Final Project Documentation

Team 8: Robinson's Karate Schools

By: Muhammad Ahsan, Pedro Bellesa, Jodie Carleton, Andre Granovsky and Maggie Ha

Table of Contents

Team 8 Members

Project Team Leader:	Muhammad Ahsan
Project Co-Leader:	Maggie Ha
Team Member:	Pedro Bellesa
Team Member:	Jodie Carleton
Team Member:	Andre Granovsky

Contributions By Team Members

Since we have five group members, we were all able to work on a number of tasks in parallel and have different roles throughout the entire project. These responsibilities are fairly consistent in regards to what each member did in terms of front-end and back-end tasks. Further, depending on the weekly tasks scheduled we would break out into sub groups of two or three and from there the work load would be split in terms of front and back-end tasks. Therefore we established a set of responsibilities and tasks defined by what our front-end and back-end includes.

Back-End Includes:

- Express route
- Node controller
- Mongodb queries
- Specify role-security for route

Front-End Includes:

- HTML page
- Angular controller
- Angular factory/service
- Add to main app
- Add to angular routes
- Specify role-security for route

1. Weekly Meetings and Status Updates Organized by Muhammad & Maggie
2. Internal Project Tracking Document (on Google Docs) maintained by All
3. GitHub
    a. Milestones and Issues Creation – Maggie & Jodie
    b. Issue Tracking – Muhammad & Andre
    c. Main Repository Management – Muhammad, Andre & Pedro
4. Client Liaison – Jodie
5. Zenit Server
    a. Install Needed Technologies – Andre
    b. Determine Server Usability – Andre
    c. Reverse Proxy to Interface Node.js and Apache – Muhammad
    d. Created RKS Daemon – Muhammad
    e. Server Admin – Muhammad
6. Back-End Security – Muhammad & Andre
7. Front-End Security – Muhammad
8. MongoDB Logical Design – Jodie & Maggie
9. Schema Creation in Node – Andre & Muhammad
10. Initial Application User Interface Design – Pedro
11. Final User Interface Consistency Audit and Modification – Pedro
12. Search Bar Functionality – Pedro
13. Admin Member Pages
    a. Member Detail and List Back-End – Muhammad
    b. Member Detail Front-End – Maggie, Muhammad & Jodie
    c. Field Validations and Testing – Jodie & Maggie
    d. Calendar Components – Maggie & Jodie
    e. Photo Upload Functionality – Muhammad
    f. Member List Front-End – Maggie, Jodie
14. Student Personal Information
    a. Front-End – Jodie & Maggie
    b. Photo Upload Functionality – Muhammad
15. Student Membership Information (Main Student Page)
    a. Front-End – Maggie
    b. Back-End – Muhammad & Maggie
16. Admin Belt Promotions (Rules)

a. Front-End – Pedro
b. Back-End – Muhammad

17. Admin Communications (Main Admin Page)
    a. Front-End – Pedro
    b. Back-End – Andre
    c. Nodemailer Configuration – Pedro & Andre
    d. Facebook Integration – Pedro
    e. Twitter Integration – Muhammad
18. Admin Events
    a. Front-End – Pedro
    b. Back-End – Andre
19. Instructor / Admin Belt Information (Main Instructor Page)
    a. Front-End – Pedro, Andre & Muhammad
    b. Back-End – Andre
20. Admin Attendance
    a. Front-End – Muhammad
    b. Back-End – Andre
21. Admin Shift Reports
    a. Front-End – Pedro
    b. Back-End – Andre
22. Admin Classes, Student Schedule
    a. Front-End – Jodie
23. Instructor Shift Reports
    a. Front-End – Pedro
24. Admin Report View
    a. Front-End and Back-End – Muhammad & Maggie
25. Help Section
    a. Created Script and Videos Tutorial – Jodie
26. Attendance Scanner
    a. Front-End – Pedro
27. Barcode Generator
    a. Front-End – Muhammad

Project Description

Develop a system for a Karate School to provide them with the means to improve communication with their students, aggregate their current internal systems into one, and improve productivity by reducing redundant work for both instructors and administrators. The application required a secure login page with three roles to protect personal information. In addition, to provide a responsive design to allow the instructors to manage the belt promotions from a tablet or handheld device.

The proposed system will allow Robinson's Karate Schools (RKS) to improve the communication with their students while managing their students in a single database. The school currently uses three different computers for three different processes and lacks communication with their students concerning notifications for upcoming events, announcements, and reminders. We want to use an internet based application solution to improve the communication between the school and their students, centralize their systems, and reduce redundant work to increase productivity and organization for the instructors and the administrators.

RKS system employs the concept of roles, where users belong to privileged groups. Groups determine level of access in regards to functional content within the system. Currently there are three types of groups: Admin, Instructor and Student. Use cases listed below belong to admin role, while they repeat within other groups with limited functional access.

Implemented Functional Cases:

- Attendance Tracking System – Manage Student Check In
- Manage Personal Information
- Manage Schedules
- Manage Student Information
- Manage Shift Reports
- Manage Students
- Manage Belt Information
- Manage Attendance (Attendance Management)
- Manage Staff Members
- Manage Staff Schedules
- Manage Shift Reports

- Manage Classes
- Manage Belt Promotions
- Manage Events
- Manage Communications (News/Announcements)

Methodologies

The methodology we used through BTR490 and BTS530 was waterfall and once we started BTS630 we started employing an Agile approach to our system modules being developed. In figure 1.0 you will see what our methodology looked like. In the first two classes of the capstone project, BTR490 and BTS530, we tried to really get a clear as image as possible from the client to avoid changes once the implementation phase came around. However, once we did start developing the system we would reiterate over functionality to see if it produced the desired result the client wanted, test it and then make any necessary changes before moving onto the next functionality.

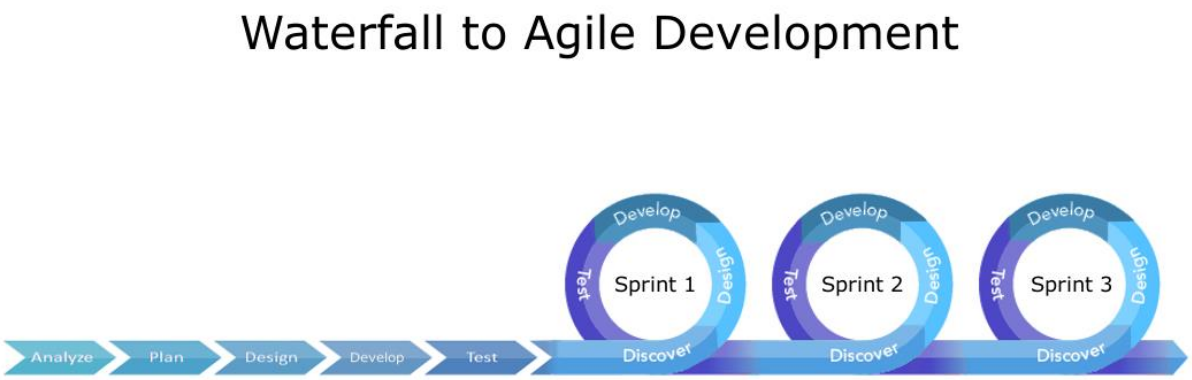


Figure 1.0 *source: http://www.statslice.com/using-both-the-waterfall-and-agile-methods*

Infrastructure

The technologies we used in our system were all open-source, more specifically we employed the MEAN stack. The MEAN stack consists of MongoDB as our data-store, Express as our web framework, AngularJS as our front-end framework and Node.js as our server. Both the front-end and the back-end, AngularJS and Node.js, are written in JavaScript. Express is a web framework that plugs into Node.js and gives us out of the box HTTP capabilities, allows us to jump right into making and handling routes.

Our system is Single Page Application (SPA) where only the initial routes are handled by the server. Once a user requests our URL they get the login page, if they successfully login, only then are they delivered the SPA. From this point of every route is handled by the Angular route handler and any data coming and going is send using AJAX. The default page is relative to the user role, which is admin, student or instructor. From this point on the server side is used like a RESTful API. Every incoming request for data is authenticated to see if the user is authorized to use that API function. If so the data is queried from the database using the MongoDB ORM known as mongoose and send back in JSON format. The Node module used for user authentication was passport.js.

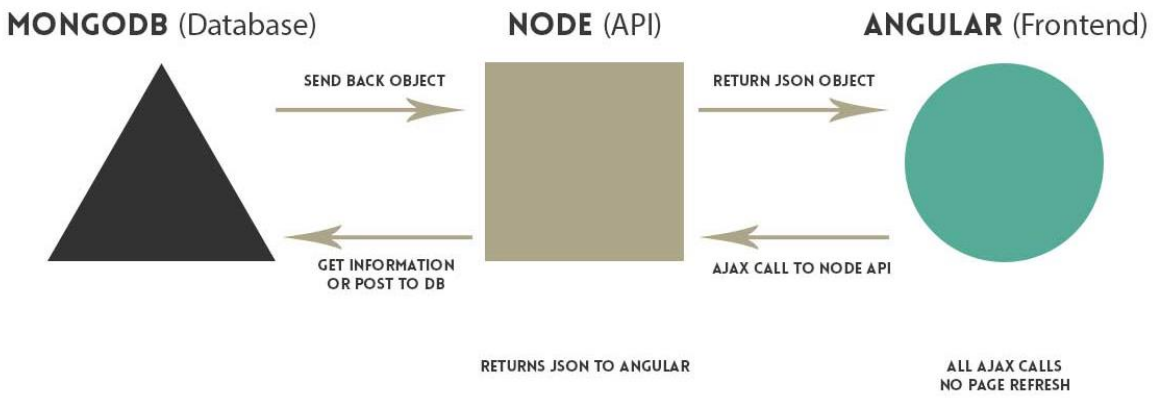


Figure 1.1 *source: https://cask.scotch.io/2013/11/mean.jpg*

This application is kept on Github and the application has two branches. One is called **master** and the other is called **productionVersion**. The master version works on Windows and Mac OS; however, in order to have it run on our Ubuntu server we had to change certain paths to absolute Linux paths. This is the only difference between the two branches and it was regarding the image upload.

We have listed below the JavaScript libraries and server-side modules used in our system, bower and NPM dependencies respectively.

Bower Dependencies

- bootstrap
- font-awesome
- animate.css
- angular
- angular-route
- angular-resource
- bootbox
- jQuery
- ng-file-upload
- bootstrap-calendar,
- angular-bootstrap-calendar
- moment
- angular-ui
- angular-bootstrap
- angular-easyfb

NPM Dependencies

- express
- mongoose
- body-parser
- method-override
- morgan
- passport
- passport-local
- passport-local-mongoose
- cookie-parser
- cookie-session
- ejs
- connect-flash
- connect-multiparty
- connect-modrewrite
- node-uuid
- mongoose-auto-increment
- nodemailer
- twitter

Considering time constraints and requirement specifics in our development iteration, we leveraged on existing open source libraries. With powerful NPM and Bower tools, we were able to construct robust, dynamic SPA in the given timeframe.

Use Case Examples

RKS Video Tutorial: https://youtu.be/8cDGr5Jrw5U

Use Cases Includes:

- Login
- System Overview
- Communications
- Members
- Belt Information

- Attendance
- Report
- Classes
- Shift Reports
- Belt Promotions
- Events
- Attendance Scanner
- Bar Code

Appendix

Final Document for BTS530

The final document for BTS530 is a requirements document for our system. This document contains the project proposal, the business problem, stakeholders, business rules, system use case diagrams, use case specifications (test cases) and non-functional requirements.

Some functionality could not be completed due to time constraints and/or the change in our client's needs.

Robinson's Karate Schools

Requirements Document

Team 8
Revised: Nov 2, 2014

Created By:
Muhammad Ahsan
Pedro Bellesa
Jodie Carleton
Andriy Granovsky
Maggie Ha

Contents

1 Project Proposal

1.1 Team Skill/Experience Inventory:

Student	Skills
Muhammad	Programming Languages: Python, Java, C/C++, ASP.NET, C#, JavaScript, SAS, SQL, HTML5, CSS5, PERL, UNIX shell scripting, XML, AS/400 CL, CUDA Database Systems: MongoDB, HBase, CouchDB, Oracle, MySQL, MSSQL, SQLite Software: IDLE, Wing IDE 101, Enthought Canopy, IntelliJ IDEA, Eclipse, Talend, Borland, Visual studio2005/2010, Dev-C++, Rational Rose
Pedro	Programming Languages: PHP, C++, C#, Java, JavaScript, jQuery, SQL, HTML, CSS, XML, Unix, PowerShell. Platforms: .NET, Oracle Database, Yii, Ajax, OSX, Android, iOS, MySQL, ActiveDirectory
Jodie	Programming Languages: PHP, C/C++, Unix(ksh/bash), Java, PL/SQL, SQL, HTML, XHTML, CSS, ASP.NET MVC, JavaScript, Testing, JUnit Applications/Systems: Oracle, Rational Rose, MySQL, Visual Studio 2013, Apache HTTP Server (LAMP Stack), UNIX, LINUX, Windows, Mac OS, Adobe Captivate
Andriy	Programming Languages: Java, JavaScript, Objective C
Maggie	Programming Languages: PHP, C/C++, C#, Java, ActionScript3.0, JavaScript, MySQL, SQL, PL/SQL, HTML, CSS Software: Adobe Photoshop/Premier/Flash, Rational Rose, Oracle, Visual Studio 2013, Maya, Unreal Engine(UT3,UDK), Eclipse

1.2 Client:

Client's Organization	Robinson's Karate Schools (RKS)
Client's name and contact	Ian Cockburn / Tyler Manning 1131 Gorham Street Unit 2 Newmarket, Ontario L3Y 8X0 Phone: 905-830-1245 Email: newmarket@teamrks.com
Client's position	General Manager / Head Instructor

1.3 Brief description of required system:

The proposed system will allow Robinson's Karate Schools (RKS) to improve the communication with their students while managing their students in a single database. The school currently uses three different computers for three different processes and lacks communication with their students concerning notifications for upcoming events, announcements, and reminders. We want to use an internet based application solution to improve the communication between the school and their students, centralize their systems, and reduce redundant work to increase productivity and organization for the instructors and the administrators.

1.4 Description of any Technological Constraints:

There aren't any specific technologies the client requires; however, one thing that will be required is a database system to keep track of everything.

1.5 Key System Components:

The major components of our proposed system includes: attendance tracking system, student web portal, instructor web portal, and admin web portal. Internet connection will be required to use the web portals.

Attendance Tracking System

The attendance tracking system will be used to check-in students when they come into the school and log their attendance. The system will run on a separate computer with a software client connected to a SQL database where all student information is held. The system will update attendance information that will be used in the web portals.

The system will contain:

- A secure connection to a SQL database for retrieving and writing data

- Card reader input that reads unique barcodes and fetches user information
- Display screen to show user information when card is swiped
- The ability to print new ID cards when a new student has been added into the system

Display
The screen will display user information such as name, photo, class, TIP progression. It will also display notifications regarding next TIP day, exams, and class suggestions.

Attendance Logging
When the card is swiped, the ID will be confirmed and attendance information will be saved locally. The online database will be updated in intervals to be defined by the client with all the records logged within that interval period.

Student Web Portal

The student web portal will be used as the main communication centre between the school and the students.

Students will be able to login to the portal and have access to:

- Personal Information
  - Name, address, photo, and contact information
- Attendance History
  - A count showing total # of classes attended
  - Break down of “A” days and “B” days
  - Detail of dates attended
- Belt Information / Promotion Status
  - Current status
  - Next TIP date
    - Countdown to next TIP
    - Countdown of classes until next TIP
  - Next belt promotion
    - Black belt date - forecast?
- Financial Information - Funds remaining on account / quick card
- General Communication
  - Message of the week
  - Your order is in
  - Reminders
- Upcoming Events
  - Movie night
  - Camp
  - Seminars

Instructor Web Portal

The instructor portal will have two main functionalities: daily events reporting and TIPs approvals. The daily event log is currently being done manually via email, which is at the end of the day broadcast to all concerning parties. The TIPs are currently being tracked via queue card with a date-time and signature.

- Daily event log
    - Central point where any instructor can log in and view the daily logs dating back to the initialization of the system
- TIP approvals
    - Will allow instructors to view the students who took the TIPs test
    - Will allow the instructor to grant the student TIPs points in increments of 0.5

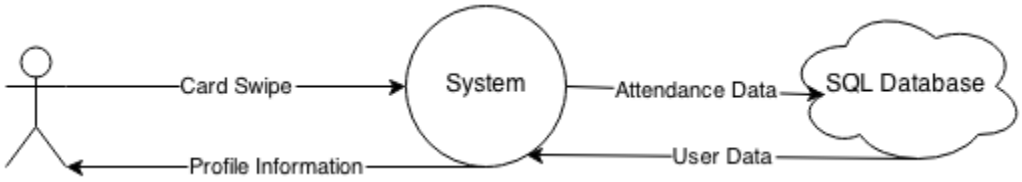
Admin Web Portal

The admin web portal will be used by administrators to manage schedules, accounts, payments, and to send out emails to groups of people.

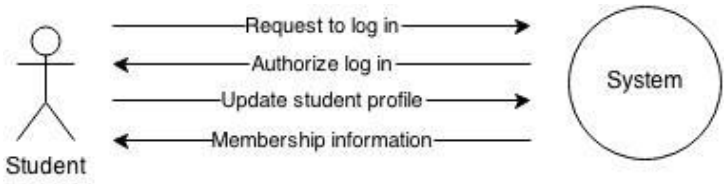
- Student Overview
    - Manage/View personal information
    - View attendance history
    - Manage/View belt information/ promotion status
    - Manage/View financial information
    - Manage/View general communication
- Staff Overview
    - Manage/View personal information
    - Manage/View schedules
    - Manage/View event log
- Payment Tracking
    - Post dated cheques
    - Credit card expiry dates
- Calendar
    - Class schedules
    - Belt promotion schedules
    - Events
- Emailing System
    - News and announcements
        - General communication
        - Upcoming events
    - TIP date reminders/notifications

1.6 Context Diagram(s):

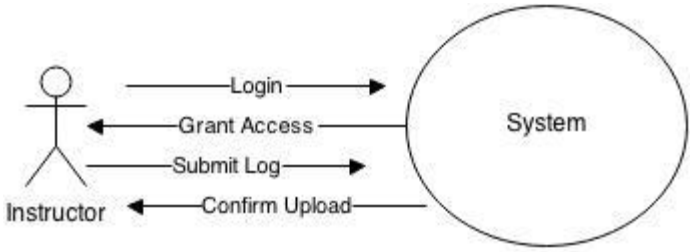
Attendance Tracking System

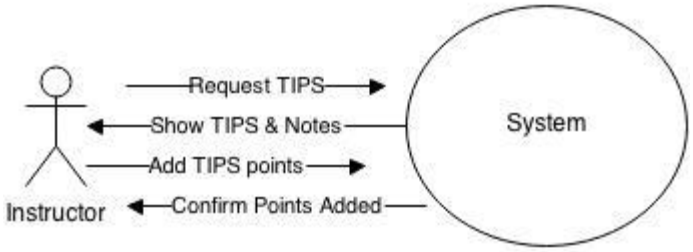


Student Web Portal

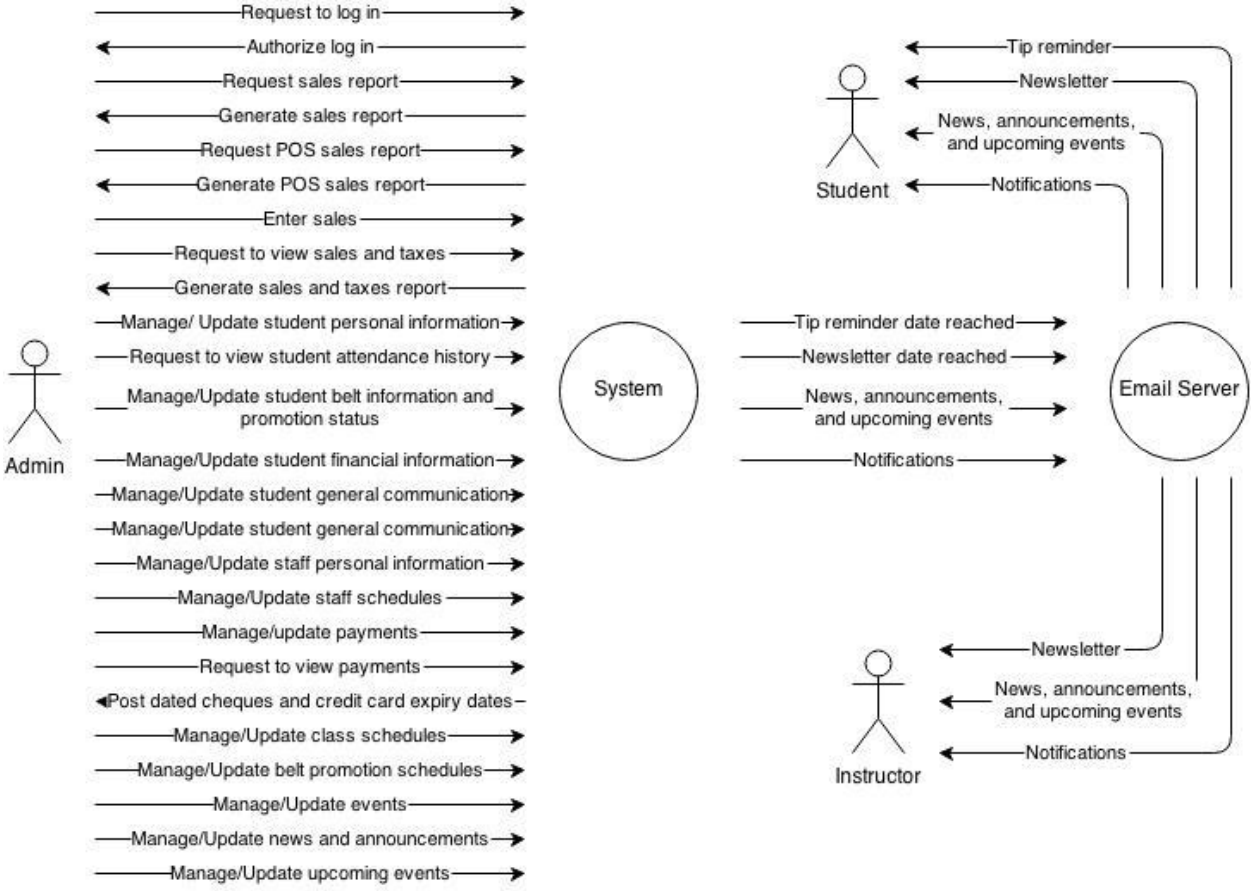


Instructor Web Portal





Admin Web Portal



2 The Business Problem

2.1 The Business

Robinsons Karate School is an organization that promotes and provides a variety of martial arts classes in Newmarket, Ontario. Robinsons Karate School teaches Karate, Jiu Jitsu, Mixed Martial Arts (MMA), Boxing, and women's self-defense to approximately 500 students. They offer classes six days a week and host a number of special events on a regular basis. New students are required to fill out a contract and their information is then entered into the Martial Arts Organizer (MAO) system, the POS system and into a variety of MS Excel spreadsheets used to track belt progression. TIPs are tracked using index cards that are placed into the student folders and MS Excel spreadsheet used by the instructors. Communication with the students is done by notes posted at the school, Facebook, and Twitter. The owner regularly prints reports to review sales and current membership statistics.

2.2 Problem Statement

Robinson's Karate School currently uses three different software solutions to handle their daily workflow.

The first system is MAO, a class management system specific for Martial Arts schools. This system is very outdated in terms of functionality and has not been updated in recent years. It has many limitations and must be run on a dedicated computer. The student information for attendance is held in this system and any user can edit and change this information unknowingly.

The other system is QuickBooks which the staffs only use to log Pro-Shop sales and memberships for tax purposes. The information on students and staff on this system has to be inserted manually as well as items for sale. It also runs on a dedicated computer.

Lastly, they use a webmail client (Hotmail) to communicate with their staff and their students. Again, contact information is on a separate system and must be inserted manually.

Our system will integrate the school's current workflow in a web-based solution coupled with a SQL database hosted online. All contact information will be consolidated and all three systems will be unified. Aside from attendance tracking, there will be no need to run dedicated machines for these systems since they will be online based and accessible from any browser. The attendance tracking will have to run on a dedicated machine in order to accommodate their current procedures and equipment.

Communication, an area that the school wants to focus on, will be improved by implementing a student portal and having all student contact information in one place. This will help them have a better context-based dialog such as notifications regarding progress or attendance.

Also communication among staff will improve by unifying the daily reports and setting up warnings or notifications.

3 Stakeholders

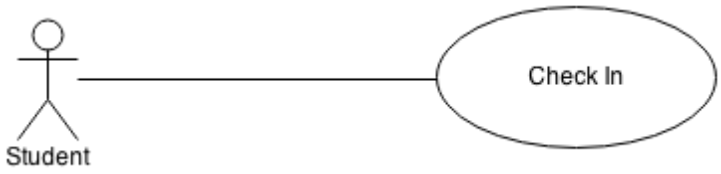
Stakeholder	Role	Power (L/M/H)	Interest (L/M/H)	Opposition (L/M/H/none)
Chris Robinson	Sponsor & Authority & Domain Expert & End User	H	H	None
Ian Cockburn (Administrator)	Authority & Domain Expert & End User	H	H	None
Tyler Manning (Administrator)	Authority & Domain Expert & End User	H	H	None
Instructors	Domain Expert & End User	L	H	None
Students	Customer & End User	L	M	None
Parents	Customer & End User	L	M	None

4 Business Rules

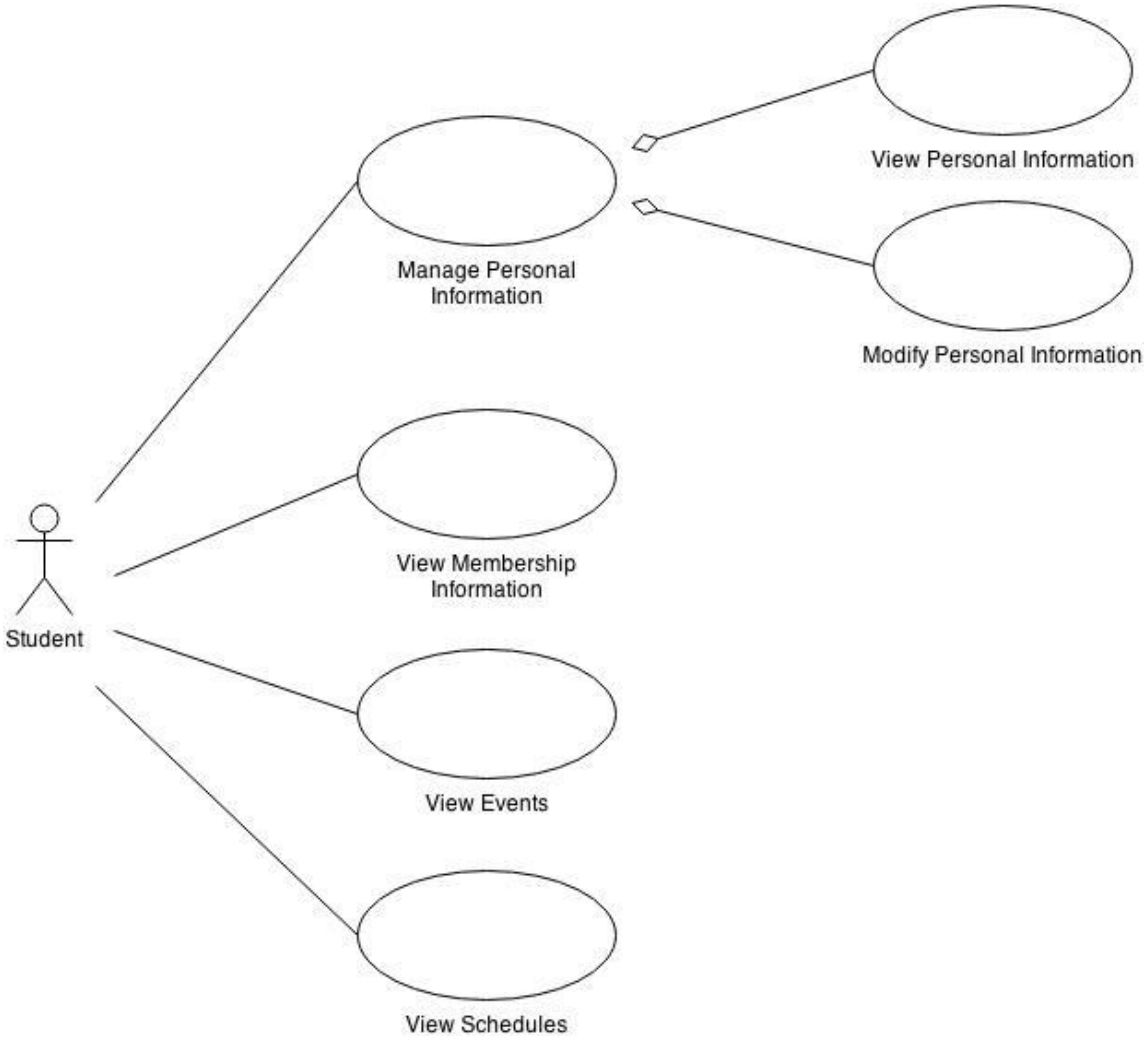
	Description
1.	In order to be promoted to the next belt, student must receive 3 TIPs. The rules for obtaining TIPs are: White - 4 weeks of classes between TIPs Yellow - 5 weeks of classes between TIPs Orange - 6 weeks of classes between TIPs Green - 7 weeks of classes between TIPs Purple - 8 weeks of classes between TIPs Red - Min. 30 classes & 4 “boot camps” between TIPs
2.	When students are tested for their TIP, they will receive either ½ a TIP or the full TIP.
3.	All members are required to sign a contract committing them to a minimum of 6 months. Contracts vary in length from 6 months to 4 years.
4.	All students are required to attend a minimum of two classes a week.
5.	All students must sign in to show attendance in a class.
6.	All programs have an initial registration fee of $199.00.
7.	Payments can be made using MasterCard, Visa, debit card, pre-authorized payments or post dated cheques.
8.	Payments are processed on the 5th and/or 20th of every month.
9.	All new members get a free uniform with enrollment.
10.	All new members get a pre-paid Quick Card with $5.00 with enrollment.
11.	Students must bring their TIP card to the instructor to be tested for a TIP.
12.	An "Instructor End of Shift Report" is due at the end of every night.
13.	If the user enters an invalid password, the user has up to three tries to get it right before being locked out.
14.	Only active students can view their personal information.
15.	All students must have an ID card.

5 System Use Case Diagram (s)

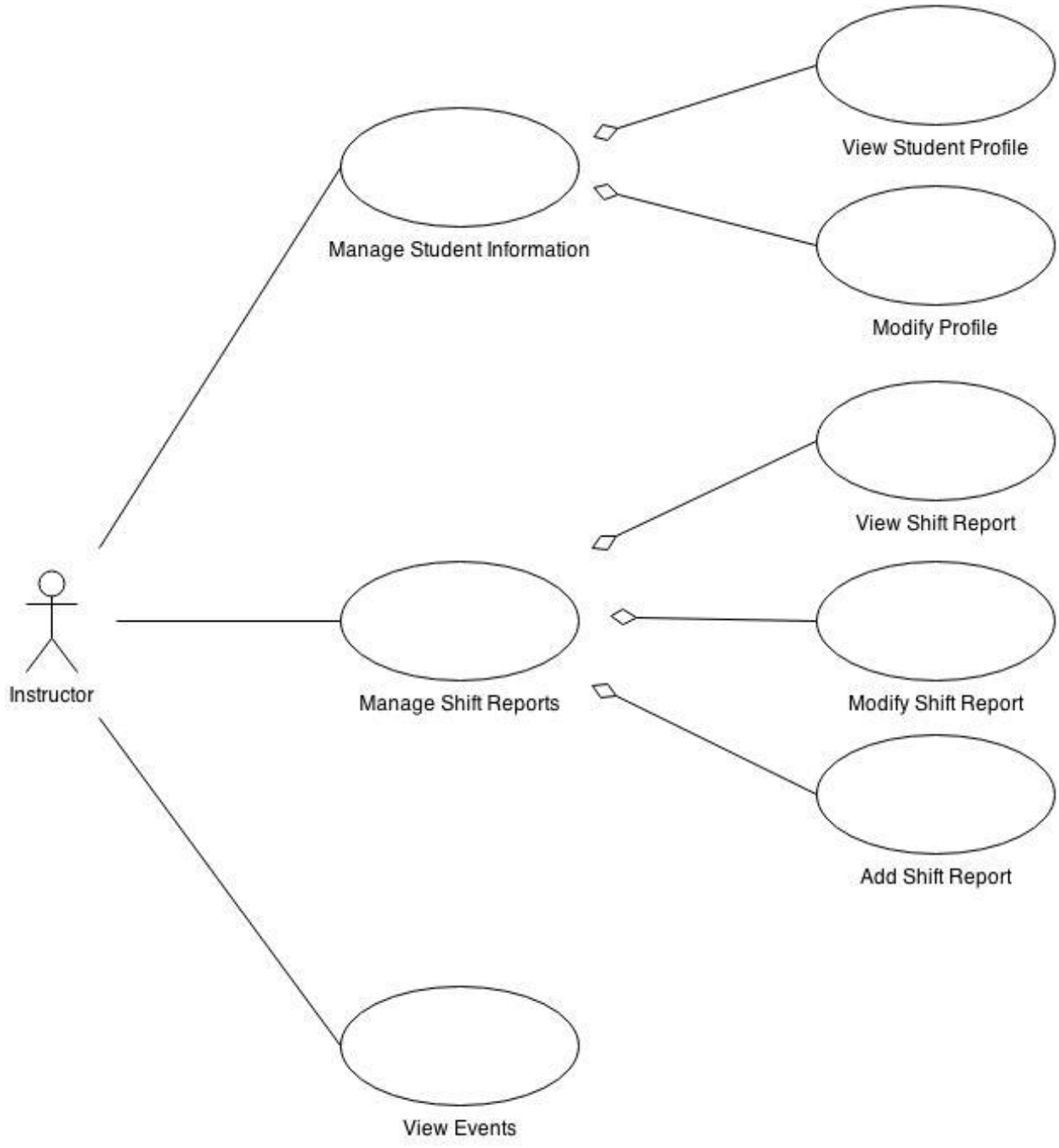
Attendance Tracking System



Student Web Portal

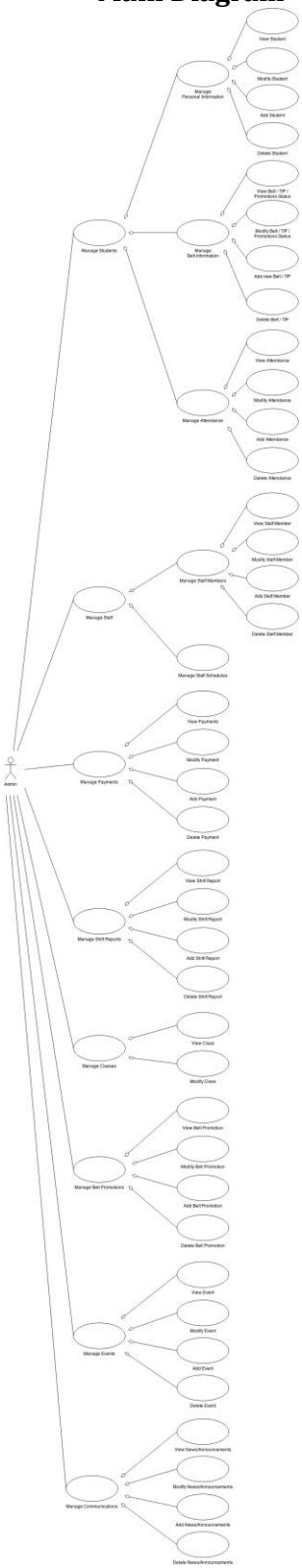


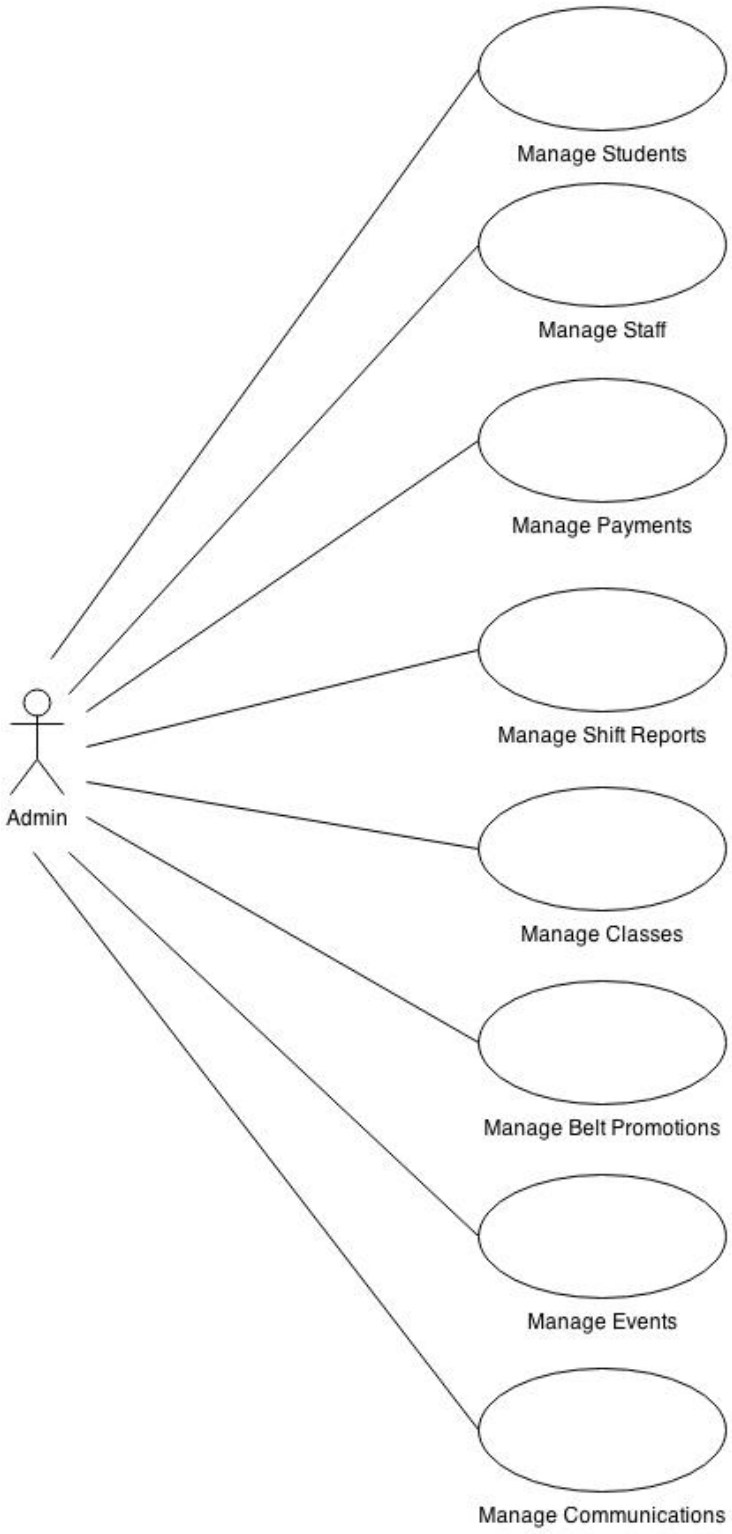
Instructor Web Portal



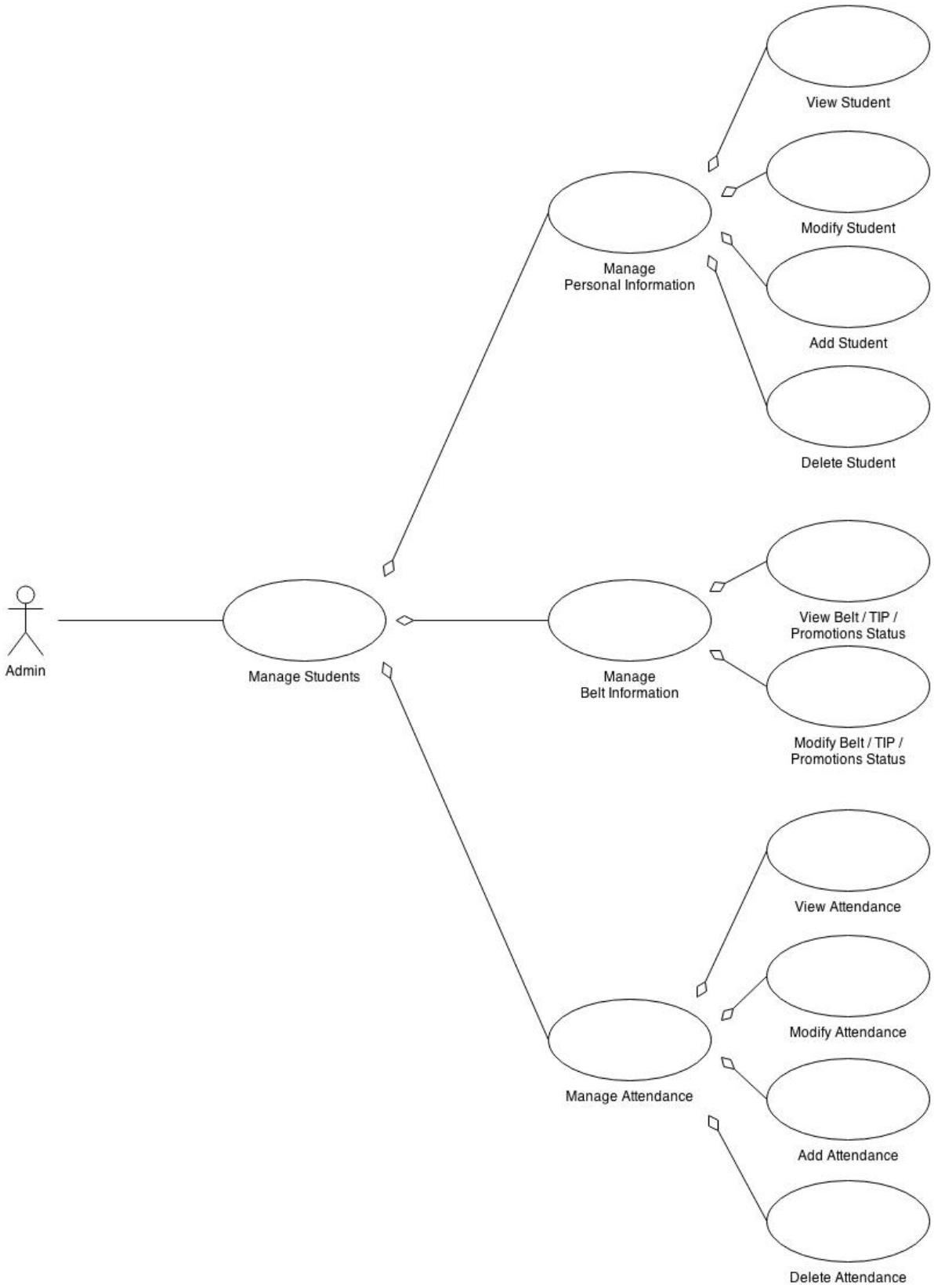
Admin Web Portal

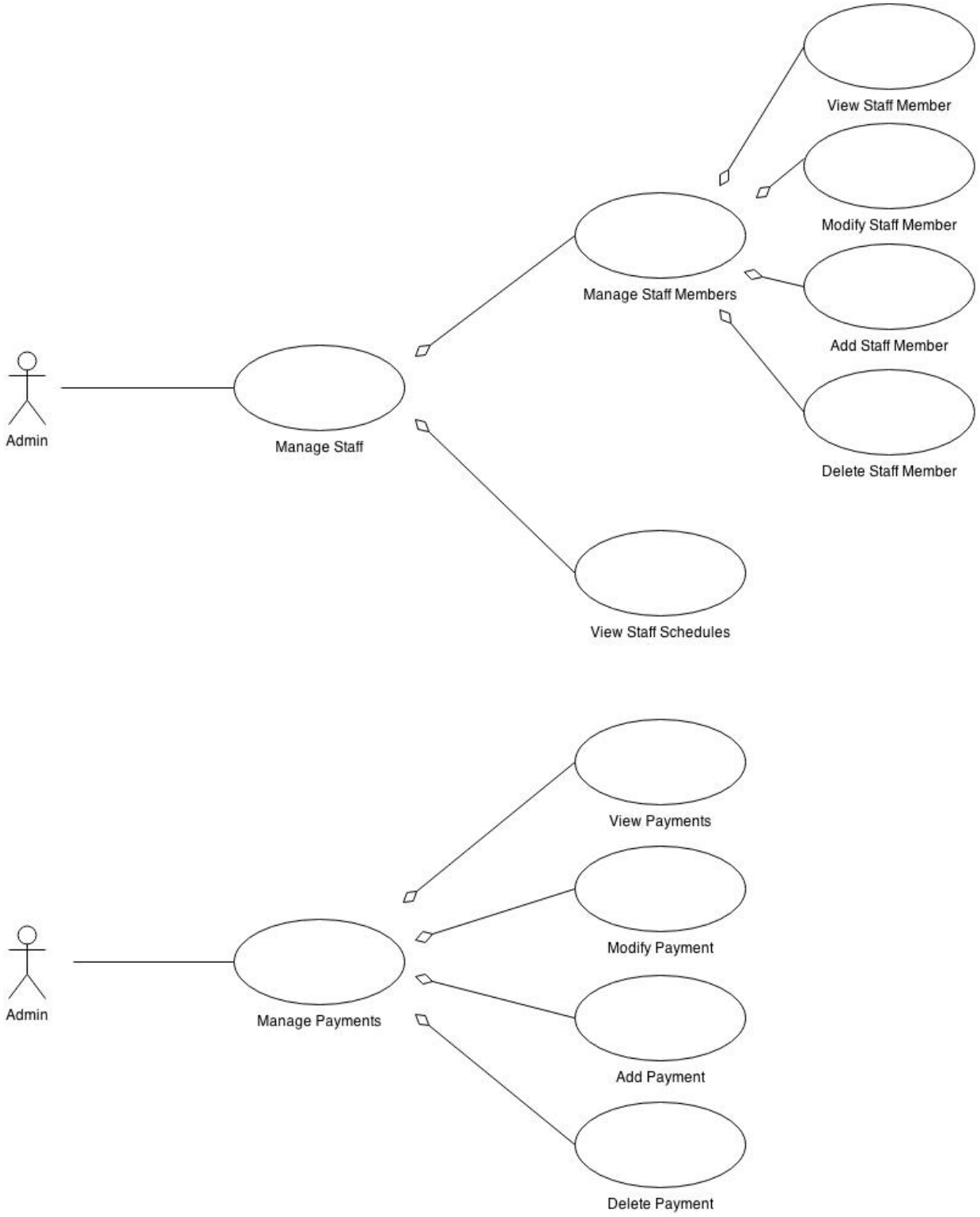
Main Diagram

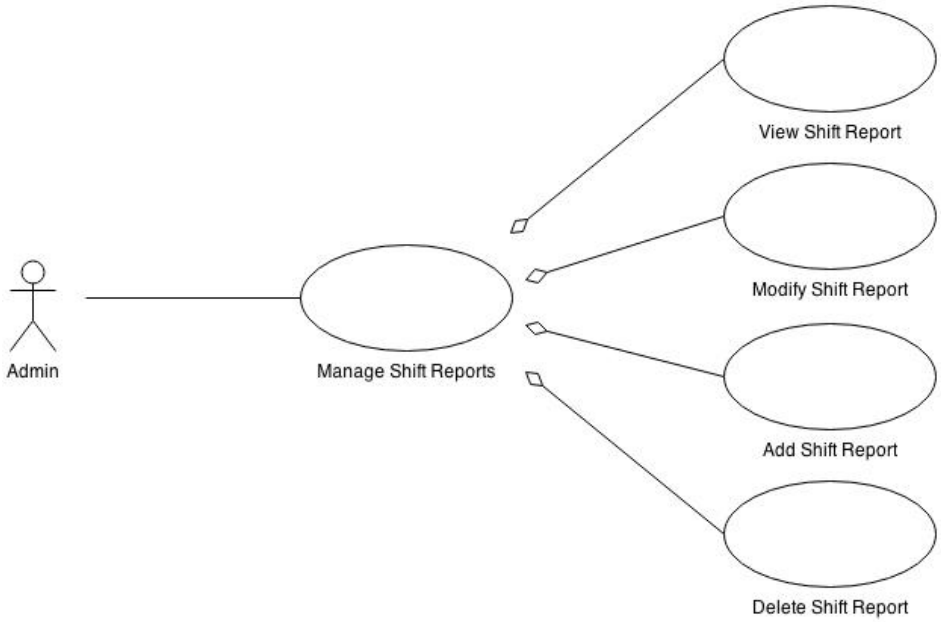


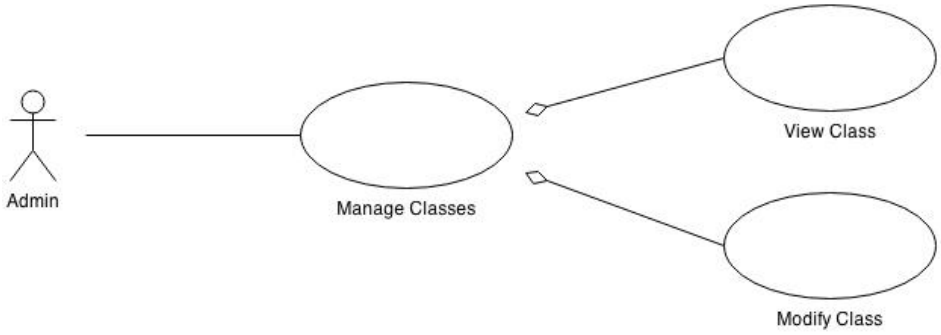


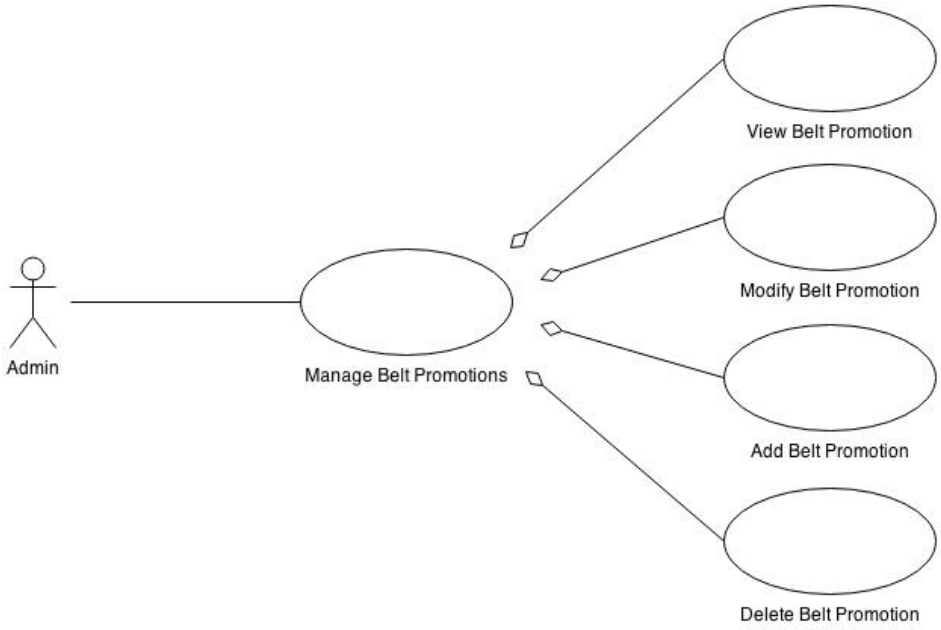
Sub Use Cases

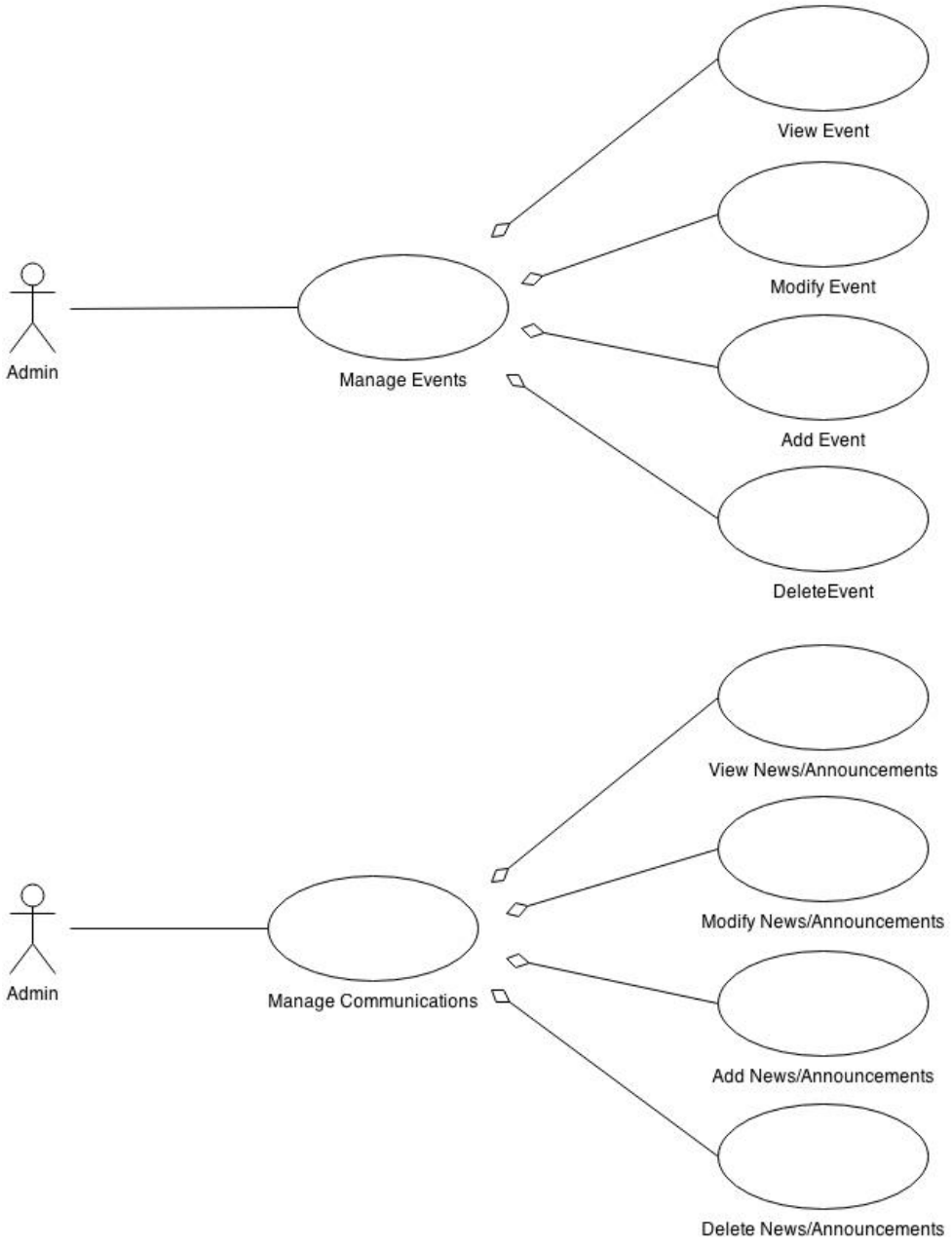



View Staff Member
Modify Staff Member
Manage Staff Members
Add Staff Member
Admin
Manage Staff
Delete Staff Member
View Staff Schedules
View Payments
Modify Payment
Admin
Manage Payments
Add Payment
Delete Payment


View Shift Report
Modify Shift Report
Admin
Manage Shift Reports
Add Shift Report
Delete Shift Report


View Class
Admin
Manage Classes
Modify Class


View Belt Promotion
Modify Belt Promotion
Admin
Manage Belt Promotions
Add Belt Promotion
Delete Belt Promotion


View Event
Modify Event
Admin
Manage Events
Add Event
DeleteEvent
View News/Announcements
Modify News/Announcements
Admin
Manage Communications
Add News/Announcements
Delete News/Announcements

6 Use Case Specifications

6.1 Web Portal - Login

Users login to the web portal.

Author: Pedro Bellesa

Actor(s)

- Student
- Instructor
- Admin

Use Case Preconditions

- User must have internet access.
- Must have working web browser.

Use Case Successful Post Conditions

- User is successfully logged in to the web portal.

Applicable Business Rules

If the user enters an invalid password, the user has up to three tries to get it right before being locked out.

Main Flow:

Additional Precondition: N/A

	Actor(s): User	System
1	User enters login information and requests to log in.	Authenticates and authorizes user with valid credentials. Redirects page to web portal according to user's access level (Student, Admin, and Instructor).

Alternate Flows

	Alternate Flow	Description
A1	User is not found.	Error message is displayed, processing resumes at step 1.
A2	Three login attempts.	Error message is displayed, processing resumes at step 1.

6.2 Attendance Tracking System - Manage Student Check In

Students self-check in to track attendance.

Author: Pedro Bellesa

Actor(s)

- Student

Use Case Preconditions

- Student must possess an ID card.
- Student must have physical access to card reader.

Use Case Successful Post Conditions

- Student attendance is registered to the system.

Applicable Business Rules

- All students must sign in to show attendance in a class.

Main Flow:

Additional Precondition: N/A

	Actor(s): Student	System
1	Student swipes ID card in the card reader.	Finds student information. Registers attendance for the student on current date. Displays user information and notifications. After 3 seconds returns to the main sign in page and waits for student to swipe ID card. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Student is not found.	Error message is displayed. After 3 seconds, returns to main sign in page and waits for student to swipe ID card. Use case ends.
A2	Attendance registration fails.	Error message is displayed. After 3 seconds, returns to main sign-in page and waits for student to swipe ID card. Use case ends.
A3	No ID card is swiped.	Display standby screen. Processing resumes at step 1 once an ID card is swiped.

6.3 Student - Manage Personal Information

Personal information can be viewed or modified.

Author: Jodie Carleton

Actor(s)

- Student

Use Case Preconditions

- User must have internet access.
- Must have working web browser
- User must be logged in as a student.

Use Case Successful Post Conditions

- Personal information is viewed or updated.

Applicable Business Rules

- If the user enters an invalid password, the user has up to three tries to get it right before being locked out.
- Only active students can view their personal information.

List of Sub Use Cases

- View personal information
- Modify personal information

Main Flow: View Personal Information

Additional Precondition: N/A

	Actor(s): Student	System
1	Requests to view personal information.	Displays personal information page. Use case ends.

Alternate Flows: View Personal Information

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Personal Information

Additional Precondition: 'View Personal Information' has been successfully executed.

	Actor(s): Student	System
1	Request to edit personal information.	Displays form for personal information and pre-populated input fields.
2	Changes personal information and saves the changes.	Validates fields, changes are saved in DB. Display student personal information. Use case ends.

Alternate Flows: Modify Personal Information

	Alternate Flow	Description
A1	Invalid input entered into the address.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into the phone number.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display personal information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.4 Student - View Membership Information

Membership information can be viewed.

Author: Jodie Carleton

Actor(s)

- Student

Use Case Preconditions
- User must have internet access.
- Must have working web browser.
- User must be logged in as a student.

Use Case Successful Post Conditions
- Membership information is viewed.

Applicable Business Rules
- If the user enters an invalid password, the user has up to three tries to get it right before being locked out.
- Only active students can view their membership information.

Main Flow:
Additional Precondition: N/A

	Actor(s): Student	System
1	Request to view membership information.	Displays membership information page. Use case ends.

Alternate Flows:

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.5 Student - View Events
Events can be viewed.

Author: Pedro Bellesa

Actor(s)
- Student

Use Case Preconditions
- User must have Internet access.
- Must have working web browser.
- User must be logged in as student.

Use Case Successful Post Conditions
- An event is viewed.

Applicable Business Rules
- N/A

Main Flow: View Event
Additional Precondition: N/A

	Actor(s): Student	System
1	Request to view events.	Display list of all events currently in the system in reverse chronological order and in summary form. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Selects to view calendar.	Display calendar of all events in the system for the month. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.
A3	Selects to view calendar and selects to “see more”.	Display calendar of all events in the system for the month and expand the selected entry. Use case ends.

6.6 Student – View Schedule

Class schedules can be viewed.

Author: Pedro Bellesa

Actor(s)

- Student

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Student-specific scheduled information is viewed.

Applicable Business Rules

- N/A

Main Flow: View Class

Additional Precondition: N/A

	Actor(s): Student	System
1	Requests to view schedules.	Displays list of classes.
2	Select class type.	Displays class information. End use case.

Alternate Flows: View Class

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.7 Instructor – Manage Student Information

Student TIP information can be viewed and updated.

Author: Andriy Granovski

Actor(s)

- Instructor

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an instructor.

Use Case Successful Post Conditions

- Student TIP information is viewed or updated.

Applicable Business Rules

- If the user enters an invalid password, the user has up to three tries to get it right before being locked out.
- When students are tested for their TIP, the student will receive either ½ a TIP or the full TIP.

List of Sub Use Cases

- View Student Profile
- Modify Profile

Main Flow: View Student Profile

Additional Precondition: N/A

	Actor(s): Instructor	System
1	Request to view students.	Displays a list of all students and their belt information. Use case ends.

Alternate Flows: View Student Profile

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Profile

Additional Precondition: 'View Student' has been successfully executed.

	Actor(s): Instructor	System
1	Selects 'No TIP', 'Half TIP', 'Full TIP', or 'Promote' button.	Student belt information is saved in the database. Use case ends.

Alternate Flows: Modify Profile

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login

		screen. Use case ends.

6.8 Instructor – Manage Shift Reports

Shift reports can be viewed, modified, added, or deleted.

Author: Andriy Granovsky

Actor(s)

- Instructor

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an instructor.

Use Case Successful Post Conditions

- Shift reports is viewed, modified, or added.

Applicable Business Rules

- The instructor end of shift report is due at the end of every night.

List of Sub Use Cases:

- View Shift Report
- Modify Shift Report
- Add Shift Report

Main Flow: View Shift Report

Additional Precondition: N/A

	Actor(s): Instructor	System
1	Requests to view shift reports.	Displays a list of all shift reports. Use case ends.

Alternative Flow: View Shift Report

	Actor(s): Instructor	System
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Shift Report

Additional Precondition: 'View Shift Report' has been successfully executed.

	Actor(s): Instructor	System
1	Selects a shift report by clicking on a class date.	Displays detailed shift report information about selected date.
2	Request to edit shift report.	Displays form with pre-populated input fields.
3	Edits information and save the changes.	Validate fields, changes are saved in DB. Display detailed shift report page. Use case ends.

Alternative Flow: Modify Shift Report

	Actor(s): Instructor	System
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display detailed shift report page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Shift Report

Additional Precondition: N/A

	Actor(s): Instructor	System
1	Requests to add shift report.	Display form for new shift report.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display list of shift reports page. Use case ends.

Alternative Flow: Add Shift Report

	Actor(s): Instructor	System
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display list of shift reports page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.9 Instructor – View Events

Events can be viewed.

Author: Pedro Bellesa

Actor(s)

- Instructor

Use Case Preconditions

- User must have Internet access.
- Must have working web browser.
- User must be logged in as instructor.

Use Case Successful Post Conditions

- An event is viewed.

Applicable Business Rules
- N/A

Main Flow: View Event
Additional Precondition: N/A

	Actor(s): Instructor	System
1	Request to view events.	Display list of all events currently in the system in reverse chronological order and in summary form. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Selects to view calendar.	Display calendar of all events in the system for the month. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.
A3	Selects to view calendar and selects to “see more”.	Display calendar of all events in the system for the month and expand the selected entry. Use case ends.

6.10 Admin – Manage Students
Student entries can be viewed, modified, added, or deleted.

Author: Maggie Ha

Actor(s)
- Administrator

Use Case Preconditions
- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions
- Student information is viewed, modified, added, or deleted.

Applicable Business Rules
- N/A

List of Sub Use Cases
- View student
- Modify student
- Add student
- Delete Student

Main Flow: View Student

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view students.	Displays list of registered students. Use case ends.

Alternate Flows: View Student

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Student

Additional Precondition: ' View Student' has been successfully executed.

	Actor(s): Administrator	System
1	Request to view specified student by clicking on the student number.	Displays detailed student personal information about selected student.
2	Request to edit student personal information.	Displays form for student personal information and pre-populated input fields.
3	Edits information and save the changes.	Validate fields, changes are saved in DB. Display student personal information page. Use case ends.

Alternate Flows: Modify Student

	Alternate Flow	Description
A1	Selects to view parent/guardian information and requests to edit.	Displays parent/guardian information page with form pre-populated input fields. Processing resumes at step 3.
A2	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A4	Cancel at any point.	Nothing is updated; Display student personal information page. Use case ends.
A5	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Student

Additional Precondition: ' View Student' has been successfully executed.

	Actor(s): Administrator	System
1	Requests to add student.	Displays input fields for new student.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display student personal information page. Use case ends.

Alternate Flows: Add Student

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.

A3	Cancel at any point.	Nothing is updated; Display student personal information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Student

Additional Precondition: View students have been executed.

	Actor(s): Administrator	System
1	Selects checkboxes for students to delete. Requests to delete.	Request confirmation.
2	Confirms.	Flag selected students as inactive in DB. Display main view student page (list of all students). Use case ends.

Alternate Flows: Delete Student

	Alternate Flow	Description
A1	Selects a student by clicking on the student number. Request to delete student and confirms.	Displays detailed student personal information about selected student. Request confirmation. Flag student as inactive in DB. Display main view student page (list of all students). Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.11 Admin – Manage Belt Information

Belt information can be viewed or modified.

Author: Andriy Granovski

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Belt information is viewed or modified.

Applicable Business Rules

- N/A

List of Sub Use Cases

- View Belt Information
- Modify Belt Information

Main Flow: View Belt / TIP / Promotion Status

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view belt information.	Displays belt information. Use case ends.

Alternate Flows: View Belt / TIP / Promotion Status

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Belt / TIP / Promotion Status

Additional Precondition: View Belt / TIP / Promotion has been executed.

	Actor(s): Administrator	System
1	Selects 'No TIP', 'Half TIP', 'Full TIP', or 'Promote' button.	Promotion information is saved in the database. Use case ends.

Alternate Flows: Modify Belt / TIP / Promotion Status

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.12 Admin – Manage Attendance (Attendance Management)

Attendance entries can be viewed, modified, added, and deleted.

Author: Jodie Carleton

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Attendance information is viewed, modified, added, or deleted.

Applicable Business Rules

- N/A

List of Sub Use Cases

- View attendance
- Modify attendance
- Add attendance
- Delete attendance

Main Flow: View Attendance

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view attendance information.	Displays list of students and attendance information. Use case ends.

Alternate Flows: View Attendance

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Attendance

Additional Precondition: 'View Attendance' has been successfully executed.

	Actor(s): Administrator	System
1	Selects a student by clicking on the student number.	Displays detailed student attendance information about selected student.
2	Request to edit attendance entry by clicking on attendance entry date.	Displays form for attendance information and pre-populated input fields.
3	Edit information and save the changes.	Validates fields, changes are saved in DB. Display attendance detail information for selected student. Use case ends.

Alternate Flows: Modify Attendance

	Alternate Flow	Description
A1	Cancel at any point.	Nothing is updated; Display student attendance information. Use case ends.
A2	Invalid input entered into any of the input fields.	Error message Is displayed, processing resumes at step 3.
A3	Illegal characters entered into any of the input fields.	Error message Is displayed, processing resumes at step 3.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Attendance

Additional Precondition: 'View Attendance' has been successfully executed.

	Actor(s): Administrator	System
1	Selects a student by clicking on the student number.	Displays detailed student attendance information about selected student.
2	Requests to add attendance information.	Displays form for attendance information and empty input fields.
3	Edit information and save the changes	Validates fields, changes are saved in DB. Display attendance detail information for selected student. Use case ends.

Alternate Flows: Add Attendance

	Alternate Flow	Description
A1	Invalid input entered into any of the	Error message is displayed, processing resumes at

	input fields.	step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display student attendance information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.
A5	Attendance information already exists.	Display error message. After 3 seconds, display list of registered students. Use case ends.

Main Flow: Delete Attendance

Additional Precondition: View Attendance has been executed

	Actor(s): Administrator	System
1	Selects a student by clicking on the student number.	Displays detailed student attendance information about selected student.
2	Selects checkboxes for attendance information to delete. Requests to delete attendance information.	Requests confirmation.
3	Confirms.	Flags attendance information as inactive in DB. Displays attendance detail information for selected student. Use case ends.

Alternate Flows: Delete Attendance

	Alternate Flow	Description
A1	No entries are selected.	Error message is displayed, processing resumes at step 1.
A2	Cancel at any point.	Displays list of registered students. Use case ends.
A3	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.13 Admin – Manage Staff Members

Staff Members can be viewed, modified, added, and deleted.

Author: Maggie Ha

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Staff Information is viewed, modified, added, or deleted.

Applicable Business Rules

- N/A

List of Sub Use Cases

- View staff member
- Modify staff member
- Add staff member
- Delete staff Member

Main Flow: View Staff Member

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view staff members.	Displays a list of staff members. Use case ends.

Alternate Flows: View Staff Member

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Staff Member

Additional Precondition: 'View Staff Members' has been successfully executed.

	Actor(s): Administrator	System
1	Requests to view specified staff member by clicking on staff number.	Displays detailed staff member personal information about selected staff member.
2	Request to edit staff member personal information.	Displays form for staff member personal information and pre-populated input fields.
3	Edits information and save the changes.	Validate fields, changes are saved in DB. Display detailed staff member personal information page. Use case ends.

Alternate Flows: Modify Staff Member

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display staff member personal information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Staff Member

Additional Precondition: 'View Staff Members' has been successfully executed.

	Actor(s): Administrator	System
1	Requests to add staff member.	Displays input fields for new staff member.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display staff member personal information page. Use case

		ends.

Alternate Flows: Add Staff Member

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display staff member personal information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Staff Member

Additional Precondition: 'View Staff Members' has been successfully executed.

	Actor(s): Administrator	System
1	Selects checkboxes for staff members to delete. Requests to delete.	Request confirmation.
2	Confirms.	Flag selected staff members as inactive in DB. Displays main view staff member page (list of all staff members). Use case ends.

Alternate Flows: Delete Staff Member

	Alternate Flow	Description
A1	Requests to view specified staff member by clicking on staff number. Request to delete staff member and confirms.	Displays detailed staff member personal information about selected staff member. Request confirmation. Flag staff member as inactive in DB. Displays list of all staff members page. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.14 Admin – View Staff Schedules

Staff member schedules can be viewed.

Author: Muhammad Ahsan

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Staff schedule information is viewed.

Applicable Business Rules
- N/A

Main Flow: View Staff Schedule
Additional Precondition: N/A

	Actor(s): Administrator	System
1	Request to view staff schedules.	Displays a calendar with staff schedule information. Use case ends.

Alternate Flows: View Staff Schedule

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.15 Admin – Manage Payments
Student payment information can be viewed, modified, added, can deleted.

Author: Muhammad Ahsan

Actor(s)
- Administrator

Use Case Preconditions
- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions
- Payment information is viewed, modified, added, or deleted.

Applicable Business Rules
- N/A

List of Sub Use Cases:
- View Payment
- Modify Payment
- Add Payment
- Delete Payment

Main Flow: View Payment
Additional Precondition: N/A

	Actor(s): Administrator	System
1	Request to view payments.	Displays list of all active students and summary of their payment information.

2	Request to view student payment information by clicking on student number.	Display selected student's detailed payment information. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Payment

Additional Precondition: 'View payments' has been successfully executed.

	Actor(s): Administrator	System
1	Requests to edit payment information.	Displays form for payment information and pre-populated input fields.
2	Edits information and save the changes.	Validate fields, changes are saved in DB. Display student's payment information page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display student's payment information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Payment

Additional Precondition: 'View payments' has been successfully executed.

	Actor(s): Administrator	System
1	Request to add payment.	Displays form for a new payment.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display student's payment information page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display student's payment information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Payment

Additional Precondition: 'View payments' has been successfully executed.

	Actor(s): Administrator	System
1	Request to delete payment information.	Request confirmation.
2	Confirms.	Flag selected payment information as inactive in DB. Display student's payment information page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Cancel at any point.	Nothing is updated; Display student's payment information page. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.16 Admin – Manage Shift Reports

Shift report can be viewed, modified, added, and deleted.

Author: Andriy Granovsky

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Shift report is viewed, modified, added, or deleted.

Applicable Business Rules

- An instructor end of shift report is due at the end of every night.

List of Sub Use Cases

- View Shift Report
- Modify Shift Report
- Add Shift Report
- Delete Shift Report

Main Flow: View Shift Report

Additional Precondition: N/A

	Actor(s): Admin	System
1	Requests to view shift reports.	Displays a list of all shift reports. Use case ends.

Alternate Flows: View Shift Report

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Shift Report

Additional Precondition: 'View Shift Reports' have been successfully executed.

	Actor(s): Administrator	System
1	Selects a shift report by clicking on a class date.	Displays detailed shift report information about selected date.
2	Request to edit shift report.	Displays form with pre-populated input fields.
3	Edits information and save the changes.	Validate fields, changes are saved in DB. Display detailed shift report page. Use case ends.

Alternate Flows: Modify Shift Report

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display detailed shift report page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Shift Report

Additional Precondition: N/A

	Actor(s): Admin	System
1	Requests to add shift report.	Display form for new shift report.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display list of shift reports page. Use case ends.

Alternate Flows: Add Shift Report

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display list of shift reports page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Shift Report

Additional Precondition: N/A

	Actor(s): Admin	System

1	Selects checkboxes for shift reports to delete. Requests to delete.	Request confirmation.
2	Confirms.	Flag selected shift reports as inactive in DB. Display list of shift reports page. Use case ends.

Alternate Flows: Delete Shift Report

	Alternate Flow	Description
A1	Selects a shift report by clicking on a class date. Request to delete.	Displays detailed shift report information about selected date. Request confirmation. Flag selected shift reports as inactive in DB. Display list of shift reports page. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.17 Admin – Manage Classes

Classes can be viewed, modified, added, or deleted.

Author: Pedro Bellesa

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Class information is viewed, modified, added, or deleted.

Applicable Business Rules

- N/A

List of Sub Use Cases

- View class
- Modify class

Main Flow: View Class

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view classes.	Displays list of classes.
2	Select class type.	Displays class information. End use case.

Alternate Flows: View Class

	Alternate Flow	Description

A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Class

Additional Precondition: 'View Class' has been successfully executed.

	Actor(s): Administrator	System
1	Requests to edit class information.	Displays form for class information and pre-populated input fields.
2	Edit information and save the changes.	Validates fields, changes are saved in DB. Display class information page. Use case ends.

Alternate Flows: Modify Class

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display class information page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.18 Admin – Manage Belt Promotions

Belt promotion information can be viewed, modified, added, or deleted.

Author: Maggie Ha

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as an administrator.

Use Case Successful Post Conditions

- Belt promotion information is viewed, modified, added, or deleted.

Applicable Business Rules

- In order to be promoted to the next belt, student must receive 3 TIPs.
  The rules for obtaining TIPs are:
  - White - 4 weeks of classes between TIPs
  - Yellow - 5 weeks of classes between TIPs
  - Orange - 6 weeks of classes between TIPs
  - Green - 7 weeks of classes between TIPs

- Purple - 8 weeks of classes between TIPs
- Red - Min. 30 classes & 4 “boot camps” between TIPs

List of Sub Use Cases

- View belt promotion information
- Modify belt promotion information
- Add belt promotion information
- Delete belt promotion information

Main Flow: View Belt Promotion Information

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Requests to view belt promotions.	Displays belt promotion rules page. Use case ends.

Alternate Flows: View Belt Promotion Information

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify Belt Promotion Information

Additional Precondition: ‘View Belt Promotion Information’ has been successfully executed.

	Actor(s): Administrator	System
1	Request to edit belt promotion information.	Displays form for belt promotion rules and pre-populated input fields.
2	Edit belt promotion rules and save the changes.	Validate fields, changes are saved in DB. Display belt promotion rules page. End use case.

Alternate Flows: Modify Belt Promotion Information

	Alternate Flow	Description
A1	Belt already exists (same name).	Error message is displayed, processing resumes at step 2.
A2	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A4	Cancel at any point.	Nothing is updated; Display belt promotion rules page. Use case ends.
A5	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add Belt Promotion Information

Additional Precondition: ‘View Belt Promotion Information’ has been successfully executed.

	Actor(s): Administrator	System
1	Request to add new belt rule.	Displays input fields for new belt rule.
2	Enters new belt rule and saves the changes.	Validate fields and checks to see that the belt does not already exist (it does not). Request confirmation.

3	Confirms.	Belt promotion rule is added to the DB. Display belt promotion rules page. Use case ends.

Alternate Flows: Add Belt Promotion Information

	Alternate Flow	Description
A1	Belt already exists (same name).	Error message is displayed, processing resumes at step 2.
A2	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A4	Cancel at any point.	Nothing is updated; Display belt promotion rules page. Use case ends.
A5	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Belt Promotion Information

Additional Precondition: 'View Belt Promotion Information' has been successfully executed.

	Actor(s): Administrator	System
1	Selects checkboxes for rules to delete. Requests to delete.	Request confirmation.
2	Confirms.	Selected belt promotion rules are deleted from DB. Displays belt promotion rules page. Use case ends.

Alternate Flows: Delete Belt Promotion Information

	Alternate Flow	Description
A1	No entries are selected.	Error message is displayed, processing resumes at step 1.
A2	Cancel at any point.	Nothing is deleted; Display belt promotion rules page. Use case ends.
A3	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.19 Admin – Manage Events

Events can be viewed, modified, added, and deleted.

Author: Pedro Bellesa

Actor(s)

- Administrator

Use Case Preconditions

- User must have internet access.
- Must have working web browser.
- User must be logged in as administrator.

Use Case Successful Post Conditions

- An event is viewed, modified, added, or deleted.

Applicable Business Rules

- N/A

List of Sub Use Cases:

- View Event
- Modify Event
- Add Event
- Delete Event

Main Flow: View Event

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Request to view events.	Display list of all events currently in the system in reverse chronological order and in summary form. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Selects to view calendar.	Display calendar of all events in the system for the month. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.
A3	Selects to view calendar and selects to “see more”.	Display calendar of all events in the system for the month and expand the selected entry. Use case ends.

Main Flow: Modify Event

Additional Precondition: ‘View Event’ has been successfully executed.

	Actor(s): Administrator	System
1	Request to edit event.	Displays form for event and pre-populated input fields.
2	Edit information and save the changes.	Validate fields, changes are saved in DB. Display event page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display event page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login

		screen. Use case ends.

Main Flow: Add Event

Additional Precondition: 'View Event' has been successfully executed.

	Actor(s): Administrator	System
1	Request to create event.	Displays form for creating a new event.
2	Enter information and saves the changes.	Validate fields, changes are saved in DB. Display events page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 2.
A3	Cancel at any point.	Nothing is updated; Display event page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete Event

Additional Precondition: 'View Event' has been successfully executed.

	Actor(s): Administrator	System
1	Request to delete event.	Request confirmation.
2	Confirms.	Delete event in DB. Display event page. Use case ends.

Alternate Flows

	Alternate Flow	Description
A1	Cancel at any point.	Nothing is updated; Display event page. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

6.20 Admin – Manage Communications (News/Announcements)

News/Announcements can be viewed, modified, added, and deleted.

Author: Muhammad Ahsan

Actor(s)

- Admin

Use Case Preconditions

- User must have internet access.
- Must have working web browser.

- User must be logged in as administrator.

Use Case Successful Post Conditions

- News/Announcements are viewed, modified, added, or deleted.

Applicable Business Rules

- N/A

List of Sub Use Cases

- View News/Announcements
- Modify News/Announcements
- Add News/Announcements
- Delete News/Announcements

Main Flow: View News/Announcements

Additional Precondition: N/A

	Actor(s): Administrator	System
1	Request to view communications.	Displays list of all news/announcements. Use case ends.

Alternate Flows: View News/Announcements

	Alternate Flow	Description
A1	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Modify News/Announcements

Additional Precondition: 'View communications' has been successfully executed.

	Actor(s): Administrator	System
1	Select 'see more' to view detailed news/announcements information.	Display detailed news/announcement entry of selected entry.
2	Request to edit entry information.	Displays form for news/announcement information and pre-populated input fields.
3	Edits information and saves the changes.	Validates fields, changes are saved in DB. Display detailed news/announcement page. Use case ends.

Alternate Flows: Modify News/Announcements

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 3.
A3	Cancel at any point.	Nothing is updated; Display news/announcements detailed page. Use case ends.
A4	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Add News/Announcements

Additional Precondition: 'View Communications' has been successfully executed.

	Actor(s): Administrator	System
1	Enter information and select to post.	Validate fields, changes are saved in DB. Display communications page. Use case ends.

Alternate Flows: Add News/Announcements

	Alternate Flow	Description
A1	Invalid input entered into any of the input fields.	Error message is displayed, processing resumes at step 1.
A2	Illegal characters entered into any of the input fields.	Error message is displayed, processing resumes at step 1.
A3	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

Main Flow: Delete News/Announcements

Additional Precondition: 'View communications' has been successfully executed.

	Actor(s): Administrator	System
1	Select 'see more' to view detailed news/announcements information.	Display detailed news/announcement entry of selected entry.
2	Request to delete entry.	Request confirmation.
3	Confirms.	Flag news/announcement entry as deleted. Displays communications page. Use case ends.

Alternate Flows: Delete News/Announcements

	Alternate Flow	Description
A1	Cancel at any point.	Nothing is updated; Display news/announcements detailed page. Use case ends.
A2	Requests to log out.	The user is logged out of the system. Display login screen. Use case ends.

7 Non-Functional requirements

	Requirement Type	Description and Implementation Plan
1.	Look and Feel	- Aesthetically pleasing - Easy to read text: font colour and font type will be appropriate - Responsive design to scale to different devices **Implementation Plan:** - Do not make it too cluttered; space things out - Use appropriate colour palette and font types - Use HTML5 and CSS3 to make it responsive
2.	Usability and Humanity	- Intuitive design to allow users to easily find what they are looking for and transverse to different regions - all error messages must be informative **Implementation Plan:** - Categorize the different site locations into categories and subcategories for a well-organized and easy to use navigation bar
3.	Performance	- Handle high workloads (ex. Number of users logged in, many DB transactions) - 0.1-1 second response time for processing student swipe-in - System must provide high availability **Implementation Plan:** - Handling high workloads will be done by choosing the right database - Quick swipe time is handled by the hardware - High availability will be provided by choosing the right hosting company
4.	Operational	- Systems must stay up regardless of power outage - Systems must allow for multiple users to be accessing the software at the same time **Implementation Plan:** - Make sure hosting company has UPS and auto recovery after power failure
5.	Maintainability and Support	- A specific part of the system (TIP approvals) shall be readily portable to Android and/or iOS **Implementation Plan:** - When designing system, keep in mind of portability of the TIP approvals part of the system; must design a system that can be portable to Android and/or iOS for TIP approvals

6.	Security	**Access:** - Only registered users can use this system - Each account type has certain accessibility and functions they are allowed to use - Only administrator roles can use the delete functions/operations **Privacy:** - All customer information must be protected. - All collected information must be protected. - Customer information is not the be released to others (third party) - Must comply to PIPEDA standards **Integrity:** - Financial transactions are accurate and secure **Audit:** - All sales information and transactions are stored **Implementation Plan:** - Create a secure and trusted account management and log in system with a lot of testing to ensure accessibility and functions are proper - Use a trusted and secure payment processing system (Moneris) - Ensuring all sales information and transactions are stored and secure by choosing the right database
7.	Legal	- System shall comply with Canada tax and business regulations **Implementation Plan:** - Ensure all tax and business regulations are applied in the system