

AIO-utomate

Final Documentation Report

From 1st Semester Work
Jan. 2018 – Apr. 2018

By:

Grant Warr (Leader)

Heather Brunell

Sitanun (Dream) Changhor

Chris Connors

Scott Martell

Joshua Morash (Coordinator)

Alex Clarke

Yide Ge

Sam Irvine

Nathan Robertson

Submitted Apr. 8th 2018

Table of Contents

List of Deliverables	2
Project Background Summary	3
Project Description	4
Comparison of Planned Versus Actual	5
Original Scope and History of Approved Changes	6
Burn-Up Chart	7
Outstanding Issues	8
Lessons Learned	8
Ongoing Support Required and Duration	10
System Documentation	10
User Manual	11
List of Training Material	16
Project Sign-off	17

List of Deliverables

The AIO-automate project concludes with several key deliverables. These deliverables are to be presented to the client and passed on to the next team that undertakes work on the AIO project. An itemized list of deliverables is as follows:

1. The Code

Our first key deliverable is the project code itself. All the code that was completed by the AIO team this term (both front-end and back-end) is stored in the GitLab repository. This repository contains separate directories for HTML, CSS, JS and PHP functions. These directories contain the code that achieve our outlined functionality (See Comparison of Planned Versus Actual). There also exists separate directories for LaTeX form publishing, evidence submission, and code tests. This code repository is a representation of what was achieved by the AIO team this semester.

2. The Database

The Database was designed by the back-end team. It is hosted on db.cs.dal.ca and can be accessed through a phpMyAdmin interface. It contains several tables including: Professor, AIO, Admin, Student, Cases, and Case History. As of the conclusion of this project semester, the only table that is not being used is the History table. The database will be used to authenticate user roles, define permissions, and update any case related information display.

3. The Server

The server is used to host the live version of the AIO project. It is hosted on projects.cs.dal.ca. At the end of this project, the server will host the master version of the AIO code. The server is an Apache 2.2.3 HTTP server running on CentOS 5.11 (Final) and is maintained by the CS help desk.

4. The Presentation

The final presentation is the first deliverable of our closing phase. The purpose of this was to provide the client with an overview of the AIO project, what the team has achieved, and what future work will need to be completed. The presentation includes a comparison of the planned scope versus the actual achieved scope. A burn-up chart detailing actual hours work versus planned. The presentation also covers topics such as project changes, outstanding issues, lessons learned, and test scripting. The presentation concludes with a live-demo in which the team demonstrates for the client the current functionality of the AIO project.

5. The User Manual (Included in the Report)

The user manuals will provide the client, and future teams, with the know-how to navigate the current state of the AIO project. The manual will contain, in detail, documentation of the code and provide insight on specific functions. The manual will also include an overview of the knowledge required in future AIO development.

6. The Report

The final key deliverable is this project report. The report marks the conclusion of work on the AIO project for this term. The purpose of the report, like the final presentation, is to provide an overview and summary of what was completed on the project. The report will contain descriptions of the project, an evaluation of scope, issues and lessons learned, as well as any supplementary information that the client or future teams may require (i.e. System Documents, Manuals, training material, etc).

Project Background Summary

An ongoing problem at our university is that students are frequently plagiarizing their academic papers and projects. When this happens, a new case is opened. This case needs to be resolved professionally and quickly for all students involved in the committed offence. The process involves a Professor submitting an accusation that then get assigned to an Academic Integrity Officer (AIO). The AIO then confers with the Senate to determine if they can move forward internally with the case. If this is so, they schedule a meeting with the student and then render a verdict that the student can then either accept or decline, the latter will result in being retried by the Senate. This is currently all done via email using an Administrator as an intermediary who also assigns AIOs to cases.

This system works; however, it is extremely inefficient. This system is prone to documents being lost in mailboxes or even potentially deleted and makes it difficult for AIOs to stay on top of their cases. It is a process that is outdated in our modern world of web services and is due for an upgrade with the technology of today.

The solution to address this problem is to create a brand new web portal service that can automate the series of form exchanges to alleviate the stress on the AIOs and to streamline the process to help complete cases faster and more efficiently.

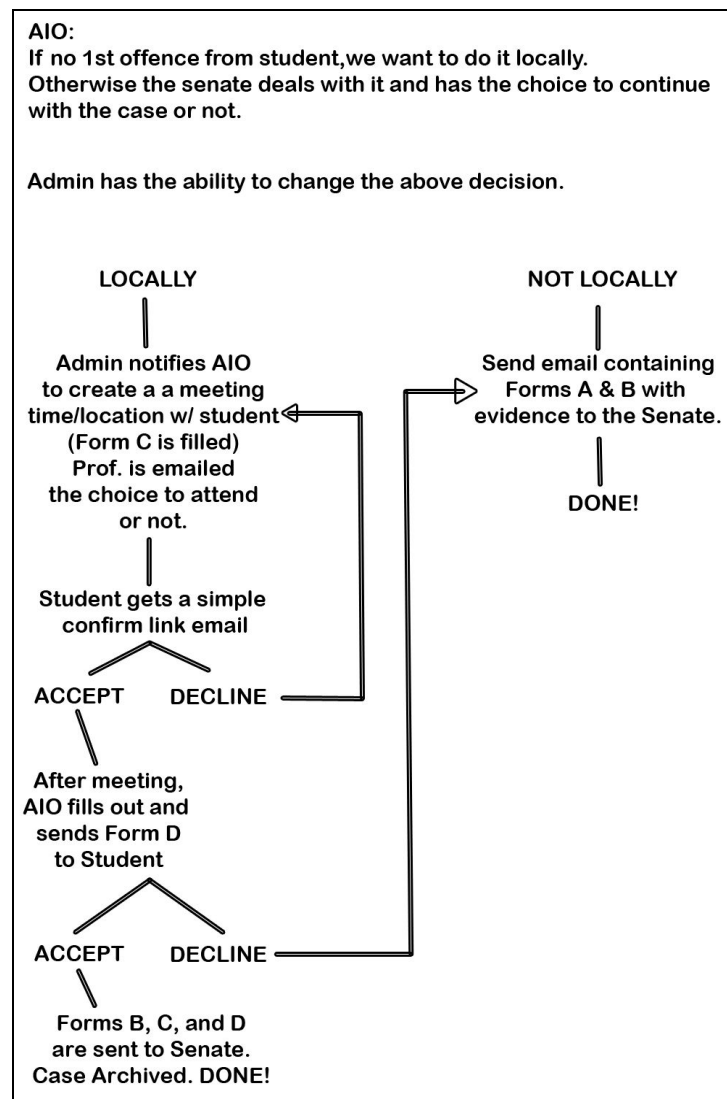


Figure. 1 - Lo-fidelity tree diagram composed during the planning phase of AIO-utomate. This diagram was made to help the entire team understand the flow of a student academic integrity offence case.

Project Description

AIO-utomate provides a web portal for professors, AIOs, and Administrators to monitor and pursue cases of academic integrity violations. The portal allows professors to submit an accusation against a student through a provided Form A interface, where all details of the account can be recorded.

Form A is then submitted to be reviewed by an Academic Integrity Officer who can view the case in their active cases screen, which can be seen upon logging in. The AIOs can then move the case forward by filling out forms B, C, or D through the portal itself. If necessary they can also let the administrator know if there is not enough evidence present.

The Administrator has the ability to view all active cases and assigned AIO's on their home screen in the active cases table. From this screen, they have the ability to view cases, delete cases, assign AIO's, and reassign AIO's on a case by case basis.

Comparison of Planned Versus Actual

The table below shows what the group had planned and what we were able to successfully complete over the semester.

Completed

Feature was implemented into the web portal successfully.

Partially Completed

Certain parts of the feature doesn't work/requires additional time to polish.

Outstanding

Feature was not started.

Planned	Actual
Develop first version of the web portal's user interface	Completed
AIO ability to forward evidence and forms to the Senate to resolve cases	Partially Completed
Auto-filling text content into forms upon typing them in	Partially Completed
Accounts for AIOs, Professors, and Admins respectively	Completed
Content viewable on web portal without the need to download them	Completed
Efficiently login and logout of accounts	Completed
Quick access back to assigned/pending cases to resume progress	Partially Completed
Ability to save case progress	Partially Completed
Automatic form signature created by CS account properties	Outstanding
Professor can submit a new case concerning a student's academic integrity	Partially Completed
Ability to upload evidence (images, PDFs) into the case	Completed

Original Scope and History of Approved Changes

This section will outline the original scope of the project decided on at the start of term, and agreed upon by the client. Throughout any agile project though, there are guaranteed to be changes that need to be made. This section will discuss the changes made during this project that were approved by the client, and why they were made.

The original scope of the project is outlined in the section above “Comparison of Planned vs Actual.” Some of these sections however, did go through some changes throughout the development of this project. Changes were made to “Efficiently login and logout of accounts,” “Develop first version of the web portal’s user interface,” and others that do not fit into a specific category.

At the start of the project, we planned to have a tabbed login screen where the user would select which type of user they were (AIO, Professor, or Administrator) and would then be brought to the appropriate login screen. Early on, this was changed to a single login screen with a drop-down for the user to select their role. This was done because it is possible for a user to have multiple roles; a professor who is also an AIO, for example, and they needed to be able to login as either one at any time.

There were also a number changes made to the forms, UI, and functionality of the portal. The largest was probably adding forms to the case information page. Originally each form was going to have its own page, but it was decided to have a tabbed section in the case information page with all of the forms embedded. This was done in order to simplify the layout of the site, and decrease the number of pages that a user would have to navigate. Furthermore, because each case has one Form A, but each student has Forms B, C, and D, in cases where there are multiple students, a student information page was made with all forms embedded, and the case information page was changed to only include Form A. Lastly, there were a number of layout changes to all of the forms, but most notably, there was a total shift in the format of Form B from two columns of inputs to one, and the addition of checkbox activated accordion sections for optional parts of the form. Both of these changes were made to both make the form less cluttered and easier to understand from the users perspective.

Outside of the frontend changes mentioned above, there was also a notable change made on the backend. This was to the History table, which tracks data about all of the completed cases. The client had requested that more data metrics be added to the table to increase the amount of information available about closed cases.

Another major change to the backend was to the database structure itself. Originally, we attempted to consider the situation in which a student may be involved in two different cases. As such, we had assigned each individual student a case in the database. This was very easy to maintain from a database perspective because we could simply delete the entire row from the Case table as soon as the student's case was closed. However, we realized that this would generate too much unnecessary data. For example, excess and duplicated data would be generated in a case that involved two or more students. This information would be: professor

information, class details, student names, student ids, etc. Now all students are simply linked to their cases and their personal information is stored in the Student table.

Burn-Up Chart

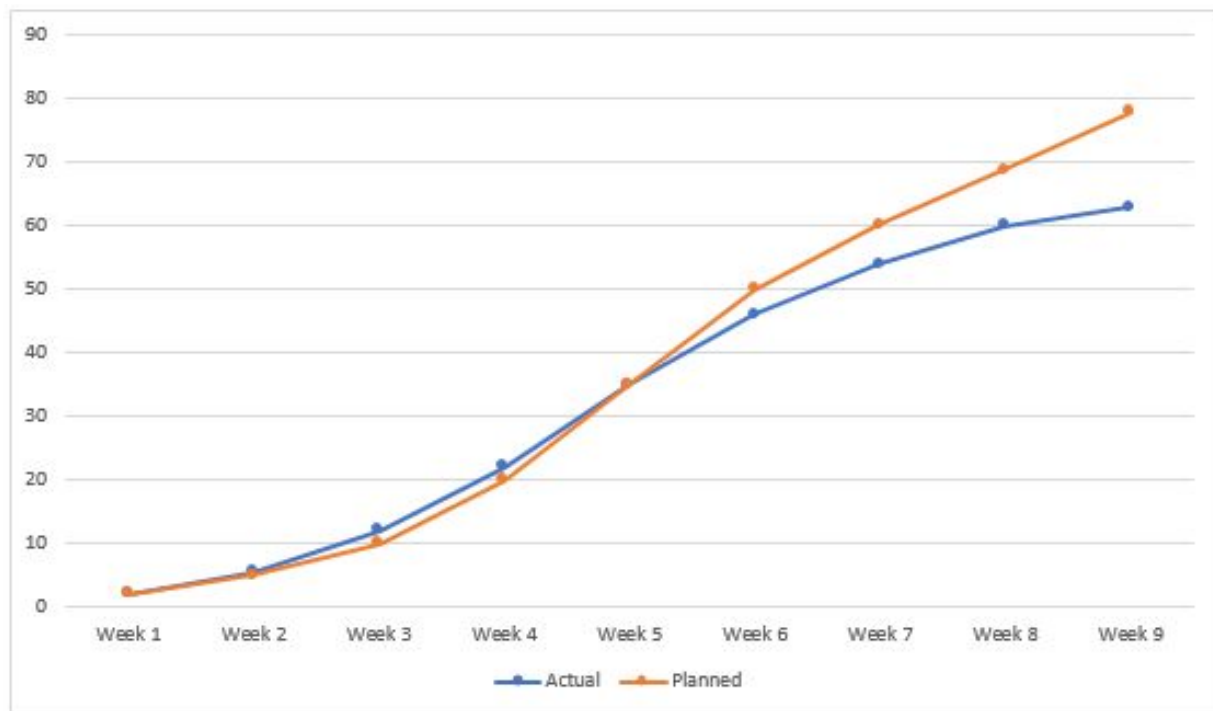


Figure. 2 - Line graph displaying the nine work week progress of the work hours that were planned and the work hours that were actually executed.

While composing the project plan, we came up with user stories, estimated and assigned the number of work hours it would take to complete each individual story. The result came out to be a total of 78 dedicated, successful work hours over the course of the four month 2018 winter semester period. However, as shown by the comparison of the actual progress line and the planned progress line, the end result that happened for us by the end of the semester was around 62 work hours of successful content created. For the first half of the nine work weeks, we managed to be a little ahead of schedule due to using the prototype program, MockFlow, to create hi-fidelity designs that were also accurate to what they would look like with the markup languages. As a result, everyone was able to quickly understand and make choices regarding the aesthetics and user interface elements of the webpages. The turning point came at week 5 where difficulties in database implementation, page loads, lessons in learning new languages, additional user interface edits across all pages, and an entire database restructuring to incorporate more than one student to a case. Some planned goals, such as proper automation of forms and computer science identification authentication, had to be sacrificed because of the longer than planned time it took to overcome the aforementioned

obstacles. As shown above in Figure. 2, the once rising actual curve dips down after week 5, showing the loss of 16 successful work hours.

Outstanding Issues

Login

Login works apart from authenticating the password using the faculty's LDAP system. This became an issue because we thought we would be able to use the Central Authentication System, which is being developed by Vlado Keselj. Because the user would be required to visit a separate webpage to login after the initial login on the main website, we didn't think that this was the right time to implement this system.

All active case pages

The "Action required" field and the "Change AIO" button in the tables are not functional.

Forms B-D

Need form processing functionality (saving/submitting forms) and form auto-filling.

Form A

Need further tests to confirm that the current form processing logic does what is expected. Submitting a new form should be fully functional, but the logic for saving and resuming a form is yet to be tested further.

All forms

The LaTeX form creation files need to grab the pertinent data from the database when certain forms are submitted. The LaTeX form creation files need to also be able to grab the data in an active form when clicking the "preview form" button on a form. There also needs to be a future discussion about removing some fields that are in the form or display them as being uneditable (e.g. the professor name field in Form A could be removed as there is no field in the active_case table for it).

Lessons Learned

From the beginning of tackling the first version of AIO-utomate during the first semester of work development, the 10-man team had high hopes in believing and delivering AIO-utomate as a fully working web service for users to login and create/complete cases. However, by around work week 5/9, it became clear that a fully functional web portal service was not going to arrive at that time. Another Community Outreach semester will be needed and therefore, a new team will need to be assembled and pick up where our team's work left off.

The result of this came from overestimating the user stories and end goals at the beginning of the semester. This overestimation came from having a large 10-man team who believed that more workers equals more work successfully completed. Ironically, this was not

really true because despite having more roles and tasks assigned to people, the time it takes to finish the tasks can take longer than with less people due to communication and conflicting work schedules. In addition to the difficulties in quick communicating and various work schedules per team member, many team members had new experiences to learn while working on the project itself, effectively delaying the delivery of code and/or editing of code. However, this is also a very good realization because it prepares everybody for their next team-based project they will be working and delivering for another client. Examples of new concepts some team members learned included: PHP coding, Web-markup designing and coding, Latex, etc.

Despite having overestimated the end goals, at least it is better than underestimating the end goals because the more goals that were left for us to complete, the more incentives there were for us to continue pursuing the project with more completed work. If we underestimated, a limited scope of end goals would have only been completed within the first four months.

While planning, the other big takeaway lesson the entire team learned was knowing how much time it takes for prototyping the first visual designs for all of the webpages and developing how all of the webpages will interact and flow between one another (i.e. When a user clicks on “View Case” button, it will take them to that case’s summary page). For the first three/four weeks of the execution phase, 95% of the work hours were spent on creating user interface prototype visuals, deciding on the shapes and colours of the user interface elements, deciding on how the data of each page will be displayed as (i.e. a dropdown, a navigation menu, a table, a list, etc.), and learning how the pages will interact between one another, and lastly, coding the HTML, CSS, and JavaScript for the live website. Furthermore, because the project concept was brand new and we do not know what happens between AIOs and Professors when they are dealing with a offence case, we had to direct a lot of questions to our client to learn more about the process that goes on. Once we had more of this knowledge, we were then able to progress with our designs and edit them to address the issues of the current process accordingly. Now that the end of the first work semester has arrived, we will all leave this project knowing that the next group business project we will be commissioned to do (individually) will involve a lot of work time dedicated towards prototype designing and learning more about the project’s concept if the project is being start-from-scratch.

Finally, when we considered what the work hours should involve, we did not include the work it would take for the team to learn new languages and how to use them to their maximum potential. When working on a business related group project, it is important to know that learning about new languages and new ideas is all a part of the work experience and therefore, they are part of the actual project work. There is an initial fear during planning when coming up with a number of work hours and a number of goals that can be completed within an allotted amount of time, but in reality, things are bound to go wrong and adding more hours and goals relating to individual team member growth is never a bad thing at all. To conclude, there is no need to be afraid to add a lot more hours/goals to the plan because they do not reflect signs of damaged pride amongst the team members.

Ongoing Support Required and Duration

The ongoing support required for our project is minimal. The work done thus far relies solely on the existence of the database and server. The database must be maintained since it holds all case information that is used by the web portal and sent to the users. The web portal would be unusable if the server was not maintained so its continued existence is necessary as well. As long as these two components are kept running, the other components of the project should continue to run as intended.

Action has been taken to ensure that the database and server are maintained for the time being. An email was sent to the help desk, the source of both the database and server, explaining that they would need to be maintained. The duration they need to be maintained is as long as this system is planned to be used. Both the server and database can be terminated if the use of AIO-utomate is ever to be ceased.

System Documentation

Website

The live website can be viewed at **projects.cs.dal.ca/aio**. In order to access the directory for the site you must SSH into either nano.cs.dal.ca or projects.cs.dal.ca with the **username: aio** and the **password: ge7ochooCae7**. When you are logged in, you will notice a folder called **html**. This is a soft link to the real home directory of this website, which is located at **/var/www/html/aio**. Once you go into this folder you will see all of the projects folders and files.

When adding a new php file to this directory or any of its subdirectories you should make sure that the group write permission is removed. You can remove it using: **chmod g-w filename.php**. You can also use the command **chmod -R g-w .** while in the html directory to remove the group write permission on all files in the website. The “.” above is not a period but specifies to start in the current directory. This will be necessary anytime you pull from a remote git repository as all files and directories will have the group write permission. If you try to visit a php page in your browser with the permission on, you will get an error.

Database

The database is hosted on **db.cs.dal.ca**. You can access it at <https://myadmin.cs.dal.ca/> with the **username: aio** and the **password: ge7ochooCae7**. You can also connect to it through the MySQL Workbench using the information provided in Figure 3. You can only connect to this database from the live website. You can't access it through a local server, so it will be necessary to export the live database and import it locally if you wish to do local development.

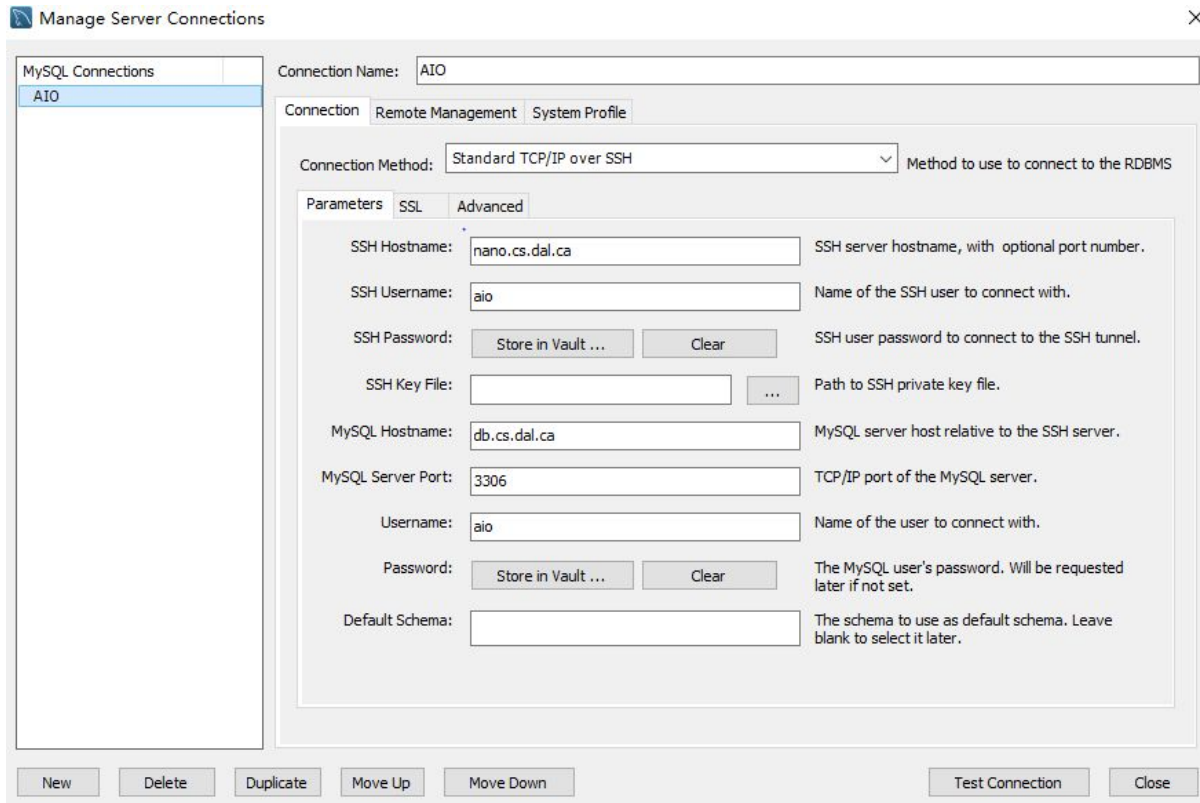


Figure. 3 - Connecting to the aio production database in MySQL Workbench

User Manual

Login Page

This is the login page. Even if you have multiple roles, you can currently only log in as one at a time, so make sure you select the right one before logging in.

Login

Enter CS ID and password

Login as:

Professor ▼

Submit

Please Note:
When logging in, make sure you switch to Professor, Academic Integrity Officer, or Administrator. Even if you have more than one role, you can only log in as one at a time!

Img. 1 - Login Page

After logging in, each user is taken to a page that shows ongoing cases. Professors can view only the cases that they have submitted. Academic Integrity Officers can view cases assigned to them, and Administrators can view all the cases.

Admin page

The administrator can view all the cases in the system, open new cases (the button is missing in the image), assign or reassign AIO's, and delete cases.

There should be a "New Case" Button, but it's missing in the live version

Logout

Active Cases

Case ID	Course	Professor Name	AIO Name	Action Required	Action Needed By	Action
5	CSCI 2100	professor A		No	N/A	Actions ▼
6	CSCI 2100	professor A		No	N/A	Actions ▼
2	CSCI0000	professor B	AIO A	No	N/A	Actions ▼
3	CSCI0000	professor D	AIO C	No	N/A	View Change AIO Delete
4	CSCI0000	professor D	AIO D	No	N/A	

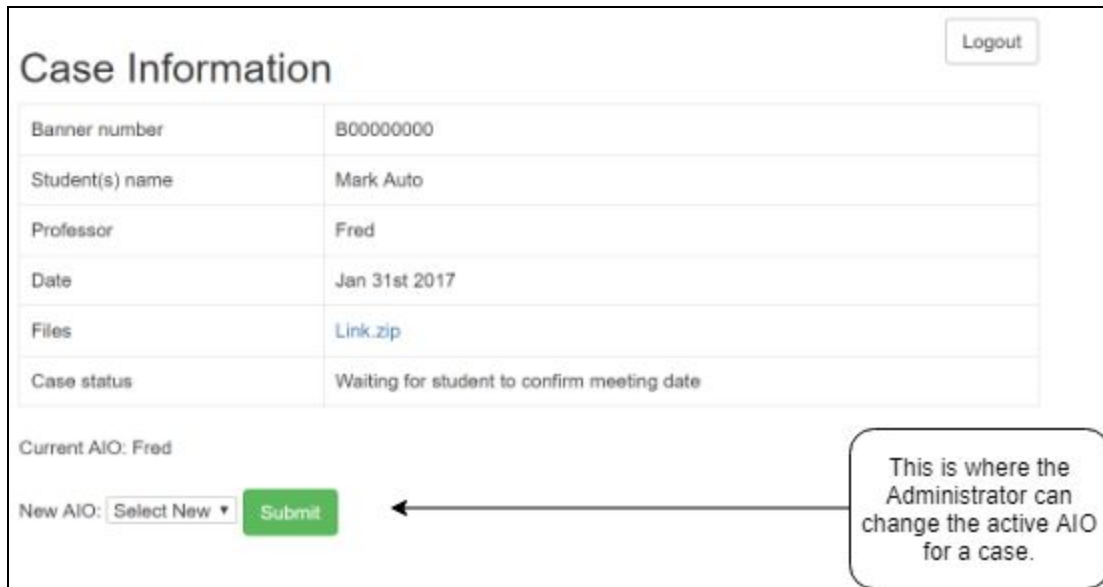
"View" leads to the case information page

Deleting a case will prompt a confirmation to remove the case

Img. 2 - Administrator Active Cases Page

Change AIO Page

The Administrator will see this page when they click “Change AIO” in a case drop-down menu from their main screen.



Case Information Logout

Banner number	B00000000
Student(s) name	Mark Auto
Professor	Fred
Date	Jan 31st 2017
Files	Link.zip
Case status	Waiting for student to confirm meeting date

Current AIO: Fred

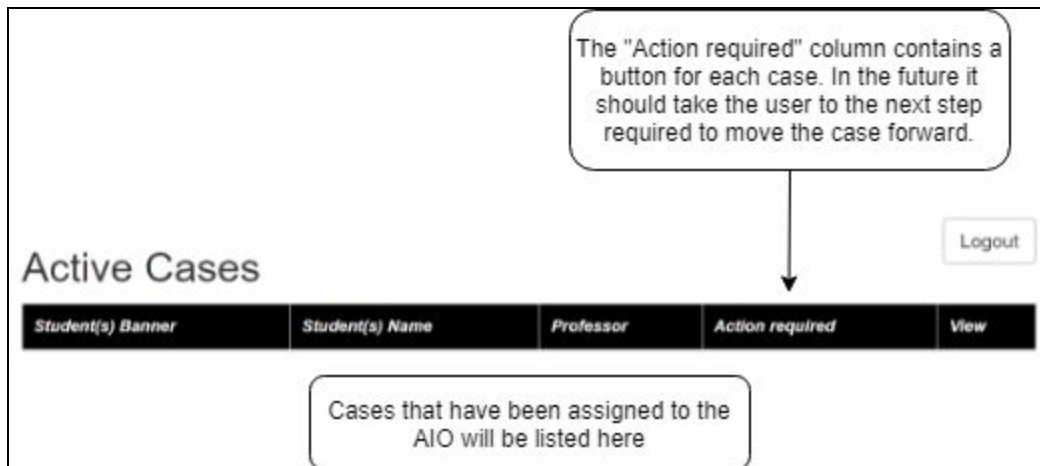
New AIO: Select New ▼ Submit

This is where the Administrator can change the active AIO for a case.

Img. 3 - Change AIO page

AIO page

The Academic Integrity Officers can view all cases assigned to them, and can navigate to the case information pages associated with each case to fill out further forms (B,C,D) to move the cases to the next stages.



Active Cases Logout

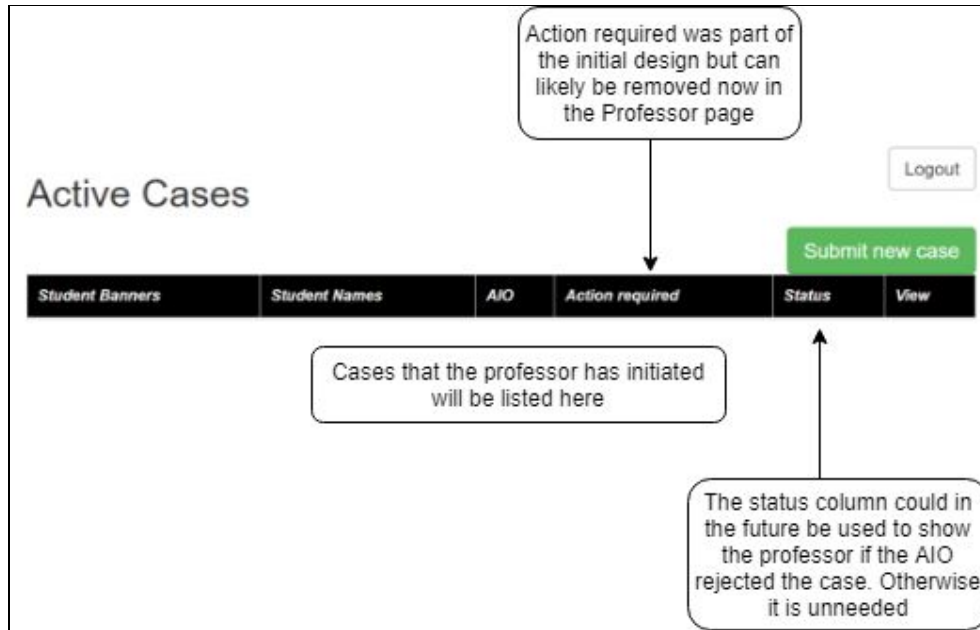
Student(s) Banner	Student(s) Name	Professor	Action required	View
Cases that have been assigned to the AIO will be listed here				

The "Action required" column contains a button for each case. In the future it should take the user to the next step required to move the case forward.

Img. 4 - AIO active cases

Professor Active Cases

The main role of this page is to enable professors to submit a new academic integrity allegation. The “Submit New Case” button will take the professor to a page with Form A, which will initiate a new case.



Img. 5 - Professor Active Cases Page

Case information page

This is the Case Information page. The top table shows an overview of identifying information about the case, and the bottom contains the original Form A data that started the case. All students who are part of the same case share the same Form A. To view or fill out the other forms in a case, you will need to view the individual case information page for a specific student in the case. You can switch between students within a case by using the drop-drop menu, which will act as a link to move you between cases.

The screenshot shows the 'Case Information' page. At the top right is a 'Logout' button. The main title is 'Case Information'. Below it is a table with the following data:

Case Number	000000
Student(s) name	<input type="text" value="Students ▼"/>
Professor	Fred
Date	Jan 31st 2017
Files	Link.zip
Case status	Ongoing

Two callout boxes are present:

- The first callout points to the 'Students ▼' dropdown menu and contains the text: 'To see the individual students within a case, and forms B-D, click this drop-down menu'.
- The second callout points to the 'Link.zip' link and contains the text: 'This contains the current evidence and form A to download and view outside the browser'.

Below the table is a section titled 'Form A' with the subtitle 'Report of Academic Integrity Violation'. It contains four input fields:

- Professor:**
- Email:**
- Phone Number:**
- Faculty:**

Img. 6 - Case Information Page

Student Information page

This page looks very similar to the case information page, but there is information specific to one student in the table now, and there are tabs to switch between each form pertinent to the student.

The screenshot displays a web interface for a student case. At the top right is a "Logout" button. The main section is titled "Case Information -" and contains a table with the following data:

Banner number	B00000000
Student name	Mark Auto
Other Students	<input type="text" value="Other Students ▼"/>
Professor	Fred
Date	Jan 31st 2017
Files	Link.zip
Case status	Waiting for student to confirm meeting date

Below the table is a row of four tabs: "Form A", "Form B", "Form C", and "Form D". "Form B" is currently selected and highlighted in blue. Two callout boxes provide instructions: one points to the "Other Students" dropdown menu with the text "Continue to use the drop-down menu to switch to other students within the same case", and another points to the tabs with the text "Click to switch between forms".

Below the tabs, the section is titled "Form A" with the subtitle "Report of Academic Integrity Violation". It contains two input fields: "Professor:" with a text box containing "Name", and "Email:" with a text box containing "Email".

Img. 7 - Student Case Page

List of Training Material

- Apache
- Bootstrap
- LaTeX
- MySQL
- PHP

Project Sign-off

By signing below, the team leader, team, and client have all gone over and agreed with all of the finished work the AIO-utomate team was able to execute and release by the end of the winter semester of January-April, 2018. All parties have a full understanding of what currently works and what currently does not work. The client also fully understands what they need to do to recruit a new team for a new semester of Community Outreach, and what they need to do to get set team ready and prepared to resume the development of AIO-utomate.

Team Leader: _____

Client: _____