Implementation of a Web and Mobile Based Application of Attendance Tracking Software Project Plan

Patrick Horne, Erik Stroud, Lukas Paradiso, and Nicholas Thompson

*Department of Computer Science, University of North Carolina Wilmington, 601 South College Rd, Wilmington, NC 28403, United States*

Table of Contents

Chapter 1: Software Project Plan

* 1. [Introduction 3](#_TOC_250013)
  2. [Project Estimation 4](#_TOC_250012)
  3. Android Activities and Error Instances 5
  4. Project Risk 8
  5. [Schedule](#_TOC_250011) 8
  6. [Project Resources](#_TOC_250010) 9
  7. [Activity Log](#_TOC_250009) 10

Chapter 2: Software Requirements Specification

* 1. Objectives and Content 13
  2. [Software Project Constraints](#_TOC_250008) 13
     1. [Money](#_TOC_250007) 13
     2. [Time](#_TOC_250006) 14
     3. [Scope](#_TOC_250005) 14
     4. [Learning New Tools and Languages](#_TOC_250004) 14
  3. [Entity Relationship Diagram (ERD) 1](#_TOC_250003)5
  4. [Process Narrative 1](#_TOC_250002)6
  5. [Use Case Diagram 1](#_TOC_250001)8
  6. [Validation and Testing 1](#_TOC_250000)9

Chapter 3: Software Design Specification

3.1 Scope 20

3.2 Data Design 21

3.3 Test Provisions 22

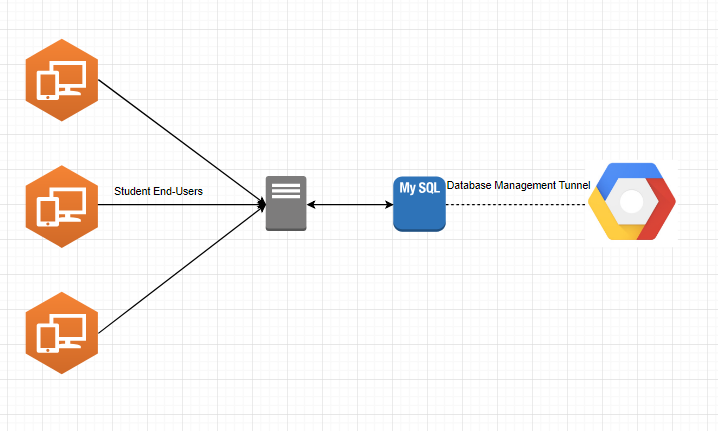
# Chapter 1: Software Project Plan

# 1.1 Introduction

We will be working to create a web and mobile based application of attendance tracking software. This software will be useful in tracking classroom attendance in schools or universities, or any other situation where attendance needs to be tracked. The web application will be accessible to an administrator/teacher who is wanting to keep a record of attendance. The students will also have access to the web application in the case that they have problems with the application on their phones. The mobile application will be used by a user/student to “check in” to a classroom or meeting.

This application relies on a few major functions. The first one being, to generate a class or meeting code for the student/user to log into a class initially. Also, this software will generate a code each day the administrator wants to log attendance. This code will be inputted by the student/user on their mobile application to log their attendance for the meeting. To ensure that a person is truly in attendance, we will use the location services on the mobile phone or specific subnets depending on the university’s implementation of their network. Another function that we will be implementing is for the administrator to view and manage their class records.

Throughout this project we expect to run into some problems including connectivity issues from the mobile application and website to the database, and performance issues with the user’s phone network. At the moment, we are unsure of how to implement that the student/user is actually in their class or meeting.



# 1.2 Project Estimation

We are currently projecting to work on this between ten to twenty hours a week, not including class time, for ten weeks. These hours will be split between two to three meetings a week. Considering that we have four team members working on this project, we have estimated that it will take between 400 and 800 total man hours.

**1.3 Android Activities and Error Instances**

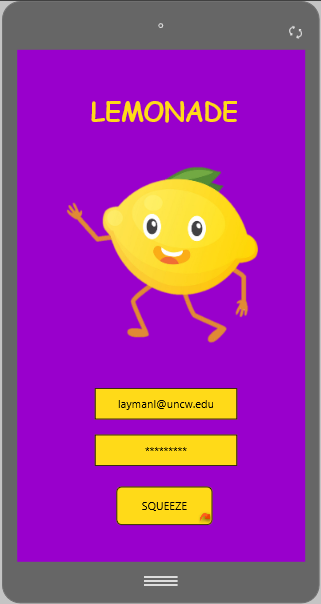
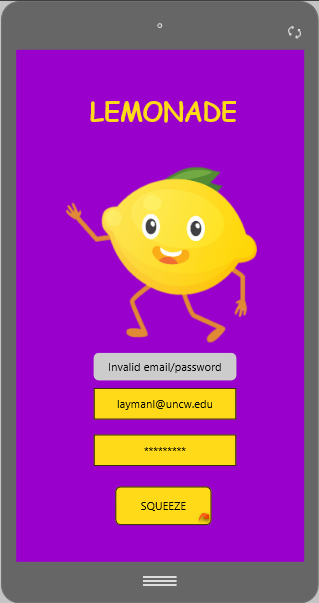
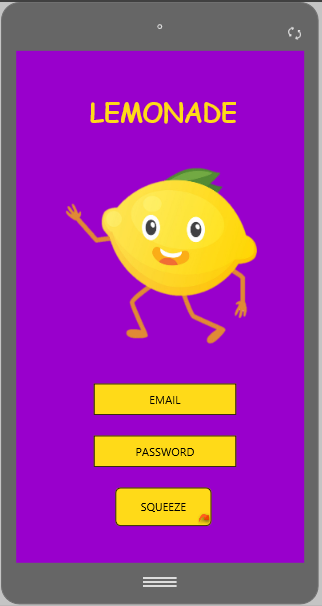
****

Figure 1.3.1 Figure 1.3.2 Figure 1.3.3

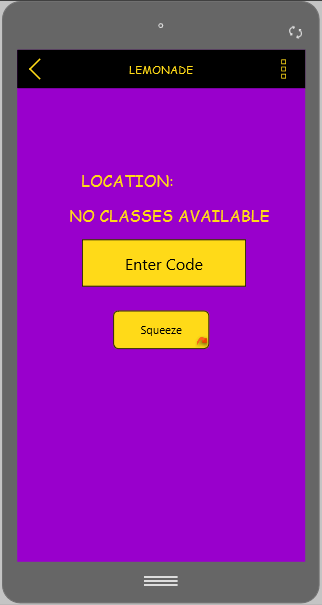
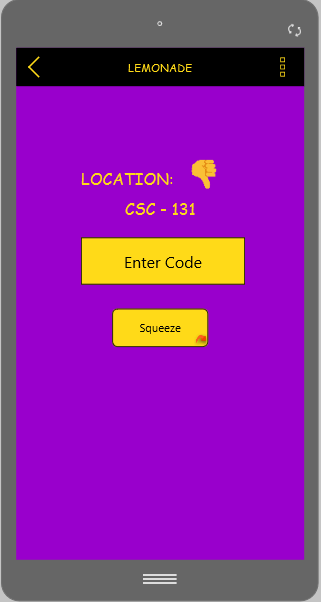
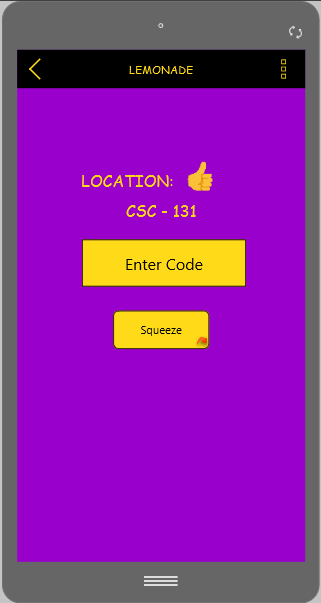
****

Figure 1.3.4 Figure 1.3.5 Figure 1.3.6

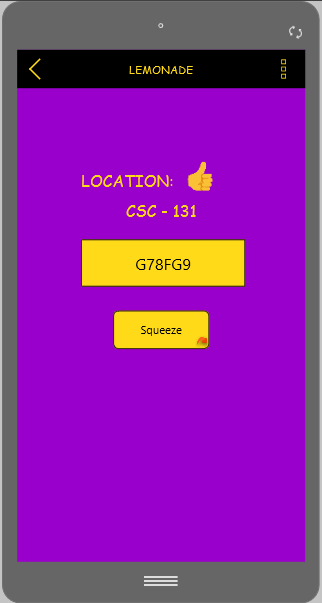
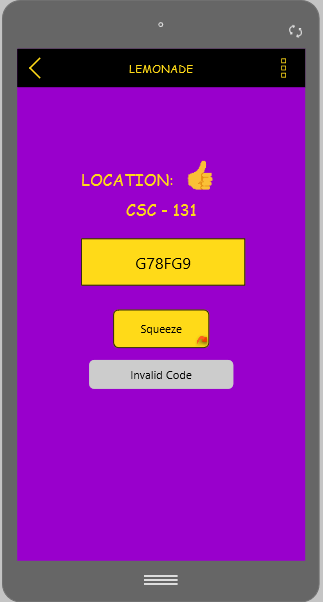
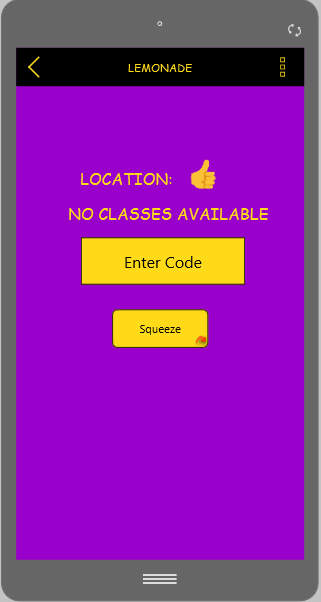
****

Figure 1.3.7 Figure 1.3.8 Figure 1.3.9

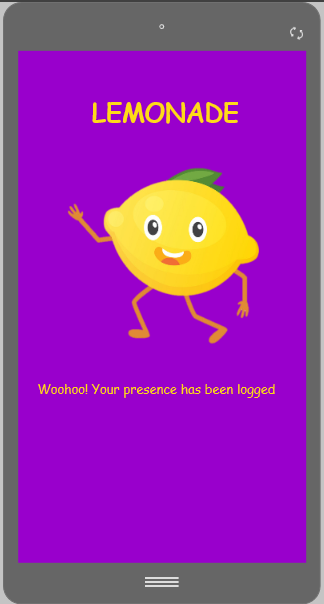
****

Figure 1.3.10

**Figure 1.3.1:** The design of the main activity that allows the user to login. This is where verification between the application and the database occurs.

**Figure 1.3.2**: A toast occurs if the user’s email or password is incorrect.

**Figure 1.3.3:** Both the user’s email and password are correct. The activity then changes to the GeneratedCodeActivity.

**Figure 1.3.4:** An example of the GeneratedCodeActivity’s interface.

**Figure 1.3.5:** An error instance which applies when the user is not located in the building where the class is taught.

**Figure 1.3.6:** An error instance which applies when the user is not located near the building where the class is taught and the time of the class does not match the time of the phone.

**Figure 1.3.7:** An error instance which applies when the user is located in the correct building, but the time on the user’s phone does not match the time of the class.

**Figure 1.3.8:** An error instance which applies when the user is located in the correct building at the correct time but entered an invalid code.

**Figure 1.3.9:** An instance which applies when the user is located in the correct building at the correct time and entered the correct code. The application then moves to the AttendenceLoggedActivity.

**Figure 1.3.10:** The AttendenceLoggedActivity which signals the user that their attendance has been logged successfully within the database.

# 1.4 Project Risks

Being that we all come from different educational and technical backgrounds, there are several risks involved in this project. We do not have much experience with databases and some of the software that we plan on using. For example, we have no experience in using Android Studio, Cloud Software (database), Bootstrap, PHP, UML, or Balsamiq. We also have no experience with the schema Agile. To eliminate these risks, we plan on spending much time researching and learning the software needed to build the software. We are also taking a course in database design and implementation. We are estimating that it will take between four and eight weeks until we are comfortable working with databases and the above software. We also plan to use Bootstrap to help eliminate some of the risk that comes with being unfamiliar with website design and development. We have assessed that some risk could occur with the Android application. One of these risks is that the connection between the application and the database could potentially fail. The way to overcome this is to throw an error message back to the user.

This will allow the user to resubmit their being present request.

# 1.5 Schedule

We are estimating that this project will take us about fifteen weeks to complete. We have these fifteen weeks broken into four sections, gathering information, design and documentation, implementation and testing, and demonstration. We are allowing four weeks for gathering information and details. Although we are only planning on four weeks, we are aware that we will be gathering information throughout the entirety of the project. We are allotting five weeks for the design and documentation section, and three weeks for the implementation and testing section. We are planning to spend three weeks in preparation for our final demonstration of the software.

|  |  |
| --- | --- |
| Week 1 | Begin deciding on group members |
| Week 2 | Discuss project ideas |
| Week 3 | Finalize group members and project idea |
| Week 4 | Research software tools to be used |
| Week 5 | Begin Software Requirements Specification document |
| Week 6 | Design Database |
| Week 7 | Functional Description |
| Week 8 | Behavioral Description |
| Week 9 | Finalize Software Requirements Specifications |
| Week 10 | Database Implementation |
| Week 11 | Mobile Application Implementation |
| Week 12 | Web Application Implementation |
| Week 13 | Documentation Revising |
| Week 14 | Presentation Revising |
| Week 15 | Review and Presentation |

# 1.6 Project Resources

The people involved in this project are Patrick Horne, Erik Stroud, Lukas Paradiso, and Nicholas Thompson. Erik used to work for a college, therefore he brings inside information on how college databases and networks are designed. He also knows how scheduling and class registration works. Nicholas brings his skills in writing and debugging code. He is also good at testing. Lukas has practical experience in database development and mobile application development. Patrick comes from a mathematical background, so he brings those skills to the group. We will also be receiving assistance from Dr. Vetter, Dr. Layman, and Mrs. Ferner.

Throughout this project we will be using our personal computers, along with the computers at the

University of North Carolina Wilmington. Some portions will be hosted in the cloud and on personal computers being used as a server for testing. Android phones will be used for testing the mobile application. We will also be using several types of software including, but not limited to, Eclipse editor for Java, Bootstrap, HTML, PHP, Android Studio, Google Cloud, and MySQL.

# 1.7 Activity Log

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Members | Start | End | Notes |
| 1/28/2019 | Nicholas, Lukas, Patrick, Erik | 2:40 | 5:30 | Begin to talk about project ideas and decided on attendance tracking software |
| 1/31/2019 | Nicholas, Lukas, Patrick, Erik | 2:00 | 3:20 | Started Software Project Plan documents. Introduction, project estimation, project risk, schedule. |
| 2/4/2019 | Nicholas, Lukas, Patrick, Erik | 3:00 | 6:30 | Finalized deliverable 1. Project resources, activity log, and revising of the document. |
| 2/12/2019 | Nicholas, Lukas, Patrick, Erik | 3:30 | 6:30 | Discussing database plan and data objects. |
| 2/15/2019 | Nicholas, Lukas, Patrick, Erik | 5:00 | 7:00 | Begin ERD and Use case diagrams. Also began working on deliverable 2. Objectives and context, software project constraints, time, scope, learning new tools and languages |
| 2/18/2019 | Nicholas, Lukas, Patrick, Erik | 3:30 | 5:30 | Revising deliverable 1 and working on deliverable 2. ERD and description. |
| 2/26/2019 | Nicholas, Lukas, Patrick, Erik | 4:30 | 8:00 | Finalized deliverable 2. Process narrative , use case diagram descriptions, validation and testing. |
| 3/11/2019 | Nicholas, Lukas, Patrick, Erik | 3:30 | 5:30 | Worked on revising deliverable 2 and started writing code for mobile app |
| 3/25/2019 | Nicholas, Lukas, Patrick, Erik | 3:15 | 6:45 | Continued revising deliverable 2 and created power point for presentation in class |
| 4/3/2019 | Nicholas, Lukas, Patrick, Erik | 3:00 | 7:00 | Software design specifications. System objective, major software requirements, design constraints, Data Objects and Resultant Data Structures, files and database structures, testing provisions. |
| 4/8/2019 | Nicholas, Lukas, Patrick, Erik | 5:00 | 7:30 | Data Design Document, Scope, System Objectives, Major Software Requirements |
| 4/11/2019 | Nicholas, Lukas, Patrick | 4:15 | 6:30 | Worked on implementation of android application |
| 4/16/2019 | Nicholas, Lukas, Patrick | 3:30 | 7:30 | Continued to work on android application and began working on web application. Research on making web applications. |
| 4/23/2019 | Nicholas, Lukas, Patrick | 3:15 | 7:00 | Continued working on android application and web application. Unit testing. |
| 4/29/2019 | Nicholas, Lukas, Patrick | 3:45 | 5:15 | Revising everything and preparing for presentation. |

**Chapter 2: Software Requirements Specification 2.1 Objectives and Context**

Our goal is to create a software solution that is easy to use and maintain within a college environment. Therefore, tracking attendance using our software must feel natural and be part of the everyday life of our users. Currently, there are other solutions available; however, they are subscription based and require the university to allocate fund to purchase such products from their budget. Our solution will be lightweight and free giving professors and administrations easier access to our software.

Once the software suite is completed, more features may be implemented in the future considering they do not negatively affect the experience of our end users. Our current goal is to build attendance tracking software that contains a web application and Android application.

# 2.2 Software Project Constraints

The main constraints that we will be focusing on with this project are as follows. Money, time, scope, and learning to use new tools.

# Money

As students, we lack the funding of a company, which means we are required to rely on free or cheap software solutions. To help alleviate the problem, we will use open source software. For example, we will be using MySQL for our database software because it is both free and widely supported in the industry. Android Studio will be used for creating the mobile application because it is free and robust in the features it contains.

In the case of unforeseen hindrances due to the use of free and open source software, we are currently researching which paid software is available within our current budget. Microsoft, for example, offers various software suites for free or at a discounted price for students. Because

many companies offer discounts for enterprise and student accounts, we will be able meet our budget requirements while still acquiring the tools we need.

# Time

Due to the fact that this is a fifteen-week course, we are only allotted that much time to complete the project. Because of our conflicting schedules, our meetings need to be scheduled far in advance. Currently, we are tracking our scheduled meeting times in the form of an activity log (Section 1.6). This will be used in the future to create a complete activity log that contains our total time spent on this project. Another factor that we are accounting for is we need to allot time to learn and familiarize ourselves with some of the tools necessary to build the software.

# Scope

We believe that the current scope of the project seems is realistic. Because we have found a couple open source solutions for certain problems that we will most likely encounter, we can easily implement those early on and finish up the base feature set. Once we know that our initially planned features will be fully implemented within the allotted time of our project, we will consider adding more features to our software.

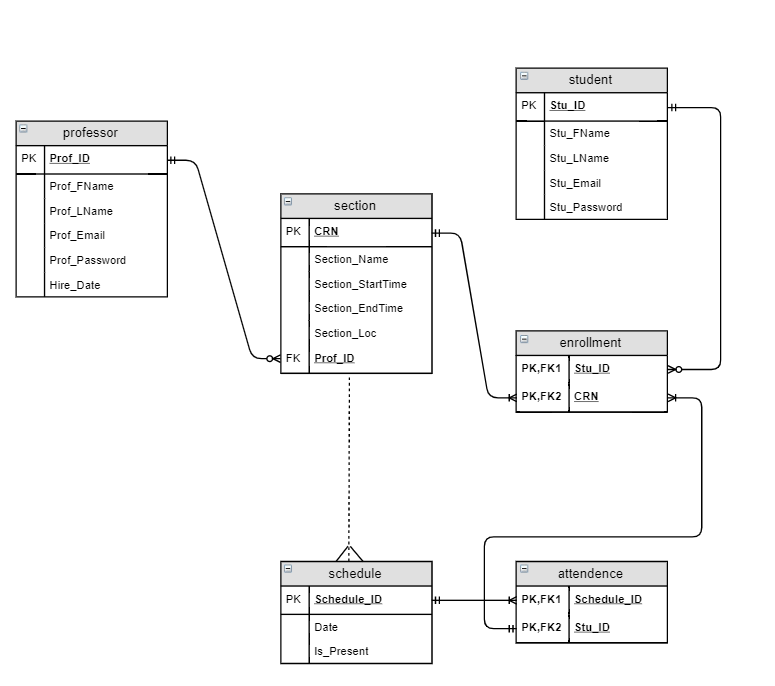
# Learning New Tools and Languages

There will be tools and software languages with which some of our team members have little experience. As a result, certain portions of our project will be delegated in order to keep a single member from being overloaded. This constraint also directly impacts the time constraint as it means less time working on the software and more time learning. We are focusing on simplicity in our project to start with to ensure that we have adequate time to learn the tools and programming languages we need to fully implement our project.

If we can keep all four of these constraints in check, the project should be a success in the end. Of course, not everything will go perfectly, but we believe our expectations are realistic.

Quality checks will be performed throughout our project in order to avoid a domino effect of problems. Problems that arise will be analyzed to determine their causes and the appropriate corrections will be made.

# 2.3 Entity Relationship Diagram (ERD)



Professor will be an entity in our database and will serve as our table of professors. Our Professor table will have the following attributes: Prof\_ID (the employee ID number of the professor), Prof\_FName (the professor’s first name), Prof\_LName (the professor’s last name),

Prof\_Email (the professor’s email address associated with the university), and Prof\_password (the professor’s password which will be encrypted) Hire\_Date (the date which the professor was hired).

Section will be an entity in our database and will serve as the table containing a section of a class. We will not have a class entity because we only care about each class’ individual sections. Section will have the following attributes: CRN (the course registration number of the section), Section\_Name (the name of the section), Section\_StartTime (the time the section will start), Section\_EndTime (the time the section will end), Section\_Loc (the location the section will be taught) and the foreign key Prof\_ID from the professor table.

Student will be an entity in our database and will serve as our table of students. Our student table will have the following attributes: Stu\_ID (the student’s ID number), Stu\_FName (the student’s first name), Stu\_LName the student’s (last name), Stu\_Email (the student’s email address associated with the university), and Stu\_password (the student’s password which will be encrypted).

Schedule will be an entity in our database and will serve as the table containing the schedule of the Section entity. Schedule will have the following attributes: Date (the date of the section on a given day), Is\_Present (the boolean value representing oi the student was present in the section on a given day). Schedule has a weak many-to-one relationship with Section.

Enrollment serves as a junction table between Section and Student. Attendance serves as a junction table between Enrollment and Schedule.

# 2.4 Process Narrative

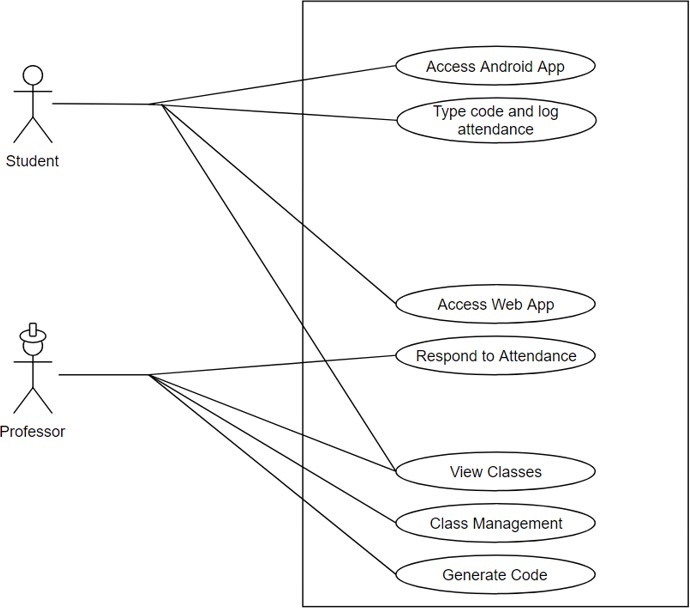
As the developers of Lemonade, we want to enhance previous iterations of attendance tracking software. Our software provides the end users comfort in knowing a student cannot log attendance without the generated code. Along with the generated code, we will have the software

use the IP address from the university’s internet service provider (ISP) to ensure the student is in the classroom. With both the generated code and the IP address, our software will require that students come to class.

To develop the software, we will start with the development of the database. This database will store the information needed for the web application and Android application (as seen in section 2.3). We plan to develop the Android application after the database is implemented. The Android application will be a user interface for the students that are logging their presences in a particular class section.

The steps needed to develop the application will start with the design of our user interfaces, the environment we want to build it in, and the functionality we want it to have. After the design and decisions have been made, we will implement them into code to build the software to our specifications. Our next step will be to repeat the same steps to implement our web application. As a team, we will develop the design of the project and assign positions for the implementation of the software. Because it is a requirement for each group member to serve as the lead developer, we will rotate that position along with the other positions that need to be filled on a bi-weekly basis.

# 2.5 Use Case Diagram



A student has a couple different options on what he/she can do with this web and mobile application. Once a student logs into the application, the student can then input a generated code to show if he/she was present on that day. The student could also log into the web application, where the student can view their class schedule.

As a professor, Lemonade offers full access to the web application which is entailed in the use case diagram above. The professor will be able to manage their classes as well as respond to attendance. This allows them to make changes that are needed on a particular basis. From the

web application, the professor will be able to generate the daily code that the students will be able to use to log their presence in the class.

# 2.6 Validation and Testing

The application should work regardless of what network the device is connected to. There should also be fallback procedures in the case that there are intermittent network issues. While most devices will be connected through WiFi while on their respective campus, many will use cell signal if the class has an outside portion or the user prefers to not connect to public WiFi networks. In case the network does go down and cellular networks are unavailable, the professor should be able to manually take role if preferred.

The mobile and web applications should work without using too many system resources. Therefore, unnecessary resource hogs such as extraneous animations and images will be avoided.

**Chapter 3: Software Design Specification**

**3.1 Scope**

System Objectives:

For this project, the goal is to create a software suite that will allow teachers to keep track of attendance using software. This system will be implemented using a database of our own design and a web applet to allow for management from both the teacher and student’s end. An Android application will also be included for ease of access since many students carry phones around and would prefer to just have a quick and easy method of authentication.

Major Software Requirements:

The software will require the ability for teachers to manually manage a student’s attendance in case there is an issue with the student’s phone or laptop. Using this web application, the teacher will need to be able to generate a random and unique code for the day. This will be included in the teacher’s view in the web application. Students will need the ability to see what classes they are in for verification and a way to type and send the code for the current day.

We were unable to gain access the the university's database to allow users to login with their UNCW credentials so we will be required to store the user’s login information in our database.

Design Constraints:

When designing the software, there are many constraints. Working around the groups availability, monetary issues, learning to use new software solutions, time management, etc. With availability, we have a schedule we can use which has allowed us to progress with designing the software. Monetary issues have been one that we luckily have had access to many free solutions with software that can be used to help create our program. We also are all new to using prototyping software since we generally had not needed to create design documentation for previous development projects for other professors as the scope of those projects was much smaller.

**3.2 Data Design**

Data Objects and Resultant Data Structures

The data objects included will be mainly from the software suites that we use in development. The database will be the main data object to manage as it will contain relevant student information. Within the database, there will be mostly strings and integers. Not everything will need to be stored, as it can be calculated when a query is ran.

Files and Database Structures:

The database ERD is shown on page 11 currently to give an idea of what and where certain data will be stored. For our program, we do not expect there to be any external files needed since all will be stored in the cloud. While there will be temporary storage created by the web browser when accessing the web app, that is handled by the web browser in question.

|  |  |  |
| --- | --- | --- |
| Schedule\_ID | INT(11) | Primary Key, Foreign Key |
| Stu\_ID | INT(11) | Primary Key, Foreign Key |

There will be globally accessible data via the database, but most of that will be hidden from the user as only certain users will be permitted to access all of the data.

|  |
| --- |
| ATTENDANCE |

|  |
| --- |
| ENROLLMENT |

|  |  |  |
| --- | --- | --- |
| Stu\_ID | INT(11) | Primary Key, Foreign Key |
| CRN | INT(11) | Primary Key, Foreign Key |

|  |
| --- |
| PROFESSOR |

|  |  |  |
| --- | --- | --- |
| Prof\_ID | INT(11) | Primary Key |
| Prof\_FName | VARCHAR(45) |  |
| Prof\_LName | VARCHAR(45) |  |
| Prof\_Email | VARCHAR(45) |  |
| Prof\_Password | VARCHAR(45) | Foreign Key |
| Hire\_Date | Date | NOT NULL |

|  |
| --- |
| SCHEDULE |

|  |  |  |
| --- | --- | --- |
| Schedule\_ID | INT(11) | Primary Key |
| Date | datetime |  |
| Is\_Present | Tinyint(4) |  |

|  |
| --- |
| SECTION |

|  |  |  |
| --- | --- | --- |
| CRN | INT(11) | Primary Key |
| Section\_Name | VARCHAR(45) |  |
| Section\_StartTime | VARCHAR(45) |  |
| Section\_EndTime | VARCHAR(45) |  |
| Section\_Loc | VARCHAR(45) |  |
| Prof\_ID | INT(11) | Foreign Key |

|  |
| --- |
| STUDENT |

|  |  |  |
| --- | --- | --- |
| Stu\_ID | INT(11) | Primary Key |
| Stu\_FName | VARCHAR(45) |  |
| Stu\_LName | VARCHAR(45) |  |
| Stu\_Email | VARCHAR(45) |  |
| Stu\_Password | VARCHAR(45) |  |

**Test Provisions**

Test Guidelines:

For testing, we will begin with some internal testing using our own preset credentials. Later on, we will do broader testing using a group of people that will use a set of predetermined user credentials created by us for testing. We plan to ask a professor for a chance to test using our premade accounts in a classroom setting. There is also a chance that we have to use a group of people in a club on campus. Both options should cover a decent portion and if we are able to perform both tests, it will give plenty of data and allow for more results.

There will also be tests that we carry out ourselves for time and geolocation authentication. For example, since a class should only be going on at a specific time at a specific location, we will attempt to log our presence while at Taco Bell or at home.

Some of the other testing that is planned will be with various Android devices and OS versions. Between the class testing and our own personal devices, we should be able to cover a decent variety of devices and OS versions for Android. We will also test Android in a virtual machine to create artificial limitations that could possibly arise allowing us to do

Integration Strategy:

For this, we will purposely create invalid attempts for logging in, logging presence, and other functions to expose flaws in the software that rely on component integration. So this will test how the Android app, web app, and database interact in case there is a flaw that can be abused to create unwanted or invalid results.

Special Considerations:

Some of the aspects we will have to take into consideration are the amount of testers and tester availability. While we can do plenty of testing just by ourselves, a test on a larger scale with more random variables would be better and more thorough as it would cover significantly more possibilities. There is also the consideration of time. The longer things take to be implemented, the less time we will have for testing which could lead to more issues making it past testing.