Lab 2 - Inclusive Classroom Product Specification

Travis Bennett, Aubrie Davie, Dalton Hanbury, Greg Hubbard, Colton Hurst, Randy Layne, Grant Ralls, Sergiy Yermak

CS 411w
Thomas Kenndey

November 8, 2021

Version 2

3 Specific Requirements

This document contains the Inclusive Classroom prototype requirements. These requirements include the functional requirements, non-functional requirements, performance requirements, and the assumptions and constraints necessary to accomplish the prototype features. The functional requirements are grouped by user types and subgrouped by prototype features and capabilities using the notation 3.1.x and 3.1.x.y respectively, where x and y are subheading numbers.

Following each requirement, there will be a line that displays the requirement authors. A requirement can have two types of authors, which include: the Originator and any Modifiers. To identify the originating and modifying authors the following notation will be used: (O: Last Name, M1: Last Name, M2: Last Name,...,MN: Last Name). In this notation, O represents the Originator and MX represents the Modifier, where X is the sequence of modification from 1 to N. This notation was defined by a previous Old Dominion University CS 411W development team named Cryptolio, and it is intended solely for software configuration management purposes (2018). Utilizing Originator-Modifier notation will allow evaluators to give credit to the authors of this document.

[This space is intentionally left blank]

3.1 Functional Requirements

3.1.1 Student Views

3.1.1.1 Student Dashboard

An Inclusive Classroom Student shall be brought to the student dashboard after login. The student dashboard will display the following selectable tabs:

3.1.1.1.1 Classwork Tab

The classwork tab shall be a central source for navigating to assignment related activities. The tab shall display an index of all assignments a student has in tabular format. Assignments will be broken into two categories, incomplete and complete, with the incomplete assignments placed first on the page layout.

1. Each row in the incomplete assignment table will link to the Student Assignment Submission View [Ref. 3.1.1.9] and display
    - The name of the assignment
    - The name of the class
    - The assignment due date
2. Each row in the completed assignment table will link to the Graded Assignment View [Ref. 3.1.1.6] and display
    - The name of the assignment
    - The name of the class
    - The date the assignment was completed

3.1.1.1.2 Meeting Tab

The meeting tab shall be the central source for navigating to video content. It will contain links to the following views:

1. Link to Video Meeting View [Ref 3.1.1.3]
2. Link to Live Video View [Ref 3.1.1.4]
3. Link to Recorded Video View [Ref 3.1.1.5]

(O: Layne)

[This space is intentionally left blank]

3.1.1.2 Live Video Call To Action

A student who is logged in and has a connection to the internet shall see a call to action button appear whenever there is a live video in progress for a class the student is enrolled in. The call to action button will contain a link to the video meeting view for that video. The call to action will display information about the video that is in progress:

- The name of the class
- The teacher of the class
- When the live video began
- The live video end time

(O: Layne)

3.1.1.3 Video Meeting View

A student will have a layout that shows past, current, and future live-streams of classes to which they have been assigned to. The classes to display will be determined by the data stored in the SQL database which is accessed by the API. The student will be sent to a live-stream or a recording depending on the button that was selected. The following functional requirements must be met:

1. Retrieve the classes in which the currently logged-in student is enrolled from the API.
2. Display the classes in a table format on the page.
    - 2.1. Display a class name.
    - 2.2. If the live stream is active, then display a join button.
        - 2.2.1. If the join button is selected, then the student will be redirected to the live-steam viewing page [Ref. 3.1.1.4].
    - 2.3. If a live stream is available, then display a watch button.
        - 2.3.1. If the watch button is selected, then the student will be redirected to a Recorded Video View [Ref. 3.1.1.5].

(O: Ralls, M1: Yermak)

[This space is intentionally left blank]

3.1.1.4 Live Video View

A student who is logged in to the application with an active internet connection will be able to view ongoing live-streams of classes to which they have been assigned. The following functional requirements must be met:

1. Display a viewing page for the meeting.
2. Display the class details of the live meeting. The class details include:
   - Name of the class
   - Title of the meeting
   - Teacher teaching the class
   - Date of the meeting
3. Utilize and display a Zoom live meeting integration for viewing the live-stream.
4. Display an exit button in the upper right corner to return to the Video Meeting View.

(O: Hanbury)

3.1.1.5 Recorded Video View

A student who is logged in to the application and has downloaded any class recordings will be able to view the downloaded recordings of classes to which they have been assigned. The following functional requirements must be met:

1. Display a viewing page for the class recording.
2. Display the class details of the class recording. The class details include:
   - Name of the class.
   - Title of the meeting.
   - Teacher teaching the class.
   - Date and time of the meeting.
3. Utilize and display a Zoom recorded meeting integration for viewing the class recording.
4. Display an exit button in the upper right corner to return to the Video Meeting View.

(O: Hanbury)

[This space is intentionally left blank]

3.1.1.6 Graded Assignment/Feedback View

A student who is logged in to the application and has downloaded feedback and grades for previously submitted assignments will be able to view the graded assignments and the feedback associated with individual assignments. The following functional requirements must be met:

1. Retrieve the graded assignment.
2. Display details associated with the graded assignment:
   - Name of assignment.
   - Class associated with assignment.
   - Overall grade received on assignment.
   - Teacher notes regarding assignment.
   - Date assignment was submitted.
   - Due date of assignment.
3. Display all files associated with the graded assignment on the page:
   - Files included by the teacher.
   - Files submitted by the student.
4. Display answers submitted by the student.
5. Display credit received for each answer submitted by the student.
6. For short answer [Ref 3.1.5.1] and essay questions [Ref 3.1.5.2], display feedback for the submitted answer with the following attributes:
   - Correctness of answer given by student: correct, incorrect, or partially correct.
   - Reason for answer being correct, incorrect, or partially correct.
7. For “true or false” questions [Ref 3.1.5.3], display feedback for the submitted answer with the following attributes:
   - Correctness of answer given by student: correct or incorrect.
   - Explanation of why an answer is incorrect.
8. For “select the only one that applies” questions [3.1.5.4], display feedback for the submitted answer with the following attributes:
   - Correctness of answer given by student: correct or incorrect.
   - The correct answer if the submitted answer was incorrect.

(O: Hanbury, M1: Layne)

[This space is intentionally left blank]

3.1.1.7 Student Profile View

Inclusive Classroom students shall be able to view and modify their profile presented to the class. The following functional requirements must be met:

1. Edit profile
    1.1. Profile Picture
        1.1.2 Add picture
        1.1.3 Change picture
        1.1.4 Remove picture
    1.2. Add preferred name and pronouns
2. Change password according to security standard

(O: Hubbard, M1: Layne)

3.1.1.8 Class Detail View

Inclusive Classroom students shall be able to view their specific class information from a top level. From this view, the following functional requirements must be met:

1. Display current grades
2. Display class announcements which include:
    - Upcoming assignments
    - Recent grades
    - Recent teacher messages
    - Upcoming class meetings
3. Display relevant class information
    - Name of the course
    - Teacher contact information
    - Classmates names
    - Syllabus

(O: Hubbard, M1: Yermak)

[This space is intentionally left blank]

3.1.1.9 Student Assignment Submission View

An Inclusive Classroom student will be able to see the classes in which they are enrolled. Each class will have a set of associated assignments. After navigating to the assignment view, a student can access the submission view by clicking on an incomplete assignment. The student will be able to submit an assignment using various means based on the requirements set by the teacher. The following functional requirements must be met:

1. Retrieve the enrolled classes of the student who is currently logged in.
2. Retrieve the selected assignment information for each course.
3. Display the assignment due date on the page.
4. Display all assignment details associated with the selected assignment on the page. The assignment details include:
   - Name of the assignment
   - Background information for the assignment.
   - Instructions for completing the assignment.
5. Display all files associated with the selected assignment on the page.
6. If short answers are required by the teacher, then display a text box for submission.
7. If essay style answers are required by the teacher, then display a text area for submission.
8. If "select all that apply" style answers are required by the teacher, then display checkboxes for submission.
9. If "select the only one that applies" style answers are required by the teacher, then display radio buttons for submission.
10. If file submissions are required by the teacher, then display an attach file or upload file button.
11. Display a submit button at the bottom of the page for all submission types.

(O: Hanbury, M1: Davie)

[This space is intentionally left blank]

3.1.2 Teacher Views

3.1.2.1 Teacher Dashboard

An Inclusive Classroom Teacher shall be brought to the teacher dashboard after login. From the teacher dashboard, they shall be able to access the following views:

1. Class Overview View [Ref. 3.1.2.5].
2. Video Recording Creation View [Ref. 3.1.2.6].
3. Assignment Overview Creation View [Ref. 3.1.2.3]

From the dashboard, with the ability to access the above views, they should be able to successfully complete all actions the teacher will need for their day to day work. This includes the following:

- Creating a pre-recorded video for a class or multiple classes
- Creating a live stream that a class or multiple classes can access
- View the settings and info for the classes they are teaching
- View the students in the classes they are teaching
- Assign assignments to a class or multiple classes
- Assign grades to submissions from students

How to access all of the above actions are documented in the individual views within this document.

(O: Hurst)

3.1.2.2 Assignment Index View

This page will display to a teacher all of the assignments that have been created within a class. This page will be accessed through the teacher dashboard within a specific class from the class overview view. The following functional requirements must be met:

1. Display all the created assignments for the specific class.
2. Provide the capability to navigate to the Grade Assignment view for each assignment.
3. Provide the capability to navigate to the Create Assignment view.

(O: Bennett)

3.1.2.3 Assignment Overview Creation View

An Inclusive Classroom Teacher can create new assignments for a class that they are assigned to teach. The results obtained from this view will be used to determine when an assignment will be available to download by a Student and what the relative priority of an assignment is based on assignment type. The following requirements shall be met:

1. Display an HTML form which will allow the user to assign the following attributes to the assignment.
   - Class
   - Start date and time
   - Due date and time
   - Type of assignment as described in section 3.1.5
2. Display a submit button
   2.1. If the submit button is selected, then redirect the user to the Assignment Content Creation View [Ref. 3.1.2.4].
3. Display a cancel button
   3.1. If the cancel button is selected, then redirect the user to the Teacher Dashboard View [Ref. 3.1.2.1].

3.1.2.1.1 The view shall utilize a HTML form to allow the Teacher to assign the following attributes to the assignment:

- Class the assignment is for
- Assignment start date and time
- Assignment due date and time
- Type of assignment as described in section 3.1.5

3.1.2.1.2 The HTML form will provide a button to submit the form results and redirect a Teacher to the Assignment Question Creation View [Ref. 3.1.2.4] and a button to cancel the form completion and redirect the Teacher back to the Teacher Dashboard View.

(O: Layne, M1: Yermak)

[This space is intentionally left blank]

3.1.2.4 Assignment Content Creation View

An Inclusive Classroom Teacher can see the existing assignments they have created, and will have the ability to create more assignments, or update existing assignments. These assignments will clearly show (like rows in a table with this data as columns) the following:

- The name of the assignment
- The class(es) the assignment is linked to
- A button to edit the assignment. The following edits can be made to an assignment:
    - Delete the assignment
    - Add a question to the assignment drop down selection. Selection of questions include:
        - Short Answer Question [Ref. 3.1.5.1]
        - Essay Question [Ref. 3.1.5.2]
        - True or False Question [Ref. 3.1.5.3]
        - Multiple Choice Question [Ref. 3.1.5.4]
    - Delete a question
    - Modify a question
- A fraction showing how many submissions have been made to the assignment as the numerator, and the denominator showing how many students the assignment is assigned to
- The creation and due date of the assignment

The view will also allow, when clicking the column headers, to sort the assignments. The teacher should be able to sort by all data columns that can be compared, swapping between ascending and descending.

(O: Hurst)

3.1.2.5 Class Overview View

An Inclusive Classroom Teacher can see the classes they are currently teaching. For each class they are teaching, they can see the students in that class.

The view should allow for the teacher to do the following:

1. View the list of classes assigned to the teacher.
2. For each class, view the list of students in that class [Ref. 3.1.3.7].
3. For each class, view the class details, which consists of the following:
   - Class Name
   - Class Start Date
   - Class End Date
   - Class weekly meeting time(s)
   - Class location / room number
   - Class description
4. For each student in the class, the teacher can click on the student name and view the student grades for that class.

(O: Hurst, M1: Davie)

3.1.2.6 Video Recording Creation View

Once authenticated, the Inclusive Classroom Teacher will have the ability to create new recordings or start livestreams for each class they teach. Each recording will be downloadable for assigned students with recommended "watch by" markings for self scheduling. Each livestream will be viewable with "Join" markings.
The following functional requirements must be met.

- The teacher view will offer a HTML form with the following attributes:
  - Drop down selection for class identification
  - Recording post date
  - Recording recommended watch-by date
- The HTML form will also have a submit form button for the teacher that redirects to the original livestreams / recordings view.
- The View will allow the teacher to start a livestream
- There will also be a "back" button to cancel without saving the HTML form that redirects to the livestreams / recordings view.

(O: Hubbard, M1: Ralls)

3.1.2.7 Grade Assignment View

An authenticated Teacher can grade the submitted assignment.
The view should allow the teacher to do the following:

- For each individual question
    - Assign full credit
    - Assign partial credit
    - Leave comments
- Submit the graded assignment

An example use case:

1. Teacher Bob views his list of classes
2. Bob then clicks on his A Day class
3. Bob then clicks the assignments button
4. Bob then selects the assignment type
5. Bob then selects an individual assignment to grade
6. Bob then gives credit as appropriate to each question
7. Bob then submits the grade by the “Submit” button at the bottom of the view

(O: Ralls)

3.1.3 Administrator Views

3.1.3.1 Administrator Dashboard

After logging in, this is the first page an administrator of Inclusive Classroom must see. From this page, the administrator shall be able to access all other functionalities of the application. The following functional requirements must be met:

1. Display link to the Class Creation View [Ref. 3.1.3.2]
2. Display link to the Teacher Evaluation View [Ref. 3.1.3.3]
3. Display link to the Teacher Index View [Ref. 3.1.3.4]
4. Display link to the Student Index View [Ref. 3.1.3.7]

(O:Yermak)

3.1.3.2 Class Creation View

An Inclusive Classroom Administrator shall be able to create new classes and assign a teacher and a group of students to those classes. The following functional requirements shall be met:

1. Display a one-line text input field for the class name.
2. Retrieve all teachers currently assigned to the administrator's school [Ref. 3.1.3.4].
    - 2.1. Display a select input field to select a teacher for the class.
3. Retrieve all students currently enrolled in the administrator's school [Ref. 3.1.3.7].
    - 3.1. Display checkbox input fields to assign students to a class.
4. Display a back button that redirects to the Administrator Dashboard [Ref. 3.1.3.1].

(O: Yermak, M1: Davie)

3.1.3.3 Teacher Evaluation View

An Inclusive Classroom Administrator shall be able to evaluate individual teachers assigned to a school. A record of the evaluation shall be stored in the database and be available for future reference. The following functional requirements must be met:

1. Retrieve all teachers currently assigned to the administrator's school [Ref. 3.1.3.4].
    - 1.1. Display a select input field to select a teacher for an evaluation.
2. Display a text area input field to write the evaluation into.
3. Display a submit button to submit the evaluation to the database.
4. Display a back button to get back to the Administrator Dashboard [Ref. 3.1.3.1].

(O: Yermak)

3.1.3.4 Teacher Index Overview

An Inclusive Classroom Administrator shall be able to view all teachers assigned to the administrator's school. An administrator shall be able to view the total number of students assigned to a teacher, all classes assigned to a teacher, and the class details.

3.1.3.4.1 The Teacher Index shall provide the administrator the capability to view all teachers and their respective total number of students and course schedule. The following functional requirements shall be met:

1. Retrieve all teachers currently assigned to the administrator's school.
2. Display a list of all teachers along the left side of the page.
3. Display a teacher's name as a link.
4. If the administrator clicks on a teacher's name, display that teacher's name, their total number of students, a list of all assigned classes, and class status.
5. Display a class's name as a link that redirects to the Class Overview [Ref. 3.1.3.5]

3.1.3.4.2 The class list component shall provide the administrator the capability to view specific class details about each class assigned to a teacher. The following functional requirements shall be met:

1. Display an expansion button next to a class name.
2. If the administrator clicks the expansion button, display a table of class details. The class details include:
   - Class ID
   - Class status
   - Class description
   - Class size
   - Class start date
   - Class end date
   - Class meeting times

3. The administrator can proceed to the Class Overview from the Teacher Index or utilize a back button that redirects the administrator to the Administrator Dashboard [Ref. 3.1.3.1].

(O: Davie)

3.1.3.5 Class Overview

An Inclusive Classroom Administrator shall be able to view all students assigned to a class and a student's current overall grade in that class after selecting an individual class in the Teacher Index [Ref. 3.1.3.4.1]. The results obtained from this view will be used by the administrator to report on a class's grades. The following functional requirements shall be met:

1. Display a class's name, status, and total number of students enrolled in that class.
2. Display an expansion button next to a class's name.
3. If the administrator clicks on the expansion button, then all of the students enrolled in a class shall be displayed in a table format with each student's name and corresponding overall grade.
4. Display a student's name as a link that redirects to the Student View [Ref. 3.1.3.6].
5. Display a back button that redirects to the Teacher Index [Ref. 3.1.3.4].
6. Display a button that allows the administrator to abandon their search and redirects to the Administrator Dashboard [3.1.3.1].

(O: Davie)

3.1.3.6 Student View

An Inclusive Classroom Administrator shall be able to see all information related to an individual student, which includes: student name, classes enrolled in, and overall class grade. Furthermore, the administrator shall be able to expand a selected class to view specific assignments and grades. The following functional requirements shall be met:

1. Retrieve the student selected by the administrator in either Student Index View [Ref. 3.1.3.7] or Class Overview [Ref. 3.1.3.5].
2. The displayed information on the student selected should include:
   - 2.1. Full name.
   - 2.2. ID number.
   - 2.3. Table of classes the student is enrolled in.
     - 2.3.1. Each row shall be an expansion button in the table and shall have a class name and an overall grade written on it.
     - 2.3.2. If the admin clicks the expansion button, then all of the student's assignments shall be displayed in table format with an assignment name and grade.

(O: Yermak)

3.1.3.7 Student Index View

An Inclusive Classroom Administrator shall be able to view all students enrolled in the administrator's school. The following functional requirements shall be met:

1. Display a table of all students enrolled in the administrator's school.
2. Display a student's name as a link that redirects to the Student View [Ref. 3.1.3.6].
3. Display a back button that redirects the administrator to the Administrator Dashboard [Ref. 3.1.3.1].

(O: Yermak, M1: Davie)

3.1.4 Multi-Role Views

3.1.4.1 Login View

This is the first page a user of Inclusive Classroom must see. The following functional requirements must be met:

1. Display a one-line text input field for the username.
2. Display a one-line text input field for the password.
3. Display a submit button to submit the username and password to the database.
4. Display a register button that will direct the user to a register page [3.1.4.2].

Furthermore, this page is responsible for authenticating the user who is trying to login. Refer to requirement 3.6.1.1 for authentication.

(O: Bennett, M1: Yermak)

[This space is intentionally left blank]

3.1.4.2 Registration View

A new user of Inclusive Classroom can register a new account if they do not already have one or want a new one. The following functional Requirements must be met:

1. Display a one-line text input field for first name.
2. Display a one-line text input field for middle name.
3. Display a one-line text input field for last name.
4. Display a date picker input field for birthday date.
5. Display a one-line text input field for email address.
6. Display a one-line text input field for phone number.
7. Display a one-line text input field for city name.
8. Display a one-line text input field for state name.
9. Display a one-line text input field for zip code.
10. Display a submit button
    - 10.1. If the user clicks the submit button, then the information listed above will be sent to the database.

(O: Yermak)

3.1.5 Assignments

Regardless of assignment type, assignments are collections of questions. Questions will have the following types and requirements.

(O: Layne)

3.1.5.1 Short Answer Question

Type of question that can be either selected by the teacher or presented to the student. The following functional requirements must be met:

1. Selected by the teacher
    - 1.1. Display a text box for writing the question.
    - 1.2. Display a submit button for submitting the question.
    - 1.3. If the submit button is clicked, then add the question to the assignment.
2. Presented to a student
    - 2.1. Display the question for the short answer assignment.
    - 2.2. Display a text box for writing the submission.
    - 2.3. Display a submit button below the text box.
    - 2.4. If the submit button is clicked, then send the submitted text to the database.

(O: Yermak, M1: Layne)

3.1.5.2 Essay Question

Type of question that can be either selected by the teacher or presented to the student. The following functional requirements must be met:

1. Selected by the teacher
    1.1. Display a text box for writing the question.
    1.2. Display a submit button for submitting the question.
    1.3. If the submit button is clicked, then add the question to the assignment.
2. Presented to a student
    2.1. Display the question.
    2.2. Display a text area for writing the submission.
    2.3. Display a submit button below the text area.
    2.4. If the submit button is clicked, then send the submitted text to the database.

(O: Yermak, M1: Layne)

3.1.5.3 True or False Question

Type of question that can be either selected by the teacher or presented to the student. The following functional requirements must be met:

1. Selected by the teacher
    1.1. Display a text box for writing the question.
    1.2. Display a submit button for submitting the question.
    1.3. If the submit button is clicked, then add the question to the assignment.
2. Presented to a student
    2.1. Display the question.
    2.2. Display two radio buttons labeled “true” and “false” that a student can select.
    2.3. Display a submit button below the radio buttons.
    2.4. If the submit button is selected, then send the selected radio button value to the database.

(O: Layne)

3.1.5.4 Multiple Choice Question

Type of question that can be either selected by the teacher or presented to the student. The following functional requirements must be met:

1. Selected by the teacher
    - 1.1. Display a text box for writing the question.
    - 1.2. Display a text box for entering a selectable answer.
    - 1.3. Display a plus button below the second text box
    - 1.4. If the plus button is clicked, then insert a text box between the plus button and the last text box for more answer options
    - 1.5. Display a submit button below the plus button
    - 1.6. If the submit button is clicked, then add the question to the assignment.
2. Presented by the student
    - 2.1. Display the question.
    - 2.2. Display a radio button and label for each of the selectable answers created by the teacher
    - 2.3. Display a submit button under the final radio button.
    - 2.4. If the submit button is clicked, then send the selected radio button value to the database.

(O: Layne)

3.2 Performance Requirements

Performance requirements will provide tangible benchmarks for user requests in order to ensure that the application is working efficiently and user experience is not compromised.

3.2.1 The API must respond to all HTTP requests in less than 1 second.
(O: Hurst, M1: Davie)

3.2.2 The database must resolve all queries within 1 second.
(O: Hurst, M1: Davie)

3.2.3 For locally stored lecture videos, the recording must be playable in under 5 seconds.
(O: Davie, M1: Ralls)

3.2.4 When navigating from any user view to a new view, the new view must load within 2 seconds.
(O: Davie)

3.3 Assumptions and Constraints

3.3.1 Inclusive Classroom assumes a student is given a device with the application already installed from their school.
(O: Davie)

3.3.2 In the worst case scenario, Inclusive Classroom assumes that a student has access to the internet at the beginning and the end of a semester.
(O: Hurst, M1: Davie)

3.4 Non-Functional Requirements

3.6.1 Security

3.6.1.1 Authentication

When a user tries to log in from the login view [Ref. 3.1.4.1], they must be verified/authenticated by the system. The following requirements must be met when authenticating:

1. The username and password submitted must be verified/found by the database.
2. If the username and password are not verified by the database, then a warning message must be displayed notifying the user that the inputs were not found.
3. The user must be able to try submitting the username/password combination up to five times.
4. If the user unsuccessfully enters the combination on the fifth attempt, then the system must restrict the user from another attempt for 24 hours.

(O: Yermak)

3.6.1.2 Account Access

When a user attempts to login to Inclusive Classroom using invalid credentials, they must receive a nondescript error message. An error message that does not tell which specific data field is incorrect follows cybersecurity best practices. The error message must state the following:

1. "Incorrect username and password combination"

(O: Davie)

3.6.1.3 Data Release Mitigation

In accordance with all Family Educational Rights and Privacy Act (FERPA) laws, student and teacher information will be back-end encrypted. The following requirements must be met when encrypting:

1. All data will be transferred via HTTPS.

(O: Hubbard, M1: Davie, M2: Yermak)

3.6.1.4 Authorization

Users must only have access to the views corresponding to their user role. If an unauthorized user attempts to access another user role's view, permission must be denied. The following requirements must be met:

1. Enforcement of user verification must be performed on the server-side.
2. Specific URLs must be inaccessible without proper authentication.

(O: Davie)

3.6.2 Maintainability

The Inclusive Classroom software will require maintenance and periodic updates to the API and client software. At minimum, it is advised that the client software is connected to the internet at least once a month so updates can be installed as needed.

As the API and database will always be available over the internet, they will also require updates. However, the client does not have to worry about this. Logging should be added to the API to ensure any errors will be documented and available for an Inclusive Classroom developer to fix any issues that arise.

3.6.3 Reliability

3.6.3.1 The client application must complete 99% of student user requests without error when all major components are stored locally in the SQLite Database.

(O: Davie)

3.6.3.2 The client application must be available to users for 99% of the school year. Since students and teachers are limited by the time constraints of a school year, the application must be available to meet their needs. Any application maintenance that requires the application to be unavailable must be planned during school breaks and holidays.

(O: Davie)

3.6.3.3 The API should have a minimum uptime of 95%, measured within a year's time. Only unplanned downtime will decrease the measured uptime percentage.

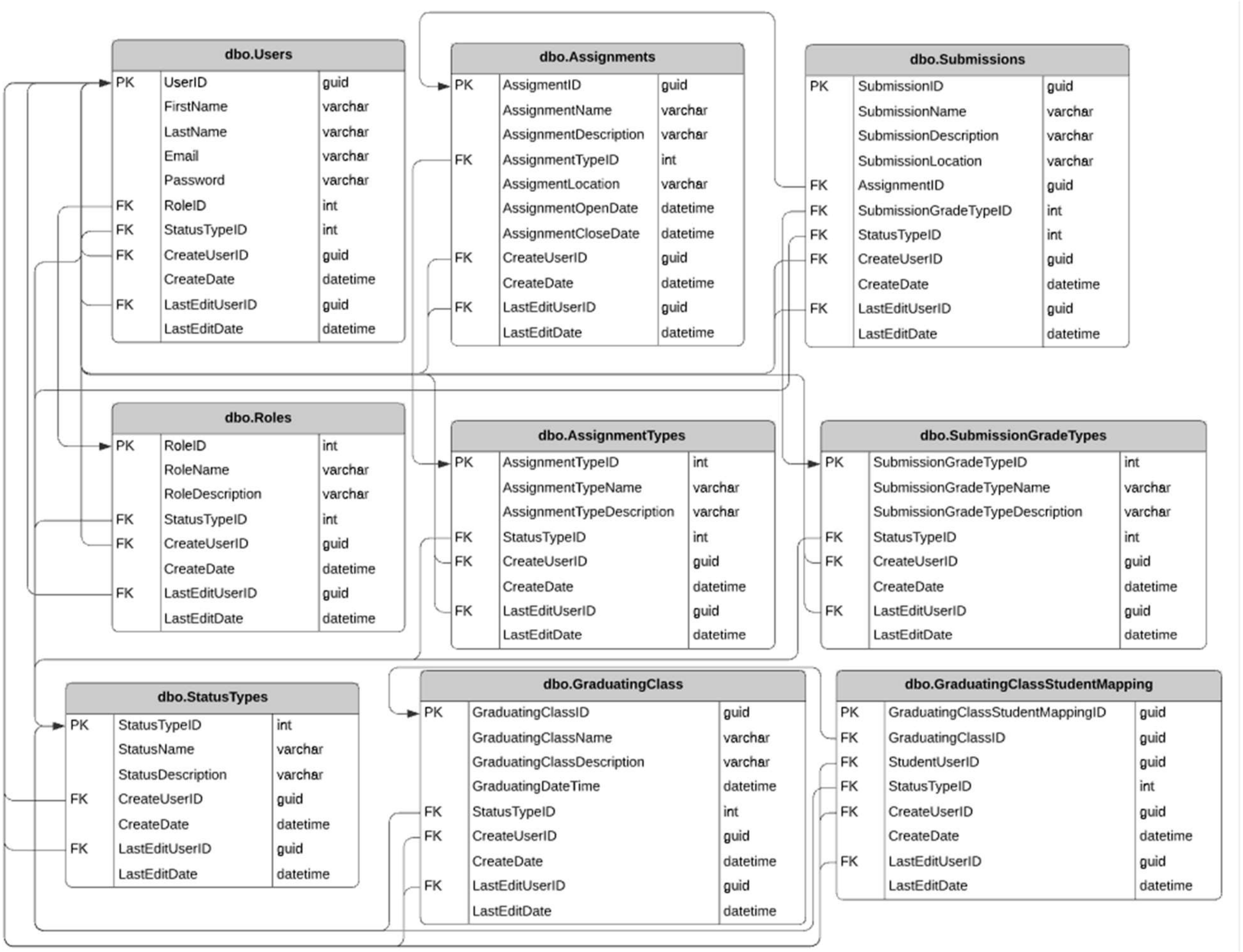
(O: Hurst)

3.6.3.4 The database should have a minimum uptime of 95%, measured within a year's time. Only unplanned downtime will decrease the measured uptime percentage.

(O: Hurst)

Appendices

Appendix A: Database Schema



	dbo.Users	
PK	UserID	guid
	FirstName	varchar
	LastName	varchar
	Email	varchar
	Password	varchar
FK	RoleID	int
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.Assignments	
PK	AssigmentID	guid
	AssignmentName	varchar
	AssignmentDescription	varchar
FK	AssignmentTypeID	int
	AssigmentLocation	varchar
	AssignmentOpenDate	datetime
	AssignmentCloseDate	datetime
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.Submissions	
PK	SubmissionID	guid
	SubmissionName	varchar
	SubmissionDescription	varchar
	SubmissionLocation	varchar
FK	AssignmentID	guid
FK	SubmissionGradeTypeID	int
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.Roles	
PK	RoleID	int
	RoleName	varchar
	RoleDescription	varchar
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.AssignmentTypes	
PK	AssignmentTypeID	int
	AssignmentTypeName	varchar
	AssignmentTypeDescription	varchar
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.SubmissionGradeTypes	
PK	SubmissionGradeTypeID	int
	SubmissionGradeTypeName	varchar
	SubmissionGradeTypeDescription	varchar
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.StatusTypes	
PK	StatusTypeID	int
	StatusName	varchar
	StatusDescription	varchar
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.GraduatingClass	
PK	GraduatingClassID	guid
	GraduatingClassName	varchar
	GraduatingClassDescription	varchar
	GraduatingDateTime	datetime
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

	dbo.GraduatingClassStudentMapping	
PK	GraduatingClassStudentMappingID	guid
FK	GraduatingClassID	guid
FK	StudentUserID	guid
FK	StatusTypeID	int
FK	CreateUserID	guid
	CreateDate	datetime
FK	LastEditUserID	guid
	LastEditDate	datetime

(O: Davie)

Appendix B: API v1 Proposal

API v1 should support the following:

- Creation of different user roles (admin, teacher, student, etc)
- Login capability authenticated via Basic Auth
- Ability for admin to assign students to classes, as well as access to teacher functionality
- Ability for teacher to create and update classes
- Ability for a teacher to create assignments
- Ability for a student to view their class
- Ability for a student to submit a submission for an assignment

(O: Hurst)

B.1 Routes

/user

Will fully allow login and user creation ability, from student to teacher to admin. Uses the *User Contract* as the json body contract.

- GET - get user info (/user/id)
- POST - create a user
- PUT - update a user (/user/id)
- DELETE - delete a user (/user/id)

/assignment

Represents assignments a teacher will create for a specified class. Uses the *Assignment Contract* as the json body contract.

- GET - get an assignment (/assignment/id)
- POST - create an assignment
- PUT - update an assignment (/assignment/id)
- DELETE - delete an assignment (/assignment/id)

/submission

Uses the Submission Contract as the json body contract.

- GET - get a submission (/submission/id)
- POST - create an submission
- DELETE - delete an submission (/submission/id)

/class

Uses the class contract as the json body contract.

- GET - get class details (/class/id)
- POST - create a class
- PUT - update a class (/class/id)
- DELETE - delete a class (/class/id)

/roles

- GET - get an array of the possible roles. Represents an enum, with the int values corresponding to the role positions in the array (0 to n, where 0 has the highest permissions). This will be an array of strings.

/statuses

- GET - get an array of the possible statuses. Represents an enum, with the int values corresponding to the role positions in the array. This will be an array of strings.

/assignment-types

- GET - get an array of the possible assignment types. Represents an enum, with the int values corresponding to the role positions in the array. This will be an array of strings. (Ex- homework, classwork, etc)

[This space is intentionally left blank]

/list

This endpoint provides easy access to lists of different resources. The output will be an array, where the array contents may be a list of strings (ex- string guids) or the list of actual contracts for that resource. This route may require change as the developers deem necessary.

- GET - get a list of the classes a teacher is teaching (/list?resource=classes&teacheruserid=x)
- GET - get a list of the students in a class (/list?resource=classstudents&classid=x)
- GET - get a list of the assignments for a class (/list?resource=assignments&classid=x)
- GET - get a list of the class submissions for a specific assignment (/list?resource=classsubmissions&classid=x&assignmentid=x)
- GET - get a list of the submissions for a specific assignment by a specific student (/list?resource=studentsubmissions&studentuserid=x&assignmentid=x)
- GET - get a list of all the students in the school (/list?resource=schoolstudents)
- GET - get a list of grades for a student by assignment (/list?resource=studentgrades&studentuserid=x)

(O: Hurst)

B.2 Contracts

User Contract

- Id (string) - will be a guid, used in some /user routes
- First Name (string)
- Middle Name (string)
- Last Name (string)
- Birthday (string) - ISO 8601 format
- Email (string)
- Phone Number (string) - as XXX-XXX-XXXX
- Street Address (string)
- City (string)
- State (string) - as XX (the two letter state code)
- Zip (string) - as XXXXX (5 number zip)
- RoleId (int) - determines what access role the user has (administrator, student, teacher), there will be an enum of possible values
- StatusId (string) - determines if the user is active / inactive / etc, there will be an enum of possible values

(O: Hurst)

Assignment Contract

- Id (string) - will be a guid
- Assignment Name (string)
- Assignment Description (string)
- AssignmentTypeId (string)
- AssignmentOpenDate (string) - ISO 8601 format
- AssignmentCloseDate (string) - ISO 8601 format
- CreatorId (string) - guid representing the user that created this assignment
- ClassId (string) - guid representing the class this assignment belongs to
- StatusId (string) - determines if the assignment is active / inactive / etc, there will be an enum of possible values. Please note this is separate from the status determined by open / close date. A teacher should be able to temporarily disable an assignment within the open / close dates… that is what this is for.

(O: Hurst)

Submission Contract

- Id (string) - will be a guid
- Submission Name (string)
- Submission Description (string)
- AssignmentId (string) - a guid representing the assignment this submission is for
- SubmissionPayload (string)
- CreatorId (string) - guid representing the user that created this assignment
- ClassId (string) - guid representing the class this assignment belongs to
- StatusId (string) - determines if the assignment is active / inactive / etc, there will be an enum of possible values.

(O: Hurst)

[This space is intentionally left blank]

Class Contract

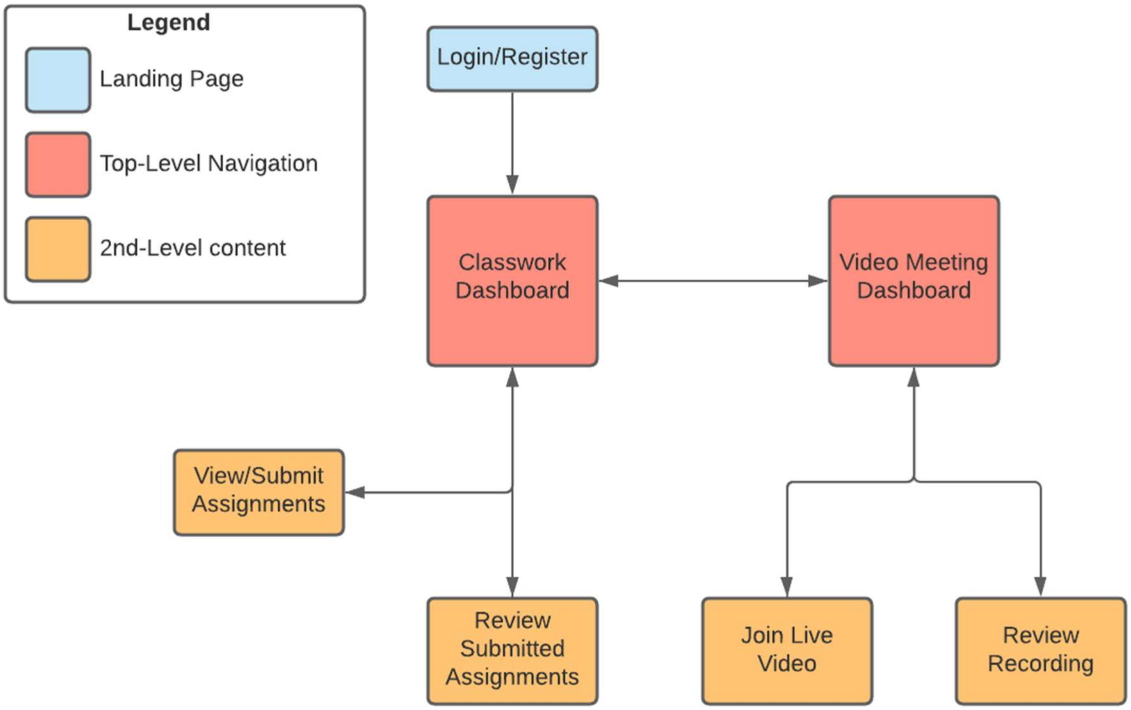
- Id (string) - will be a guid
- Class Name (string)
- Class Description (string)
- TeacherId (string) - a guid representing the teacher for this class (guid is a foreign key to the user id)
- StudentList (array of strings) - an array of strings (guids) representing the user id's of the students in the class
- Meeting Times (array of strings) - an array of strings ("DayOfWeek_ISO8601Time") representing the start time for different days of the week for this class
- ClassSize (int) - the number of students in the class
- ClassStartDate (string) - ISO 8601 format for the date the class starts
- ClassEndDate (string) - ISO 8601 format for the date the class ends
- StatusId (string) - determines if the class is active / inactive / etc, there will be an enum of possible values.

(O: Hurst, M1: Davie)

[This space is intentionally left blank]

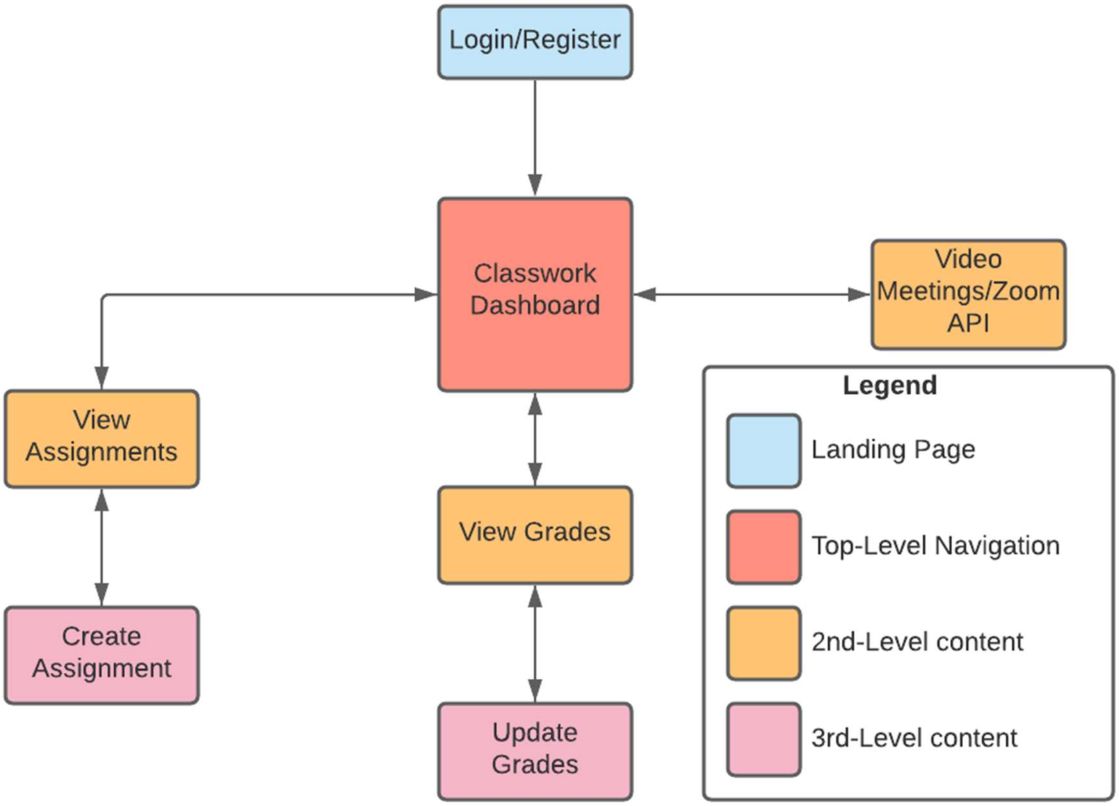
Appendix C: Site Maps

C.1 Student Site Map



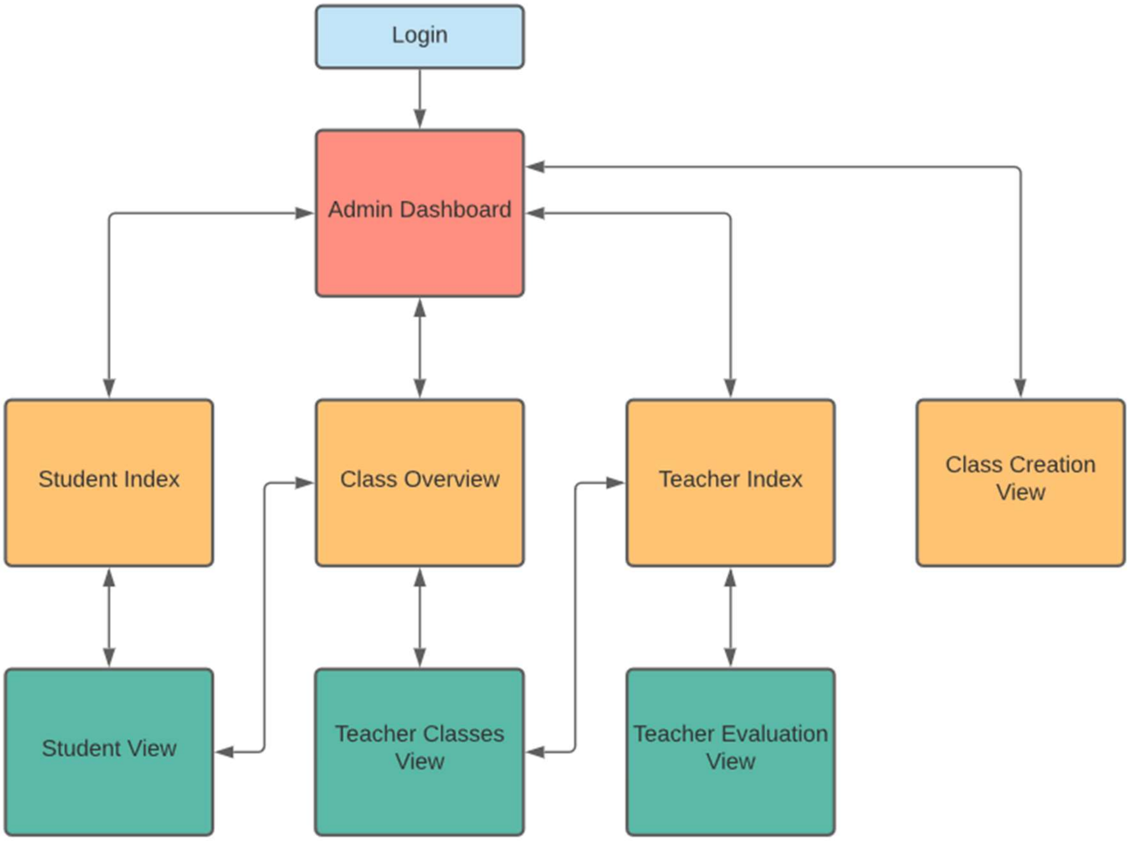
[This space is intentionally left blank]

C.2 Teacher Site Map



[This space is intentionally left blank]

C.3 School Administrator Site Map


Login
Admin Dashboard
Student Index
Class Overview
Teacher Index
Class Creation View
Student View
Teacher Classes View
Teacher Evaluation View