**Project Analysis**

**CMSC 495 #6380**

**10/11/2020**

**Group #5**

**Members:**

**David Roberts**

**Tiago Feitosa**

**Nicholas Crown**

**Contents**

[**Revision History** 3](#_Toc50495258)

[**Project Analysis for Martial Arts Attendance Tracking Application** 4](#_Toc50495259)

[**a. Outside systems** 4](#_Toc50495260)

[**b. Input Data Source** 4](#_Toc50495261)

[**c. Output data/destination** 5](#_Toc50495262)

[**d. Data Processing** 7](#_Toc50495263)

[**e. Data Processing Subsystems** 7](#_Toc50495264)

[**f. Interface Data and Subsystem Processing/Checking Solutions** 11](#_Toc50495265)

[**g. Risks** 13](#_Toc50495266)

[**h. Possible Enhancements** 13](#_Toc50495267)

[**i. Diagrams** 14](#_Toc50495268)

# **Revision History**

|  |  |  |  |
| --- | --- | --- | --- |
| **Version** | **Date** | **Revisor** | **Changes made** |
| 1.0 | 9/8/20 | Nicholas Crown | Proposed Analysis |
| 1.1 | 9/8/20 | Nicholas Crown | Added diagrams |

# **Project Analysis for Martial Arts Attendance Tracking Application**

**a. Outside systems:** User, browser, AWS services including EC2 and RDS

**b. Input Data Source:** Authorized Users

1. **Login credential form:**
   1. Text input boxes
      1. Username
      2. Password
   2. Submit button
2. **New student form:** 
   1. Text input boxes
      1. Student first name, student last name, student phone number, student address, username, password, password confirmation
   2. Calendar input boxes
      1. Student date of birth, student start date
   3. Drop down selections
      1. Student belt rank, student active status
   4. Submit button
3. **New instructor form:**
   1. Text input boxes
      1. Instructor first name, instructor last name, instructor phone number, instructor address, username, password, password confirmation
   2. Calendar input boxes
      1. Instructor date of birth, instructor start date
   3. Drop down selections
      1. Instructor belt rank, instructor active status
   4. Submit button
4. **New class form:**
   1. Text input boxes
      1. Class name
   2. Calendar input boxes
      1. Class start date, class end date
   3. Drop down selections
      1. Class instructor, class start time, class end time
   4. Submit button
5. **Modify student form:**
   1. Text input boxes
      1. Student first name, student last name, student phone number, student address, username, password, password confirmation
   2. Calendar input boxes
      1. Student date of birth, student start date
   3. Drop down selections
      1. Student belt rank, student active status
   4. Checkbox
      1. Delete student confirmation
   5. Submit button
6. **Modify instructor form:**
   1. Text input boxes
      1. Instructor first name, instructor last name, instructor phone number, instructor address, username, password, password confirmation
   2. Calendar input boxes
      1. Instructor date of birth, instructor start date
   3. Drop down selections
      1. Instructor belt rank, instructor active status
   4. Checkbox
      1. Delete instructor confirmation
   5. Submit button
7. **Modify class form:**
   1. Drop-down box
      1. List of classes will be selectable
   2. Text input boxes
      1. Class name
   3. Calendar input boxes
      1. Class start date, class end date
   4. Drop down selections
      1. Class instructor, class start time, class end time
   5. Checkbox
      1. Delete class confirmation
   6. Submit button
8. **Class attendance form:**
   1. Drop-down box
      1. List of classes will be selectable
   2. Radio buttons
      1. Selections to mark attendance of students from active roster.
   3. Submit button
9. **Calendar display data:**
   1. Month, day, year from system data
   2. Class schedule data
      1. Class start date, class end date, class start time, class end time, class instructor
   3. Attendance data from class attendance roster
      1. Number of students attending class from class attendance record

## **c. Output data/destination**

1. **Calendar display:**
   1. Date
      1. Months, days, years
   2. Scheduled class blocks on calendar grid
      1. Class name, class date, class start time, class end time, class instructor, attendance summary
2. **New student form:**
   1. Message prompt
      1. Update success
      2. Update failed
3. **New instructor form:**
   1. Message prompt
      1. Update success
      2. Update failed
4. **New class form:**
   1. Message prompt
      1. Update success
      2. Update failed
5. **Modify student form:**
   1. Text
      1. Current student record data
         1. Student first name, student last name, student phone number, student address, student date of birth, student start date, student belt rank, student active status, username
   2. Message prompt
      1. Deletion warning
      2. Modification warning
      3. Update success
      4. Update failed
6. **Modify instructor form:**
   1. Text
      1. Current instructor record data
         1. Instructor first name, instructor last name, instructor phone number, instructor address, instructor date of birth, instructor start date, instructor belt rank, instructor active status, username
   2. Message prompt
      1. Deletion warning
      2. Modification warning
      3. Update success
      4. Update failed
7. **Modify class form:**
   1. Drop-down box
      1. List of classes will be selectable
         1. Class list include class name, date, and time
   2. Text
      1. Current class record data
         1. Class name, class start date, class end date, class instructor, class start time, class end time
   3. Message prompt
      1. Deletion warning
      2. Modification warning
      3. Update success
      4. Update failed
8. **Class attendance form:**
   1. Drop-down box
      1. List of classes will be selectable
         1. Class list include class name, date, and time
   2. Text
      1. Current active student first names, last names, and belt ranks
   3. Message prompt
      1. Modification warning
      2. Update success
      3. Update failed
9. **Login form:**
   1. Message prompt
      1. Login attempt unsuccessful message

## **d. Data Processing**

The user will log into the application. Based on the login credentials associated to the database, the application will grant either a student or instructor view. The system will take user inputs to the AWS EC2 server instance and communicate these inputs across the web application interface to make queries to the AWS RDS MySQL database. Queries made to the database are output to generate form data and build display components for the calendar interface. Access to the menu selections will be displayed for instructors. Menu items will allow for instructors to create, modify, and delete records from the database.

## **e. Data Processing Subsystems**

1. **Login form:**
   1. The login form will take the credentials input by the user and will submit them securely to the database to verify the hashed and salted password matches the username and will grant access to the system.
   2. The database will determine if the login is associated to a student or an instructor and provide applicable access controls to application.
      1. Students will be able to view the calendar of classes.
      2. Instructors will be able view the calendar of classes and menu.
2. **New student form:**
   1. Instructors will be able to input new student information via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will be required inputs.
            1. Missing input data will result in a failed message prompt upon submission.
         2. Successful submission of data will result in a success message prompt upon submission.
            1. Student data is submitted to RDS Students table.

A new numeric student ID is created upon successful submission based on last student ID in the students table.

1. **New instructor form:**
   1. Instructors will be able to input new instructor information via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will be required inputs.
            1. Missing input data will result in a failed message prompt upon submission.
         2. Successful submission of data will result in a success message prompt upon submission.
            1. Instructor data is submitted to RDS Instructors table.

A new numeric instructor ID is created upon successful submission based on last instructor ID in the instructors table.

1. **New class form:**
   1. Instructors will be able to input new class information via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will be required inputs.
            1. Missing input data will result in a failed message prompt upon submission.
         2. Successful submission of data will result in a success message prompt upon submission.
            1. Class data is submitted to RDS Classes table.

A new numeric class ID is created upon successful submission based on last class ID in the classes table.

1. **Modify student form:**
   1. Instructors will be able to modify the data of a student via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will not be required inputs.
            1. Blank inputs will be ignored for update.
            2. If the deletion checkbox is not checked, modified inputs will be compared to the original data.

If new values are different from the old, system will display a confirmation prompt.

Confirming changes will update the student record in the Students table.

If new values are the same as the old values, no changes will be made to current record.

Failed message prompt will alert user that inputs were not new changes to the student record and prevent updating Students table.

If new values are a mixture different and same values, system will display a confirmation prompt.

Confirming changes will update the student record in the Students table.

* + - * 1. If the deletion checkbox is checked, system will prompt the user to verify deletion selection.

Confirming deletion will delete the student record from the Students table.

1. **Modify instructor form:**
   1. Instructors will be able to modify the data of an instructor via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will not be required inputs.
            1. Blank inputs will be ignored for update.
            2. If the deletion checkbox is not checked, modified inputs will be compared to the original data.

If new values are different from the old, system will display a confirmation prompt.

Confirming changes will update the instructor record in the Instructors table.

If new values are the same as the old values, no changes will be made to current record.

Failed message prompt will alert user that inputs were not new changes to the instructor record and prevent updating Instructors table.

If new values are a mixture different and same values, system will display a confirmation prompt.

Confirming changes will update the instructor record in the Instructors table.

* + - * 1. If the deletion checkbox is checked, system will prompt the user to verify deletion selection.

Confirming deletion will delete the instructor record from the Instructors table.

1. **Modify class form:**
   1. Instructors will be able to modify the data of a class via form input boxes and drop-down selections and submit it.
      1. Form submission
         1. All fields will not be required inputs.
            1. Blank inputs will be ignored for update.
            2. Modified inputs will be compared to the original data.

If new values are different from the old, system will display a confirmation prompt.

Confirming changes will update the class record in the Classes table.

If new values are the same as the old values, no changes will be made to current record.

Failed message prompt will alert user that inputs were not new changes to the class record and prevent updating Classes table.

If new values are a mixture different and same values, system will display a confirmation prompt.

Confirming changes will update the class record in the Classes table.

1. **Class attendance form:**
   1. Instructors will be able to select the class in which they can take attendance.
   2. Classes for that instructor will be listed via a drop-down box.
      1. Drop down selections will include class names, dates, and times, built from a database query of classes tied to that instructor’s instructor ID.
   3. Instructor can take attendance for classes via radio buttons and submit it.
      1. Two radio buttons for attended and absent will be displayed
         1. The radio buttons will be defaulted to absent if attendance has not been taken.
         2. The radio buttons will be defaulted to the previous attendance submission for the class if attendance has already been taken.
         3. The instructor will select the attended button next to each student on the roster that attends the class.
      2. The system will display the active students in a list format from Students table database query.
         1. Student information will include last name, first name, student ID, and belt rank.
      3. Upon submission, system will check for changes in the Attendance field of the specified class in the Classes table.
         1. If no changes were made, system will display an update failed prompt to the user.
         2. If changes were made, system will prompt the user to confirm selection to update attendance.
            1. Confirming changes will update the Classes table record for the selected class, adding the selected student IDs to the Attendance field
2. **Calendar display:**
   1. Upon login, students and instructors will be able to view the calendar of classes.
      1. The calendar will query the Classes table in the database for dates and times correlating to the displayed month.
         1. Visual blocks will be displayed on the calendar indicating a class name, instructor, start and end times, start and end dates, and attendance statistics.
3. **Menu display:**
   1. Menu will be displayed across the top of the application for instructor level access only.
      1. Menu selections include options to navigate to create or modify classes, create of modify student records, and create or modify instructor records.

## **f. Interface Data and Subsystem Processing/Checking Solutions**

1. **Login form:**
   1. The login form input for password will be salted and hashed by the web application front end on the login page and submitted along with the username to a function that queries the database for the hashed password value stored in the database matching the username. If the password hashes match, the application will load the calendar view. If the password does not match, the web application login page will display an authentication error.
2. **New student form:**
   1. The user will input the desired data in the respective fields. The password fields will be salted and hashed in a similar method to the login form, then compared to each other to ensure they match. Upon submission, new student form page will verify that all input fields have been satisfied, prompting the user if data is missing and halting form submission. If data is sufficient, the user confirms the request and a query is made to the database to get the next available student ID number. The data is then submitted to the Students table with the new student ID value. The hashed and salted password value is also stored in the database associated to the student ID.
3. **New instructor form:**
   1. The user will input the desired data in the respective fields. The password fields will be salted and hashed in a similar method to the login form, then compared to each other to ensure they match. Upon submission, new instructor form page will verify that all input fields have been satisfied, prompting the user if data is missing and halting form submission. If data is sufficient, the user confirms the request and a query is made to the database to get the next available instructor ID number. The data is then submitted to the Instructors table with the new instructor ID value. The hashed and salted password value is also stored in the database associated to the instructor ID.
4. **New class form:**
   1. The user will input the desired data in the respective fields. Upon submission, new class form page will verify that all input fields have been satisfied, prompting the user if data is missing and halting form submission. If data is sufficient, the user confirms the request and a query is made to the database to get the next available class ID number. The data is then submitted to the Classes table with the new class ID value.
5. **Modify student form:**
   1. The user will input the desired student ID in the respective field and submit it for change. The modify student form page will then query the student ID to the Students table in the database. If a record exists, the current data will be loaded on the screen with input selections next to the data. If a new password is entered, the password fields will be salted and hashed in a similar method to the login form, then compared to each other to ensure they match. Upon submission, the modify student form page will verify at least one input has been entered for change with a new value, prompting the user if data is missing and halting form submission. If data is sufficient, the user confirms the request and the data is sent via query to update the database record in the Students table. The new hashed and salted password value is also stored in the database associated to the student ID.
6. **Modify instructor form:**
   1. The user will input the desired instructor ID in the respective field and submit it for change. The modify instructor form page will then query the instructor ID to the Instructors table in the database. If a record exists, the current data will be loaded on the screen with input selections next to the data. If a new password is entered, the password fields will be salted and hashed in a similar method to the login form, then compared to each other to ensure they match. Upon submission, the modify instructor form page will verify at least one input has been entered for change with a new value, prompting the user if data has not been entered or has not changed and halting form submission. If data is sufficient, the user confirms the request and the data is sent via query to update the database record in the Instructors table. The new hashed and salted password value is also stored in the database associated to the instructor ID.
7. **Modify class form:**
   1. The modify class form page will query the database of classes for class IDs, times, and dates and add them to a drop-down selection. The user will input the desired class from a drop-down selection and submit the request. Upon submission, the modify class form will query the Classes table in the database for the selected class information and display it alongside input selections. The user inputs the data that they would like to change. Upon submission, the modify class form page will verify at least one input has been entered for change with a new value, prompting the user if data has not been entered or has not changed and halting form submission. If data is sufficient, the user confirms the request and the data is sent via query to update the database record in the Classes table.
8. **Class attendance form:**
   1. The class attendance form page will query the database of classes for class IDs, times, and dates and add them to a drop-down selection. The user will input the desired class from a drop-down selection and submit the request. Upon submission, the class attendance form will query the Students table for active students and display their first name, last name, and belt rank alongside two radio buttons labeled attended and absent. The class attendance form page will also query the Classes table in the database for the selected class information, including the current attendance field. If attendance has already been accounted for, the radio buttons will update to display if a student was in attendance. If no attendance data exists, the default state of the radio buttons will be on the absent radio button. The user inputs the data that they would like to change. Upon submission, the modify class form page will verify at least one input has been entered for change with a new value, prompting the user if data has not been entered or has not changed and halting form submission. If data is sufficient, the user confirms the request and the data is sent via query to update the attendance field for selected class in the Classes table, associating the student IDs to the class.
9. **Calendar display:**
   1. The calendar display page will query the system for the current date and display that calendar month in a grid format respective to the current year. The calendar display page will also query the Classes table in the database for classes scheduled for the current month displayed. The calendar will then build blocks that display data about classes on their respective calendar days. The total number in attendance for a class will be calculated by identifying the number of student IDs associated to the class’s attendance field.
10. **Menu display:**
    1. Upon login, the application will query the database to determine if the username is associated with a user in the Students or Instructors table. If the user is associated with the Instructors table, the menu will be displayed. If the user is associated with the Students table, the menu will not be displayed.

## **g. Risks**

1. Unauthorized access by students to instructor features.
   1. This can be possibly mitigated by employing a means of session control.
2. Instructors all hold the same level of access across the application.
   1. Modify the access controls to allow for a third level of application control, granting administrative access on select functions.

## **h. Possible Enhancements**

1. Create a means for students to sign up for classes.
   1. Would allow instructors to have a preemptive headcount on attendance.
2. Create a means for students to have access to change their own information.
   1. Students would have their own profile-like interface to change their own account details, including passwords.
3. Create functions to generate reports for statistics on class attendance, student activity, and instructor activity.
4. Create data to track student and instructor progress through belt ranking.

## **i. Diagrams**

A close up of a logo

Description automatically generated

A picture containing game

Description automatically generated

A picture containing game

Description automatically generated

A picture containing game, drawing

Description automatically generated

A picture containing game

Description automatically generated

A close up of a logo

Description automatically generated

A picture containing game, mirror

Description automatically generated

A picture containing mirror

Description automatically generated

A picture containing mirror, drawing

Description automatically generated

A picture containing game

Description automatically generated

A close up of a piece of paper

Description automatically generated