Project2Collaborate Documentation

Final Demo Link

Project Development Manual

PHASE 1 (Requirements/Analysis)

- ❏ Brainstorming of project concepts
- ❏ Presentation of project concepts
- ❏ Decision of project concept
- ❏ Gathering of System and Software Requirements
    - ❏ Requirements Document
    - ❏ Planning document

PHASE 2 (Design)

- ❏ Design Documents
    - ❏ Technical Specification Document
        - ❏ UML Diagram
        - ❏ Flow Diagram
    - ❏ Database Diagram
    - ❏ Makeshift Demo
- ❏ Division of roles & project technology final decisions
    - ❏ Tech Stack
    - ❏ Team allocation
    - ❏ Project roadmap (Team goals)
- ❏ Github & Version control standards setup

PHASE 3 (Implementation)

- ❏ Database Setup
- ❏ Software Training
- ❏ Back End Development
- ❏ Front End Development
- ❏ Quality Assurance
- ❏ Software Integration

PHASE 4 (Testing)

- ❏ Debugging of defects
- ❏ Bug reports

- ❏ Documentation
- ❏ Contribution logs
- ❏ Product shipment
- ❏ Maintenance

Requirements Document

1. Introduction

In the course we partake in today, software projects are not very well accommodated for D2L. D2L is a great tool for courses that rely on a homework, quiz, exam assignment manner but is not a credible tool for more intricate collaborative projects. Our intention with D2L being an error-prone platform for software development projects is to create a platform that encompasses how we would like to develop software. There are various tools out there that already exist for project management although not many of them are specifically attributed to the collegiate space.

By taking inspiration from software development practices such as SCRUM, AGILE, Kanban, Github, and various other concepts/tools. We intend to develop a website capable of addressing such a software development oriented course so that the professor and students alike are able to organize their development process.

1.1 Purpose

The purpose of the system is to provide a platform where a team in a given course is able to virtually manage a software project with regards to their software development plan. We intend to enable the software development methodology, the exterior tools the team intends to use, as well as provide functionality that even we as software developers found useful. The purpose of the system is to provide flexible organization of their project while also permitting the professor to check in on the progress.

1.2 Scope/Audience

The scope of P2C involves accommodating the students, professors, and university administrators. Professors will be able to observe the software development process while students will be able to create an environment they will be comfortable developing in. Overall P2C should be able to accommodate any software development framework or methodology for the students while the professor should be able to observe the process while also assigning grades.

1.3 Criteria

The objective and success of the system overall is to have a fully functional system, with the criteria involving the ability to login as a student or faculty, interact with classes, and collaborate on projects. One criteria is that the faculty should have the ability to add or remove students to courses, modify class content, assign deadlines, create submission folders for assigned work, see student project collaboration, and to add grades to a student's submission. While, the students should be able to see the classes they are in, the class content, upcoming deadlines, upload assignments given by the faculty, and have access to a project page to collaborate with fellow students.

The final product should also have:

- Secure data collection and overall site security
- Communication features - comments
- 24/7 Reliability
- Chrome browser/ mobile device support
- Visualization for data usage and display
- Gamification to lessen the "work feel"

2. Current System

As of today, D2L is the current system used in DePaul University to access course content and academic resources posted by faculty. The system is primarily used by the students of DePaul University in order to access documents posted by their professors and to submit assignments to them. The weakness of this system is that it follows the model of faculty uploading and students submitting assignments, which makes it unsuitable for software projects as one cannot collaborate or interact with the project on D2L itself.

Description:

- Faculty and Students can communicate via email and phone
- In order to collaborate, students need to have the know-how of third-party communication methods

Disadvantages:

Faculty:

- Unable to consolidate all course related work into one website
- Professors have to use a third-party website or software to evaluate or view our project work (i.e. GitHub)

Students:

- Unable to consolidate all course related work into one website
- Limited ability to use D2L as a way to work on a project
- Limited ability to communicate with fellow students
- Need to have the know-how of third-party communication methods in order to effectively collaborate

Proposed Solutions:

- The platform should allow students to collaborate on project via a dedicated project tab for each course
- The platform should allow a comment system for projects in order to communicate on-site
- The platform should allow students to create a project-phase based checklist in order to create a timeline for their project

3. Proposed System

3.1 Overview

The functionality of the system is to be an Project-Oriented website specifically for students and professors in college.

The organization of the site will split between students and professors. For students, the site consists of class home pages, course content, project planning page, and a grades page. Professors will have all the same access students have, but in addition they can also manage courses, manage student class enrollment, and submit grades.

Project2Collaborate - Feature by Feature

- ❖ Sign-in Page
  - ➢ Signup
  - ➢ Login
    - ■ Student
    - ■ Admin
- ❖ Class Selection Page
- ❖ Class Pages (note: swp = shared with project page, yellow = useful, green = required)
  - ➢ Course Page
    - ■ Course Content
      - ● Content
      - ● Submissions
      - ● Grades
      - ● Groups
    - ■ Settings (swp)
    - ■ Notifications (swp)
  - ➢ Project Page
    - ■ Software Development Methodology choice
    - ■ Checklist - editable (_Task, _Assigned_Member, _Date)
      - ● Delete task
      - ● Assign member
      - ● Move checkmark

        - Progress Bar - colored and updates with checkmark list completion
        - Text Box - real time interaction between users, visible on given stage of methodology
        - Comment Box - user should be reflected in comment
        - File Storage - upload/download files

3.2 Functional requirements

The student can:

- Manage their account:
    - Login or register
    - Opt-In to responsibilities/agreements
    - Edit their profile
    - Delete their account
- Select which class to navigate to
- View course page
    - See upcoming deadlines
    - Upload assignments due to the faculty
    - Click uploaded content from professor
    - View grades
- View project page (students can edit this)
    - Can edit project page
        - Choose software methodology (only at start)
        - Edit goals in checklist
        - Edit progress bar
        - Can give other team members feedback or rewards
    - Click check marks - completion will fill the progress bar
    - Fill in text boxes - real time updates
    - Comment - users reflected in comments
    - Upload/Download files to the file storage

The faculty can:

- Manage their account:
    - Login or register
    - Opt-In to responsibilities/agreements
    - Edit their profile
    - Delete their account
- Create a class section
- Add or Remove students to and from courses and project groups

- Create project groups
- Add or remove class content
- Assign deadlines
- Create submission folders for assigned work
- Add grades to a student submissions
- See student project page
    - Comment - users reflected in comments

3.2.1 Technical requirements

Student:

- Login or register
    - Store user information in database
    - Confirm true/false credentials with database
    - Opt-in to responsibilities/agreements
- Edit their profile
    - Database, dropdown, buttons
- Select which class to navigate to
    - Text Box, Selection Button, and Database Storage
- View Course Page
    - Interact Button, Text View, List View, Option bar
- See upcoming deadlines
    - Text View, List View
- Upload assignments due to faculty
    - File Storage
- View Project Page
    - Comment
        - User Input Text Field with Send Button
        - Scrollable Text Field for Comments
    - Progress Bar
        - Generated with software development methodology choice
        - GUI with colors dependent on project phase checklist
    - Click Checklist
        - Checkbox, Text View, List View
    - Text box
        - Text view, database
    - Upload/Download files (for project group)
        - File Storage

Faculty:

- Manage their account
    - Store user information in database
    - Confirm if login is true/false with credentials in database
    - Opt-in to responsibilities/agreements
    - Database, dropdown, buttons
- Create a class section
    - Database Storage, Add or Remove Button
- Create project groups
    - Database Storage, Add or Remove Button, List View
- Add or Remove students to and from courses and project groups
    - Database Storage, Add or Remove Button, Text View, List View
- Add or remove class content
    - File Storage, Add or Remove Button, Text View, List View
- Assign deadlines
    - Add or Remove Button, Text View, List View
- Create submission folders for assigned work
    - File Storage, Add or Remove Button, Text View, List View
- See student project page
    - Text View, List View
- Add grades to a student submissions
    - Database Storage, Add or Remove Button, Text View, List View

3.3 Non Functional requirements

3.3.3 Usability

The design and organization is intuitive and easy to pick up for all users. The proposed system is designed for both students and professors to be able to manage all course related content on one website, including projects.

3.3.2 Reliability

All course and project content should be readily available, in addition to having all submitted data onto the website be secure and saved properly.

3.3.3 Performance

User actions provide feedback within seconds and the performance of the website is optimized towards organization of course content and the ability to collaborate on projects directly on the website.

3.3.4 Implementation

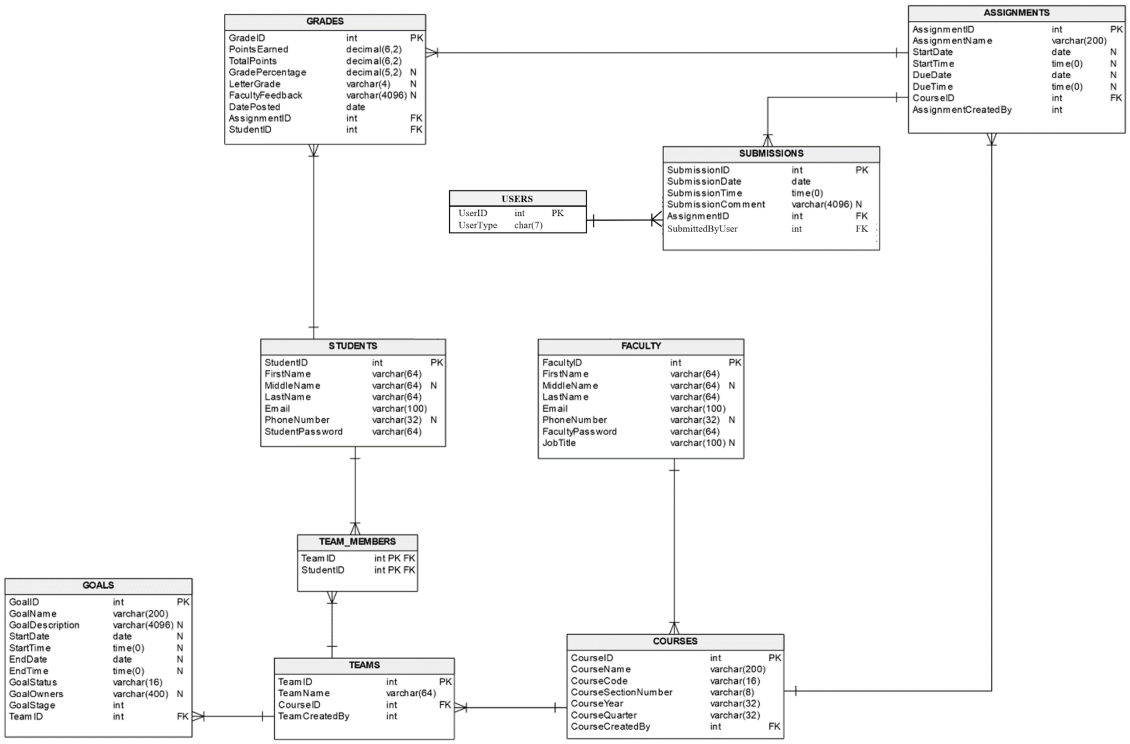
This system is being implemented using JQUERY as the front-end and a Python & MS SQL back end.

3.3.5 Interface

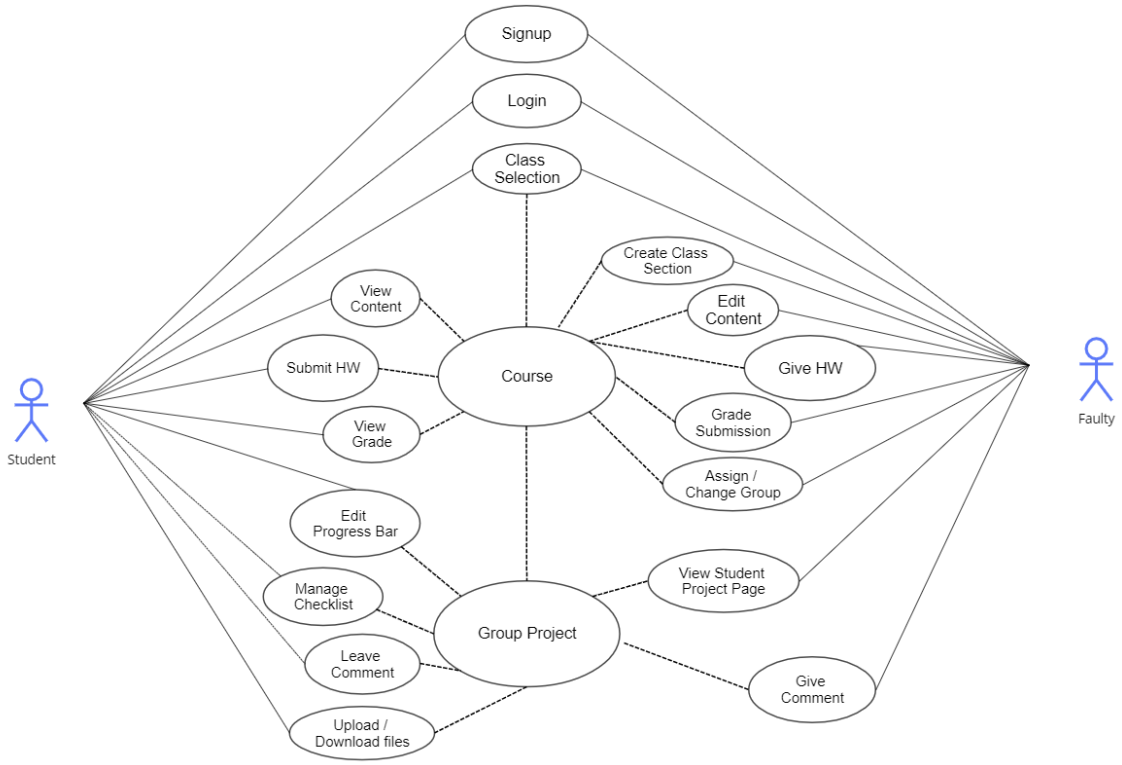
All information and user actions are conveniently placed and clearly labeled.

3.4 System Models

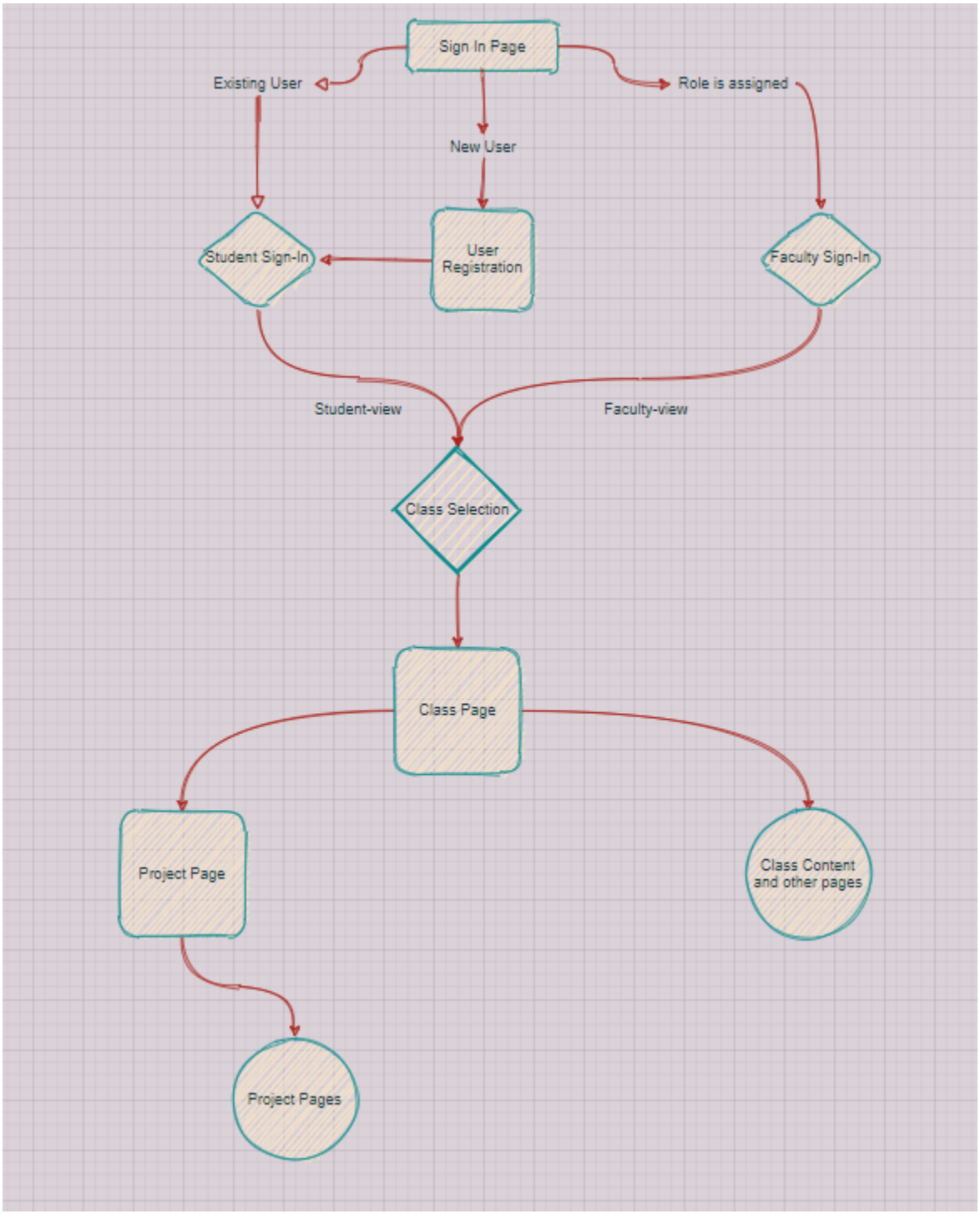
3.4.1 - Database Diagram



3.4.2 - Use case flow chart

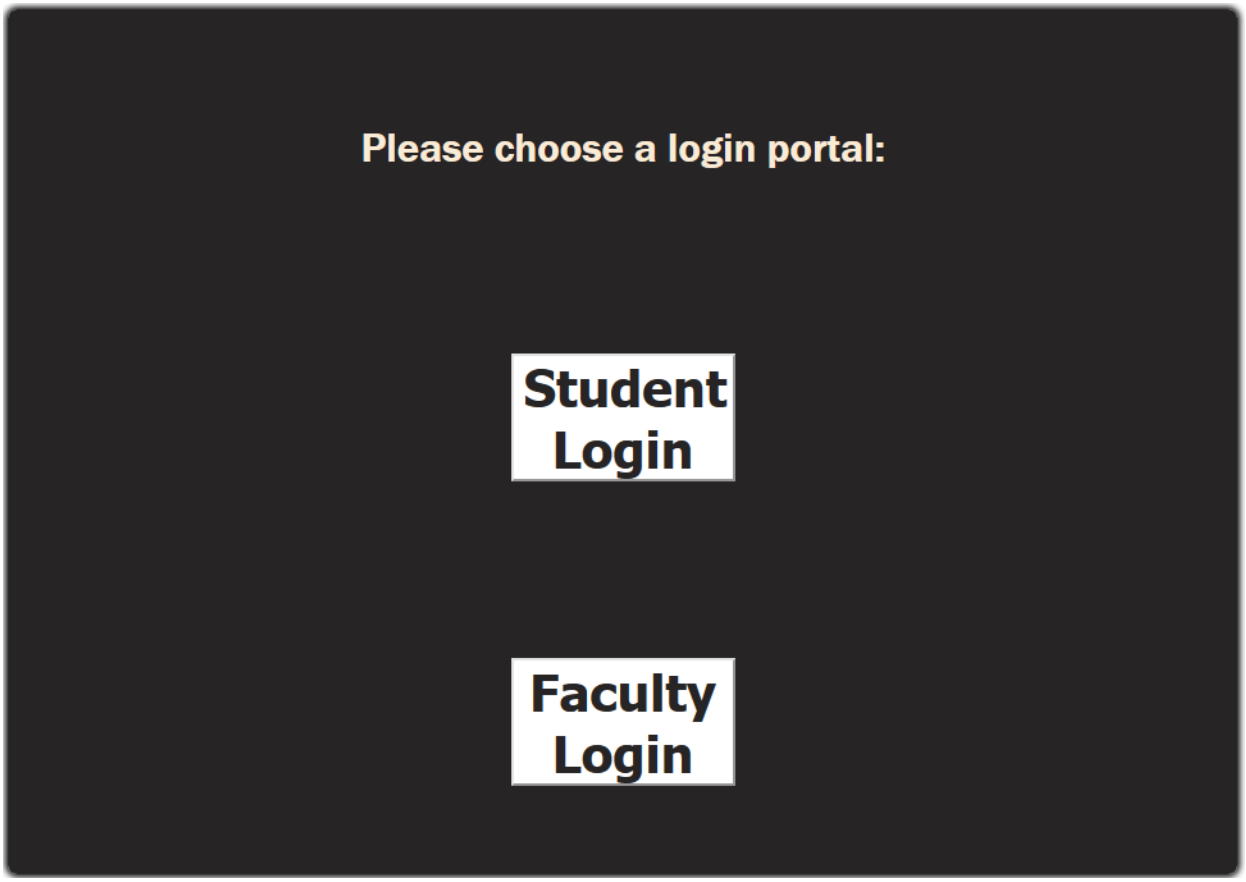


3.4.3 - Flow Chart



3.4.4 User Interface - Actual Website

- The Login Page



Student Login

Email

Password

Forgot your password?

Create Account

Login

Reset Your Password

Email: Email

Current Password: Current Password

New Password: New Password

Submit

Create A Student Account

First Name: First Name

Middle Name: Middle Name

Last Name: Last Name

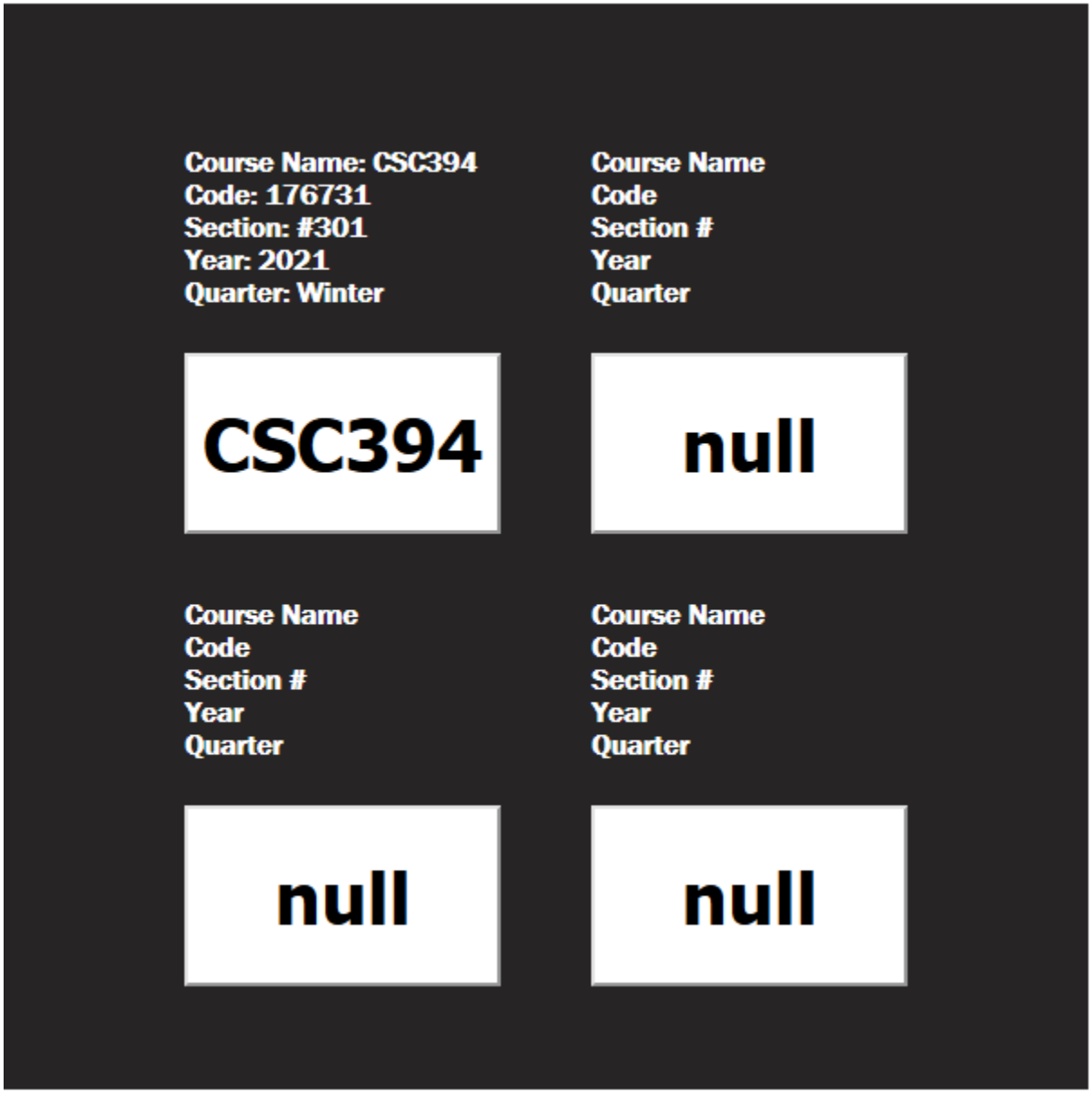
Email: Email Address

Password: Password

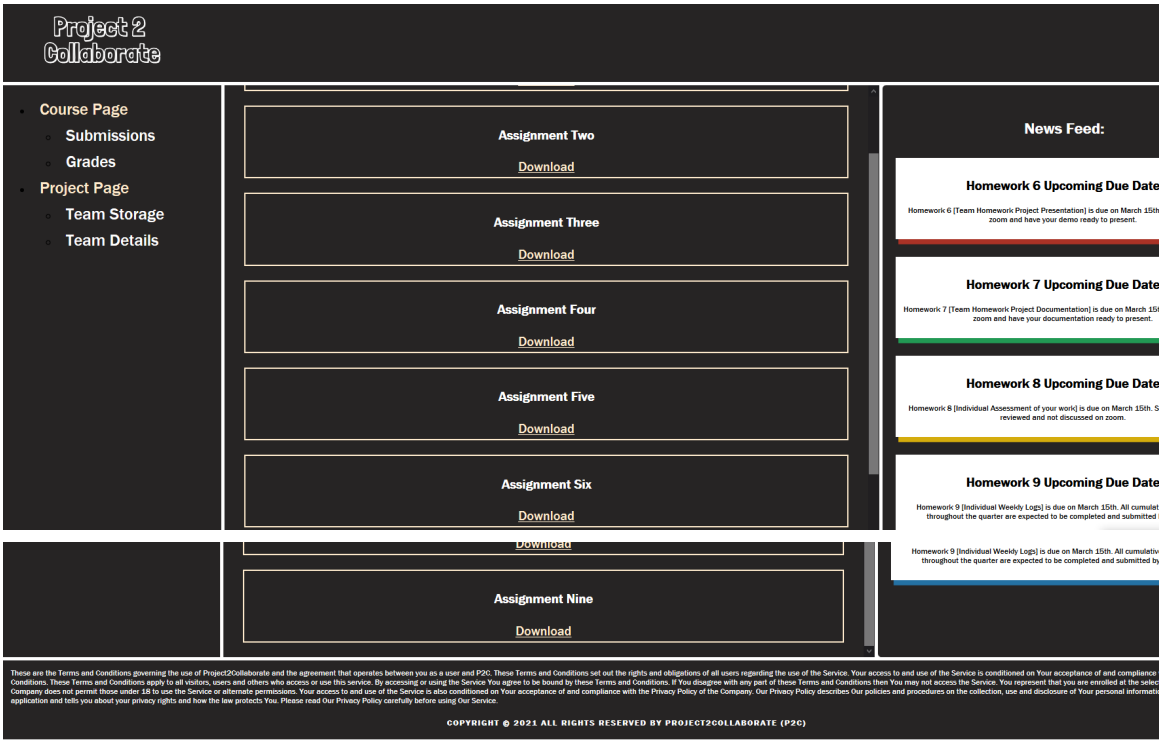
Phone Number: Phone Number

Sign Up!

- The Class Selector Page



- The Class Homepage


Project 2 Collaborate
Course Page
Submissions
Grades
Project Page
Team Storage
Team Details
Assignment Two
Download
Assignment Three
Download
Assignment Four
Download
Assignment Five
Download
Assignment Six
Download
Assignment Nine
Download
News Feed:
Homework 6 Upcoming Due Date
Homework 6 [Team Homework Project Presentation] is due on March 15th
zoom and have your demo ready to present.
Homework 7 Upcoming Due Date
Homework 7 [Team Homework Project Documentation] is due on March 15t
zoom and have your documentation ready to present.
Homework 8 Upcoming Due Date
Homework 8 [Individual Assessment of your work] is due on March 15th. S
reviewed and not discussed on zoom.
Homework 9 Upcoming Due Date
Homework 9 [Individual Weekly Logs] is due on March 15th. All cumulat
throughout the quarter are expected to be completed and submitted
Homework 9 [Individual Weekly Logs] is due on March 15th. All cumulativ
throughout the quarter are expected to be completed and submitted by
These are the Terms and Conditions governing the use of Project2Collaborate and the agreement that operates between you as a user and P2C. These Terms and Conditions set out the rights and obligations of all users regarding the use of the Service. Your access to and use of the Service is conditioned on Your acceptance of and compliance
Conditions. These Terms and Conditions apply to all visitors, users and others who access or use this service. By accessing or using the Service You agree to be bound by these Terms and Conditions. If You disagree with any part of these Terms and Conditions then You may not access the Service. You represent that you are enrolled at the selec
Company does not permit those under 18 to use the Service or alternate permissions. Your access to and use of the Service is also conditioned on Your acceptance of and compliance with the Privacy Policy of the Company. Our Privacy Policy describes Our policies and procedures on the collection, use and disclosure of Your personal informati
application and tells you about your privacy rights and how the law protects You. Please read Our Privacy Policy carefully before using Our Service.
COPYRIGHT © 2021 ALL RIGHTS RESERVED BY PROJECT2COLLABORATE (P2C)

- The Project Homepage

Systems Technical Specs

The current system, D2L, is a great tool for courses that rely on a homework, quiz, exam assignment manner but is not a credible tool for more intricate collaborative projects. Therefore to fix these problems, our purpose system is meant to provide a platform where a team in a given course is able to virtually manage a software project with regards to their software development plan.

Goals our new system would include:

1. The ability for students to collaborate on a project via a dedicated project tab for each course.
2. The ability for a comment system for projects in order to communicate on-site.
3. The ability for students to create a project-phase based checklist in order to create a timeline for their project.

Project Requirements:

1. The students should be able to do all project related tasks on the website.
2. The students shouldn't need to use third-party communication systems to effectively work on projects and should be able to use only website communications for projects effectively.
3. The website should be simple and quick to navigate, i.e. users shouldn't need to go down for more than 2 or 3 levels of tabs to access something.

Out of Scope:

1. There are no video or voice related communication systems implemented
2. There is no ability to purchase course materials on the website
3. Quizzes and Surveys are not available on the website

Open Questions:

1. How will the website update & maintenance function?
2. Does the website support multiple languages?
3. Will there be guides/tutorials for the non-tech savvy?

Approach:

We will organize the site between students and professors. Therefore allowing for dedicated features and functionally dependent on the type the user is. For students, the site consists of class home pages, course content, project planning page, and a grades page. Professors will have all the same access students have, but in addition they can also manage courses, manage student class enrollment, and submit grades. This system allows us to finely control what each user can do on the website and reduce workload by not always having to account for both when adding a new feature.

Other Options Considered:

We considered adding different forms of communications aside from our text-based one, such as adding links to third party services. However, this would go against the idea of a consolidated project website, therefore the idea was scrapped.

Measuring Impact:

We will compare the time users report using the website for managing their project compared to that of D2Ls. Additionally, we compare the amount of files and communications are being used on the website during project times. This will allow us to see if our website encourages more usage on the schools network and improves productivity.

Security and Privacy:
User's username, password, and any related private information will be hosted and protected by the managing school database.

Front-end Team:

1. Write website code
2. Design the website
3. Help implement back-end code

Back-end Team:

1. Write back-end functions
2. Write FTP code
3. Implement back-end code

Design/Architecture

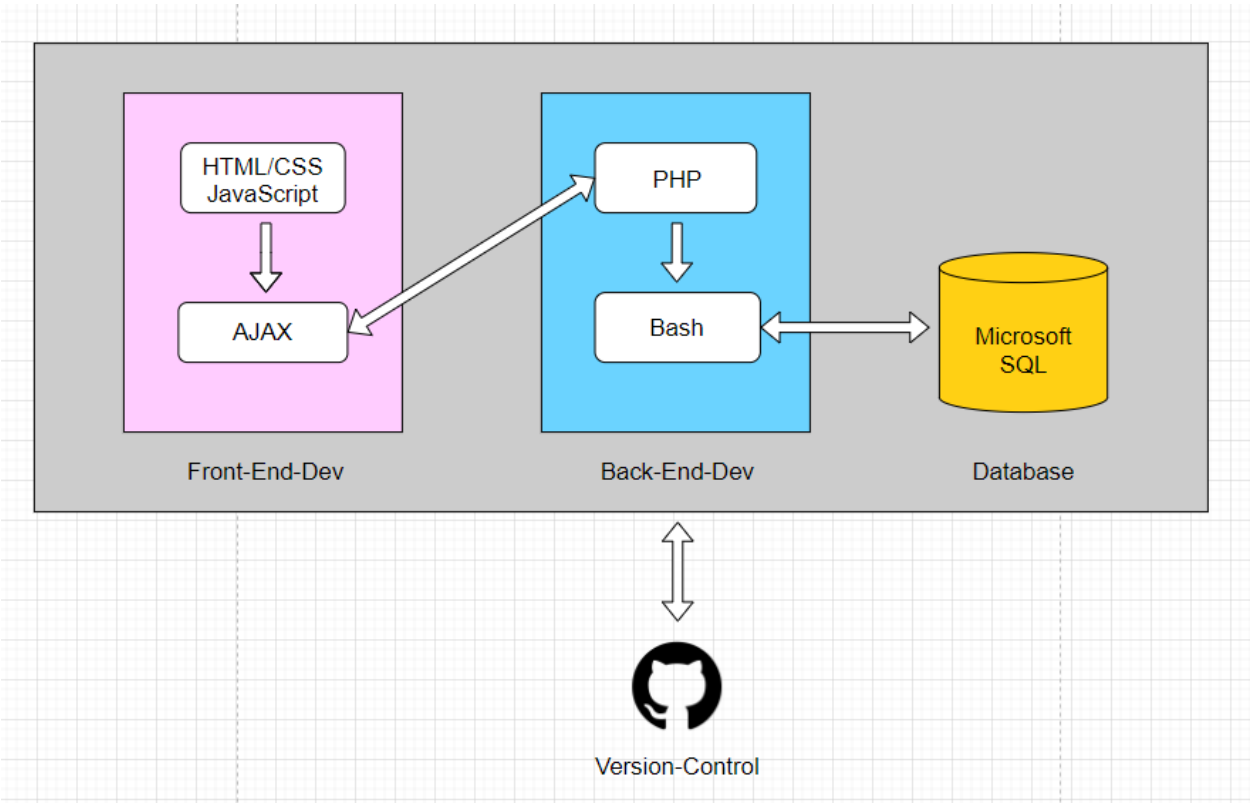
User-View System Model:

Use Cases Document

Hardware Design

We are using a web server, a database server, and an FTP server on a virtual machine. The app will be hosted on the web server. The database will be hosted on the database server. The FTP server will provide the ability to transfer files.

System Hardware/Software Architecture



Software Design

Overview:

Software and integration related aspects to be included here once implementation occurs. This section discusses the Technology Stack and software package integration based on the relations between front-end and back-end.

Technology Stack:

- Backend: PHP / Bash - sqlcmd
    - Bash was chosen for its compatibility with MS SQL. Sqlcmd will be used to connect Bash to the Database, and Ajax/PHP will be used to connect Bash to the front-end.
- Frontend: HTML / CSS / JavaScript / Ajax

    - HTML, CSS, JavaScript, and Ajax are all languages that will be used in creating the GUI and will use text fields to query the Database.
- Configuration: Ajax / PHP
- Database: MS SQL
    - MS SQL queries will be used in Bash functions via sqlcmd to send and receive information between the front-end and the Database.

Software Package Structure:

Website Interaction:

- RegisterQuery
    - (Inserts first name, middle name, last name, email, phone number, and password into either faculty or student table. Faculty must also include job title.)
- LoginQuery
    - (Searches for matching email and password)
- ResetPasswordQuery
    - (Searches for matching email and current password, then allows update)
- UpdateProfileInfoQuery
    - (Edits a user's details.)
- CreateTeamQuery
    - (Insertion of team name into teams table.)
- UpdateTeamQuery
    - (Edits a team's details.)
- DeleteTeamQuery
    - (Removes a team from the teams table.)
- GetTeamQuery
    - (Displays a list of teams. This list can be filtered according to specified conditions.)
- AddStudentToTeamQuery
    - (Inserts a unique combination of team ID and student ID into the "team members" table.)
- RemoveStudentFromTeamQuery
    - (Removes the specified combination of team ID and student ID from the "team members" table.)
- CreateCourseQuery
    - (Inserts course name, course code, course section number, course year, course quarter, and the creator of the course into the courses table.)
- UpdateCourseQuery
    - (Edits a course's details.)
- DeleteCourseQuery
    - (Removes a course from the courses table.)
- GetCourseQuery

    - (Displays a list of courses. This list can be filtered according to specified conditions.)
- CreateGoalQuery
    - (Inserts goal name, goal description, start date and time, end date and time, goal status, goal owners, and goal stage into the goals table.)
- UpdateGoalQuery
    - (Edits a goal's details.)
- DeleteGoalQuery
    - (Removes a goal from the goals table.)
- GetGoalQuery
    - (Displays a list of goals. This list can be filtered according to specified conditions.)
- CreateAssignmentQuery
    - (Inserts assignment ID, assignment name, start date and time, and due date and time into the assignments table.)
- UpdateAssignmentQuery
    - (Edits an assignment's details.)
- DeleteAssignmentQuery
    - (Removes an assignment from the assignments table.)
- GetAssignmentQuery
    - (Displays a list of assignments. This list can be filtered according to specified conditions.)
- CreateGradeQuery
    - (Inserts points earned, total points, grade percentage, letter grade, and faculty feedback into the grades table.)
- UpdateGradeQuery
    - (Edits a grade's details.)
- DeleteGradeQuery
    - (Removes a grade from the grades table.)
- GetGradeQuery
    - (Displays a list of grades. This list can be filtered according to specified conditions.)
- CreateSubmissionQuery
    - (Inserts submission date, submission time, and submission comment into the submissions table.)
- GetSubmissionQuery
    - (Displays a list of submissions. This list can be filtered according to specified conditions.)

Providers:

- HTML Provider
- URL Provider

Database Interaction:

- RegisterNewUser
    - Inserts
- CheckPassword
- ResetPassword
- StudentLogin
- FacultyLogin
- Misc.

Other Module:

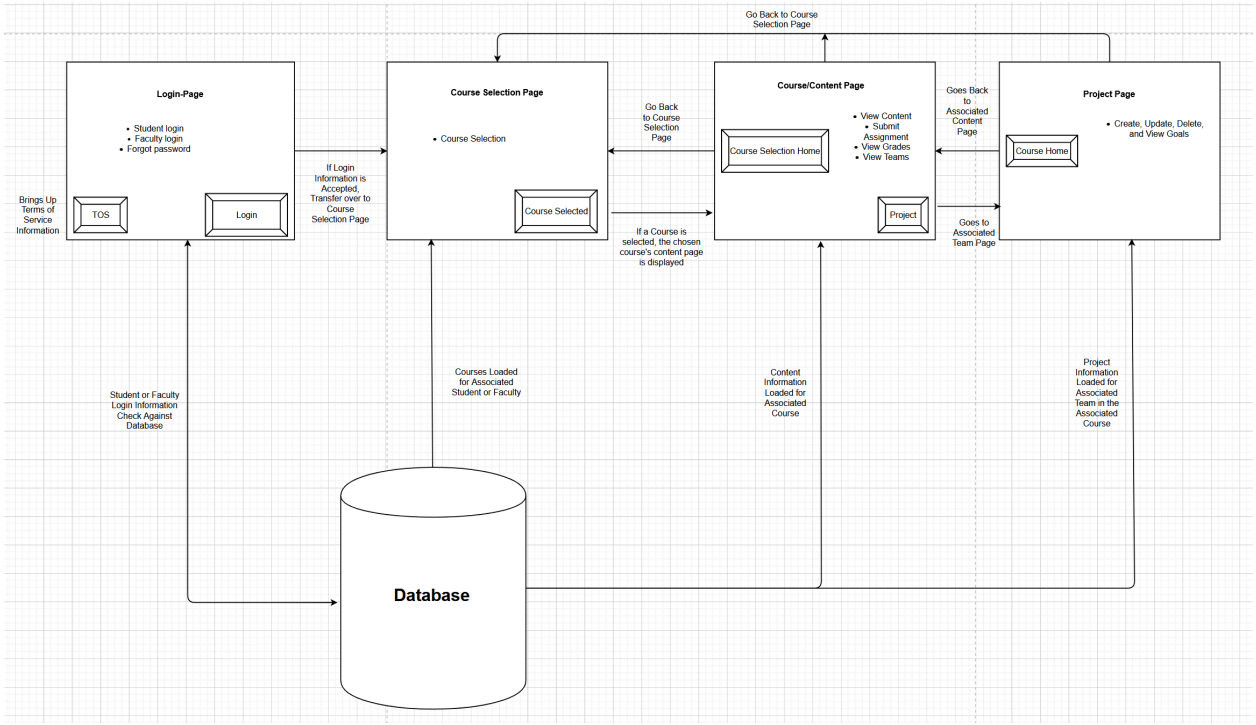
- Chatbox

Software Integration:

HTML/JavaScript → AJAX → PHP → Bash → Database

Based on the input received from the front-end, specific functions will be executed. These functions will be written in JavaScript and they will connect to the back-end using Ajax and PHP. When the JavaScript functions are executed, they will execute Ajax web requests. These web requests will use PHP to send data from the front-end to the back-end. Bash functions will accept the data from PHP and they will use this data to either make changes to the database or read data from the database. Then by using Ajax and PHP, the query results will be sent back to the HTML website.

Wireframe

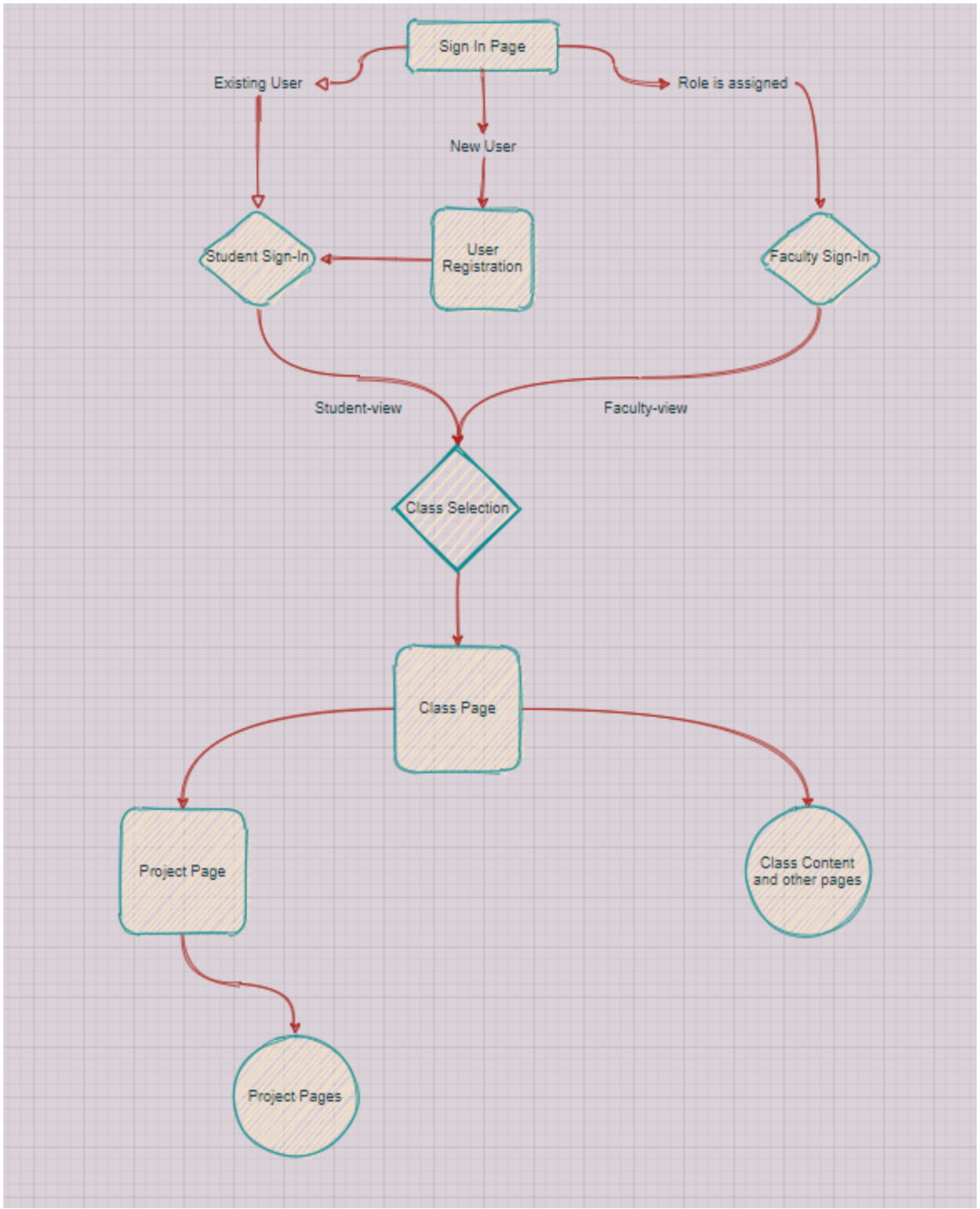


Any Test Design Plan and Results

We didn’t really have test design procedures formally and usually had the most involved people approving aspects of code that we were working on. A finalized test plan would be to replicate going through our project using Project2Collaborate to test the project and team aspects of it, as well as simulating assignments from both faculty and student sides to test the rest of the project's functionality.

Use Cases


Signup
Login
Class Selection
Create Class Section
Edit Content
Give HW
Grade Submission
Assign / Change Group
View Student Project Page
Give Comment
View Content
Submit HW
View Grade
Course
Edit Progress Bar
Manage Checklist
Leave Comment
Upload / Download files
Group Project
Student
Faulty


Sign In Page
Existing User
New User
Role is assigned
Student Sign-In
User Registration
Faculty Sign-In
Student-view
Faculty-view
Class Selection
Class Page
Project Page
Class Content and other pages
Project Pages

Project Code Module List (Version + Development History) and Weekly Team Log

User Manual

Accounts

- New users can create Student and Faculty accounts. When a Student or Faculty account is created, a new user gains access to the application and a specific list of permissions is applied to the user. In addition, students and faculty can modify their account information and delete their accounts.

Courses

- Faculty can create, update, and delete courses. Only the user who created a course has the ability to modify and delete the course and its content.
- Students can view their grades for all assignments in their courses.

Assignments and Submissions

- Faculty can create, update, and delete assignments for all courses they have created.
- All users can submit files to assignments assigned in their courses, and they can add comments to their submissions.

Grades

- Faculty can create, update, and delete grades for any student in their courses.
- Students can see the grades that the Faculty has assigned them, but are unable to modify them in any way.

Teams

- Faculty can create, update, or delete any teams in their courses.
- Students can see the teams that they are assigned to in all of their courses.

Team Collaboration Goals

- Faculty can observe team collaboration by viewing a team's goals.
- Students can create, update, and delete goals in their teams. Only the students assigned to that specific team can create, update, and delete that teams' goals.