

JMG Student Site Check-in Application

Administrator Manual

Client



JMG

Lanet Anthony, Samantha Brink

Developer



Cyber Cookie

Elijah Caret, Michael Ferris,
Xingzhou Luo, Spencer Morse

University of Maine

April 6, 2022

Version 1.0



JMG Student Site Check-in Application Administrator Manual

Table of Contents

1. Introduction	3
1.1 Purpose of the Document	3
1.2 References	3
2. System Overview	4
2.1 Background	4
2.2 Hardware and Software Requirements	4
3. Inspected Modules	5
3.1 Installation	5
3.2 An Important Distinction	5
3.3 Routine Tasks	6
3.4 Periodic Administration	14
3.5 User Support	16
4. Troubleshooting	17
4.1 Dealing with Error Messages and Failures	17
4.2 Known Bugs and Limitations	17
Appendix A: Agreement between Customer and Contractor	18
Appendix B: Team Review Sign Off	19
Appendix C: Document Contributions	19

1. Introduction

The JMG Student Site Check-In Application is a web service that aims to automate the process of JMG students notifying teachers of their attendance at events outside of school counted towards course credit. This is a capstone project for the Cyber Cookie team, which includes Elijah Caret, Michael Ferris, Xingzhou Luo, and Spencer Morse, in partial fulfillment of the Computer Science BS degree for the University of Maine. The deliverable of this project practices the development cycle of industrial standards. During the previous semester, the Cyber Cookie team had established system requirements, architecture definitions, and user interface designs. The goal for the team this semester is to implement and integrate these modules into the desired functional system with supportive documentation.

1.1 Purpose of the Document

This document is the administrator manual for the JMG Student Site Check-In Application designed by Cyber Cookie. It lays out the responsibilities of the system administrator. These responsibilities include any routine maintenance required, which may occur daily. There are also descriptions of any periodic maintenance required, which may occur every few months. Also included in this document are troubleshooting tips as well as the current status of user support. Any software or hardware requirements for setup are included as well, although not much is required for this system.

1.2 References

See *System Requirements Specification* for information regarding system requirements.

See *System Design Document* for information regarding system design.

See *User Interface Design Document* for information regarding user interface design.

2. System Overview

2.1 Background

As the administrator of the JMG Student-Site Check-in Application, most maintenance and work will be done through the interface provided by the platform upon which the app is built: Budibase. As of right now, a good amount of manual interaction is required to keep the system running for students as there are functionalities that have not been implemented yet due to the system being in its beta stages.

There are various tasks that will need to be routinely performed. All data needs to be frequently saved in another location other than the Budibase database, either locally on the administrator's machine or uploaded to a cloud file hosting service. All past data must be saved in order for JMG to either help students or JMG to use it for future endeavors. Data clean-ups must also be semi-regular, as keeping the amount of memory used within the server small will help increase system performance. All new accounts will need to be set up manually, as there isn't any functionality for automated account creation. Any missed check-ins by students, when notified by the JMG representative at the user's school, will need to be manually entered into the system by the administrator, as a core functionality of the system is that the student can't check-in once the session has finished.

2.2 Hardware and Software Requirements

All that is required for the administrator to effectively maintain and run the system is to have a device that can connect to the internet (phone, computer, or tablet), a web browser, and a link to access the actual application. No other software or hardware is required.

3. Administrative Procedures

3.1 Installation

Installation of the application itself is quite simple. Just visit the eventual URL to the application. No other installation is required, aside from the actual hosting of the application.

3.2 An Important Distinction

There are two interfaces that will be referred to throughout the rest of this document. The Budibase development portal allows administrators and developers to edit the user base, create new applications as well as set up additional login functionality such as single sign on.

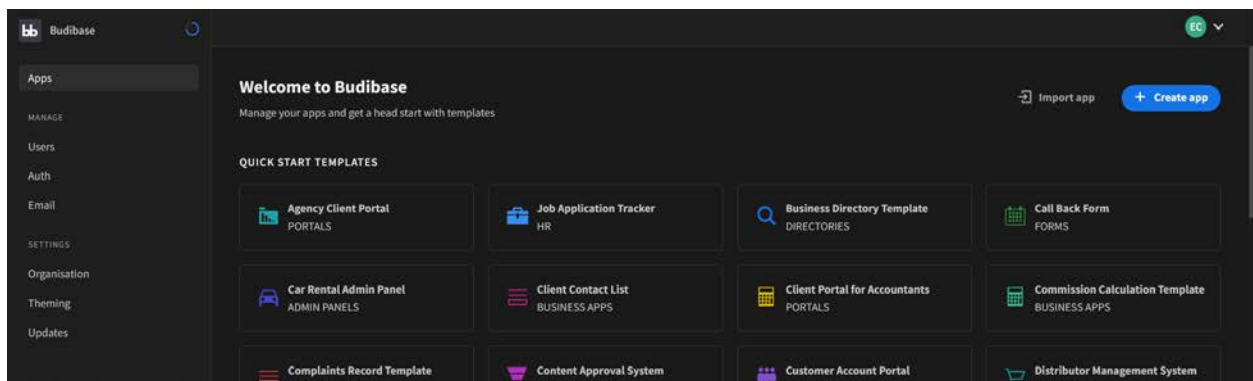


Figure 3-1: The Budibase Development Portal - This is where new applications can be created if needed, as well any additional development.

The other interface is within the application itself; this can be accessed by scrolling down to the bottom of the development portal. This is where the JMG Student Site Check-in Application and other applications (Backup copies, applications for testing, etc.).

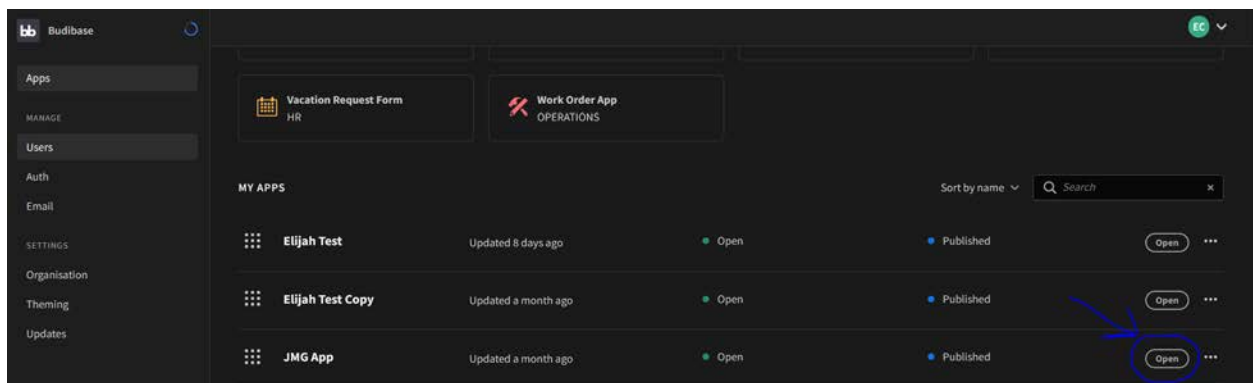


Figure 3-2: How to Access and Application - All apps created under the portal will be found at the bottom of the development portal page.

The application builder interface is used to “code” the application. Most of the administrator maintenance is done in the data tab labeled in figure 3-3. This is where all data and tables can be accessed.

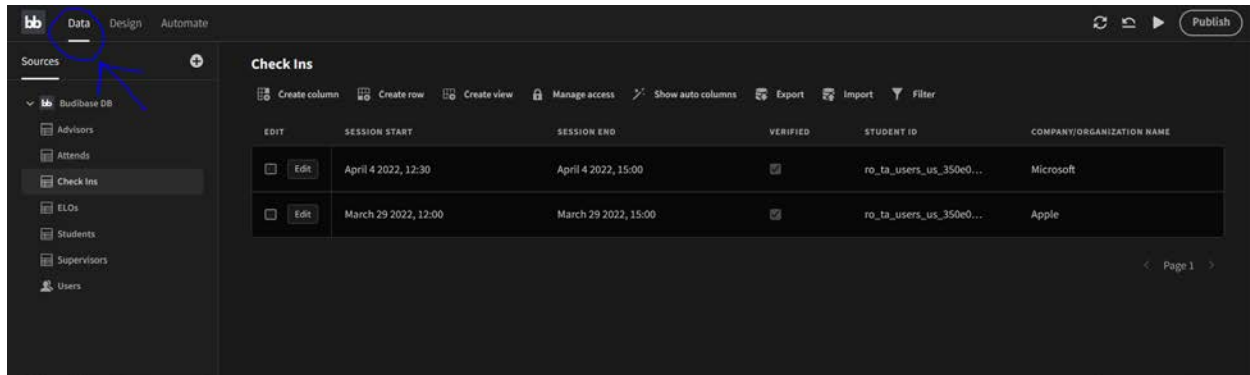


Figure 3-3: Application Interface - Most app maintenance and routine responsibilities center around the data and tables found under the data tab.

3.3 Routine Tasks

Adding Users:

To add a user to the application, the administrator must first sign into their admin privileged account. Successful login brings administrators to the Budibase development portal. From there, click the user’s button on the left side navigation bar.

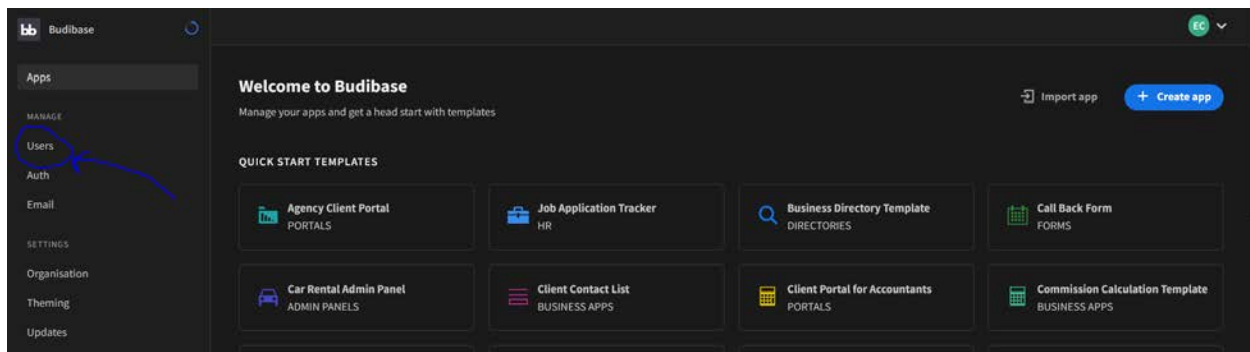
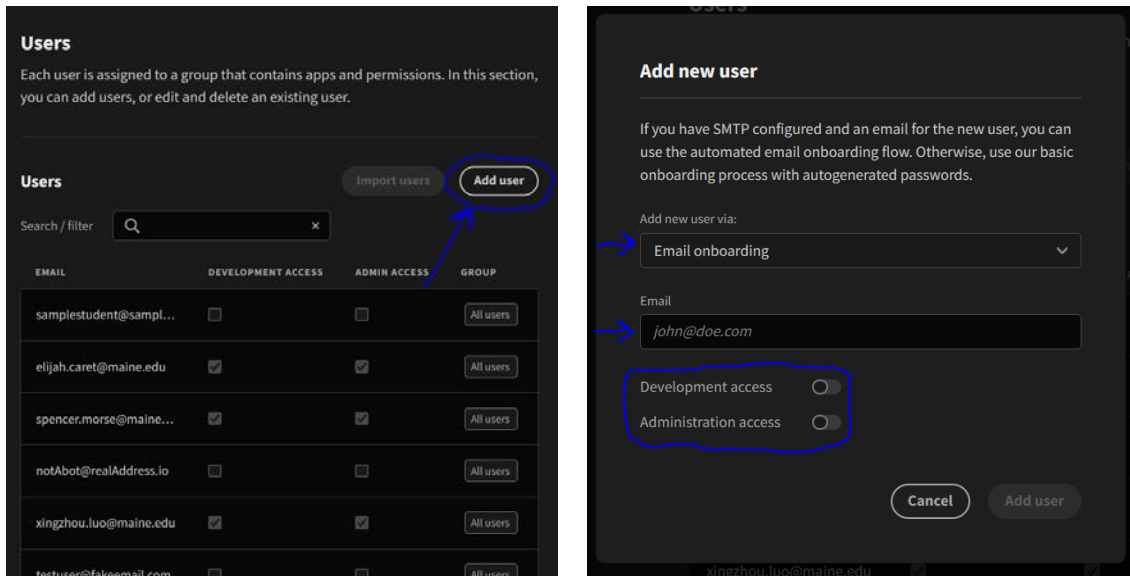


Figure 3-4: Users Tab Location - This is used to edit the user base

Click on the “Add User” button. This will bring up the window to add a new user. There are two options to add users: email onboarding or basic onboarding. Currently email onboarding is not in use so all users will be added via basic onboarding. There are two optional access permissions when creating a new user, development access and

administrator access. These should only be selected if a new administrator is being added to the system.



Figures 3-5 and 3-6: Adding a User - Currently email onboarding is not functional on our application, which means the user will need to be added via basic onboarding.

The administrator will enter the email that the user will use (most likely their school email or business email). A default password will be generated for the user which they will need for their first login attempt. Once a user successfully logs in for the first time, they will be prompted to change their password to something of their choosing. Clicking the “continue” button finalizes the creation of the new user.

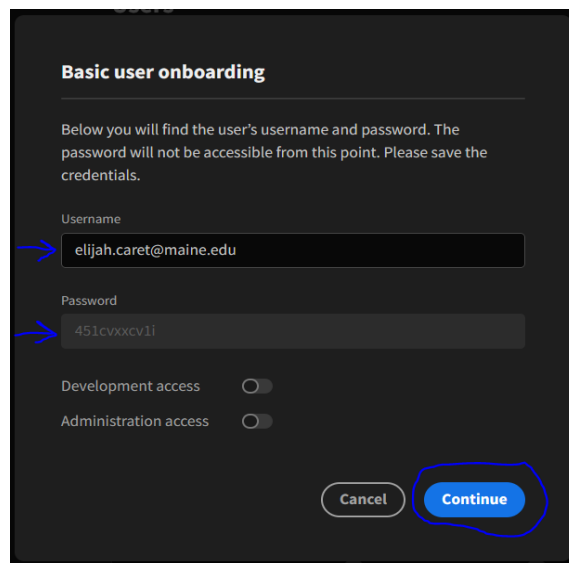
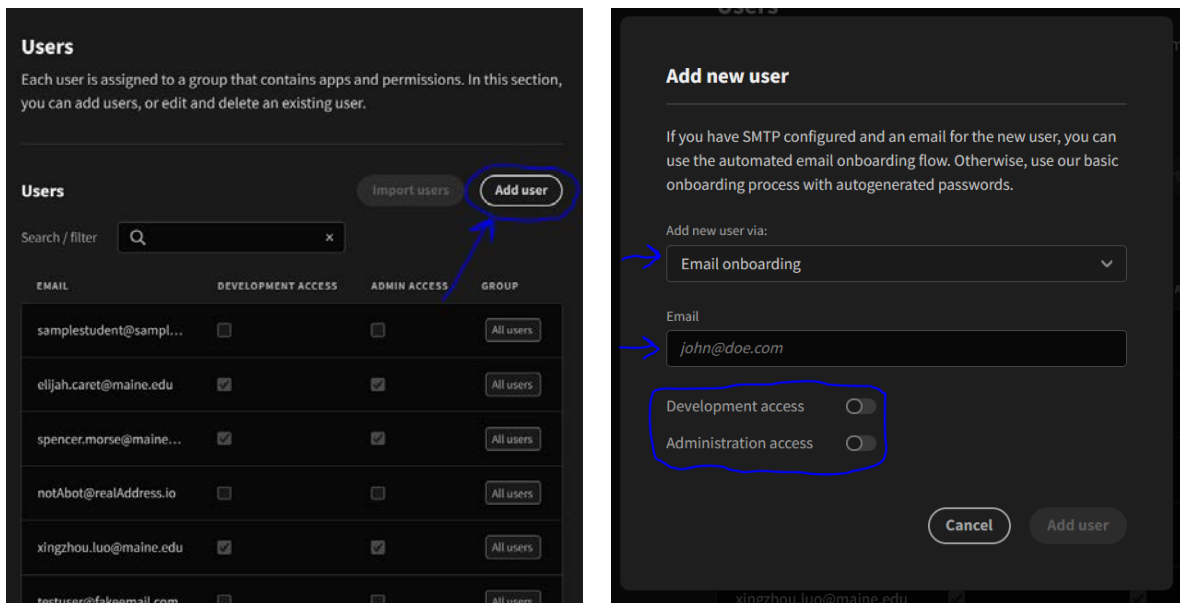


Figure 3-7: Creating a Username and Password via Basic Onboarding - The username must be an email address, which will make transition to email onboarding easier.

Lastly, the user needs to be added to the application itself. Clicking on the recently added user will open up their information. Scrolling down this page gives access to the “Configure roles” section. This section gives the administrator the ability to establish an access role for the particular application they want the user to be enrolled in. To add a user to a specific app, select the application under the “No Access” section. Then select the user’s access role for the application. In most cases, users will be given basic access to only the JMG application unless the user is an administrator. The user should now be able to access the application.



Figures 3-5 and 3-6: Adding a User - Currently email onboarding is not functional on our application, which means the user will need to be added via basic onboarding.

The administrator will enter the email that the user will use (most likely their school email or business email). A default password will be generated for the user which they will need for their first login attempt. Once a user successfully logs in for the first time, they will be prompted to change their password to something of their choosing. Clicking the “continue” button finalizes the creation of the new user.

Basic user onboarding

Below you will find the user's username and password. The password will not be accessible from this point. Please save the credentials.

Username
elijah.caret@maine.edu

Password
451cvxxcv1i

Development access ☐

Administration access ☐

Cancel Continue

Figure 3-7: Creating a Username and Password via Basic Onboarding - The username must be an email address, which will make transition to email onboarding easier.

Lastly, the user needs to be added to the application itself. Clicking on the recently added user will open up their information. Scrolling down this page gives access to the “Configure roles” section. This section gives the administrator the ability to establish an access role for the particular application they want the user to be enrolled in. To add a user to a specific app, select the application under the “No Access” section. Then select the user’s access role for the application. In most cases, users will be given basic access to only the JMG application unless the user is an administrator. The user should now be able to access the application.

Configure roles

Specify a role to grant access to an app.

APP	ROLE
No rows found	

No Access

Apps do not appear in the users portal. Public pages may still be viewed if visited directly.

APP
JMG App
Michaels_Build
Elijah Test
test

Update App Role

Update a@e.com's role for JMG App.

Role
Choose an option

Cancel Update role

Figures 3-8 and 3-9: Configuring User Permissions for App Access - The available roles to select from include basic, power, and admin. All users will either have basic or admin access.

Removing Users:

Going back to the user information for a particular user (found under the user tab in the development portal - see figure 3-4) there is an option at the bottom of the page (scroll all the way down) to delete that user. Click the red button, and then click confirm and the user will be removed.

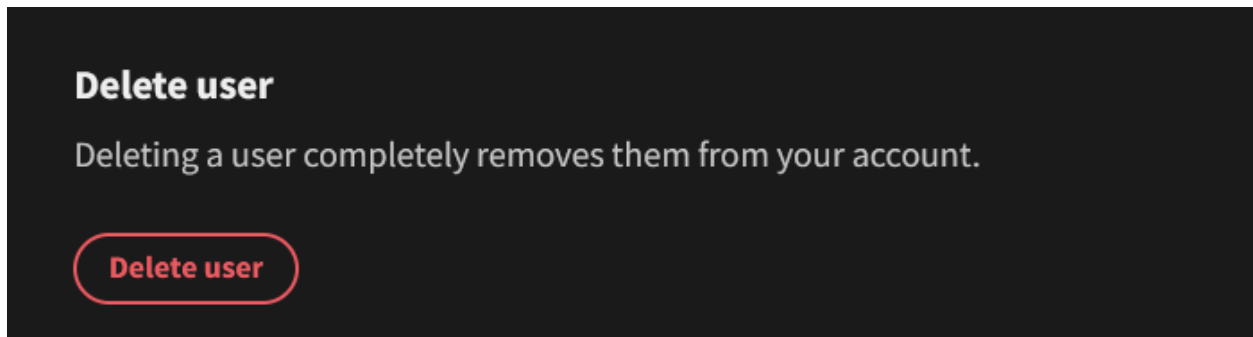


Figure 3-10: Delete User Option - Once opening up the user information, this option is located all the way at the bottom of the page (scroll all the way down).

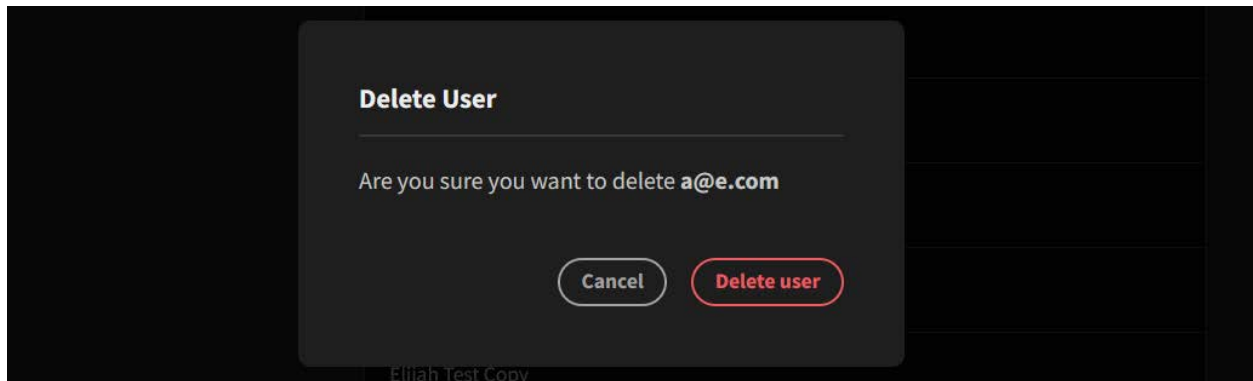


Figure 3-11: Delete User Confirmation

Fixing any Database Errors:

If any error in the database of the application is detected, such as incorrect information (according to the user whom the data is centered around), it can be fixed by manually editing the particular row that contains the data. With a fairly large dataset, the filter feature can be used to help locate the error quicker. This filter feature is located at the top of the table as shown in figure 3-12.

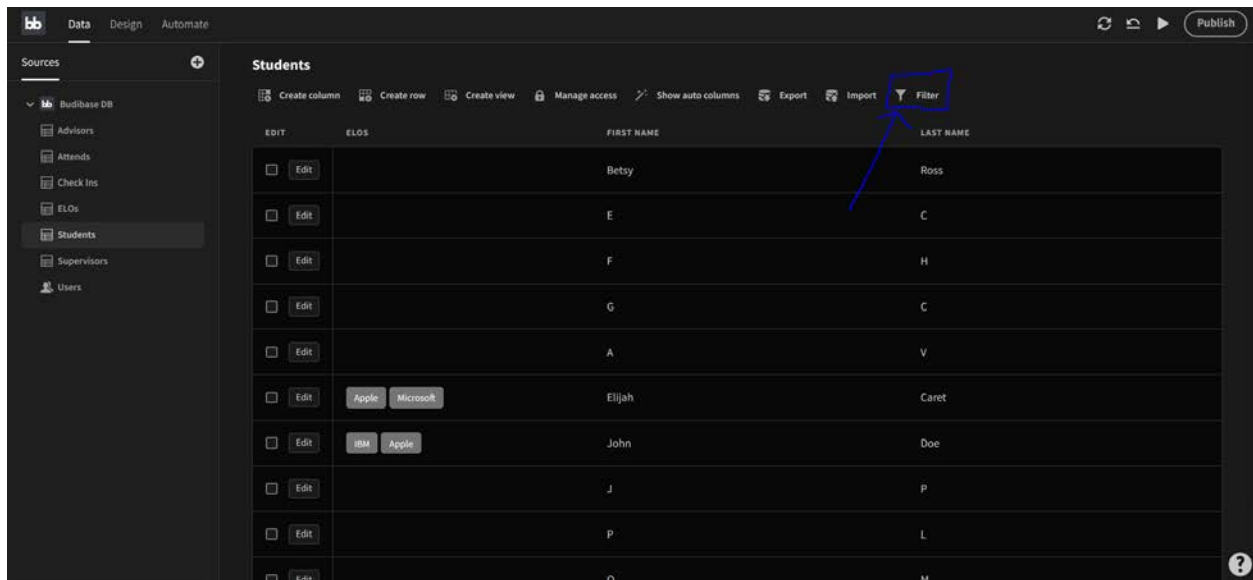


Figure 3-12: Location of Filter Feature - This feature is particularly useful to help locate a particular row.

Once the filter feature is opened, click “add filter”. Then select a particular column and value to search for. For example, figure 3-13 shows a query to search for users with the first name John.

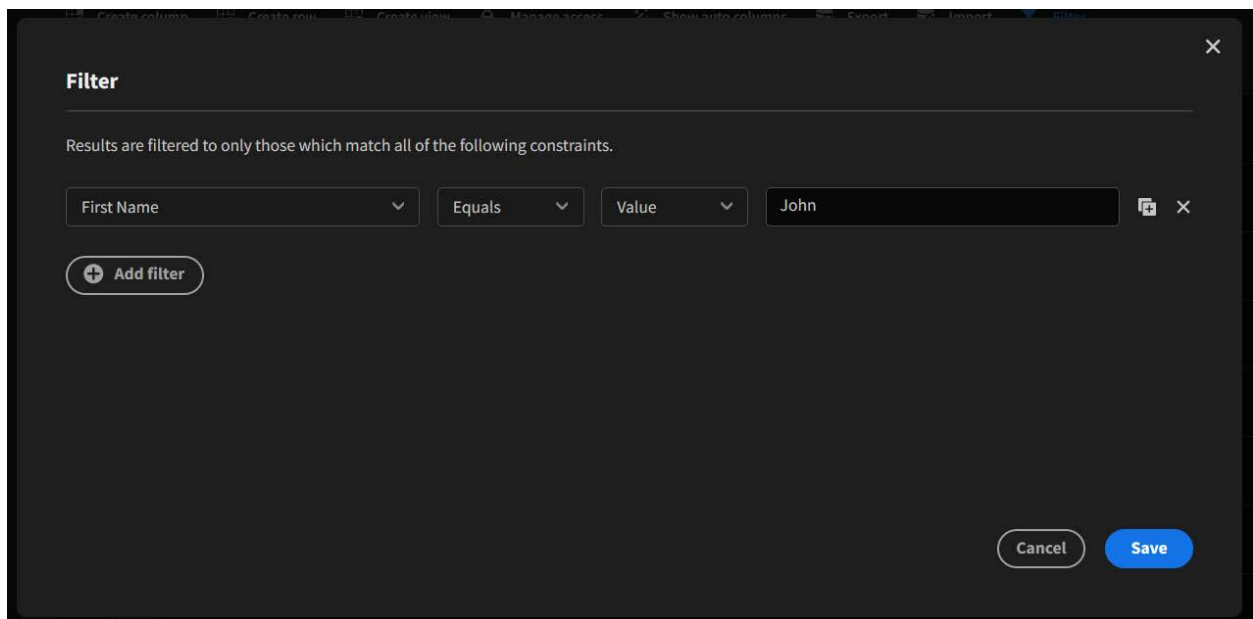


Figure 3-13: Filtering the Table - The first dropdown option is the column, while the next three options set the condition to filter the table by.

Once the proper row has been located, it can be edited by clicking on the edit button on the left side of the row. All visible attributes can now be changed based on what is incorrect. For example, in the table shown in figure 3-14 a student might have been assigned to the wrong company/organization (ELOs attribute).

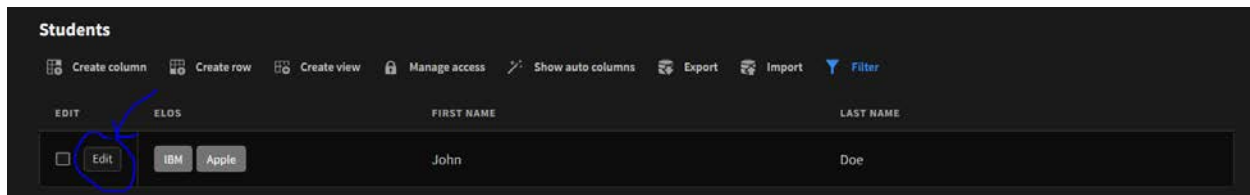


Figure 3-14: Edit Row - This option allows the administrator to edit the values of the attributes of a row.

A screenshot of a 'Edit Row' pop-up screen. It has a title bar with a close button (X). The form contains three input fields: 'ELOs' with a dropdown menu showing '(2) IBM, Apple', 'First Name' with the text 'John', and 'Last Name' with the text 'Doe'. At the bottom of the form are two buttons: 'Cancel' and 'Save Row'.

Figure 3-15: Edit Row Pop-up Screen - If John was actually attending Microsoft instead of Apple, the value of the ELOs attribute would be changed by accessing the respective dropdown bar.

Manually Adding a Missed Check-In:

If a student happens to forget to check in to their respective ELO, a JMG representative from their school, who is responsible for keeping track of their respective student's performance should contact the administrator to notify them. The administrator will then add that particular check-in, with the proper information, into the proper data table. This table is found under the name "Check-Ins" under the data tab of the application itself (Not the development portal homepage).

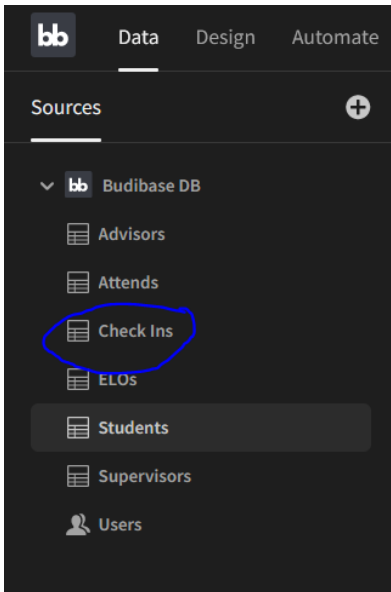


Figure 3-16: Location of Check-Ins Table - This contains all of the information about every recorded check-in on the system.

To add a row, click the “create row” icon above the table and enter the necessary information.

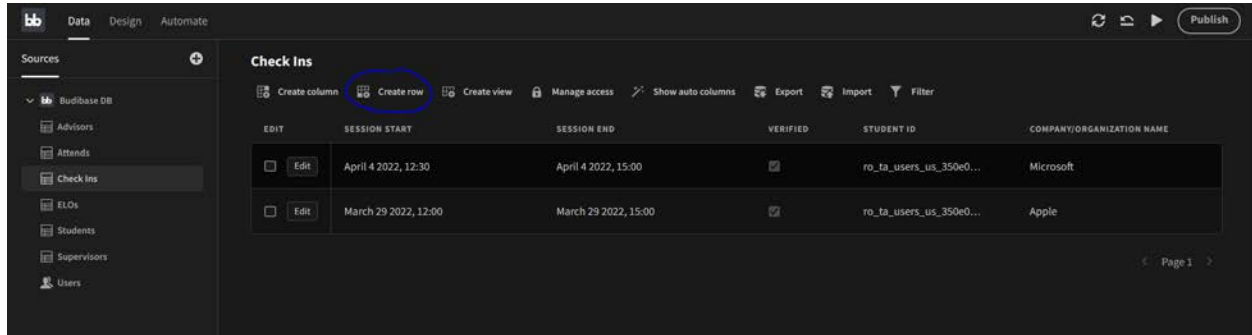
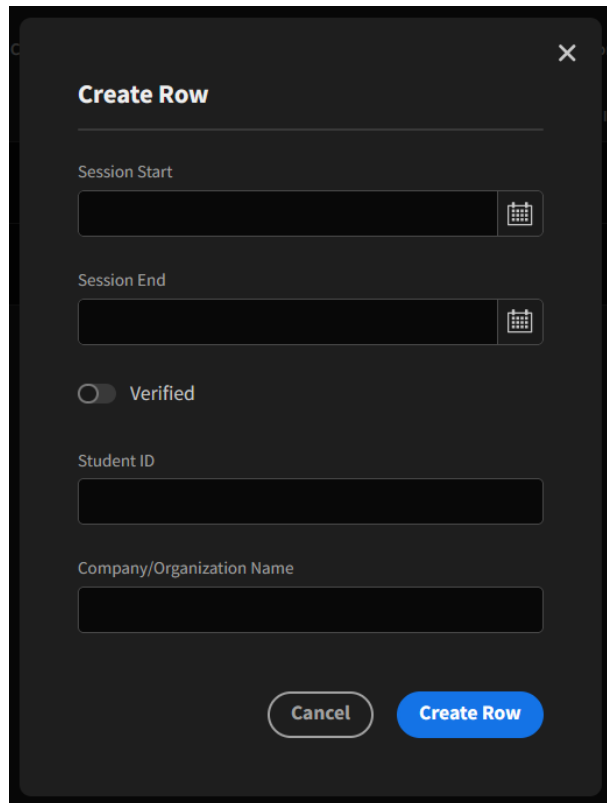


Figure 3-17: Location of Create Row Icon - This feature allows the administrator to manually add a new row to the table.

The information required here includes the time the session began, the time it ended, whether it was verified by the supervisor (this should be left unchecked, as the supervisor is responsible for verifying their respective student’s check-ins), the ID of the student and the company/organization they attended.

A dark-themed pop-up window titled "Create Row" with a close button (X) in the top right corner. The form contains five input fields: "Session Start" and "Session End" each with a calendar icon to its right; a "Verified" toggle switch currently turned off; "Student ID"; and "Company/Organization Name". At the bottom are two buttons: "Cancel" and "Create Row".

Create Row

Session Start

Session End

☐ Verified

Student ID

Company/Organization Name

Cancel Create Row

Figure 3-18: Create Row Pop-up Screen - This is where all check-in information is entered (with the exception of the verified switch, which remains off).

3.4 Periodic Administration

Database Backups:

Budibase provides a feature that allows administrators to export any table to a .csv/excel file. Navigate to the table you want to save. Then, at the top of the table, there will be several icons to edit the table itself. Click on the export icon as shown in figure 3-19. It can either be exported as a .csv file or .json file if needed. After that, it is recommended that this file gets backed up to either a cloud storage service or on a flash drive. This is usually done at the end of the student's semester/term.

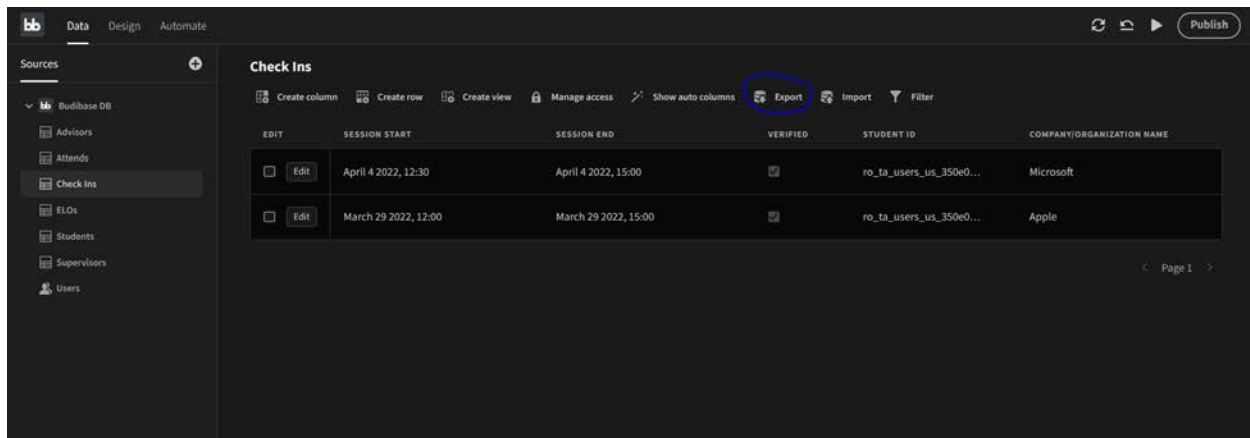


Figure 3-19: Location of Export Icon - This feature is used to download any table in the application database to a local machine.

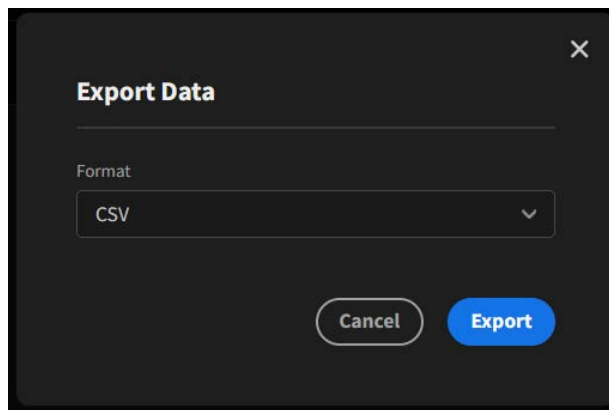


Figure 3-20: Choose Format to Export Table - The dropdown allows for the selection of either a .csv or .json file.

Database Cleanup:

The administrator must ensure that all desired data has been backed up and saved before this is done. The only way to clean up the database is to remove each row individually. The Cyber Cookie team is currently investigating a potential alternative to this and will update this document accordingly. As of now however, the cleanup is done through deleting rows. To delete the row, select it by clicking anywhere on it, or select the checkbox at the left side of the row.

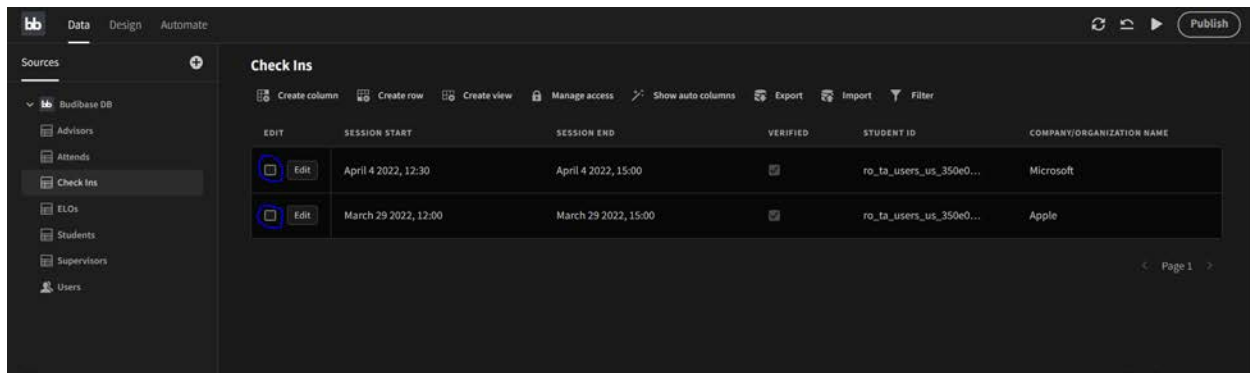


Figure 3-21: Selecting Rows - It can be done either by selecting the checkbox to the left of each row or by clicking anywhere on the row itself.

A new option will appear with the other icons at the top of the table labeled “Delete (x) row(s)”. Click on this option to remove the selected rows.

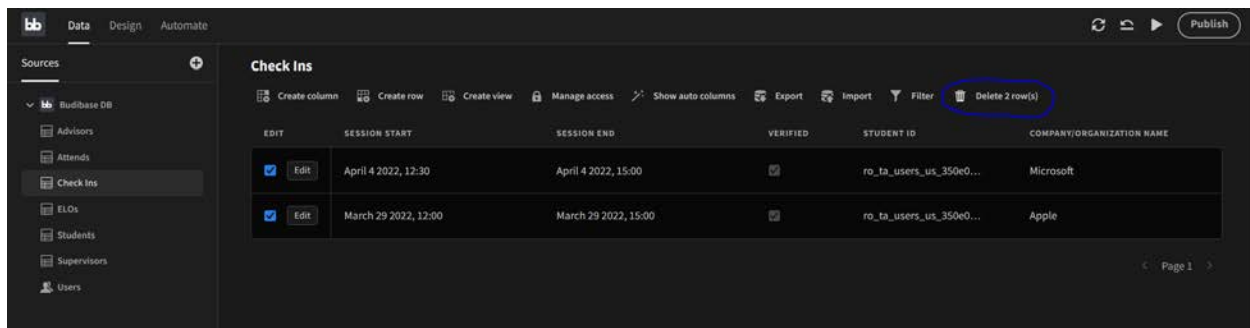


Figure 3-22: Deleting Rows - Once the desired rows to be deleted have been selected, a new icon will appear at the top of the table as circled

3.5 User Support

Currently any user support can be directed to any members of the Cyber Cookie team.

Elijah Caret: elijah.caret@maine.edu
 Michael Ferris: michael.ferris@maine.edu
 Xingzhou Luo: xingzhou.luo@maine.edu
 Spencer Morse: spencer.morse@maine.edu

Eventually support will be transferred over to the JMG ELO technologist via a support email (elohelp@jmg.org for example) as well as a phone number.

4. Troubleshooting

4.1 Dealing with Error Messages and Failures

Build Breaks:

In the event of a broken update to the system being published. Budibase provides a version control system that will allow administrators to revert back to a previous version of the application. At the top right of the screen, there are three icons as well as the publish button. Selecting the middle icon will revert the application back to the previously published version.

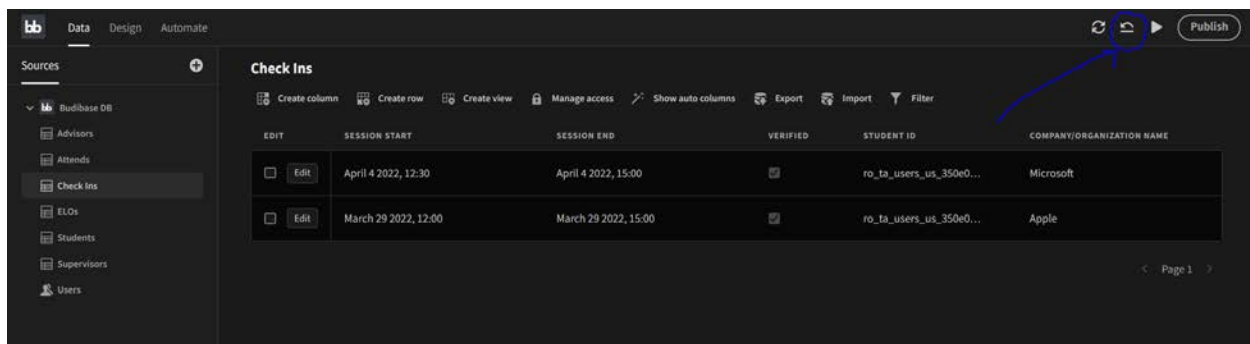


Figure 4-1: Version Control Feature - Most system breaking errors are mitigated by clicking this icon, which reverts back to the previous working version

Other helpful tips here are to run/make any new updates on an unpublished private copy of the application. It's also important to save all data prior to publishing a new update in the event of a system failure.

4.2 Known Bugs and Limitations

The user interface of the system is very limited and basic to serve functionally. There is currently no student performance assessment functionality implemented, although that will change before acceptance testing. Table linking is not fully fleshed out, as the Cyber Cookie team is still trying to understand how the database works in that respect. Relationships are the definite feature to use to link tables, and more specifically columns together, but there has been a challenge trying to utilize these relationships in the design module of the application. The established roles have also not been fully implemented, the student and supervisor roles are in place, but school supervisor/JMG representative roles have not. This is also something that will be most likely finished by acceptance testing.

Functionality to adjust for phone screens is not finished but is currently under development. This simply was something that appeared later on in the team's checklist for things to complete.



Appendix A: Agreement between Customer and Contractor

By signing this document, all parties agree that this is a complete list of system architecture and design for the JMG Student Check-in Site application. In the case that system architecture and design or any other information in this document need to change for the contract to be fulfilled, the following procedure will be followed: The party that believes a change is necessary shall contact the other party, explaining the situation. A meeting between the two parties will be held to discuss the problem and possible solutions. Once an agreement has been reached, modifications to this document will be made to reflect it, and all members of each party will sign the new document, which will then replace this one.

Team Members:

Name: Elijah Caret	Signature: <u></u>	Date: 4/06/2022
Name: Michael Ferris	Signature: <u></u>	Date: 4/06/2022
Name: Xingzhou Luo	Signature: <u></u>	Date: 4/06/2022
Name: Spencer Morse	Signature: <u></u>	Date: 4/06/2022

Customers:

Name: Samantha Brink	Signature: <u></u>	Date: 04/06/2022
Name: Lanet Anthony	Signature: <u></u>	Date: 04/06/2022

Appendix B: Team Review Sign Off

By signing below, all members agree that they have reviewed this document and agree on its content and format.

Name: Elijah Caret	Signature: <u></u>	Date: 04/06/2022
Name: Michael Ferris	Signature: <u></u>	Date: 04/06/2022
Name: Xingzhou Luo	Signature: <u></u>	Date: 04/06/2022
Name: Spencer Morse	Signature: <u></u>	Date: 04/06/2022

Appendix C: Document Contributions

- **Elijah Caret** - 65%
Section 1, Section 2, Section 3, Appendix A, Appendix B, and Appendix C
- **Xingzhou Luo** - 10%
Proofread and Formatting
- **Spencer Morse** - 25%
Section 3, Appendix A, Appendix B, and Appendix C