COLLEGE OF ENGINEERING AND COMPUTER SCIENCE

FLORIDA ATLANTIC UNIVERSITY

PRINCIPLES OF SOFTWARE ENGINEERING

CEN4010 – Spring 2023

(#23) Team Nightshift:

Milestone 1 Project Proposal and High-Level Description

Name	Email
Adam Clark	*cclark36@fau.edu*
Quang Le	*leq2019@fau.edu*
Nicholas Moussa	*nmoussa2017@fau.edu*
Mahmood Sakib	*msakib2022@fau.edu*
Kyla Tolentino	*ktolentino2021@fau.edu*

Revision History	Date
n/a	*n/a*

➢ **Submitted on 03/03/23 by Mahmood Sakib**

Table of Contents

Executive Summary

As students, we all know from personal experience that there are always moments toward the end of any given semester in which we all inevitably whip out our calculators to perform a series of "doomsday" GPA number crunchings. Questions arise such as "What do I need to score on this final to pass the course?" or, hopefully, "What's the absolute *worst* I can do and still retain my 'A'?" Such questions are important in prioritizing one's time and establishing goals, especially when coinciding project and exam deadlines roll in for those already juggling multiple courses and responsibilities from family to work! "But *where* are the answers to all these pressing issues?" you might ask. Why, the how, the what, the when, and the where are all within the Grade PlanAlyzer, of course – it just needs you!

The Grade PlanAlyzer application is our solution to students' woes by creating a platform tailored to providing them a more robust, proactive, and quantitative approach to academic planning throughout each semester. By retaining all of one's pertinent academic information, the student not only avoids having to manually transcribe large quantities of data on a case-by-case basis to calculate grading hypotheticals, but is also afforded the luxury of attaining more dynamic and impactful results via the application of data analytics. In other words, the student will remain keenly aware of one's academic standing throughout the entirety of a given semester such that they may make informed decisions to guide their path forward, rather than defensively access such information to backpedal a semester gone awry. The Grade PlanAlyzer will enable students to optimize their time management skills and dedicate a more appropriate amount of time to each assigned task. This approach will enable them to ensure that they are allocating their efforts efficiently and effectively, ultimately resulting in better overall performance and academic outcomes.

Competitive Analysis

The analysis of competitors' web sites will be quantitatively assessed based upon the following six primary features: User Experience, UI Design, Data Storage, Calculation Complexity, Customization, and What-If? functionalities. The criteria are scored on a 0 - 5 scale (0 = not present, 1 = awful, 2 = poor, 3 = fair, 4 = good, 5 = outstanding) and are applied to five software applications chosen for their thematic similarities involving grade calculations. The results of this assessment are displayed in Table 1, below.

Table 1 - Concept Scoring Matrix

	Grade PlanAlyzer	Canvas	Calculator.net	Fourpoint	Grades	The Grade Calculator
User Experience	4	5	2	4	4	2
UI Design	5	3	2	4	3	2
Data Storage	4	5	0	4	4	0
Calculation Complexity	5	4	3	4	3	1
Customization	5	3	0	5	4	0
What-If?	5	3	2	0	0	0
Mean	**4.67**	**3.67**	**1.50**	**3.50**	**2.83**	**0.83**

Canvas (3.67) - https://www.instructure.com/canvas

At present, Canvas falls short in providing students with essential features to optimize their academic performance. Specifically, the platform lacks the capability to calculate grades per assignment, limiting students' ability to allocate their time efficiently and accurately gauge their progress. Additionally, Canvas does not provide students with the ability to determine the minimum score required to achieve an A in a specific assignment, depriving them of essential information needed to set realistic goals and make informed decisions. By contrast, an advanced grade calculator app can address these shortcomings, empowering students to take a more proactive approach to their studies, optimize their learning experience, and achieve their full potential.

Fourpoint (3.50) - https://apps.apple.com/us/app/fourpoint-a-gpa-calculator/id383417299

Fourpoint is an efficient and user-friendly mobile application designed to make calculating your GPA easy and accurate. It is beneficial for students who attend schools that use the supported GPA calculation system and allows you to add semesters and courses as you progress through your academic career. Fourpoint has several useful features, including the ability to designate a course as Pass/Fail or not count toward your GPA, setting the weight of grades, and determining your major GPA. You can also designate courses as in progress and add past GPA history with Q-Points and Hours. Fourpoint enables you to calculate both a single semester GPA and a cumulative GPA for your entire educational career. However, it may not be useful for students who have unique academic circumstances or situations that require more complex GPA calculations.

Grades (2.83) - https://apps.apple.com/us/app/grades-view-your-scores/id1434492452

The Grades App is a helpful tool for students who are looking to track their academic progress. With features such as automatic GPA calculation, the app can provide users with a quick

and easy way to get an overview of their semester grades. However, it seems that the app may have some limitations. For example, the user interface may lack personality and customization options, which could be a disadvantage for some users who are looking for a more personalized experience. Additionally, the app may not account for customizable grading scales, which could be an issue for users who have unique grading systems in their schools or courses. Finally, the app's analytics may be somewhat inaccurate, which could impact the reliability of the data presented.

Calculator.net (1.50) - https://apps.apple.com/us/app/grades-view-your-scores/id1434492452

The visual appeal of Calculator.net is lacking and requires an excessive amount of manual data input due to the website's lack of dynamism. This means that each assignment grade must be manually selected and typed in for calculation, resulting in a time-consuming and frustrating process that can lead to errors and decreased efficiency. This is particularly problematic for individuals who need to calculate grades frequently, as it can impede their workflow and productivity. Calculator.net isn't user-based and requires complete manual entry for every grading query, also only represents a singular built-in grading scale so it is not customizable to the user.

TheGradeCalculator (0.83) - https://thegradecalculator.com/

The Grade Calculator tool has some useful features, but there are also some significant drawbacks that should be considered. Firstly, the user interface is outdated and not very user-friendly, which could make it difficult for some users to navigate. Additionally, the tool is not consolidated into one location and requires users to visit multiple websites to access all of the features they need. This can be time-consuming and inconvenient for users. Another major disadvantage of the tool is the lack of a login feature, which means that there is no user and data relationship. This could be frustrating for users who want to save their progress and access it later, as they will have to input their data every time they revisit the website. Finally, the tool lacks infographic capabilities, which could be a significant disadvantage for users who want to present their data in a more engaging and visual format.

Grade PlanAlyzer (4.67) - https://q-wrld97.github.io/cen4010_sp23_g23/

Planned Advantages:

The What-If? tool offered by the Grade PlanAlyzer is particularly advantageous in that its calculations will always be derived from the latest empirical dataset at one's disposal. Many competitors, on the other hand, simply attempt to provide a crude letter grade estimate based solely upon input of hypothetical score averages by weighted category. Moreover, such tools are further limited due to their complete lack of customization options to account for various calculation intricacies, such as a unique grading scale, extra credit, or even complex rules such as replacing the lowest deliverable score in a category with the average of the remainder. However, the Grade PlanAlyzer maintains the real-time authenticity of such What-If? estimates by automatically adjusting its calculations to display results under the context of remaining scoring opportunities after factoring in the weight of what events have already transpired.

Data Definitions

Table 2 - Table of Data Definitions

Name	Meaning	Usage	Comment
Registered User	Actor	*Use Case Scenarios*	A student who has registered with the system
Unregistered User	Actor	*Use Case Scenarios*	A student who has not registered with the system
System	Platform Hardware and Services	*Use Case Scenarios*	The collective intercommunicating software and hardware components forming a computer system, including front-end and back-end code, configuration files, documentation, and Firebase database
Website	User Interface	*User Interface*	Front-end display for user interaction
Homepage	User Interface	*User Interface*	First web page the user sees when not logged in
Navigation Bar	Service/User Interface	*Site User Service*	Fixed toolbar display containing a variety of site navigation pathways (Dashboard, Support, Account Settings, Log Out)
Dashboard	Service/User Interface	*Site User Service*	Primary hub for displaying student information and tools
Account	Data	*Use Case Scenarios*	Refers to the location on a network server storing the user ID, password, and composite information associated with an individual user

Login	Service	*Site User Service*	Permits user to access their stored personal account data and grade analysis tools
Session	Service	*Site User Service*	Period of activity between a user logging in and logging out of a multi-user system
Logout	Service	*Site User Service*	Ends a user's login session and redirects to home page
Register	Service	*Site User Service*	Process by which an unregistered user may become a registered user
User ID	Data	*Use Case Scenarios*	A unique logical entity and name of respective user's email address used to identify and distinguish between users
Password	Data	*Use Case Scenarios*	A confidential, user-defined string of characters used to authorize access to the account of an associated UserID
Forgot Password	Service	*Site User Service*	A self-service account recovery process by which a user may choose to create a new password after clicking the password reset link sent to the associated user's email address
Password Reset Link	Off-site Service	*Off-site User Service*	An automated link sent to a user's registered email address that contains a unique token identifying the user which is authenticated by second factor when the link is clicked

Confirmation Email	Off-site Service	*Off-site User Service*	An automated email triggered upon registration of a new user which must be addressed within 24 hours to verify user's identity and initiate a permanent account
Account Settings	Service	*Site User Service*	The configurations and settings associated with an authorized account which are established upon initial enrollment, but may also be changed at a later date
Course Form	Data	*Use Case Scenarios*	A formatted document containing blank fields in which users must input corresponding data necessary for site function, which is then stored in the database
Support	Service	*Site User Service*	Means by which a user may report bugs and issues experienced during website operation to its creators
GitHub	Repository Hosting	*Use Case Scenarios*	An internet hosting service for software development and version control using Git whereupon the project code repository and documentation are stored and organized
GitHub Pages	Production Server	*Use Case Scenarios*	The website hosting service on which this project is published
Firebase	Database/Back-End Framework	*Use Case Scenarios*	A backend-as-a-service (Baas) which provides real-time database hosting and user authentication services

Overview, Scenarios, and Use Cases

Use Case – Register

A new unregistered user arrives at the homepage and wishes to create an account via the 'Register' button in order to gain access to the website's many features.

1. Description:

Use case describes the process of how the unregistered user will become a registered account holder with the system.

2. Actors:

2.1 Unregistered User
2.2 System

3. Preconditions:

3.1 User has an active internet connection
3.2 System is available

4. Primary Flow of Events:

4.1 User arrives on website homepage
4.2 User clicks 'Register' button from navigation bar
4.3 User enters new 'UserID', 'Password', and 'Repeat Password'
4.4 User clicks 'Submit' button
4.5 Temporary account is created to expire after 24 hours without email confirmation
4.6 System sends automated confirmation email to address listed as UserID
4.7 User clicks confirmation email link to successfully establish a permanent account
4.8 Terminate Use Case – Register

5. Alternate Flows:

5.1 User Enters Prohibited Format into UserID Field

If in Step 4.3, user enters a UserID without including '@' and ending in a domain

1. Website notifies the user that a valid email address must be utilized
2. Return to Step 4.3

5.2 User Enters Prohibited Format into Password Fields

If in Step 4.3, user enters a Password that is less than six characters

1. Website notifies the user that passwords must be at least six characters
2. Return to Step 4.3

If in Step 4.3, user enters a Repeat Password that does not match Password

1. Website notifies the user that current passwords do not match
2. Return to Step 4.3

5.3 User Neglects to Address Confirmation Email

If in Step 4.7, does not complete confirmation email within 24 hours

1. The system deletes the user's temporary account
2. Return to Step 4.2

Use Case – Login

A registered user arrives at the homepage and wishes to log into their existing account via the 'Login' button in order to gain access to the website's many features.

1. Description:

Use case describes the process of how the registered user will access their account.

2. Actors:

2.1 Registered User
2.2 System

3. Preconditions:

3.1 User has an active internet connection
3.2 System is available
3.3 User has a permanent account

4. Primary Flow of Events:

4.1 User arrives on website homepage
4.2 User clicks 'Login' button from navigation bar
4.3 User enters their account 'UserID' and 'Password'
4.4 User clicks 'Login' button
4.5 Website successfully redirects to user's account dashboard
4.6 Terminate Use Case – Login

5. Alternate Flows:

5.1 User Performs Invalid UserID and Password Combination

If in Step 4.4, user submits any combination which is not an exact system match

1. Website notifies the user that the current UserID and Password pairing does not exist in the database
2. Return to Step 4.3

5.2 User Does Not Know Account Password

If in Step 4.3, user knows their UserID but is unable to recall its Password

1. User clicks 'Forgot Password' button beneath 'Login'
2. User enters UserID into the 'Email Address' field
3. User selects 'Submit' button
4. Website notifies the user that a password reset link has been sent
5. User clicks password reset link from their primary email address
6. Redirected user enters new matching 'Password' and 'Repeat Password'
7. Go to Step 4.1

5.3 User Performs First-Time Login

If in Step 4.5, user is logging into their account for the first time

1. Website successfully redirects user to complete 'Account Settings' form
2. User completes 'Account Settings' form and clicks 'Submit'
3. Website successfully redirects user to complete new 'Course Form(s)'
4. User completes 'Course Form(s)' and clicks 'Submit'
5. Go to Step 4.5

Use Case – Add Course

A registered user is in possession of their course syllabi and wishes to input their course data into the system.

1. Description:

Use case describes the process of how the registered user will add courses.

2. Actors:

2.1 Registered User
2.2 System

3. Preconditions:

3.1 User has an active internet connection
3.2 System is available
3.3 User has a permanent account
3.4 User is logged into the system

4. Primary Flow of Events:

4.1 User logs into their account
4.2 Website successfully redirects to user's account dashboard
4.3 User selects 'Course Info' button ('ⓘ' icon) from the dashboard
4.4 User selects 'Add Course' tile ('**+**' icon)
4.5 User accurately inputs course information
4.6 User clicks 'Submit'
4.7 User assigns due dates for each course deliverable specified on prior form
4.8 User clicks 'Submit'
4.9 Course information is successfully stored in the database
4.10 Terminate Use Case – Add Course

5. Alternate Flows:

5.1 User Enters Prohibited Format into Numerical Fields

If in Step 4.5, user enters a percentage value which is not between 1 and 100

1. Website notifies the user that percentages must adhere to $1 \leq \% \leq 100$
2. Return to Step 4.5

If in Step 4.6, user has toggled a 'Custom' grading scale and left thresholds blank
1. Website notifies user that custom grading scale thresholds cannot be empty
2. Return to Step 4.5

If in Step 4.6, user has toggled a 'Custom' grading scale and entered a grade threshold which is higher than the letter grade which alphabetically precedes it
1. Website notifies user that subsequent letter grades must have lower bounds
2. Return to Step 4.5

If in Step 4.6, user has toggled a grading category but failed to specify a quantity
1. Website notifies user that numerical quantity value must be ≥ 1.
2. Return to Step 4.5

If in Step 4.6, user has toggled a grading category but failed to specify a weight
3. Website notifies user that a weight value between 1 ≤ % ≤ 100
4. Return to Step 4.5

If in Step 4.6, user has input weight category values which do not add up to 100
1. Website notifies user that cumulative weight category values must be 100
2. Form submission is rejected
3. Return to Step 4.5

5.2 User Neglects to Address Any Form Data Request

If in Step 4.6, user has failed to address any data request within the form
1. Website notifies the user that all requests for information must be addressed
2. Form submission is rejected
3. Return to Step 4.5

Use Case – Grade Entry

A registered user has received their latest grade feedback and wishes to input its data into the system.

1. Description:

Use case describes the process of how the registered user will enter grade data.

2. Actors:

2.1 Registered User
2.2 System

3. Preconditions:

3.1 User has an active internet connection
3.2 System is available
3.3 User has a permanent account
3.4 User is logged into the system
3.5 User has added at least one course

4. Primary Flow of Events:

4.1 User arrives on website homepage
4.2 User logs into their account
4.3 Website successfully redirects to user's account dashboard
4.4 User selects 'Grade Entry' button (' ' icon) from the dashboard
4.5 System automatically generates shorthand list of eligible score input fields entitled "<Category Name> #<Number>:", grouped by course, for any deliverable which is both currently past its due date and lacking existing grade data
4.6 User inputs the associated score(s) as percentage value(s) up to 4 decimal places
4.7 User clicks 'Submit'
4.8 Grade data is successfully stored in the database
4.9 Terminate Use Case – Grade Entry

5. Alternate Flows:

5.1 User Enters Prohibited Format Into Grade Entry Field

If in Step 4.6, user enters a numerical score which is not between 0 and 100

1. Website notifies the user that scores must adhere to $0 \leq SCORE \leq 100$
2. Return to Step 4.6

5.2 User Needs to Adjust Prior Grade Submission

If in Step 4.5, the user needs to adjust a prior grade submission, the desired field will not automatically generate due to its having existing grade data

1. User selects 'Show All Previous' toggle option within 'Grade Entry' dashboard
2. User inputs updated score(s) in place of existing scores for relevant field(s)
3. Return to Step 4.7

5.3 User Cannot Locate Desired Grade Entry Field

If after Alternate Flow 5.2, the user still cannot locate desired grade entry field

1. User selects 'Course Info' button from the dashboard
2. User selects the 'Edit' (icon) button from the course tile corresponding to the missing grade field
3. User ensures the appropriate grade categories and quantities are designated for inclusion of the desired grade entry field
4. User clicks 'Submit'
5. User ensures an appropriate past deadline is attributed to the desired grade entry field
6. User clicks 'Submit'
7. Proper course configuration data is successfully stored in the database
8. Website redirects to user's account dashboard
9. Return to Step 4.4

List of High-Level Functional Requirements (Initial)

Unregistered User

1. Account Creation

1.1 Procedural Steps

I. User selects 'Register' button from the homepage navigation bar

II. User enters a new UserID (same as valid email address)

III. User enters a Password:

- A. System will check if password is *at least* six characters in length:
  - i. Meets Criteria – Borders of password input field box returns to default settings; Go to Step IV
  - ii. Fails Criteria – Borders of password input field box turn red and a notification appears informing the user that passwords must be *at least* six characters in length; Go to Step III

IV. User re-enters Password:

- A. System will check if both password fields match:
  - i. Passwords Match – Borders of password input field boxes return to default settings; Go to Step V
  - ii. Passwords Do Not Match – Borders of password input field boxes turn red and a notification appears informing the user of non-matching passwords; Go to Step III

V. User confirms via 'Submit' selection

VI. System will check if UserID is available:

- A. Available –
  - i. System will send email prompting confirmation to user.
  - ii. A temporary account will be created in Firebase
  - iii. User will be notified to address confirmation email within 24 hours to prevent account deletion; Go to Step VII
- B. Unavailable – borders of UserID input field box turn red and a notification appears informing the user of UserID unavailability and to try another UserID; Go to Step II

VII. User may choose to confirm account validity via primary email:

- A. User Confirms –
  - i. User clicks confirmation link within 24 hours
  - ii. System permanently saves UserID and password → End of process
- B. User Does Not Confirm –
  - i. User does not click confirmation link within 24 hours
  - ii. System will delete account information at 24 hour mark → End of process

Registered User

2. Account Login

2.1 Login Procedure

I. User selects 'Login' button from the homepage navigation bar
II. User enters UserID
III. User enters Password associated with UserID
IV. User selects 'Login' button:
V. System will check if the entered UserID & Password pairing matches:
   A. Valid Login Combination – System shall redirect user to the dashboard → End of process
   B. Invalid Login Combination – System shall alert user that current UserID & Password combination is not found in the database; Go to Step II

2.2 Password Recovery

I. User selects 'Forgot Password?' button beneath the 'Login' button
II. System will redirect user to 'Password Reset' page
III. User enters UserID into the Email Address field
IV. User selects 'Submit' button
V. System will notify user that "a password reset link has been sent to the provided email address if the UserID exists within the system"
VI. System will check if UserID exists in the system:
   A. UserID Exists – System shall send a password reset link to the provided email address; Go to Step VII
   B. UserID Does Not Exist – System shall take no further action → End of process
VII. User clicks password reset link from their primary email address
VIII. System will redirect user to 'New Password' page
IX. User enters new Password:
   A. System will check if password is *at least* six characters in length:
      i. Meets Criteria – Borders of password input field box returns to default settings; Go to Step X
      ii. Fails Criteria – Borders of password input field box turn red and a notification appears informing the user that passwords must be *at least* six characters in length; Go to Step IX
X. User re-enters new Password:
   A. System will check if both password fields match:
      i. Passwords Match – Borders of password input field boxes return to default settings; Go to Step XI

ii. Passwords Do Not Match – Borders of password input field boxes turn red and a notification appears informing the user of non-matching passwords; Go to Step IX

XI. System redirects user to the application homepage to login with new password → End of process

3. Data Entry

3.1 Account Settings (‘⚙’ icon)

I. User selects ‘Account Settings’ button (cogwheel icon) from the navigation bar
II. User enters name of school
III. User selects current semester and year
IV. User confirms existing GPA scale or enters new parameters
V. User may opt to enter phone number for text notifications
VI. User clicks ‘Submit’ → End of process

3.2 Course Form (‘+’ Add Course icon)

3.2.1 Course Name

I. User inputs a three-letter acronym
   A. On input system will automatically capitalize and limit input length to 3 characters

3.2.2 Term Length

I. User clicks on field
II. Dropdown appears
III. User may choose from list of pre-designated options

3.2.3 Credit Hours

I. User clicks on field
II. Dropdown appears
III. User may choose from list of pre-designated options

3.2.4 Class Days

I. User selects either day(s) of the week or remote checkbox option
   A. System will check if remote option is selected:
      i. Selected – unchecks and prevents further interaction with options involving days of the week
      ii. Unselected – allows user to select day(s) of the week

3.2.5 Grading Scale

I. User selects either from two pre-made grading scales or a custom grading scale
   A. System will check if custom option is selected:
      i. Selected – custom grading scale input fields appear
      ii. Unselected – custom grading scale input fields disappear

3.2.6 Weighted Categories

I. User selects the categories of weighted groupings
   A. System will check if any options are selected:
      i. For EACH Selected – two fields appear:
         a. Quantity – User inputs the numerical total of items corresponding to category, and system will check:
            1. < 1 – notifies user numerical value must be greater than or equal to 1
            2. > 1 – nothing
         b. Percentage – User inputs the numerical percentage of total grade corresponding to category, and system will check:
            1. < 1 or > 100 – notifies user numerical value must be between 1 and 100
            2. > 1 and < 100 – nothing
      ii. For EACH Unselected – two fields disappear

3.2.7 Form Submission

I. User clicks 'Submit' and input validation is executed
   A. System will check if 'Custom' grading scale custom is selected:
      i. True – System will check grading bounds for each field in hierarchal order from A-F for numerical legitimacy:
         a. If ANY Above < Below, > 100, or < 1 – Notify user of invalid parameters; Go to Step 3.2.5
         b. If ALL Above > Below and between 1 and 100 – Condition Passed; Go to Step B
      ii. False – Go to Step B
   B. System will sum Weighted Category percentage fields and evaluate if total equals 100:
      i. True – Go to Step II
      ii. False – Notify user that total percentages must equal 100; Go to Step 3.2.6
II. User data is successfully stored in the database → End of process

3.3 Course Dates ('🗓' icon)

I. The Course dates form immediately follows submission of Course Form
II. A new date field is generated entitled "<Category Name> #<Number>:" for each individual course deliverable based upon the item quantity designated for each weighted category on the previous Course Form
III. User interacts with the calendar API to assign the deadline for each item
IV. User clicks 'Submit' and data is successfully stored → End of process

3.4 Grade Entry (‘ ’ icon)

I. User selects ‘Grade Entry’ button (report card icon) from the dashboard
II. A new grade field is generated entitled “<Category Name> #<Number>:” for each deliverable item with a deadline prior to the current date
III. For each deliverable item with known scoring feedback, the user may input the corresponding percentage value between 1 and 100 up to four decimal places
IV. User clicks ‘Submit’ and data is successfully stored → End of process

List of Non-Functional Requirements

Interoperability Requirements

Browser Compatibility – The system will be a web-based web app that operates on major browsers, including Google Chrome, Mozilla Firefox, Safari, Opera, Brave, and Internet Explorer.

Operating System Compatibility – The system will operate on multiple operating systems, including Windows, Linux, and OS X.

Mobile Operating Sytem Compatibility – The system will operate on all major mobile operating systems, including iOS and Android.

Storage Requirements:

Back-End Maximum Database Load:

- *Storage Capacity* – 1 GiB
- *Document Reads* – 50,000 per day
- *Document Writes* – 20,000 per day
- *Document Deletes* – 20,000 per day
- *Network Egress* – 10 GiB per month

Back-End Usage Load for Authentication and Authorization:

- *Daily Active Users* – 3000 per day
- *SMS Sent* – 10 per day per user
- *Multi-Factor Authentications* – 10 per day per user

Security Requirements

Our user authentication/authorization system will maintain the private integrity of site users' academic information and preferences on an individual basis. When a user creates an account or changes their password.

Firebase Authentication provides backend services, easy-to-use SDKs, and ready-made UI libraries to authenticate users to your app. It supports authentication using passwords, phone numbers, popular federated identity providers like Google, Facebook and Twitter, and more. The Firebase Authentication software development kit (SDK) provides methods to create and manage users that use their email addresses and passwords to sign in. Firebase Authentication also handles sending password reset emails.

Firebase Security Rules leverage extensible, flexible configuration languages to define what data your users can access for Realtime Database, Cloud Firestore, and Cloud Storage. Firebase Realtime Database Rules leverage JSON in rule definitions, while Cloud Firestore Security Rules and Firebase Security Rules for Cloud Storage leverage a unique language built to accommodate more complex rules-specific structures. Firebase provides a rule-based access control system that allows developers to enforce fine-grained access control to data stored in Firebase services. Use these rules to restrict access to sensitive data and ensure that only authorized users can access.

Supportability Requirements

Coding Standards – Our system will be coded in a range of 75-80% of coding standards for HTML 5 and CSS3. The code will be routinely reviewed, tested, and validated via third-party compliance software.

Naming Conventions – HTML classes and id tags will be coded in lowercase except in naming situations involving multiple words, for which the camelCase convention will be applied. Firebase collections, documents, and fields will also adhere to this convention.

Availability Requirements

Accessible Times – Our system should be available for use 24 hours per day, 7 days per week, unless catastrophic unforeseen circumstances arise due to our application being hosted upon GitHub.

Downtime Impact – General downtime will be largely non-existent, save for brief events in which our team updates the software, which should only result in about 5 to 10 minutes of unavailability.

Support – There will be a support feature accessible via the navigation bar which will pull up a template that allows the user to express concerns/report bugs which will automatically be routed to the team's email account.

High-Level System Architecture

Visual Studio (IDE)

- Hyper Text Mark-up Language (HTML) – will be the language that will allow the browser display the website
- Cascading Style Sheets (CSS) – will be the language used to decor the web pages
- JavaScript – language used for client side funcitonality that will be handled for User Interface (UI) needs to make the user experience enjoyable
- jQuery – a fast, small, and feature-rich JavaScript library designed to simplify HTML DOM tree traversal and manipulation, as well as event handling, animation, and Ajax much simpler with an easy-to-use API that works across a multitude of browsers.
- Bootstrap – a front-end framework that is used to create responsive, flexible, and mobile-first websites, which means that web pages created with Bootstrap will automatically adjust their layout and content to fit different screen sizes and resolutions.. It provides a set of pre-designed HTML, CSS, and JavaScript components, such as buttons, forms, menus, and modals, that can be easily customized and integrated into web pages.

- chart.js - provides a set of frequently used chart types, plugins, and customization options. In addition to a reasonable set of built-in chart types, you can use additional community-maintained chart types . On top of that, it's possible to combine several chart types into a mixed chart (essentially, blending multiple chart types into one on the same canvas).
- FullCalendar – a JavaScript library that provides a full-featured calendar solution for web applications. It allows developers to easily create and customize interactive calendars, with support for events, scheduling, and more.

GitHub/Git

A web-based platform for version control and collaboration that is widely used by software developers to manage and share their code. It provides a centralized location where developers can store their code, track changes, and collaborate with others on projects. At its core, GitHub is based on the Git version control system, which allows developers to create and manage multiple versions of their code. Developers can use Git to create "branches" of their code, which allow them to work on different features or aspects of a project independently. They can also use Git to "merge" changes from different branches back into the main codebase.

Firebase

Backend framework developed by Google which allows for automated authentication, authorization, and database functionality using NoSQL.

Jira

Software application that allows teams to track issues, manage projects, and delegate tasks.

Discord

Primary internal communication application between team members.

Canvas

Primary external communication application between team and instructor for upcoming Milestone assignments and deliverable requirements.

Team Members

- **Adam Clark –** Back-End Lead/Scrum Master/Product Owner
- **Quang Le –** Team Lead/GitHub Master/Full-Stack Developer
- **Nicholas Moussa –** Front-End Developer
- **Mahmood Sakib –** Front-End Lead/Full-Stack Developer
- **Kyla Tolentino –** Front-End Developer/Graphic Designer

Project Links

GitHub Repository:

- **https://github.com/Q-Wrld97/cen4010_sp23_g23**

Team Web Page:

- **https://q-wrld97.github.io/cen4010_sp23_g23/**

Jira Dashboard:

- **https://teamnightshift.atlassian.net/**

Checklist

a) Team decided on basic means of communications – *DONE* ✓

b) Team found a time slot to meet outside of the class – *DONE* ✓

c) Front and back end team leads chosen – *DONE* ✓

d) GitHub master chosen – *DONE* ✓

e) Team ready and able to use the chosen back and front-end frameworks – *DONE* ✓

f) Skills of each team member defined and known to all – *DONE* ✓

g) Team lead ensured that all team members read the final M1 and agree/understand it before submission – *DONE* ✓

Appendix

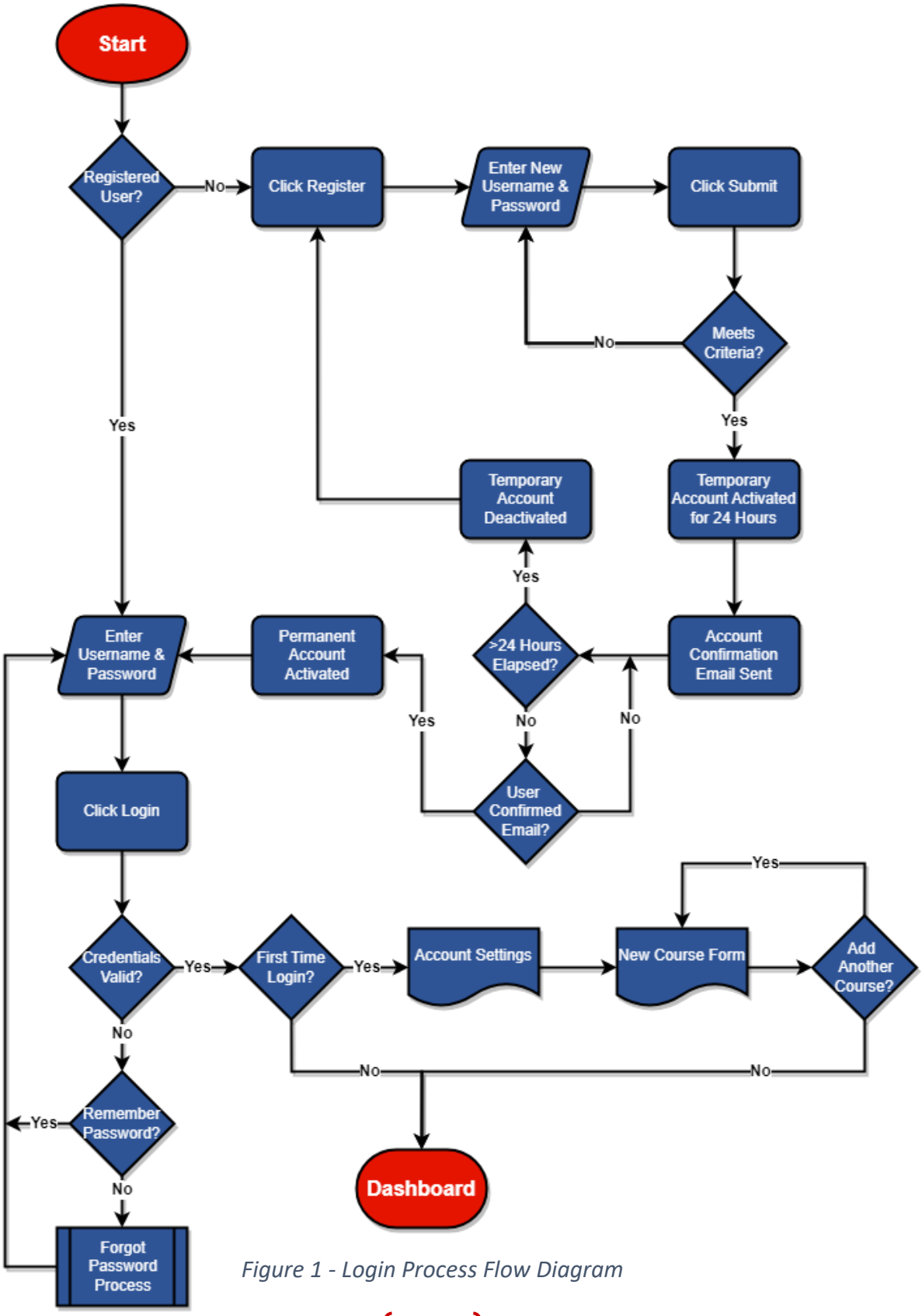

Figure 1 - Login Process Flow Diagram

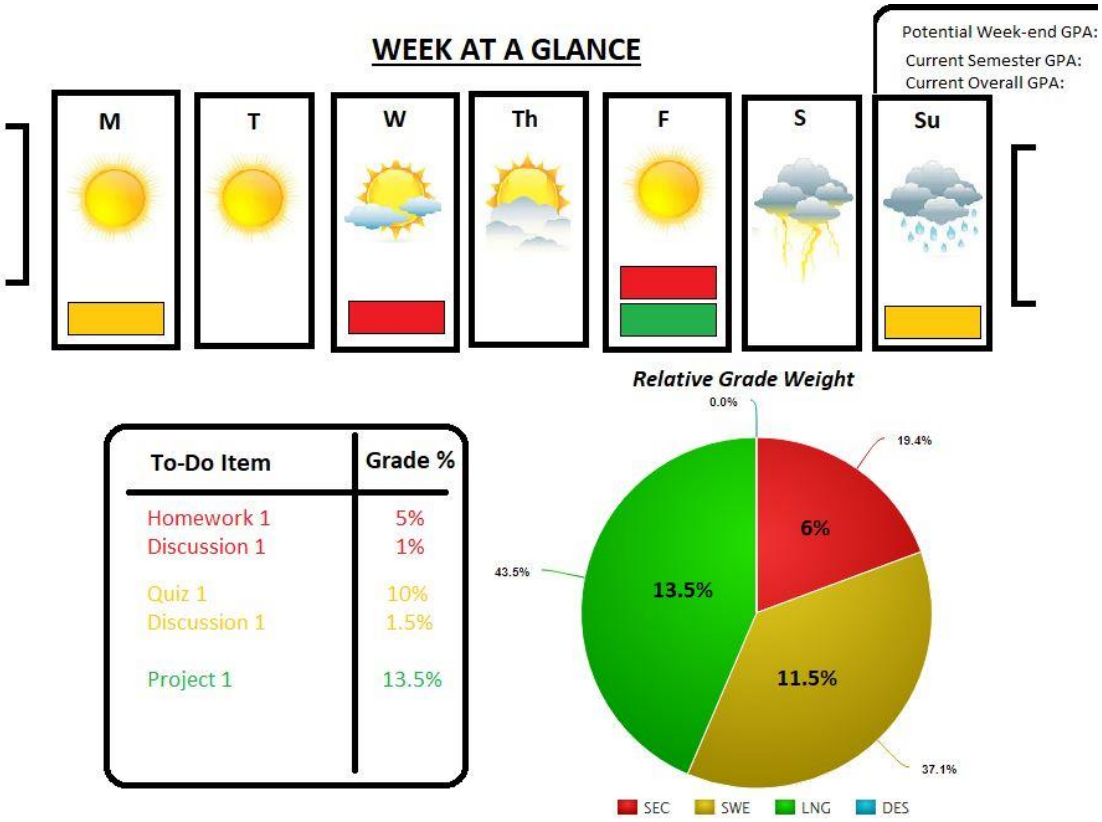

Figure 2 - Week At A Glance Concept Art

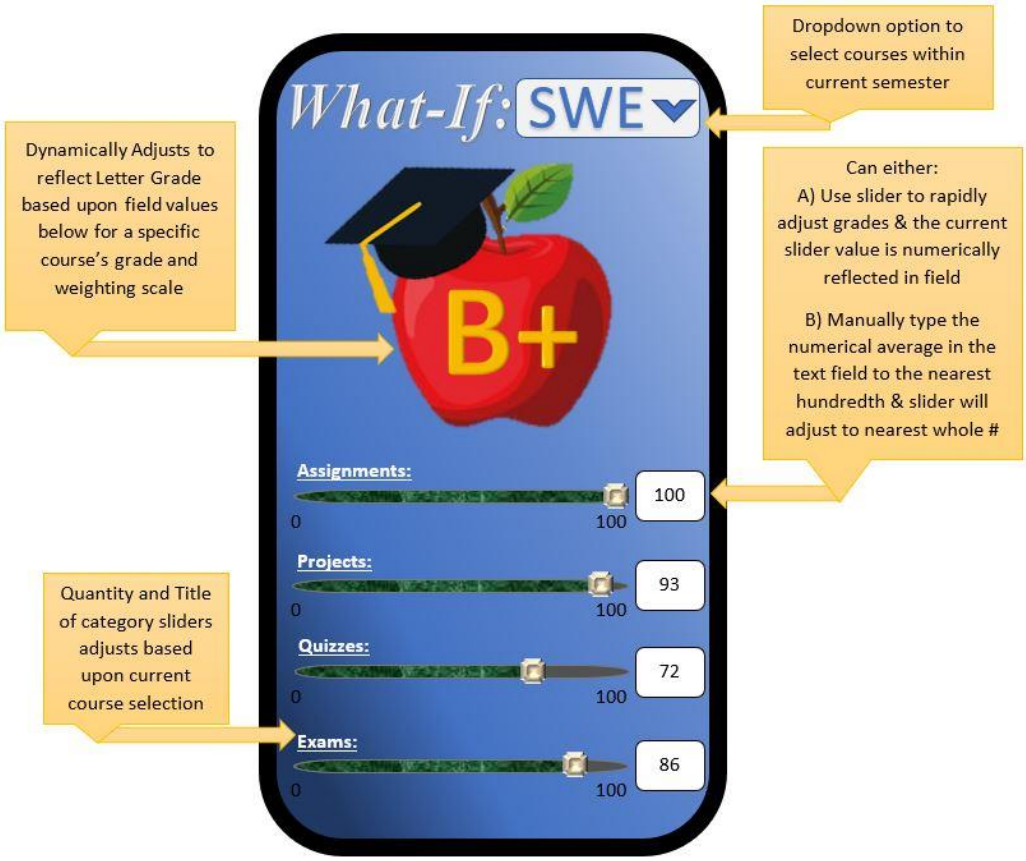

Figure 3 - What-If Concept Art

Figure 4 - Grade PlanAlyzer Logo