Requirements Specification Document - v.4.0

GAMING ED.

CAPSTONE 2022

Grace Hsieh, Brett Lewerke, and Chayson Spigarelli

16 April 2022

Sponsor Signature

Team Lead Signature

Project Sponsor: Terry E Baxter

Faculty Mentor: Dr. Michael Leverington

Table of Contents

1.0 Introduction

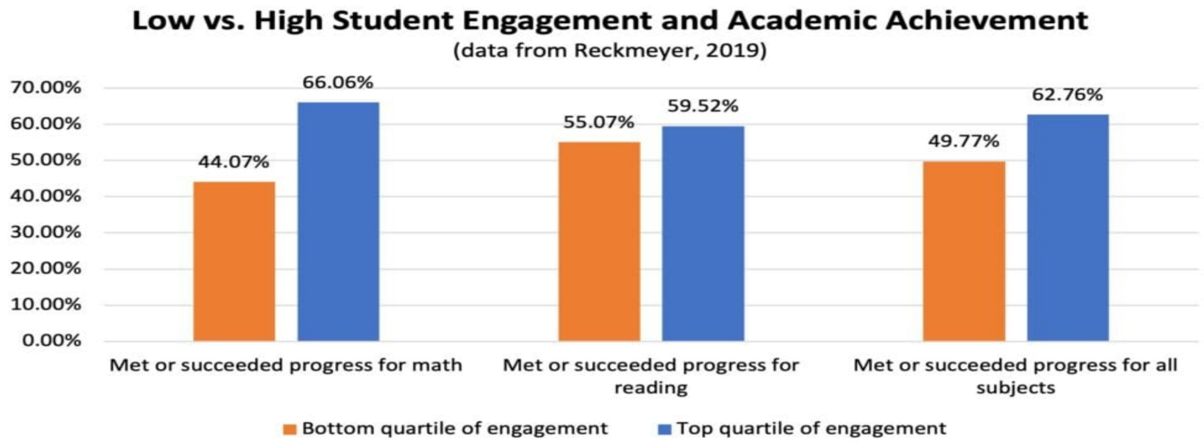
Many students are required to submit work, track course progress, and view grades through an online tool provided by their university, called a learning management system (LMS). A LMS is a piece of software used for the administration, documentation, reporting, automation, and delivery of educational courses, training programs, or learning & development programs. The number of LMS users is estimated to be 73.8 million with the primary participants being corporate-level executives [2]. Most LMS's include a gradebook, calendar, assignments, quizzes, tests, and user account info. LMS's are becoming extremely popular for administrators because they can help keep track of students' grades and progress automatically. A LMS can be a great tool for students so they can work on assignments from home, track upcoming assignments, and view their grades. Blackboard Learning Management System (BbLearn) is the current learning management system at Northern Arizona University (NAU) that both faculty members and students use.

Our client's name is Professor Terry E Baxter who studies Civil and Environmental Engineering (CENE) at NAU. Professor Baxter teaches a course called CENE 150L Environmental Engineering Computations Lab. Professor Baxter wants to make students more engaged in his CENE 150L class so that students earn a higher grade. His initial solution was to gamify the course, making his course feel more like a game by allowing students to earn coins (currency system) through taking quests (exams) as part of a game level (course module), using BbLearn. The in-course currency system made up of coins can be spent on advantages that help students achieve a higher grade in the class. Examples of in-course advantages include: technical notes, redo on exam or assignment, gifting coins to other players, exemption medals, and explorer badges. The team will discuss what these advantages are in further detail in section 4.1.5b. Improving student engagement by adding features to make the course feel more like a game will help students become more successful in the course. Please refer to figure 1.0 below which graphically shows the impact of student engagement on students' grades.

The risks are analyzed in section 5.0 which discuss the possible problems the team could encounter while working on the project down the road. Risks need to be evaluated early on in the project development so that they are not a surprise to the team and the risks can be quickly mitigated.

Figure 1.0 - How Student Engagement Impacts Grades [1]

This figure graphically shows the success level of students based on engagement and subject. The bars on the left represent the bottom quartile of engagement and the bars on the right represent the top quartile of engagement. The percentages that the graph is showing on the left side of the figure represent a student's grade. You can clearly see from the figure that students who are more engaged receive a higher grade in the class.

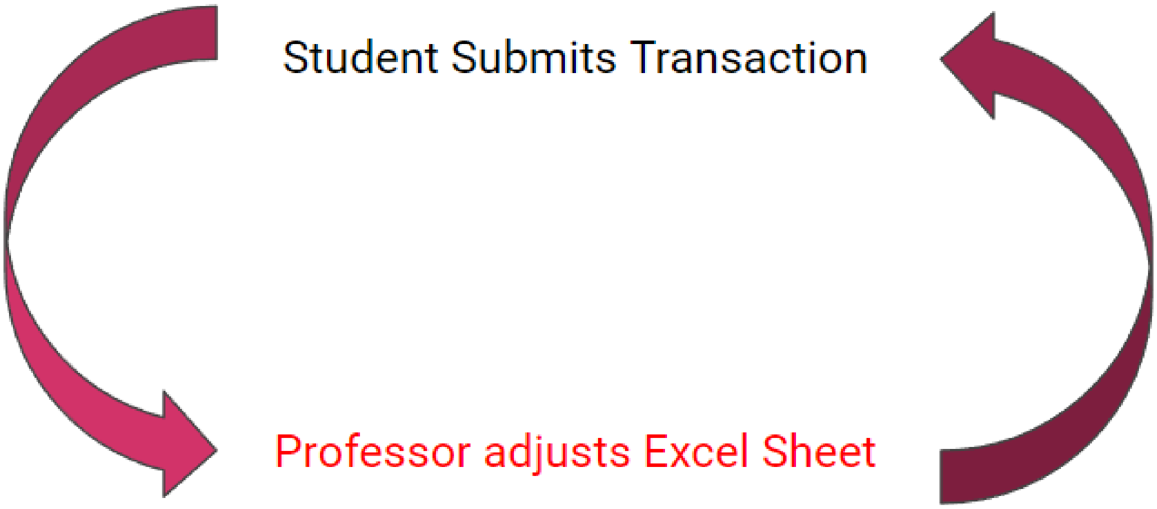


2.0 Problem Statement

The problem our client is currently facing is managing a gamified course. BbLearn currently does not allow for teachers to manage a gamified course without introducing a lot of extra maintenance to keep the course gamified. Currency is one of the features our client introduced in order to gamify his course. Our client keeps track of each student's total currency using an Excel spreadsheet which takes an extensive amount of time to maintain and adjust. For example, if a student wants to trade in their coins, our client has to manually go into the Excel spreadsheet to change the value of a student's coins. Figure 2.0 gives a graphical example of our clients current workflow to manage the in-course currency.

Figure 2.0 - Client's Workflow

This figure shows the process of managing the coins for each student in order to gamify a course. Professor Terry E. Baxter needs to adjust his Excel Spreadsheet every time a student wants to submit a transaction. This is the tedious part that costs our client a lot of extra time to gamify his course.



3.0 Solution Vision

Most learning management systems are not set up in a way that they can be easily gamified. Our proposed solution is to create a new LMS that has the capability of being gamified by introducing a currency system, allowing students to pick their own avatars, and flexibility for the administrator to create levels. The LMS will be divided into two parts: a desktop application and a website application. The desktop application will be where the teacher will create a course, while the website application will be where the students take the course. In order to create a gamified LMS, Team GamingEd has identified key technologies that will help in creating a new LMS such as Unity for the development of the desktop application and .Net Frameworks for the development of the website application. By making the website application and desktop application separate the teacher can use the desktop application to create courses while students are currently taking the course on the website application.

In order to better understand how the team will write the application, team Gaming Ed. has looked into project requirements such as: functional requirements, non-functional requirements, and environmental requirements. The functional requirements discuss the functions of our application. The non-functional (performance) requirements discuss how the program will be expected to perform. The environmental requirements will finally discuss the non-functional requirements related to constraints imposed on the project, usually coming directly from the client.

The client's issue of having to manually deal with the coins will be a thing of the past. Our new LMS will keep track of coins which will be automatically saved through the use of GET and POST requests to our database using PHP and SQL. This will save our client valuable time so that he can be focused on other tasks. Each module will be referred to as a "level". Each assignment will be referred to as a "quest". Each quiz will be referred to as a "challenge quiz". Each exam will be referred to as a "final boss". Each student will be referred to as a "player". These references will all be done in order to make the players feel like they are playing a game instead of taking a course. Lastly the marketplace will be a section on the website where students can spend their earned coins on advantages. These advantages will include technical notes, redo on exam or assignment, exemption medals, and explorer badges. Technical notes are explanations on how to do certain problems in detail. Exemption medals can be bought and used in order to make a question not be graded on the final. Explorer badges are awarded to students who bother to try to go through extra material. Explorer badges will also be shown on students profiles and can be cashed in for coins. By implementing these features Team Gaming Ed. hopes to improve overall student engagement and lower maintenance time for our client.

4.0 Project Requirements

There are three categories of requirements the team has to consider for a successful project. Functional requirements are things the program will do, non-functional requirements are metrics for how well the system will perform, and environmental requirements are imposed or necessary features imposed on the project by an outside source, either client or users.

A design choice across the entire project that the team has made is how the team references its users. Although the primary purpose of the project is for professors and teachers to create classes for students, in functionality the courses and levels do not have to be related to education. Henceforth, the users making the courses will be referred to as administrators and the users who complete the content will be referred to as players.

4.1 Functional requirements

Functional requirements describe what the team promises the system will do. The headers here are what the team believes to be the core of what the project must allow users to do.

4.1.1 Administrator Profile Management

For the following requirements, it is assumed that a teacher has already downloaded the Unity application and opened it. In this section we will review the processes of the super admin adding new teachers to the database to be logged in, what the teachers can do once logged in, and how they can personalize their profile for the website. A super admin is in charge of all other admin and will have the ability to add/delete admins with their courses

4.1.1a Super Administrator Controls

The super admin is the ruler over all other admins. They have unrestricted access over other teachers, and can even view their login credentials. This super admin is crucial because admin's can get fired from their jobs resulting in unused space in our database. This section will go over the roles and responsibilities of the super admin.

- **Super Admin creates/deletes admin:** The super admin is in charge of the creation of new admins and the deletion of an admin from the database. The super admin does not have any restrictions on the deletion of an admin, so a very trusted person will be in charge of this. In the super admin page, there will be a list of all admins (teachers) that are currently using the application. There will be a "plus" sign at the bottom of the page where the super admin can add new admins and a "minus" sign next to their name to be deleted. An extra window will pop up confirming that the super admin wants to delete an admin and they will have to type in their username to delete it. An example deletion prompt would be: “Type ‘Mr. Smith’ to delete ‘Mr. Smith’”.

- **Super Admin can delete any course:** The super admin can also delete any course that a teacher has created. The super admin can view any course to delete it by clicking on an existing teacher's name and the information about them is displayed. The reason for this is that if an admin is deleted, there will be junk data about a course inside of our database that is taking up unnecessary room. We cannot rely on the fact that if an admin gets fired, they will delete their course information. In the super admin page, they will have to click on the teachers name in order to delete a course. Once the super admin has selected one of the administrators a window will open, displaying a list of all courses that admin is hosting. There will be a “minus” button next to the course name to delete the course. An extra page confirming the deletion will pop up including the name of the course and the super admin will be prompted to type in the name of the course to delete it. An example deletion prompt would be: “Type ‘CS476’ to delete the ‘CS476’ course.

- **New Administrator Creation:** When a new admin is created by the super admin, they will give them a temporary username and password. It is then up to the admin to change this immediately once they log in for the first time.

4.1.1b Administrator Login

The administrator (teacher) will have to change their login information once their new account is created. This step is assuming that an admin has downloaded the Unity Application and has opened it. They will be on the main login page once the application is opened.

- **Admin will log in with username and password:** Once the super admin has created this admin, they will log in with their temporary username and password. If the login information is not correct they will not be able to go to another screen and will be prompted to try again.

- **Admin will then change username and password:** After being logged in with their temporary username and password, the admin will then be prompted to change their credentials to a new username and password.

4.1.1c Personalizing Administrator Profile

Once logged in to the Unity Application, an admin will be able to start using the application. They will be given the option to add a new course, view the existing courses, delete a course, or customize their admin profile to be displayed to the students. They will be taken to their profile page once logged in and from there they will be able to add a new course. There will be a screen showing the existing courses that they have created and if none exist a "add new course" button will be the only information available.

- **Add a new course:** Once a teacher has logged in and their profile page is being displayed to them, they will see the existing courses that exist. If none exist then there will just be a button called "add new courses" in the box where they should be displayed. Once this button is clicked they will be taken to the course creation page (see 4.1.2)

- **View existing courses to be edited:** From their profile page, a teacher will be able to see the courses that they have already created. This will be a box in the center-top of their screen. If no courses exist then no courses will be displayed to them. They can then edit the course by clicking on one of their existing courses (see 4.1.3)

- **Delete course:** The admin will be given the option to delete a course. This would happen if the school has reassigned them courses to teach or they have lost their job as a teacher. In their main profile view, there will be a "minus" button on the right side of the existing courses. If no courses exist this will not be displayed. The teacher can click on the button to delete the course. A confirmation page will appear prompting them to type the name of the course in order to delete it.

- **Change user information:** Inside of the admin profile page, the admin can change their profile picture and display name to be displayed to the students. These names would pop up in a discussion board or any posts made available to the student. The avatar that the teacher can choose will be respective of the ones they put into the marketplace. This means that a teacher will not be able to change the profile picture until after a course has been created with a marketplace.

4.1.2 Creating Course Content

This part is assuming that a teacher has already logged into the application and clicked on creating a course. This section will break down the different levels that can be created in a course including all of the possible values that can be inputted. A "level" can be easily understood as a unit, section, or chapter in a course. It can have different criteria for unlocking the next level for the students to be able to view. It will also describe the functionality of the marketplace, if it will be included in the course at all, and the different types of ways coins can be spent/earned. A teacher user will be automatically created when making the course. The teacher can access the website like a student and create discussion board posts.

4.1.2a Creating the Levels

The admin will need the ability to actually create the levels for the students. The main course creation page is the largest part of the application. There will be a main course timeline where the admin can drag and drop icons into the line. The order in which the icons are placed does not reflect the order in which the box of blank icons are located.

- **Allows for the creation of a course with different attributes:** A teacher can use the course creation screen to start building the course. Admins will have a box containing icons located on a certain side of the screen representing different parts of course. They will be able to click and drag these icons into a major timeline line location in the middle of the screen. These icons will include **Tests/Quizzes**, **Lessons**, **Assignments**, and **Levels.** In order to add information to these icons, the teacher will double click on the icons and add information into them. The information will include things like the name, start date, and unlock criteria. The teacher also has the opportunity to link marketplace items to these icons

- **Allows for the admin to save the course:** The admin will be able to build the course while not being connected to the internet. With Unity being a desktop application, it allows the admin to download the application and use it offline. However, an internet connection is required to submit the completed course.

4.1.2b Creating the Marketplace

When the teacher first opens the course creation page, they will be prompted to add a marketplace. The marketplace can be used by the admin in order to make the course more engaging for their students. If an admin wants to add a marketplace after the course is completed, they can do this through the edit course page. The admin can link marketplace items to certain icons and/or have them available to students to purchase other things (etc. avatars).

- **Admin clicks on the option to have a marketplace:** Before entering the main course creation screen, the admin will have the option to add a marketplace feature to the course. Inside the course creation screen, there will be a side icon indicating the marketplace. The teacher can click on this and add the information for the marketplace. Some examples of marketplace items would include: option to add image for another avatar, option to add hints/cheats pertaining to a certain quiz/exam, option to add calculator/conversions for questions. The marketplace will have some default avatar’s set and the teacher can add more if they want.

- **Admin links marketplace item to icon:** The admin will have the option to drag one of the marketplace items to a specific icon. If the icon has a marketplace item attached to it it will have a marketplace icon right next to it.

- **Admin views marketplace items:** The admin will have the option to view all marketplace items that have been created. The marketplace icon will be in the top left corner or the course creation screen. Once clicked on, a screen will display each marketplace item, the amount of coins that it costs, and if it is linked to a specific icon. If a marketplace item is not linked to an icon it means that it can be purchased by the student at any time from the student's home screen. An example of an unlinked marketplace item would be a profile picture.

4.1.3 Managing Existing Course Content

For the following requirements, it will be assumed that a complete course already exists. This completed course will be accessible from a list of all completed courses owned by the administrator. According to the specifications and requests of the sponsor, the team assumes that an entire course from the administrator perspective at its core will contain levels, a marketplace, a roster of players taking the course, and a correlating table with the grades of each player. Any changes to the levels or marketplace must be uploaded by clicking a confirmation button. Even if no changes are made, a prompt that says "Any changes made will be lost. Continue?" if the administrator attempts to close the level. The options in this menu will be "Save and Close", " Close", and "Return". Any changes to players or player grades will save after a change is detected.

4.1.3a Editing Levels in the Course

To reiterate from 4.1.2a, levels are described as a series of stages that the player must clear in order to complete the course. Using the structure of a highschool textbook for reference, each level would be one chapter in the book and each level would most likely contain information about the chapter and practice questions pertaining to that material. This section will cover adding levels, deleting levels, rearranging levels, and modifying the contents in the levels.

- **Adding Levels:** From the list of completed courses, a button to view the course will display the current layout of the levels in the order determined by the administrator. The menu for editing will be the same as the screen described in 4.1.2a. On the side of the screen, the administrator will drag the icon that correlates to new levels into the main timeline which will understand, based on the position of the mouse holding the icon, which side of a level to create a new level on. There will be a visual indicator showing that the attempt is being made to put a level where the mouse is holding the icon to be determined later in development.

- **Deleting Levels:** The administrator will grab the icon of the existing level with the mouse and drag it into a red box with a trash can symbol over it. A confirmation box asking if the administrator really meant to delete this level will appear to prevent accidents from occuring.

- **Moving Levels:** It has been established in the previous two sections that it is possible to drag the level icons on the screen. In order to rearrange levels, the administrator needs to drag the desired level's icon to an area between two other levels. There will be a visual indicator showing that the attempt is being made to put a level where the mouse is holding the icon to be determined later in development.

- **Adding Content:** Tests, assignments, and learning material can be added with the same steps as seen in 4.1.2a.

- **Moving Content:** There will be an indicator on the left hand side of any object in the level that will allow the object to be moved up and down on the list by dragging the indicator. The program will know to rearrange the content based on the position of the mouse holding the object.

- **Editing Tests:** After clicking on a level, any exams in the level will be displayed with icons and the title of the exam. On the line that displays the exam, there will be two buttons: "Edit" and "Delete". The administrator will be able to click a button that says "Edit" on it which will bring up the same screen that is used for initially creating the test except all of the current information, listed in 4.1.2a, will be filled in instead of default values. All values can be changed in this menu. Going back to the level display, the red "Delete" button will bring up a confirmation screen to avoid accidents. It will then remove the test from the level overview.

- **Editing Assignments:** After clicking on a level, any assignments in the level will be displayed with icons and the title of the exam. On the line that displays the assignment, there will be two buttons: "Edit" and "Delete". The administrator will be able to click a button that says "Edit" on it which will bring up the same screen that is used for initially creating the assignment except all of the current information, listed in 4.1.2a, will be filled in instead of default values. All values can be changed in this menu. Going back to the level display, the red "Delete" button will bring up a confirmation screen to avoid accidents. It will then remove the assignment from the level overview.

- **Editing Learning Material:** After clicking on a level, any resource in the level will be displayed with icons and the title of the resource. On the line that displays the resource, there will be two buttons: “Edit” and “Delete”. The administrator will be able to click a button that says “Edit” on it which will bring up the same screen that is used for initially creating the resource except all of the current information, listed in 4.1.2a, will be filled in instead of default values. The menu layout will vary based on the type of resource. It will not be possible to change the type of learning material from this screen. The administrator must delete the undesired resource and drag and drop the appropriate type from the level display screen. Going back to the level display, the red “Delete” button will bring up a confirmation screen to avoid accidents. It will then remove the material from the level overview.

4.1.3b Editing Marketplace Functions

Each marketplace will have some number of possible purchases based on the administrator’s preference. Similar to 4.1.2b, new sections can be added to the marketplace. The team will start with the client’s desired core functionality, listed in 4.1.2b, and add more possible actions in the future when possible. It should be noted that not all currency transactions in a course are found in the marketplace. For example, some transactions may be found in levels to buy new quiz attempts or to exempt certain questions. Even if no changes are made, a prompt that says “Any changes made will be lost. Continue?” if the administrator attempts to close the marketplace. The options in this menu will be “Save and Close”, “ Close”, and “Return”. This section will cover adding, removing, editing, and rearranging parts of the marketplace.

- **Adding Functions:** A button on the course will open the marketplace view for the administrator. From a menu on the right side of the screen, it will be possible to drag and drop icons that relate to different commands into the existing list of commands. Based on the position of the mouse holding the icon, the program will know where to put the new action and place it accordingly. There will be a visual indicator showing that the attempt is being made to put a level where the mouse is holding the icon to be determined later in development.

- **Removing Functions:** The administrator will grab the icon of the existing function with the mouse and drag it into a red box with a trash can symbol over it. A confirmation box asking if the administrator really meant to delete this object will appear to prevent accidents from occuring.

- **Moving Functions:** It has been established in the previous two sections that it is possible to drag the marketplace icons on the screen. In order to rearrange levels, the administrator needs to drag the desired module to an area between two other modules. There will be a visual indicator showing that the attempt is being made to put a module where the mouse is holding the icon to be determined later in development.

- **Modifying Functions:** Any module in the marketplace will be displayed with icons and the title of the action. On the line that displays the action, there will be two buttons: “Edit” and “Delete”. The administrator will be able to click a button that says “Edit” on it which will bring up the same screen that is used for initially creating the action except all of the current information, listed in 4.1.2b, will be filled in instead of default values. The menu layout will vary based on the type of action. It will not be possible to change the type of action from this screen. The administrator must delete the undesired module and drag and drop the appropriate type from the marketplace display screen. Going back to the marketplace display, the red “Delete” button will bring up a confirmation screen to avoid accidents. It will then remove the action from the marketplace overview.

4.1.3c Grading Completed Course Content

A choice between manual grading and automated grading will be selected by the administrator when creating questions for tests and assignments that do not require uploading a file, such as a virtual document or video. Questions will be categorized into multiple choice, multiple select, text answers, and number answers.

- **Manual grading:** Administrators can view completed assignments in a grading table accessible in the course viewing screen. They can then open an assignment that is associated with a player to judge each question and submit a point value into a text box next to the answer. The value should be a number associated with the correctness of the answer given by the player.

- **Automated Grading:** If the administrator chose automated grading while creating questions, then the answer needs to be recorded for future use. For example, if the question is multiple choice then there would be bubbles next to each answer that will fill in when the administrator clicks them. The program would then record the filled in bubbles to be used when it grades later.

4.1.3d Modifying Existing Grades

When viewing a course, there will be a button that allows the administrator to view the current grades and roster of the course in a table. The first row will have text saying what is in that column. The first column will be a player’s real name, which the administrator can click to access and view that player’s profile which includes their chosen display name, profile picture, and any provided information by the player. The columns after will include all graded material in order of due date. The last column will be the player's total weighted score. Other methods of sorting the table will be discussed at a later time to evaluate plausibility.

4.1.3e Exporting Player Grades

The sponsor has requested that the grades be accessible outside of the course for documentation reasons. As mentioned in 4.1.3c, the grades are currently being stored in a table. Several methods of printing will be available such as excel, PDF, txt, and others. Windows explorer will be prompted to use built in save and renaming functionality to allow the administrator to save the data they need.

4.1.3f Opening & Closing Course Content to Players

The levels have due dates saved to individual assignments. However, it may be necessary to manually open and close material to players due to technical difficulties or exceptions outside of the purview of the program. When viewing a specific level in a course, a button will appear on the same line as the assignment or exam that says "Open" if the content is currently closed and "Close" if the assignment is currently open. When hitting this button, the administrator will be prompted with a screen that asks to choose to change its viewability to all players or to choose a single player to change. The administrator then needs to confirm the change and it will be moved into the database. If the administrator is closing an assignment that a player is simultaneously interacting with, after the player finishes the assignment the program will have already automatically checked if the assignment or exam is open or closed to them and will report failure if it is closed. However, the assignment will still be submitted with a written tag that says late when the administrator views it. It will then be the administrator's discretion to choose what happens to the assignment.

4.1.3g Adding & Removing Players

This may change based on cooperation from NAU and how successful this program is to be determined in the near future. As of right now, the team is not moving to integrate with NAU systems. The process of adding and removing players must be manually done. A player's account should already have a predetermined student ID that administrators can use to add new players to the system manually.

- **Adding Players Manually:** In a course, the administrator will navigate to the grade screen that also includes the list of current players. On the screen, there will be a button labeled "Add Player" and prompt for a student ID or email. With an ID, the program will then compare the ID to existing accounts in the database to find the correct account. If the account does not exist, the program will display an error message explaining that it does not exist. If it does exist, a message will display saying that the account has been added and it should also be reflected in the grading table. The player will then be able to access the account from their account from then on. If the "Add Player" prompt is an email, the program will follow the steps listed in the Creating New Player Accounts section in 4.1.3g. A successfully created player account will then have access to the course from then on.

- **Importing Lists of Players**: In a course, the administrator will navigate to the grade screen that also includes the list of current players. On the screen, there will be a button labeled "Import Players." This will prompt for a file on the administrator's file explorer. That file will contain a list of player email addresses, which will not be saved after the following process. An automated bot will email the list of players containing a unique ID to join the class and the administrator's email address. If the email added is already linked to an existing player, then no additional steps will need to be taken by the player. If the email added is not linked to an existing player, the steps detailed in Creating New Player Accounts in section 4.1.3g will be followed in order to add the player to the course.

- **Removing Player:** In a course, the administrator will navigate to the grade screen that also includes the list of current students. The first column of the table is the list of player names currently in the course. Clicking on these will display that player's profile along with a button in red that says "Remove Player". After clicking, a prompt will appear to confirm the action to avoid accidents.

- **Creating New Player Accounts:** The first time an email is attached to a player account is what the team is considering a new player account. When an administrator attempts to add a player that does not have an account already within the system, they must insert the player's email address into the respective prompt, see Adding Players Manually and Importing Lists of Players in 4.1.3g. After the program confirms that the new email is not already in the database, a screen will open to the player welcoming them to the program and displaying information about the program, and in the future potentially a tutorial. The next screen will display a premade login name that the student must use. Following this on that same screen will be text boxes for an in-game name, also called a display name which will be publicly visible to any classes the player joins, a password, and a selection of default profile pictures to choose from. After successfully creating a profile, they will be redirected to their home page which will display the class they attempted to join initially.

4.1.3h Emptying the Course

To clarify, a full course means that there are players in the course with completed grades. Emptying the course is clearing all of the current players to prepare for the next group of players to take the course. In the level's grade screen, there is a red button that says "Empty Course". This will prompt for a first screen to ask if the administrator is sure. A second screen will appear asking for which version the administrator wishes to save the grades in, the same method explained in 4.1.3d. Finished grades will be exported to player profiles, which only they can view on their own profile screen. The administrator's grading screen will reflect the change with an empty table and a default message asking to add players instead of the grid the table would occupy.

4.1.4 Player Profile Management

This section will discuss the player login in section 4.1.4a as well as how players will be able to customize their profiles in section 4.1.4b. Customization is important for students in order to keep students engaged by making the course feel more like a game than a course. Customization lets users make their own selections about what they want to see, or set preferences for how information is organized or displayed. Customization can enhance user experience because it allows users to be able to interact more with the LMS. Having a profile picture inside the LMS that students can customize will help keep students engaged because they are more invested in the course. Having a secure login for each profile will be important in order to maintain the integrity of the data. Section 4.1.4 will discuss the main customization features the students will be able to use on the website application. Later section 4.1.5 will discuss how the players will interact with the course material on the website application.

4.1.4a Player Login

A player login will be used to limit access to people outside the organization or users that aren't registered. The login page will verify credentials in order to securely put the students into their correct account. Without a login page then students could not work on their own assignments, quizzes, or tests. Each time a student comes to the website they will be prompted to enter their username and password to get started with the course. The login page will prompt the user for a username and a password to enter and submit for approval to the database as well as include a forgot password button under where the user will enter their password.

- **Securely logs in:** Everytime a user comes to the website a login screen will appear asking for a username and password so that the system can securely log the user in. No one except the owner of an account will be able to access the password for that account. Users will not be logging in through the NAU subsystem but instead through a secure web client that our team will develop. The students will be given their username and password for securely logging into their individual account by the administrator. The team will provide strong and consistent security with .NET Frameworks.

- **Lost Password:** The "forgot password" button will be below where the player enters their password on the login screen. The username will not be allowed to change but if someone does forget their password they can click "forgot password" where they will get an email confirming that they do actually want to change their password. In the email players will have the option to change their password by clicking on a link confirming they would like to change their password and then by entering a new password they would like to use. If a player still can not access their account then the player can contact their administrator which in turn can ask the super-admin to create a new username and password for the student to use for the remainder of the course.

- **Logs in to correct account:** The login page will appear every time a user first enters the website which will correctly retrieve login information from the teams database so that students can only access an account they own. We want to maintain the integrity of the data and not let students who do not belong on an account tamper with it. It would be unfair to have students sabotaging other students' accounts by making them fail a bunch of assignments, quizzes, or tests. Without correctly logging in to an account the whole LMS is broken and would not be usable. Logging in to a specific profile would consist of GET and POST requests using PHP and SQL to access the database and check to make sure it is a valid login. .NET Frameworks has a lot of security features that will help us hide and protect sensitive student information. After logging in students will be able to access and work on their course material.

4.1.4b Personalizing Player Profile

By making profiles more customizable the team hopes that students will be more engaged. If the team can get students to customize their profiles more, then students should do better in the class since they are now more invested. Player profiles will include the students name, the class they are in, as well as their selected avatar. The key to having a personalized profile is so that students can easily identify other students without even having to look at their name and keep students engaged by having them work on their profile. Students will not be able to change their usernames.

- **Profile Picture:** All players will be able to choose their own profile picture after a profile picture is bought from the marketplace. Profile pictures can be selected in the player's profile section of a player by navigating to their profile and clicking on the profile picture that the player wants and clicking the "save" button. After the profile picture is saved to a player's profile, the profile picture can be viewable by all other players taking the course. Player profiles will help all users stand out from the rest of the class making it easier to identify them by other students. All player profiles will be school appropriate since the avatars will be predetermined by the website application.

4.1.5 Players Interacting with Course Material

Players will be interacting with the course material on the website application. The course will only be available after it is created on the desktop application. Players will only be allowed to interact with the course material available to their specific account. Some players may be working asynchronously because a student might be working farther ahead into the course than another student. For example if a player wants to get ahead on assignments they can work as far as ahead as they like as long as the player completes the previous level. Therefore some players may be working on different material at the same time if one of the players is working ahead.

4.1.5a Completing Course Material

Course material for courses include: quizzes, assignments, and exams. Quizzes are short exams meant to give players practice before taking an exam. An assignment is something that a player can work on to gain more knowledge on the topic at hand by showing the administrator they can complete an assigned task on the topic being studied. Exams typically count the most for any class and this is where an administrator can see if a student is doing well in the course by having the player test their skills. Course material will be unlocked for that particular player as long as the previous level has been completed. Players will also earn coins for successfully completing course material such as passing quizzes, assignments, or tests. Completing extra course material that is only recommended not required by the administrator could also earn a player an explorer badge.

- **Quizzes:** Quizzes will be available to players after course creation and can be accessed in the quiz section of the website application. Quizzes will also be viewable to a player inside their corresponding level as long as a player has unlocked the level. Questions for each quiz are going to be dependent on what the professor wants each player to answer. Quiz questions could be multiple choice, fill-in-the-blank, or a hotspot. Multiple choice questions will have a series of answers that the player can choose which answer they think is correct. Multiple choice questions can have multiple correct answers, if that is what the administrator wants upon course creation or while updating the course. Fill-in-the-blank questions are typed in an answer box which the system will check to see if the entered answer matches a correct answer set by the administrator. A hotspot is the area of correctness a student could click and still get a question correct. After completing a quiz a player can submit their quiz by clicking the “submit quiz” button at the end of the quiz. Quiz questions will be graded automatically if that is what the administrator chooses upon course creation or course updating. If a quiz consists of a written file, then the predetermined checking system would not be eligible and the quiz would need to be sent to the administrator for grading. Quizzes can have a designated due date set by the administrator and players would not be able to submit after this assigned due date unless an administrator deems otherwise, see 4.1.3f. Multiple submissions of the same quiz will be permitted if that is what the administrator prefers when creating or updating the course.

- **Assignments:** Assignments will be allowed to be uploaded to the website application for administrator grading in the assignments section of the website. Assignments will also be viewable to a player inside their corresponding level as long as the player has unlocked the level. A player can upload any type of file to the assignment for administrator grading. An assignment is uploaded by clicking the “browse files” button and selecting a file to upload from the player's computer. After the file is uploaded to the website a player can hit the “submit” button which uploads their file to the LMS and the administrator will then be able to access that file for grading. Assignments will be manually graded by the administrator since the website will not be able to grade any handwritten assignments. Assignments can also have a designated due date set by the administrator and players would not be able to submit after this due date unless an administrator deems otherwise, see 4.1.3f. Multiple submissions of assignments will be allowed if the administrator would like to allow multiple submissions for players when creating or updating the course in the desktop application.

- **Exams:** Exams can either be automatically graded or manually graded by the administrator depending on what the administrator prefers. Exams can be completed in the exam section or their corresponding level as long as the player has unlocked the corresponding level. Exams can have multiple submissions if that is what the administrator prefers when creating the exam or updating the course. Exams can also have a set due date created by the administrator. Exam questions will be similar to quiz questions but will also be able to handle files to be uploaded as an answer to an exam question. Exams can be submitted by going to the end of the exam and clicking the “submit exam” button which submits the exam to the course administrator for grading and tells the system to automatically grade any questions that the system can grade.

- **Earn Coins:** Players can earn coins by passing assignments, quizzes, or tests after being reviewed by their administrator. Coins can also be earned by winning the in-course lottery. The lottery is where players can submit some coins with the chance of earning a larger sum of coins. Each coin a player submits to the lottery is a chance they could win the entire pot, so for every coin a player submits to the lottery the greater the chance of that player winning the lottery. The system will randomly select one coin that was submitted and the owner of that coin will win the lottery. If someone drops the course, then all the coins go into the lottery's pot. Coins can also be gifted from one player to another player. The total amount of coins a player has will be displayed on each player's profile that only the owner of the coins will be able to view.

- **Explorer Badges:** Extra content can be made available for students to view if the administrator adds it into the course. The student will click the extra content they would like to complete inside the extra content section of the website. The extra content could be from any level as long as the player has unlocked the level that is associated with it. After completing the extra content the player has chosen, the program will automatically reward that player with an explorer badge. These badges are achievements that can be displayed on the player's profile if a player chooses to show their explorer badges by selecting "show badges" inside the "profile" section of the website. These badges can then be traded in for coins to the marketplace by going in the "marketplace" section of the website and selecting a badge the player wants to sell to the marketplace. Any achieved badges can be displayed as long as the player has received a least one at any time in the course.

4.1.5b Purchasing from Marketplace

The marketplace is a section in the LMS where students will be able to purchase items to help them achieve a higher grade in the class. These items inside the marketplace will be referred to as "advantages". The advantages that will be listed in the marketplace for purchase via coins are described below.

- **Technical Notes:** Grants the player valuable extra learning materials on a chosen topic as long as that topic has been unlocked by the player. Players are not permitted to use the technical notes on any exam but the information on the notes could greatly improve a players understanding of exam content before taking the actual exam. Technical notes would be useful for students who missed a class and need more knowledge on a certain topic before taking an exam. The technical notes can be purchased from the marketplace by selecting the topic of the notes they would like to purchase and clicking on the "purchase" button. The technical notes can be accessed from a player's profile at any time for review. Only the player who purchased the technical note will be able to access it. There is no limit on the amount of technical notes that can be bought from any one player as long as the notes are on a level that the player has unlocked.

- **Redo on Exam or Assignment:** A redo on an exam or assignment is beneficial to players who did not perform as well as they would have liked to when they first completed the exam or assignment. A redo buys that player an extra attempt on an assignment or exam without being deducted for being late. After an assignment is failed or the student wants a redo, the student can purchase a redo from the page they started it by clicking the “redo” button next to the exam or assignment. A redo option will only be available once it has been attempted. The test, quiz, or assignment will open back up for that particular player so that they can resubmit their exam or assignment.

- **Gift Coins to Another Player:** A player can gift as many coins as they possess. Coins will be able to be gifted to any player in the same course as the player trying to gift the coins. A player will have to navigate to the marketplace where the player can then submit how many coins they would like to give to another player by selecting the player and how many coins they want to send. After clicking the “submit” button when attempting to send coins, the amount of coins selected will be deducted from one profile and added to the profile that should receive the coins.

- **Exemption Medals:** Exemption medals can exempt questions on the final exam, but a player is only allowed to use a maximum of two exemption medals on the final exam. To use an exemption medal a player can select their exemption medal while taking the final exam and will then be prompted to select which question they would like to use that particular exemption medal on. After submitting which questions a player wants to be exempt from, the submitted questions will not count towards the player's grade on the final exam. The number of exemption medals a player owns will be displayed on that player's profile where any player in the course can view.

4.2 Non-Functional Requirements

If functional requirements are what the project is expected to do, then non-functional, also known as performance, requirements describe the performance of the project. Using a sports example, the act of running would count as a functional requirement while running at a certain speed or covering a certain distance in an amount of time would be a non-functional requirement. For a successful product, the team prioritized **ease of use, data integrity, speed, and scalability**.

4.2.1 Ease of Use & Clarity

The project needs to consider that not all users are familiar with video games as a whole. The team has elected to use game terms such as “abilities”, “players”, and “levels” to emphasize the intended structure of the courses. Plans for the future include creating a tutorial for first time users of the program and implementing help buttons on every screen to explain sections of the program. This tutorial will take at most 30 minutes to complete and will allow administrators to comprehensively understand the program.

4.2.2 Data Integrity

There are a lot of actions that can be taken by both the administrator and the players that can affect the progress of the other. Ideally, the precedence of which action occurs first should be accounted for within the design of the program, but any flaws to this will cause serious damage. In order to support this, the team is considering version saving in order to prevent incidents from occurring. The team is considering current testing and solutions based on the two generals' paradox. Generally, it is a pitfall that all programs need to consider when dealing with information loss when traveling through information systems. To test if our implementation works, the team will forcefully try to do large data uploads and cut the internet connection between the device and the database to see how the program recovers.

More detailed information about the damage can be found in 5.2.

4.2.3 Speed

The number of actions is at most equal to the number of accounts created on the program that must be processed by the system can potentially cause a slow down. In general, it is inconvenient for the program to be slow and can cause errors due to overlapping requests by both administrators and players. The goal is to not have actions take longer than five seconds. Any action that takes longer than this must explicitly display to the user that it will be the case.

4.2.4 Scalability

The largest reason for making this project for the sponsor is scalability. The current workflow that he has is not sustainable for a growing number of students in his class. Furthermore, if the project is successful enough then it may be used by other universities, or at least other professors and administrators that wish to use a differently structured LMS. It is very difficult to measure how much information that the database would have to maintain for each account. In a perfect world, the amount of data the team can hold is infinite. For the time being, the team asks that the program can comfortably handle at least two courses of 100 users each.

4.3 Environmental Requirements

Environmental requirements are parameters imposed on the project by outside sources such as the sponsor and, based on decisions made in the tech feasibility report, on the users available technologies. As a whole, the sponsor has not imposed any outstanding requirements that would need to be implemented at this time. Therefore, the team evaluated the environmental requirements based on the technology that users would need to have in order to access or manipulate the program. For these reasons, users must have access to the following technologies in order to use the program.

4.3.1 Windows Computer

The desktop application will be made in Unity. The team is unsure if it will work for MacOS or Linux at this time. Efforts are not being made to verify compatibility at this time due to time restrictions.

4.3.2 Internet

The website application will require the player to be logged into the internet to access their progress. This is to attempt to reduce the chance that players can access information built into the course.

The desktop application will only require internet access to download or upload a finished version of the course. Edits to the course or initial creation can be done offline. Furthermore, in order to obtain the setup wizard for the desktop application internet access will also be required.

5.0 Potential Risks

Throughout any endeavor the most probable cause of risk is human error. The team may try as hard as it can to avoid any potential compromising problems, but at some point at least one fault may slip through. Out of all possible mistakes, the three most probable and dangerous risks were evaluated here.

5.1 Account Information Leak

The majority of the program relies on admins and players being able to access information related to their accounts. Consequently, if anyone not the owner of an account has access to that account it can cause damage to the progress of a course or damage to the course itself.

5.1.1 Cause of Risk

Programmer Error: The easiest way that information leaks is the programmer fails the abstraction, or separation, of functions. In other words, functions that should not have access to personal information are able to view it one way or another.

5.1.2 Likelihood

Medium: If the team is not careful, this can very easily happen. At this point in time, it will be a matter of diligence and care during the development process.

5.1.3 Future Steps

The current plan would be to disable the database the program is using and move to discover how the leak started and take steps to fill any programming holes that were used to obtain the information.

5.2 Data Integrity Compromised

There are a lot of actions that can be taken by both the administrator and the players that can affect the progress of the other. Ideally, the precedence of which action occurs first should be accounted for within the design of the program, but any flaws to this will cause serious damage. Furthermore, the team wants to circumvent problems caused by internet loss. For our sponsor at NAU, a problem in the local area is significant internet outages without warning. This must be handled correctly.

5.2.1 Cause of Risk

Programmer Error: Information corruption is the fault of the programmer alone. It means that somewhere, a teammate has failed to account for multiple accounts working on the same information at the same time and the program was not equipped to handle it. The program not being able to recover from missing packages or data is unacceptable for a successful program.

5.2.2 Likelihood

High: A lot of sections across the entire program need to allow for a countless number of users to interact with the same information. Also, the way administrators and players are set up means that any higher level actions done by the administrator can accidently wipe out player actions.

5.2.3 Future Steps

The current plan would be to disable the program and move to discover where the failure began and take steps to fix the code that allowed the incident to happen in the first place.

5.3 General Database Concerns

Presently there is a concern that the database the team is using for the duration of the Spring 2022 Capstone may have to be changed, moved, or have increased cost. Complications involving the database during development may occur for other reasons, but the team stresses that any significant changes may cause a loss in time and productivity which will hopefully be avoided.

5.3.1 Cause of Risk

NAU administrative decisions: There is some confusion among NAU staff as to how to allocate resources for the Spring 2022 Capstone. At this time, if NAU imposes that we must use a database it may take some time out of the project to understand how to interface with a different database than the one the team is currently using.

5.3.2 Likelihood

Low: The main reason for this issue to be mentioned is to document issues the team is currently having in the event that a larger issue appears during development. Databases themselves are very reliable.

5.3.3 Future Steps

The current plan would be to stop working on parts of the project that require the database and to research and discuss a cost effective alternative to what the team would be presently using.

6.0 Project Plan

Throughout the rest of college, we will be working on this project with different milestones as we go along. Our plan is to have a working demo of the project before the end of the semester with a working login page for students to access. Getting both sides of the project together and connecting it to the database is the hardest part of the project. Once completed it will be used as a Proof of Concept (POC) and adding components from there will be the easy part. After the end of the semester, the team may continue working on the project throughout the summer in order to get a jump start on the next semester.

6.1 Milestones

This project is for a university senior project, or capstone, and deadlines within the semesters are fixed milestones the team must meet or beat. However, the team has decided to adopt an agile development methodology for the project wherever possible. To explain briefly, the team has set deadlines for overarching milestones of the project, but the particulars within them that need to be completed in a set order have flexible finishing times. Every week, the team will meet to adjust the workload as necessary with the goal of meeting or preemptively finishing the task's deadline. This means if a section of a project is done faster than anticipated or there is a delay due to technical difficulties the team can easily adjust the pace required in order to meet the task's deadline. The following sections are how the team has categorized the milestones.

6.1.1 Spring 2022 Capstone

The spring semester of the team's capstone started in January and will end May 6th. A majority of the deliverables are already completed at this point. The last three remaining are the requirements specification document, the design review presentation, and the tech demo for the POC. The tech demo is lengthy and will be discussed in 6.1.2. Hard deadlines are in place for each milestone set by the capstone team's faculty mentor.

- **Requirements Specification Document:** The team needs to evaluate the core components that comprise the functional, non-functional, and environmental requirements of the project as whole. Possible risks will then be evaluated and planned

for. Lastly, the team's schedule for development will be created to ensure timeliness and completeness of the project.
Completion Date: April 29th

- **Design Review:** The design review will be a ten minute presentation to the capstone class. It will include a broad overview of everything mentioned in this document. The main goal of the review is to receive feedback from the class as well as cement the team's understanding of the project.
Completion Date: April 22nd

6.1.2 Tech Demo

The tech demo will show off the team's POC to the capstone mentors and the sponsor. It will be built out of Unity, ASP.NET Frameworks, and an Amazon Web Services (AWS) database. The three technical challenges the team needs to address are connecting AWS to both the website and the desktop application, displaying information to the website from the database using model views, and creating a login screen with Unity. As of writing, a very large majority of the POC is already completed. The actual presentation day is still being decided, but must be before May 6th.

- **Set up AWS Account:** In order to create the database for the project, an AWS account needs to be made and permissions for the team to access it must be in place. The team has not used AWS before and a secondary objective to this is understanding how to add files directly to the database. This is for convenience as well as ensuring that the team understands how to access the database remotely.
Completion Date: April 22

- **Create Login Screen (Unity):** The team does not have any experience with Unity. Starting with a login screen is the most important and basic step for the desktop application. This will ensure that the team can move data from Unity to the database.
Completion Date: April 26

- **Set up ASP.NET API:** ASP.NET is the team's chosen Application Programming Interface (API). ASP.NET is the actual name of the framework being used and not an acronym. Its purpose is to connect the database to the website application and the desktop application.
Completion Date: April 26

- **Set up ASP.NET Core Views:** ASP.NET has a built-in capability to display web pages called views. Each view is built from a ".cshtml" file which is an HTML file with CSS built into it. In other words, the team does not have to use another programming language

or software to build web pages from. The main views that the program needs to function for the players is a class view, a profile view, and possibly a settings view. There will be more components to this as the program gets more complicated.
Completion Date: April 27

- **Bug testing and fixing:** Depending on the speed of completion for all of the other sections above, the remaining time has been allocated to bug fixing and testing the application. Any errors on the tech demo risk jeopardizing the project as a whole.
Completion Date: April 29

6.1.3 Summer Break

The summer break is going to focus entirely on the project. The break ends August 28th. To make it easier on the team, large milestones will be placed over the span of each month. By the end of June, the goal is to have the core of the project, described in 4.1, completed. The month of July will then be used to polish the core and fix all possible bugs. This foundation will set up all of August to begin adding peripheral functions. Development will adjust according to the pace of the team and the level of success of finished tasks.

- **Create Administrator Profiles (4.1.1) :** The program as a whole is account centered. The first features that need to be implemented are creating, saving, and accessing accounts. Administrator profiles need to be made to allow the team to test the course creation and editing process.
Completion Date: May 31

- **Create Course Creation (4.1.2)** : After the administrator profile is created, course creation can start being developed and tested. This section is the foundation for the project. Extra caution must be taken to avoid programming errors to ensure a smooth development process. Please see 4.1.2 for a complete description of what needs to be built in this section.
Completion Date: June 30

- **Be Able to Manipulate Course Content (4.1.3) :** Now that course content has been made, it will be possible to easily edit or change it after it has been published. This will most likely be in development alongside section 4.1.2.
Completion Date: June 30

- **Create Player Profiles (4.1.4) :** The program as a whole is account centered. The first features that need to be implemented are creating, saving, and accessing accounts. Player profiles need to be made to allow the testing of course completion.
Completion Date: May 31

- **Allow Players to Complete course (4.1.5) :** After course content has been created, the players will then be able to complete it. The team needs to work on how the players view the course as well as what it means to complete assignments from a programming standpoint.
  Completion Date: June 30

6.1.4 Fall 2022 Capstone

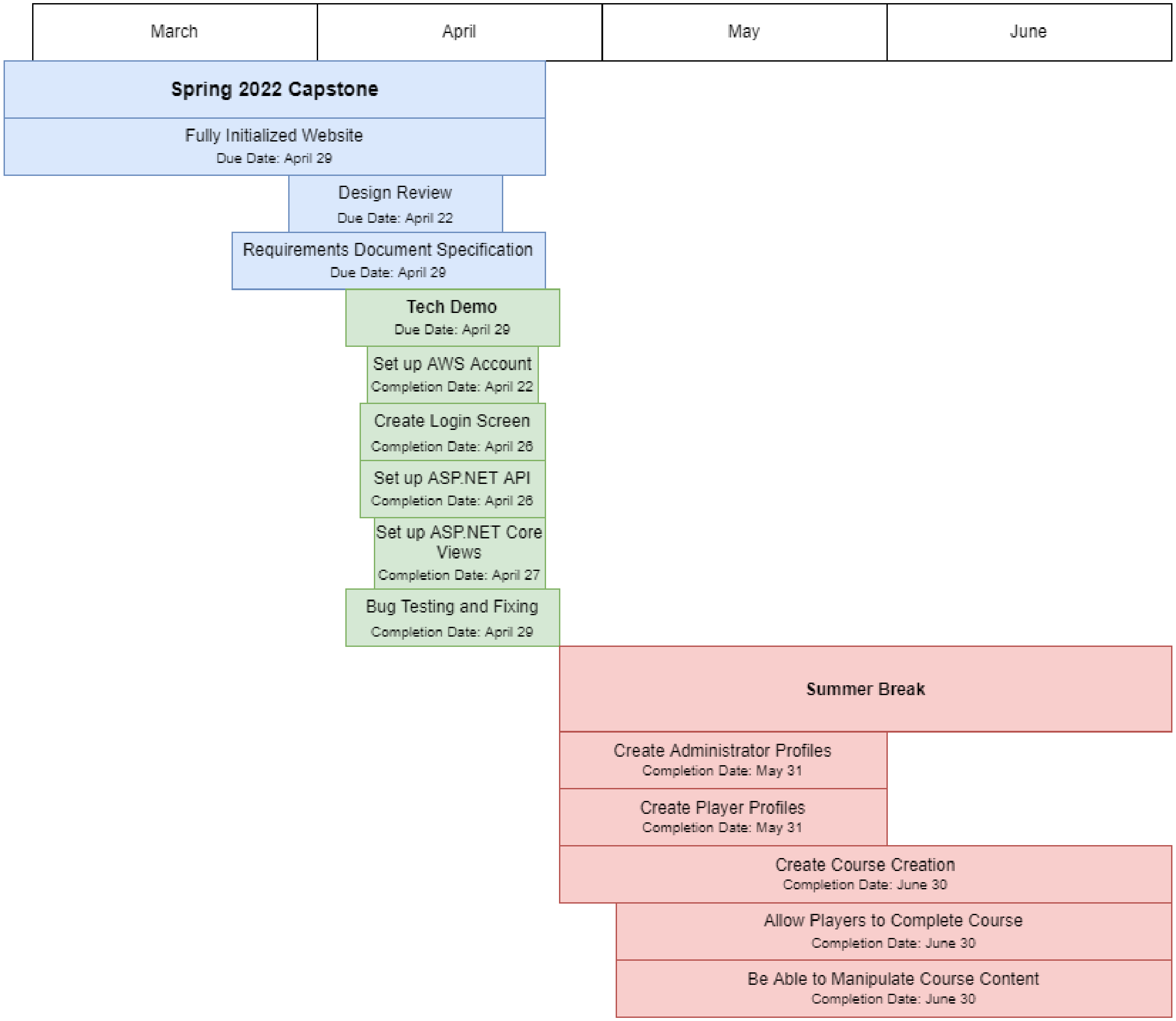
The fall semester of the team's capstone class starts in August and ends in December. The team is not privy to the full list of deliverables that the class requires them to complete. Therefore, no milestones have been added to the schedule at this time. Concerning the project itself, the start of the semester will be used to evaluate the progress of the team. New milestones will be created to continue adding peripheral ideas and to make changes more specific to the sponsor's wants and needs.

6.2 Visual Representation of Milestones

Please refer to figure 4.0.

Figure 4.0 RTX Gantt Chart

Each color represents a different set of tasks to be completed. The blue color represents the Spring 2022 Capstone, the green color represents small tasks for the tech demo, and the red color represents small tasks to be completed over summer break. The Gantt Chart starts from March and goes until the end of June. The preferred date of completion is shown under the tasks.



7.0 Conclusion

Many different LMS’s are used all over the globe for educational purposes. The main issue with BbLearn at NAU is that it is not very interactive. Gamifying a course is a great way to increase student engagement and this requires the LMS to be interactive for the students. The problems our client is having with BbLearn is that it requires too much manual labor and does not allow for much customization of a course. Without the proper customization tools, a course is much more difficult to gamify.

Our client Professor Baxter has already designed a gamified version of his course, but it takes him a lot of time to keep it updated and maintained. Team Gaming Ed. will design a new LMS that allows for administrators to build a gamified course. The LMS will also support courses that wish to not be gamified. Administrators will be able to create a course that keeps students more engaged and also saves them valuable time.

Out of all possible mistakes, the three most probable and dangerous risks were evaluated in section 5.0. The team can not allow any account information leaking out of the system so that users' credentials are not compromised. This risk will be mitigated by separating functions so only functions that are supposed to have access to personal information are able to view the sensitive information. Data integrity is important to the project and will be supported by equipping the program to handle multiple accounts working on the same information and making the code not allow for data to be compromised even under an internet outage. General database concerns include having the database change, moved, or have increased cost. The solution to the general database concerns is to discuss a cost effective alternative to what the team is presently using. Risks are evaluated so that the project stays successful over a long period of time without compromising the integrity of the project.

A new LMS is required to make a gamified course because our client spends way too much time trying to manage his gamified course using BbLearn. Our new application will be easily customizable and be able to be used for a variety of different courses while keeping students at a high engagement level. The team hopes to revolutionize the way future LMS’s are developed from now on so that students can be more successful.

8.0 Glossaries and Appendices

Glossary

Data Integrity: A measure of accuracy, completeness, and consistency of data.

Gantt Chart: A chart of due dates and project developments for the team to complete

Player: A user who works to finish the course created by the administrator.

Abbreviations

API	Application Programming Interface
AWS	Amazon Web Services
BbLearn	Blackboard Learning Management System
LMS	Learning Management System
NAU	Northern Arizona University
POC	Proof of Concept
CENE	Civil and Environmental Engineering

Appendix A - References

[1] "Why Student Engagement Is Important in a Post-COVID World." *Learning Sciences International*, 14 May 2021, https://www.learningsciences.com/blog/why-is-student-engagement-important/.

[2] Fareena Afzal, et al. "What Is Learning Management System & How to Choose Best One." *WP Minds*, 19 Jan. 2022, https://wpminds.com/what-is-the-learning-management-system/#:~:text=Currently%2C%20the%20number%20of%20LMS,purposes%20using%20the%20LMS%20platform.