# **D2L (Desire 2 Learn) 3rd Party App**

Class/Section - 4850-02

Fall Semester 2023

Date: 08/31/2023

 Sean Work

 Christopher Dymanus

 Kenneth Molinari

**Software Requirements**  
**Specification**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
3rd Party Project Planning D2L Application  
Version 1.0 approved  
Prepared by Sean Work, Kenneth Molinari, Christopher Dymanus  
KSU  
08/30/2023

Table of Contents

[**D2L (Desire 2 Learn) 3rd Party App** 1](#_Toc150975362)

[Revision History 6](#_Toc150975363)

[1. Introduction 6](#_Toc150975364)

[1.1 Purpose 6](#_Toc150975365)

[1.2 Document Conventions 6](#_Toc150975366)

[1.3 Project Scope 7](#_Toc150975367)

[1.4 References 7](#_Toc150975368)

[2. Overall Description 7](#_Toc150975369)

[2.1 Product Perspective 7](#_Toc150975370)

[2.2 User Classes and Characteristics 9](#_Toc150975371)

[2.3 Operating Environment 9](#_Toc150975372)

[2.4 Design and Implementation Constraints 9](#_Toc150975373)

[2.5 Assumptions and Dependencies 9](#_Toc150975374)

[3. System Features 10](#_Toc150975375)

[3.1 Class Creation/Deletion 10](#_Toc150975376)

[3.1.1 Description 10](#_Toc150975377)

[3.1.2 Stimulus/Response Sequences 10](#_Toc150975378)

[3.1.3 Functional Requirements 11](#_Toc150975379)

[3.2 Login/Create Account 11](#_Toc150975380)

[3.2.1 Description 12](#_Toc150975381)

[3.2.2 Stimulus/Response Sequences 13](#_Toc150975382)

[3.2.3 Functional Requirements 13](#_Toc150975383)

[3.3 Dynamic Home Header 14](#_Toc150975384)

[3.3.1 Description 15](#_Toc150975385)

[3.3.2 Stimulus/Response Sequences 15](#_Toc150975386)

[3.3.3 Functional Requirements 15](#_Toc150975387)

[3.4 Import Projects/ Import Students 16](#_Toc150975388)

[3.4.1 Description 17](#_Toc150975389)

[3.4.2 Description 17](#_Toc150975390)

[3.5 Import/Export 17](#_Toc150975391)

[3.5.1 Description 17](#_Toc150975392)

[3.5.3 Functional Requirements 17](#_Toc150975393)

[3.6 Project Selection 18](#_Toc150975394)

[3.6.1 Description 18](#_Toc150975395)

[3.7 Peer Review 19](#_Toc150975396)

[3.7.1 Description 19](#_Toc150975397)

[3.7.3 Functional Requirements 19](#_Toc150975398)

[3.8 Backend API 19](#_Toc150975399)

[3.8.1 Description 19](#_Toc150975400)

[3.8.2 Stimulus/Response Sequences 20](#_Toc150975401)

[4. Data Requirements 20](#_Toc150975402)

[4.1 Logical Data Model 20](#_Toc150975403)

[4.2 Data Dictionary 20](#_Toc150975404)

[4.3 Reports 22](#_Toc150975405)

[4.4 Data Acquisition, Integrity, Retention, and Disposal 22](#_Toc150975406)

[5. External Interface Requirements 22](#_Toc150975407)

[5.1 User Interfaces 22](#_Toc150975408)

[5.2 Software Interfaces 24](#_Toc150975409)

[6. Quality Attributes 25](#_Toc150975410)

[6.1 Usability 25](#_Toc150975411)

[6.2 Performance 25](#_Toc150975412)

[6.3 Security 25](#_Toc150975413)

[7. Internationalization and 25](#_Toc150975414)

[Localization Requirements 25](#_Toc150975415)

[8. Other Requirements 25](#_Toc150975416)

[Appendix A 26](#_Toc150975417)

# Revision History

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Date | Reason for Change | Version |
| Final | 11/14 | Needs update after project completion | 2.0 |
|  |  |  |  |

# 1. Introduction

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
The Software Requirements Specification (SRS) document has been updated to reflect the pivot from the development of the 3rd Party Project Planning D2L Application (D2LPP) to an independent project management web application. This change comes in response to the prohibitive cost of the D2L API development license. The revised document will outline the new scope, user base, and functional requirements of the self-contained application, now disassociated from D2L API integration but maintaining the core intent of facilitating project planning within an educational environment.

## 1.1 Purpose

This SRS document now specifies the software requirements for the KSU D2L Project Planning Web Application, Version 2.0. It serves as a comprehensive guide for the project team, including software developers, project managers, QA/testing teams, and technical writers, and provides insight for end-users on the application's capabilities and use.

## 1.2 Document Conventions

**Bold**: Indicative of sections/sub-sections.

**Font**: Calibri

**Size**: For sub-headings and general text font size 13.

For main headings font size 16.

3rd Party KSU Project Planning D2L Application appreciation will be (D2LPP).

(tentative) will be used to specify items or features not in the current version.

Faculty END and Student END will be used to simplify long running explanations to further reiterate the user being discussed.

## 1.3 Project Scope

The D2LPP Application aims to deliver a standalone solution for project management within our educational institution, independent of the D2L platform. It will support group project management, customizable surveys, and automated notifications and deadlines without requiring integration with external APIs. This application will provide a streamlined process for faculty and students, aiming to improve upon and eventually replace existing project coordination mechanisms.

## 1.4 References

D2L Platform Official Documentation

KSU IT Policies

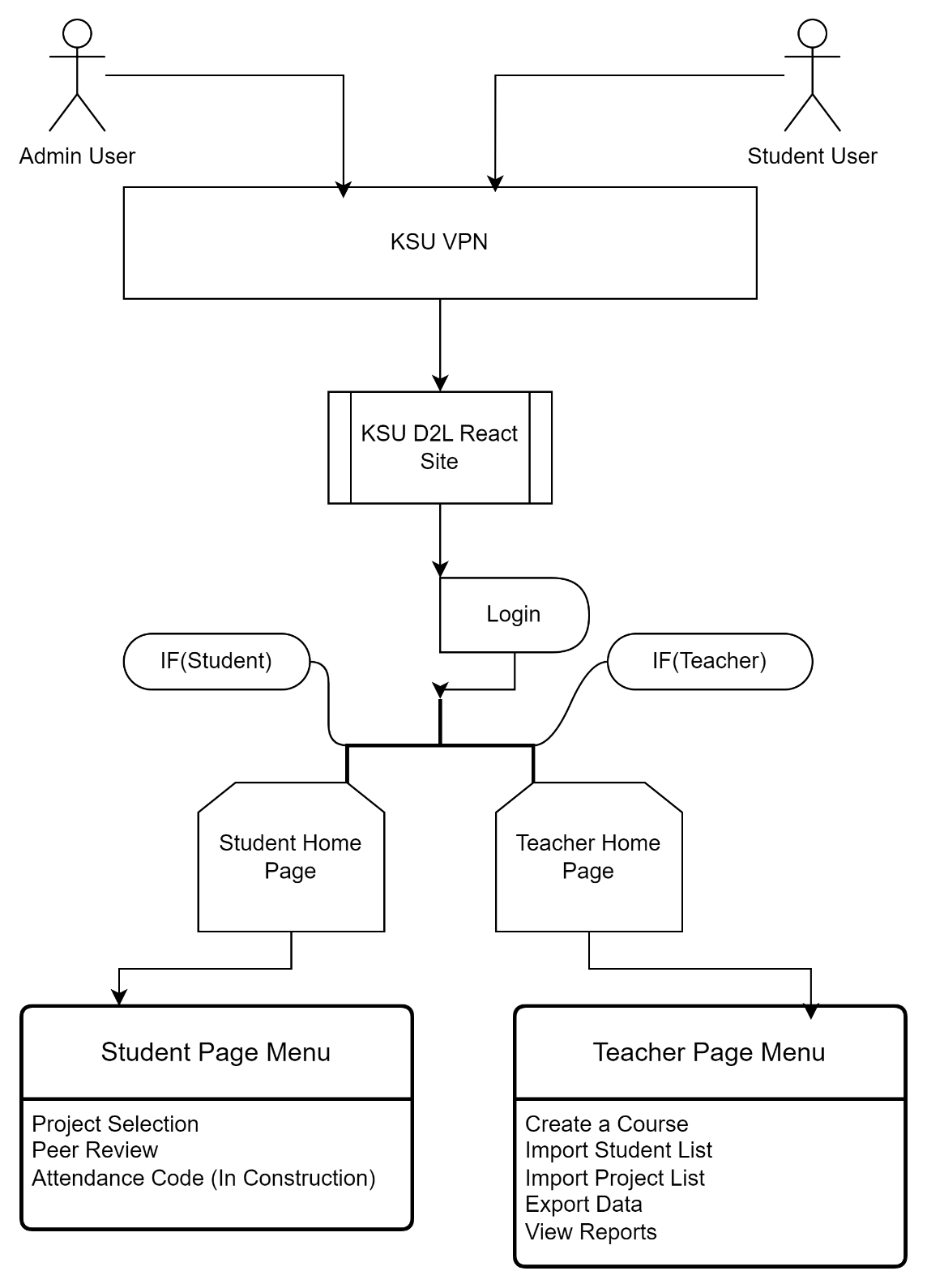
Technical Documentation for Used Technologies (e.g., React, Axios, C#, Azure, MySQL)

# 2. Overall Description

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
The D2LPP Web Application is designed to address and enhance project planning and management processes for educational institutions. Developed independently of the D2L platform, it aims to introduce modern ease of use and functionality that was previously unattainable due to the high cost of API integration and the lack of alternatives in the current D2L marketplace.

## 2.1 Product Perspective

D2LPP is a novel product that operates independently, no longer positioned as an extension to the D2L platform. The product is tailored to meet all user requirements and is intended to function autonomously, providing a comprehensive project planning solution.



*Figure 1: Flow Chart of D2LPP Web Application*

## 2.2 User Classes and Characteristics

D2LPP caters to several user classes:

**Faculty**: The primary users who will utilize the application for creating, managing, and tracking projects efficiently from inception to completion.

**Students**: Engage with the application primarily as respondents to surveys and notifications, contributing data for faculty analysis.

**Administrators**: Users responsible for overseeing the application's maintenance and ensuring its alignment with institutional needs.

## 2.3 Operating Environment

**Hardware**: Compatible with any device capable of accessing modern web applications.

**Software**: Designed to be compatible across all modern web browsers (Chrome, Firefox, Edge, Safari, Opera, etc.).

**Servers**: Hosted on cloud services to ensure scalability, security, and compliance with institutional standards.

## 2.4 Design and Implementation Constraints

* Must adhere to institutional IT policies and best practices for web application development.
* Must be compliant with FERPA and other relevant security regulations and laws.
* User interface and experience must be accessible and intuitive for all targeted user classes.

## 2.5 Assumptions and Dependencies

**Assumptions**

* Faculty and students are familiar with web application interfaces and possess the necessary credentials to authenticate and use IPPWA.

**Dependencies**

* No longer reliant on D2L's API, the application's functionality is self-contained, with a dedicated backend infrastructure.

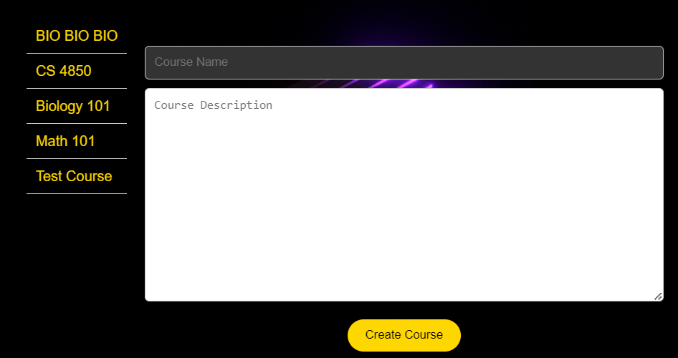
The language for development now includes a stack of React for the frontend and C# for backend services, with data management facilitated by MySQL, and the application's architecture is designed with these technologies in mind.

# 3. System Features

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

D2LPP is designed to provide a suite of independent features that streamline the project management and information collection processes for faculty and students in an educational setting.

## 3.1 Class Creation/Deletion



*Figure 2: Manage Courses section of Admin/Faculty Role*

### 3.1.1 Description

A process that allows for the creation/deletion of Classrooms that will be a bucket for Students and Projects. It will provide the existing classrooms on the left and on the right the ability to create a new project that will then be added to our database and returned to the list.

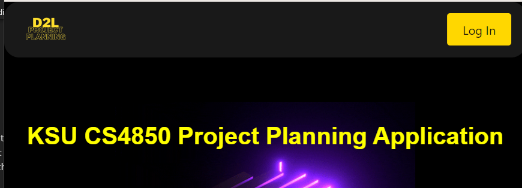
### 3.1.2 Stimulus/Response Sequences

* Faculty logs into the app and navigates to the “Manage Courses” section.
* The app provides a user interface that displays the current classrooms under the teacher and a section to input a course name/ course description with a “Create Course” button.
* Faculty selects one of their classrooms and the option to delete will populate to the right.
* Faculty may select delete to remove the class and all its contents from the database. If so then the website will demand via pop up “Are you sure you want to delete this course?” with a yes/no to finally submit or cancel.
* If the faculty decides to choose nothing for 5 seconds, the delete button will disappear.
* Faculty can create a new classroom under their account by typing any name and any course description.
* Faculty can press “Create Course” which will submit to the database and then the website will refresh the list of Classrooms on the left showing the change.

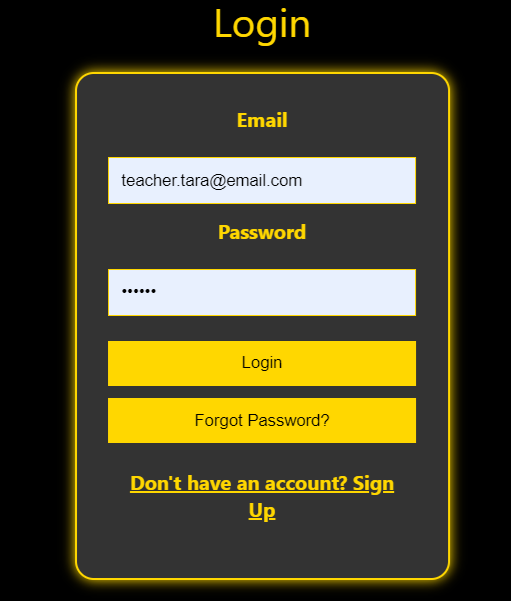
### 3.1.3 Functional Requirements

* The system must provide an interface for faculty to manage (create/delete) classrooms.
* Class creation must include input fields for class name and description, with a validation mechanism to ensure data integrity. (tentative)
* Upon class creation, the system must update the database in real-time and reflect the new class in the list of existing classrooms.
* For class deletion, the system must implement a confirmation mechanism to prevent accidental deletions, including a pop-up with “yes/no” options.
* A timeout feature for the delete action must be present, where the delete option disappears if no action is taken within a specified time frame (e.g., 5 seconds).
* Deletion of a class must remove all associated data (students, projects, etc.) from the database securely and irreversibly.
* The interface must refresh automatically to show changes immediately after any addition or removal of a class.
* The system must handle concurrent class creation and deletion requests in a manner that maintains database integrity.
* Adequate error handling must be in place for database transactions to provide feedback to the user in case of any issues during the class creation or deletion processes. (tentative)
* All actions must comply with institutional data handling and privacy policies. (tentative)

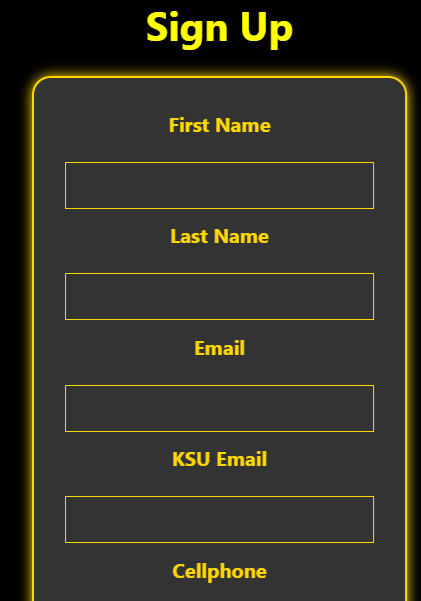
## 3.2 Login/Create Account



*Figure 3: Log in button located on top right of header*



*Figure 4: Login section after Login button is pressed.*



*Figure 5: Signup section after “Don’t have an account? Sign Up”*

## 

## 3.2.1 Description

Will allow new and existing users access to the application through a portal login that checks for email/password integrity prior to logging in. New users are able to create an account and specify their data prior to its addition to the database. All users must be logged in to utilize the D2LPP Application.

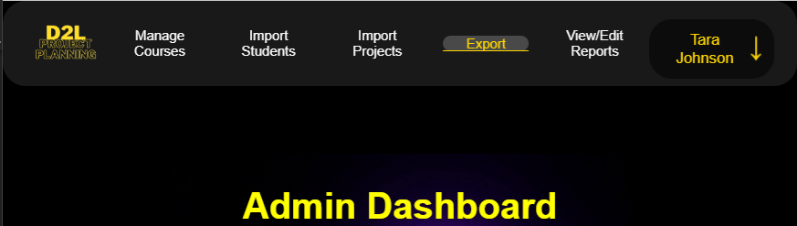
## 3.2.2 Stimulus/Response Sequences

* Faculty presses the Login button on the top right.
* Faculty is brought to a component that asks for an email address and password. Below are a few buttons ( Login – to submit data for validation, Forgot Password – (tentative), Don’t have an account? Sign up – link to the Signup Section )
* For the time being there is only going to be **ONE** user for the faculty that will be both Admin and Faculty user so the signup process is irrelevant until new features are updated to accommodate further users (tentative).
* Faculty will input credentials and press Login to submit.
* The Email and Password is then validated by the back-end database.
* A message will prompt letting the user know whether their login attempt was successful or not. (tentative) Faculty END
* Student presses Login button on top right.
* Student is brought to a component that asks for an email address and password. Below are a few buttons ( Login – to submit data for validation, Forgot Password – (tentative), Don’t have an account? Sign up – link to the Signup Section )
* Student will input credentials and press Login to submit.
* The Email and Password is then validated by the back-end database.
* A message will prompt letting the user know whether their login attempt was successful or not. (tentative)
* If student has no account, they press “Don’t have an account? Sign up”.
* The Signup page will provide a form that will ask for information(First Name, Last Name, Email, KSU Email, Cellphone, Term, Section, Year, Password) and a Signup button below. (tentative)
* The student should be logged in and returned to the homepage with their name displayed on the top right. Student END

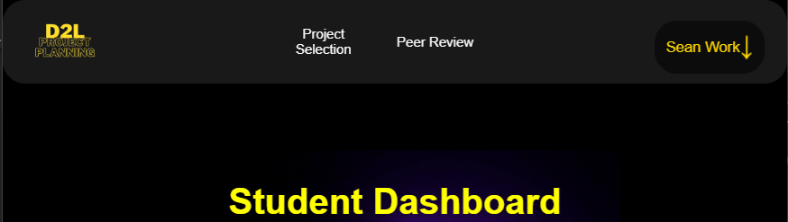
### 3.2.3 Functional Requirements

* ***User Authentication*** – System must allow users to login using their email and password. The system must also validate the credentials against the back-end database.
* ***Account Creation*** – The system must provide a mechanism for new users to create an account. The account creation process should fill out all required information to be used in furtherance of project planning organization.
* ***Input Validation*** – The system must validate input for both login and account creation forms to ensure data integrity and security. Validation will include checking the format of the email and the strength of the password. (tentative)
* ***Error and Success Messaging*** – Upon attempting to login, the system must provide feedback to the user indicating whether the login attempt was unsuccessful. Similarly, feedback should be provided to the user in the account creation process.
* ***Navigation and Accessibility*** – The login button will be visible and easy to see according to the same font/color style as the website itself. After successful login, users should be redirected to the homepage with their first and last name displayed on the top right.
* ***Security Considerations*** – The system must securely handle and store user credentials and should implement measures to prevent unauthorized access.
* ***Password Recovery*** – The system should provide a mechanism for users to recover or reset their password. (tentative)
* ***Account Types and Permissions*** – Initially and for the purposes of the prototype the system will have a single user account that serves as both Admin and Faculty. The system will be scalable to support different user roles and permissions in the future.
* ***User Interface*** – The UI will be user-friendly and intuitive for our end-users.
* ***Scalability and Future Enhancements*** – The system will be designed in such a way to allow for future additions and enhancements to refine the final project.

## 3.3 Dynamic Home Header



*Figure 6: Admin Header with the user on the top right, the logo on the top left, interactive menu buttons for accessibility.*



*Figure 7: Student Header with the user on the top right, the logo on the top left, interactive menu buttons for accessibility.*

### 3.3.1 Description

The header for the website will dynamically adapt to the role of the logged-in user displaying menu items relevant to their access level. On the left-hand side will be the website logo that will return any user regardless of role to the home page. On the right-hand side depending on whether the user is logged in or not will either display the logged in users first and last name or the Log In button. If a user is a Faculty/Teacher/Admin user their header will contain menu items (Manage Courses, Import Students, Import Projects, Export, View/Edit Reports) and if the user is a student their header will contain menu items (Project Selection and Peer Review). All these menu items will update the home screen while maintaining the header above.

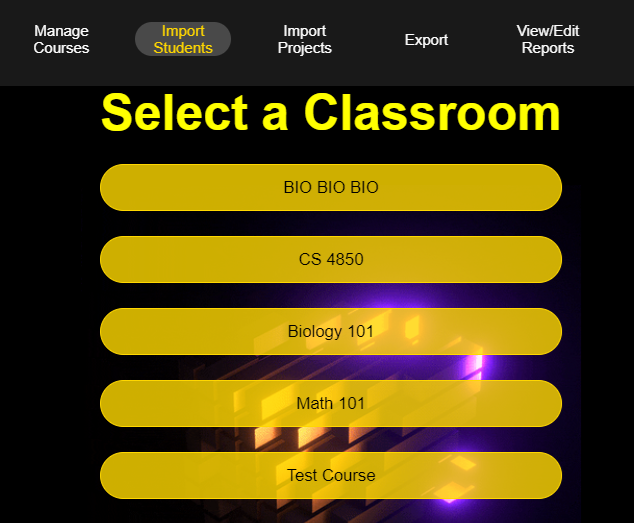
### 3.3.2 Stimulus/Response Sequences

* Logged in user selects menu item.
* The home screen spawns a transparent overlay with the relevant menu item components.
* User can simply press the Logo on the top right to clear the page and return to the main dashboard.
* User selects their name and a dropdown menu populates.
* The options provided are My Account and Log out.
* If the user selects My Account, they are taken to a page that shows all the information the backend database contains with GET requests.
* User will be able to edit the contents of their account and change their information. (tentative)

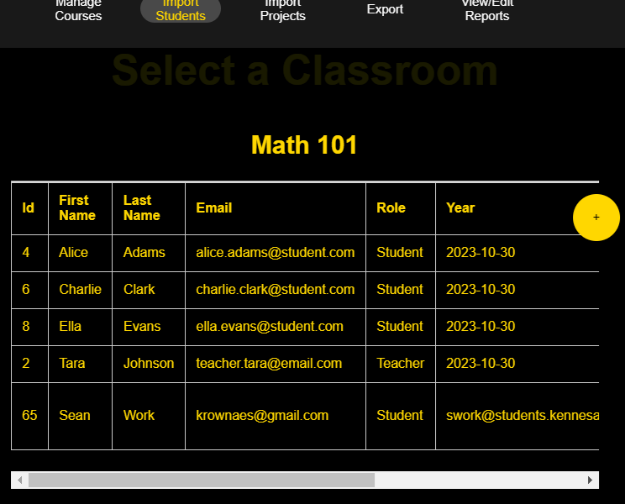
### 3.3.3 Functional Requirements

* ***Dynamic Role-Based Header*** – The system must display a dynamic header that adapts to the user's role (Faculty/Teacher/Admin or Student). The header must show different menu items based on the logged-in user's role.
* ***Header Content*** – For Faculty/Teacher/Admin users, the header should include menu items like Manage Courses, Import Students, Import Projects, Export, View/Edit Reports. For Students, the header should include menu items like Project Selection and Peer Review.
* ***Interactive Menu Buttons*** – The header should contain interactive menu buttons that are accessible and easy to use. Selecting a menu item should update the home screen content accordingly while maintaining the header.
* ***Transparent Overlay for Menu Items*** – When a menu item is selected, the system should display a transparent overlay with the relevant components.
* ***User Dropdown Menu*** – Clicking on the user's name should open a dropdown menu with options like My Account and Log Out.

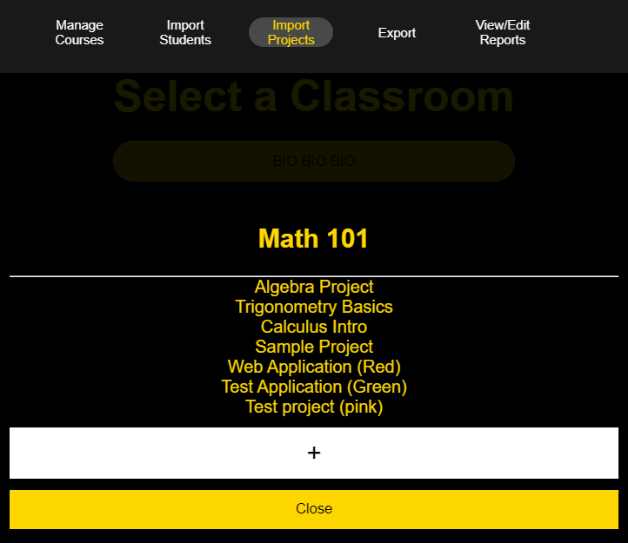
## 3.4 Import Projects/ Import Students



*Figure 8: Admin Import Students classroom select page.*



*Figure 9: Admin Import Students page displaying students within a classroom.*



*Figure 10: Admin Import Projects displaying projects within a classroom.*

### 3.4.1 Description

In the Admin/Faculty section, there are two distinct features this will touch on: Import Projects and Import Students. Both sections initially display a list of classes associated with the logged-in admin, faculty, or teacher user.

***Import Projects***: Allows the user to select a class, view its current projects, and add new projects via a form (project name, description). These additions are updated in the database and made available to students.

***Import Students***: Currently, this section only displays the students enrolled in a selected class, with no further functionality added yet.

### 3.4.2 Description

* Display lists of classes associated with the user.
* Allow adding/deletion of new projects to a class with proper database integration.
* Show enrolled students in a selected class (further functionalities are tentative).

## 3.5 Import/Export

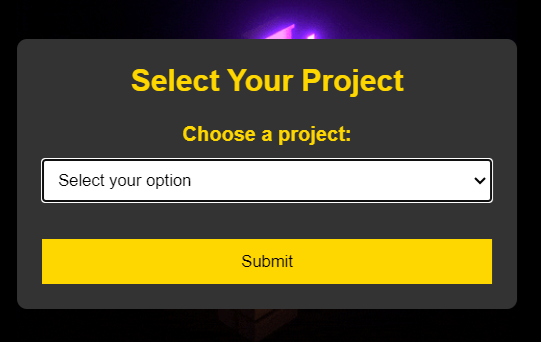
### 3.5.1 Description

The Import/Export function, integrated into various sections of the website, is intended to facilitate mass data imports/exports, including documents, reports, and peer reviews. This data can then be used in external platforms like D2L, Excel and csv files that can be reverted back at anytime. (tentative)

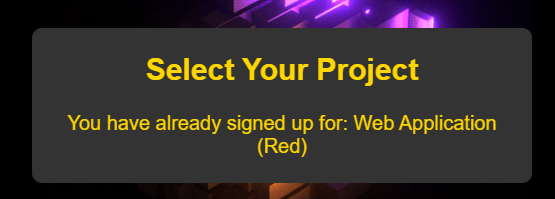
3.5.3 Functional Requirements

* Ability to export various data types in a user-friendly format.
* Integration with external platforms for seamless data transfer.
* The feature is currently non-functional and in the planning phase.

## 3.6 Project Selection



*Figure 11: Student Project selection page displaying projects within a classroom and allowing the user to assign themselves.*



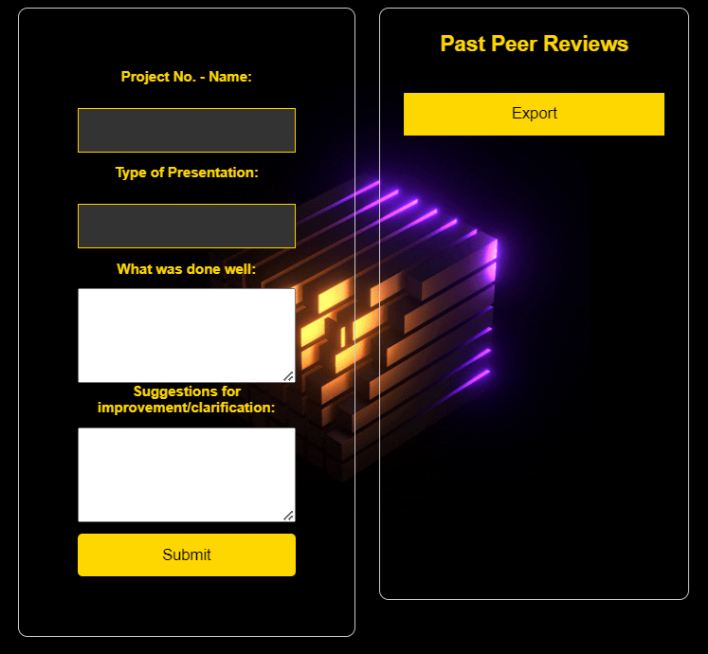
*Figure 12: Student Project selection page displaying successful signup.*

### 3.6.1 Description

Students can choose their projects from a dropdown menu listing all projects loaded into their assigned class by an admin. This selection links the student to the chosen project.  
  
3.6.3 Functional Requirements

* Display a list of projects available for selection to the student.
* Enable students to confirm their project choice.
* Update the admin view to reflect student-project associations.

## 3.7 Peer Review



*Figure 13: Student Project peer review page.*

### 3.7.1 Description

This feature, currently in the front-end development stage, aims to allow users to submit peer reviews. Reviews are linked to both the reviewer and the relevant project

3.7.3 Functional Requirements

* Enable users to select a project and submit a peer review.
* Associate reviews with both the user and the project.
* Export functionality for peer reviews to submit to D2L.
* Phase two plans to implement backend connectivity and functionality.

## 3.8 Backend API

### 3.8.1 Description

The backend API was built using a .net framework that connects to a SQL database, both of which are hosted by Azure.

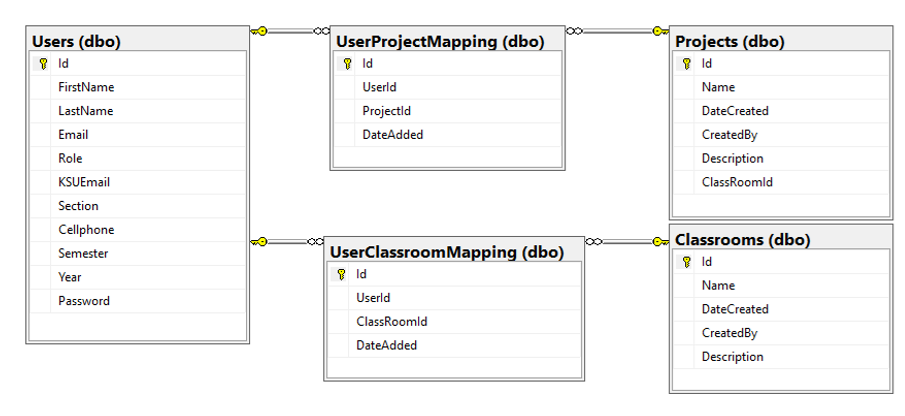
### 3.8.2 Stimulus/Response Sequences

Using IActionResult responses, each API endpoint receives a JSON payload and sends a JSON payload in response.

# 4. Data Requirements

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 4.1 Logical Data Model

****

*Figure 13: Data Model for our backend database utilizing SQL and C#*

## 4.2 Data Dictionary

**Users Table:**

Id: Primary Key, Int

FirstName: VarChar(50)

LastName: VarChar(50)

Email: VarChar(100)  
Role: VarChar(50)  
KSUEmail: VarChar(100)  
Section: VarChar(3)

Cellphone: VarChar(15)

Semester: VarChar(25)

Year: Date

Password: VarChar(25)

**Classrooms Table:**

Id: Primary Key, Int

Name: VarChar(50)

DateCreated: Date

CreatedBy: Int

Description: VarChar(max)

**PeerReviews Table:**

Id: Primary Key, Int

ProjectId: Int

UserId: Int

Positive\_Remark: VarChar(max)

Suggestion\_Remark: VarChar(max)

DateAdded: DateTime

**Projects Table:**

Id: Primary Key, Int

Name: VarChar(50)

DateCreated: Date

CreatedBy: Int

Description: VarChar(max)

ClassRoomId: Int

**UserClassroomMapping Table:**

Id: Primary Key, Int

UserId: Int

ClassRoomId: Int

DateAdded: Date

**UserProjectMapping Table:**

Id: Primary Key, Int

UserId: Int

ProjectId: Int

DateAdded: Date

## 4.3 Reports

The D2LPP Web Application will provide numerous reports that will be organized from our database. These will all be visible and exportable from the View/Edit Reports under the Admin Role. (tentative)

**Project Selection Reports**: Will provide reports on which students in a certain classroom has or has not successfully completed project selection and assigned themselves to a group. (tentative)

**Peer Review Reports**: Will aggregate obtained data from all peer reviews under a specific project and make it visible to the student accounts that are associated with the project that was reviewed. (tentative)

**Peer Review Sheet**: Will aggregate a user’s singular reviews into a single sheet that will be submittable to D2L for their grade. (tentative)

## 4.4 Data Acquisition, Integrity, Retention, and Disposal

Originally, this was intended to function as part of the D2L ecosystem and would follow their security guidelines. However, since this is a stand-alone product that is not attached and is ultimately a prototype we have simply implemented a test database that will be revisited later on to ensure security and data integrity with encryption and FERPA applicable best practices for full release.

# 5. External Interface Requirements

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 5.1 User Interfaces

**General Features Across All Pages**

***Responsive Design***: Ensures compatibility with various devices and screen sizes.

***Dark Color Scheme***: Enhances visibility and distinguishes different UI elements without being too bright.

***Accessible and Interactive Elements:*** Large, responsive buttons; clear and readable fonts; high-contrast color combinations for better visibility.

***Simple Navigation***: Intuitive layout for easy navigation across different pages and functionalities.

**Home Screen**

***Dynamic Content Based on User Role***: The content and options displayed are tailored to the user's role (Admin, Faculty, Student).

***Role-Specific Dashboard Access***:

*Admin/Faculty Dashboard*: Includes options like Manage Courses, Import Projects, Import Students, Export, View/Edit Reports.

*Student Dashboard*: Features Project Selection and Peer Review.

***Overlay Panels for Menu Selections***: Selecting a dashboard item brings up an overlay panel on the home screen with the relevant functionality or information.

**Admin/Faculty Dashboard**

***Manage Courses***: Interactive interface for course management tasks.

***Import Projects/Students***: Tools for importing project and student data.

***Export Features***: Options for exporting reports or data.

***View/Edit Reports***: Interface for accessing and editing various reports.

**Student Dashboard**

***Project Selection***: Interface for students to select and view projects.

***Peer Review***: Tools for conducting and managing peer reviews.

**Common Elements**

***Overlay Mechanism***: Each selection from the dashboard spawns an overlay on the home screen. Users can interact with the features within the overlay without leaving the home screen.

***Home Button Functionality***: A prominent Home button allows for quick return to the main dashboard. Closes the overlay and refreshes the home screen content.

This approach creates a user interface that is both functional and visually engaging, with clear distinctions between different user roles and their respective dashboards. The overlays provide a seamless experience, keeping the user oriented while interacting with various features. The dark-mode, accessible design ensures that the application is user-friendly and easy to navigate.

## 5.2 Software Interfaces

- **Microsoft Visual Studio 2022**: This IDE will be used for developing the front and back-end of the web application. This will be tightly coupled with the .NET 7 Framework.

*Purpose*: Code Development and Debugging

- **Microsoft SQL Management Studio(Latest)**:Will be used as database management tool.

*Purpose*: Database Schema Design and Data management

- **Azure Web Services**:Used to host the web application as well as the project repository.

*Purpose*: Web Hosting, CI/CD, Repository Management

- **.NET 7 Framework**:Will be used for server-side logic.

*Purpose*: Web Application development

- **React Framework**:Will be used for front-end logic and component structure.

*Purpose*: Web Application development

- **Axios**:Will be used for end-point connections.

*Purpose*: Web Application development

- **GitHub**:Will be used for version control.

*Purpose*: Version Control

- **Netlify**:Will be used for hosting the website and passing changes directly from the GitHub repo.

*Purpose*: Web Hosting

5.3 Communications Interfaces

- **Email**:SMTP Protocol for notifications and alerts. (tentative)

*Security*: SSL/TLS encryption

- **Web Browser**:HTTP/HTTPS for web application access. (tentative)

*Security*: SSL for HTTPS

- **Database Connections**:Connection strings via .NET to SQL Server. (tentative)

*Security*: Restricted IP access and String encryption

- **Azure/GitHub Repository**:Git Protocol for Version Control. (tentative)

*Security*: Two-Factor Authentication

- **Survey/Forms**:Built in forms.

*Security*: SSL encryption

6. Quality Attributes

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

## 6.1 Usability

The (D2LPP) web application will be providing a simple UI for Students that will prioritize “minimal clicks” to navigate and a much more verbose menu section for creation, editing and deletion for Faculty/Admin. Decision has been solidified as to make every part of our web application to conform to the (SPA) or Single Page Application methodology. The data collected will be archived and modifiable even after current projects are closed to allow full access regardless of the setting. (tentative)

## 6.2 Performance

There are no expectations for the performance of the web application since it’s very lightweight and not too complex. Depending on the number of users we might have to scale our data storage, but as this is a prototype it is negligible.

## 6.3 Security

Security will be focused in phase 3 when we are closer to a full launch of the D2LPP Application. (tentative)

# 7. Internationalization and

# Localization Requirements

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Not applicable currently as there should be no localization issues yet since we are primarily focused on D2L United States. If that changes this section will be updated.

# 8. Other Requirements

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_  
Not Applicable as the previous sections tend to contain the full scope of requirements.

# Appendix A

D2LPP – Desire 2 Learn Project Planning