5.4.6.1 Web Admin Portal

The administrative portal allows the instructor access to control and configure various elements of the attendance application. The following screen mockups are presented to illustrate the system behaviors as well as interactions with the instructor.

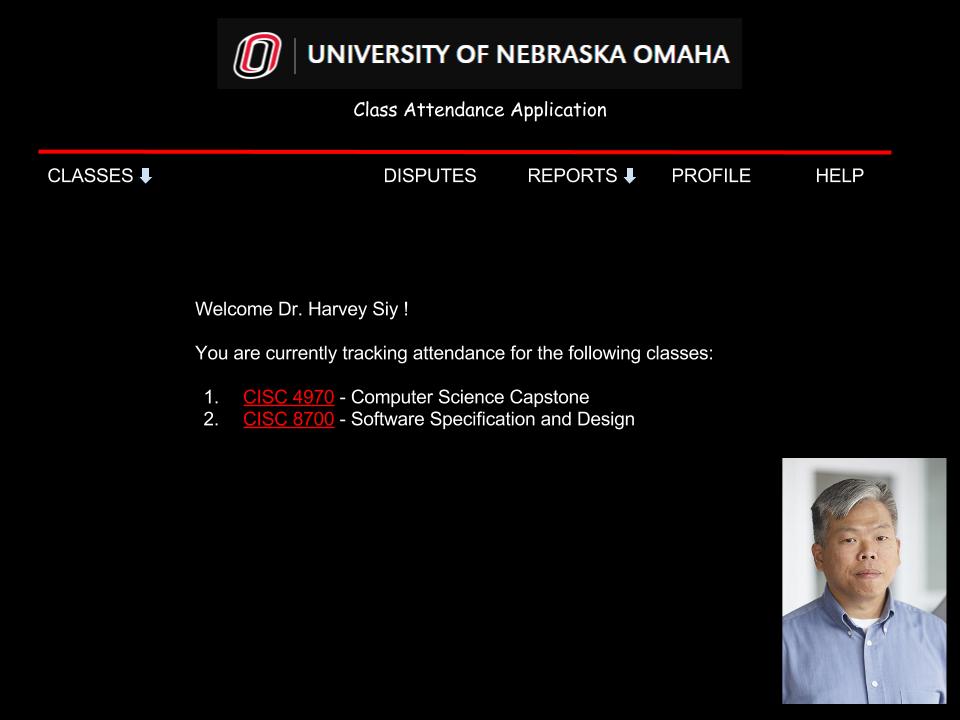
##### 5.4.6.1.1 Login Screen

The login screen follows existing UNO credential authentication methods established. A NUID is required to login along with a password. Since this is an administration portal, only instructors shall be granted access. This will be controlled via (stubbed out) UNO Authentication API. The below mockup shows the standard UNO login screen, with an option for “Forgot Password”.



##### 5.4.6.1.2 Welcome Page

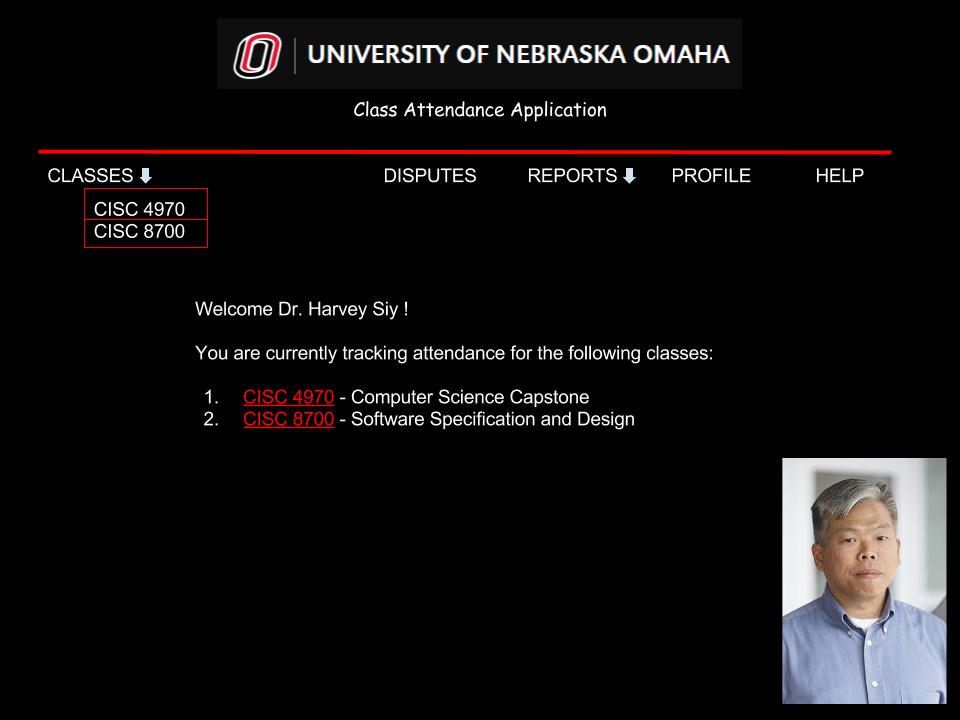
Once the instructor has been authenticated, the welcome page is displayed. A menu of various options is displayed, along with a center screen welcome message presenting those classes in which the instructor is currently tracking attendance. Upon initial login to the attendance application, an instructor would be required to navigate to the Profile tab to configure classes. The below mockup shows a welcome page where 2 classes are already set up. Note the classes listed are hot links, allowing navigation to the class page.



##### 5.4.6.1.3 Classes Page

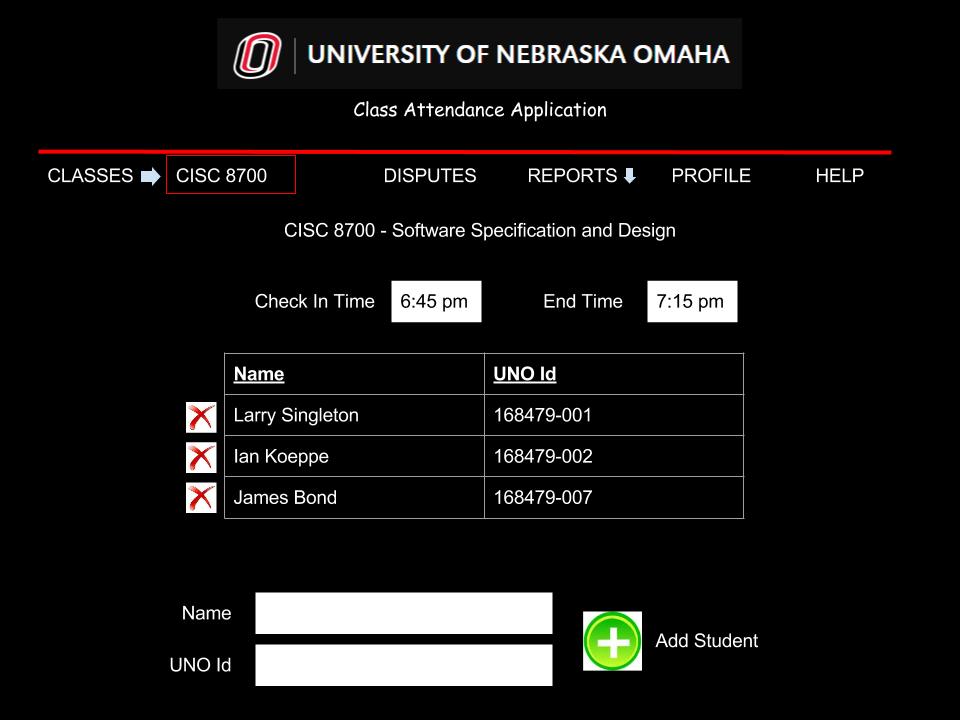
The classes page mockup shows menu navigation to switch contexts between classes.

This same functionality exists from the list of classes shown in the center of the welcome screen. The top level Classes navigation menu item serves as a breadcrumb or context for the disputes and reports menu.



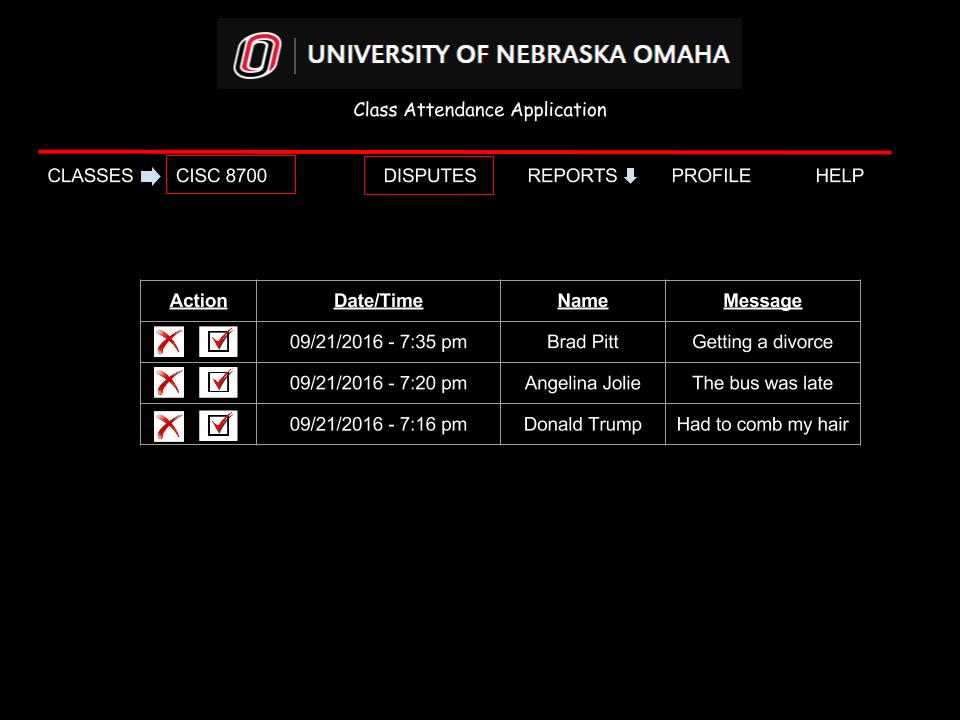
##### 5.4.6.1.4 Class Page

The class page mock up is shown below, note the breadcrumb in the menu showing the context of the page. After class selection is made, a page of users is displayed. The below snapshot shows a page with 3 students entered, an option to delete a student, as well as add a new student to the class.



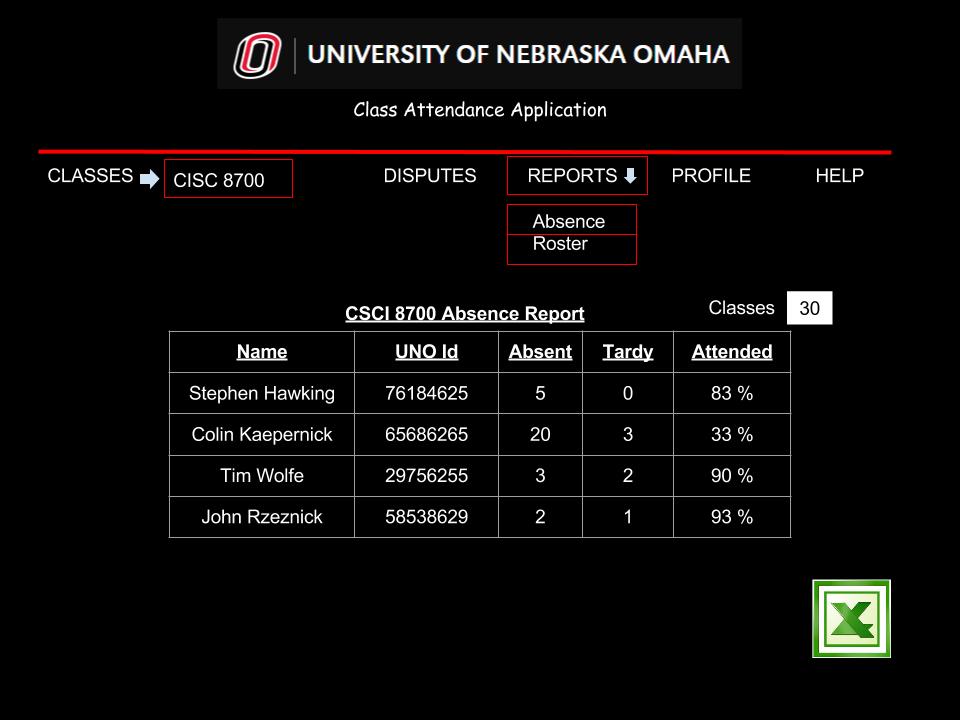
##### 5.4.6.1.5 Disputes

The disputes section, allows the instructor to view notes from students how have checked in after the configured deadline. This allows instructors to review an explanation, and decide if an allowance should be made. Selecting the “X” would deny the request, and provide an email feedback to the student which would count as an occurrence of being “Tardy”. Selecting the check mark however, would accept the reason, and remove the tardy condition.



##### 5.4.6.1.6 Reports

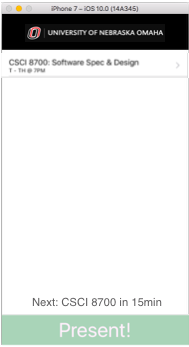
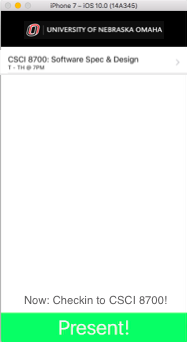
The reports page allows the instructor to select either an absence report or view a class roster. The absence report is shown below, providing details on students that missed a class or were late. A class roster report would include a listing of all students, along with their respective properties.



#### 5.4.6.2 Mobile Application

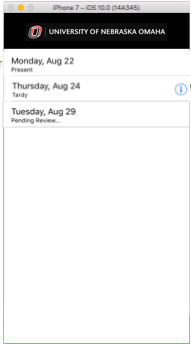
##### 5.4.6.2.1 Welcome Page

After login, the user is presented with the welcome dashboard. This view presents at a glance all the classes a student is currently enrolled in for the active semester. Furthermore, it does not require the student to locate the class they wish to check in to. The application knows, based on the check in times set by the professor, which class is about to start and allows him or her to mark attendance simply by hitting the “Present!” button. Notice when there is not an active check in window for the student’s class, it provides a helpful message of when the student will be able to check in next. By clicking on a specific class in the table, the student can navigate to the class page.



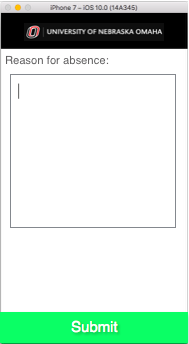
##### 5.4.6.2.2 Class Page

When a class is selected from the welcome page, it displays the class page. This view provides the student with the historical data of their attendance for each class day which has passed. Note we can clearly see the status of the student’s attendance below each date heading. Statuses can include: present, tardy, absent, and pending review. The pending review status signifies the student has disputed a tardy or absence and the teacher has not yet responded. The “i” indicator button in the right margin of the row is only available for data points with the status tardy and absent and allows the student to navigate to the dispute page to dispute the record.



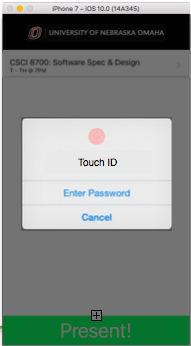
##### 5.4.6.2.3 Dispute Page

When a student taps the “i” indicator button in the class page, it takes them to the dispute page. Here the student can input justification for why they were late or missed class. Through the written justification, the professor will be able to assess the reason and either accept or reject the dispute. When a student disputes an attendance record, it enters “pending review” status. If the teacher rejects the justification, the student should no longer be able to dispute it.



##### 5.4.6.2.4 Attendance Page

When a student pressed the “Present!” button to mark their attendance for class, we will require some form of authentication at that time. The screenshot below demonstrates how we could require a biometrics fingerprint scan. In addition, we could allow students to authenticate with their UNO credentials. In either case, the authentication helps enforce students only being able to claim their own attendance, not that of another student.



# 6 GLOSSARY

* Attendance Record/Entry - Oftentimes used in reference to a single attendance datapoint. For example, a record could refer to a row in the mobile app which shows student A was absent on January 1, 2000.

1. Use Case Diagram [↑](#footnote-ref-0)