Logo

Description automatically generated

ASSIGNMENT

200 Marks

Internal Project Phase 2 and 3

|  |  |
| --- | --- |
| **Submission Details:** | Please upload this document to the Phase 2 and 3 assignment drop box. |
| **Late Penalty:** | **10% deducted each day this assignment is late so you can still submit late and get a decent mark within a reasonable time frame**. |

Contents

[The Teams 2](#_Toc126868342)

[Submission Requirements 2](#_Toc126868343)

[Supporting documentation: 3](#_Toc126868344)

[Daily Stand-Ups: 4](#_Toc126868345)

[Identity Framework User Experience 4](#_Toc126868346)

[PayPal 5](#_Toc126868347)

[SenGrid 5](#_Toc126868348)

[GitHub 5](#_Toc126868349)

[Relational Database 5](#_Toc126868350)

[Personal contribution 6](#_Toc126868351)

[Presentation 6](#_Toc126868352)

[Technology Reminder 6](#_Toc126868353)

[Debugging Tips 7](#_Toc126868354)

# The Teams

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Red | Adam | Cat | Mike | Haley |  |
| Blue | Byul | Stephanie | Kevin | Yuko | Eric |
| Orange | Tony | Ginni | David | Lisa | Ceilidh |
| Green | Sheng | Brayden | Lauren | Samuel | Sara |
| Yellow | Mason | Scott | Renz | Mario | Kriti |

# Submission Requirements

Make sure your submission package is easy to navigate. The grading rubric is as follows:

* Supporting Documentation (30 marks)
* Daily Stand ups (20 marks)
* Identity Framework (20 marks)
* PayPal (20 marks)
* SendGrid (20 marks)
* GitHub (20 marks)
* Relational Database (20 marks)
* Personal contribution (30 marks)
* Presentation (20 marks)

Marks will be deducted from your final score when:

* + Essential features are not met.
  + Design models are not current and or not consistent with the final code submission.
  + The installation package is not complete and error free.
  + Security standards are not met.
  + Responsive design is not implemented.
  + There is Poor alignment.
  + Has a sloppy appearance.
  + Uses poor naming standards.
  + There is duplicated code.
  + Values are hard coded.
  + Methods and functions are not modularized.
  + Data is unrealistic.
  + There are bugs.
  + You do not contribute to the presentation.
  + Your personal contribution to the project is much less than the other members in your group.

## Supporting documentation:

Host the documentation in a Google Drive team folder so they can be viewed and kept up to date dynamically by all members in your team.

Always keep the documents in the Google Drive folder current. Please do not show old versions of the design documents or folders. Points will be deducted if the files and folders are redundant or irrelevant to the project. These can confuse stakeholders and may even lead to misunderstandings about what is being built.

**Important:** Please make your folder public so I can access your team folder on Google Drive.

1. **Installation document should contain:**

* The project code-base location. A link to the final GitHub repository is sufficient.
* Easy to understand instructions on how to unpack and install your application so to eliminate any guess work.
  + Special instructions on the initial Identity Roles setup if applicable.

1. **SQL Script:**

Provide a document containing the SQL script. I recommend having a script to seed the database as well because it undertakes many iterations during the development process.

1. **Secret File:**

Implement a secret file to store sensitive information, such as PayPal, SendGrid API keys as well as the database connection string.

1. **Project Specification Document:**

Project Overview: In one paragraph, describe the purpose and or function of your project.

Functional Features: Provide a current listing of functional requirements under essential, important, and nice-to-have categories. Please do this on one page. Find a way to keep this organized and easy to read.

Gantt Chart: In Google sheets, build a GANNT chart with daily units. On the chart show hours for tasks. Phase one should be complete. Group all phase 2 and 3 project activities in 8 or less tasks (combined total). Highlight the critical path. A link to the

Use Case Diagram: Provide a use case diagram to show the system goals and actors involved for essential features only. Use diagrams.net’s UML option to draw your USE Case diagram. The screenshot of each diagram should fit on about a half page each and they must be presented so they are easy to read.

Provide a use case diagram for each of the following actors:

* Public facing user.
* Registered customer.
* Administrator

ERD: Provide an ERD to show the entity set relationships.

Wireframes: Use draw.io to create essential wireframes only. Marks will be deducted another tool is used. Do not waste time making graphics, fancy CSS, or JavaScript in this phase. Stay focused on page layout for link navigation, control alignment and position.

Use static realistic looking data content. Do not use ‘lorem ipsum’ or data that has no meaning such as ‘asdf’ for the address. Do not create overly silly data such as user accounts for ‘Donald Trump’ or ‘Mickey Mouse’ – this does not look professional.

Risk Assessment: Identify the top 3 risks which could reduce your team’s ability to meet their schedule deadline for implementation. Rate the likelihood, impact, and detection difficulty on a scale of 1 to 5 where 5 is the highest. Also indicate when each risk is most likely to occur.

Risk Matrix: Indicate how each risk in the risk assessment can be managed. Describe your contingency plan, the symptoms which may appear to detect if the risk event is about to or has occurred and assign more than one person to be responsible for managing this risk.

## Daily Stand-Ups:

Create an Agile scrum document and add to it each member in your group. Then I would like your group to conduct daily stand-ups (for each project phase 2 and 3 class only). Each member is to take turns updating the document with what is discussed in the Agile scrum.

Discuss the following:

* What did you do yesterday on the project.
* What will you do today on you’re the project.
* Is there anything that is concerning you with your current task.

**Note:** If you have an issue with the project, then discuss it. This is where you can identify potential roadblocks and fix them, so they don’t jeopardise the project.

## Identity Framework User Experience

Below are the minimum requirements for each user in your application.

Public Facing:

* Access to the application Landing page and other pages not restricted to registered users and administrators.
* Allows for easy navigation to registration and login pages.

Customer:

* Access to the application Landing page and other pages not restricted to administrators.
* Access to the Shopping related pages.
* Access to the logout functionality.
* After registration a customer role must be added to the new user.

Administrator:

* Access to all pages.
* Ability to add and remove sale items.
* Ability to add and remove Roles for each user.

Deductions for:

* Crashes of any kind.
* Methods that are too large. The code must be broken up into smaller methods.
* Duplicate code. Large blocks of code should not be duplicated anywhere in the application.
* Magic (hard-coded) numbers. Descriptive constants should be used instead.
* Inefficient logical flow. The application processing flow needs to be efficient.
* Unnecessary commented code.
* All currencies are displayed in correct format.
* All forms have the necessary validation.
* Your code does not adhere to the following C# standards:
  + CapitalCase – Namespaces, Filenames, Classes, Methods, Properties.
  + camelCase – local variables.
  + Constants – UPPER\_CASE.
  + Hard coded values are stored in constants at the top of the method.
  + The code is well structured and neat with correct indentation.
  + Comments are placed at the top of each class at the very least.

## PayPal

Your web application should integrate with the PayPal REST API for payment processing. The user should be able to pay for their product/order using their PayPal account.

## SendGrid

Verify user registration via email using the SendGrid email service. Also a confirmation email should be sent to customers upon successful purchase of a product.

## GitHub

Use a GitHub repository to manage the source code and track changes. Then:

* Add me to your private team repository and send me a link.
* Commit your changes frequently – at least once a day. This reduces code conflicts!

**Important:** Individual marks will be lost if team members do not make regular commits.

* Get fresh builds from GitHub frequently.

**Tip:** It is advised to occasionally back-up your source code locally. Things happen and it might just save you. Place your project in a time stamped folder like 2022\_04\_14\_*id*. Then zip it.

## Relational Database

Store the application data in a SQL Server database. Also store in the database the transaction details including the amount, currency, and payment details.

## Personal contribution

You contribution to the overall project will be assessed during the development phase. Up to 10 marks will be awarded if your effort is deemed to be equal to or greater than your colleagues.

## Presentation

Presentations will be held on the afternoon of the final day of the Internal Project Phase 3. Please be organized and practiced ahead so your team is ready to walk up and present with very little set-up time.

* Do discuss:
  + The problem your application solves.
  + How different types of users can use your application.
  + Walk us through each user experience.
* No need to discuss:
  + Obvious features like creating accounts or verifying passwords.
  + Really technical details like blocks of code or walk-throughs of the ERD.

I just ask that you be professional and show your best to all of us.

## Technology Reminder

Icon

Description automatically generatedLogo

Description automatically generated A blue letter on a white background

Description automatically generated with low confidence Icon

Description automatically generated Logo, company name

Description automatically generated

Core technologies:

* The application will be built using C# .Net.
* The data is stored in a SQL Server database.
* Users will need to log into the site and register using email verification.
* The web site will take payments using pay pal.

## Debugging Tips

* Use the debugger every day - it will make you far more productive. I guarantee anybody who does this will be glad they did.
* Read your error messages and try to understand them. Usually the first relevant error message in the list is the most important one so fix it first.
* Search Google and Stack Overflow for fixes according to your error messages. Look at the ratings in Stack Overflow to help determine if the answer is credible.
* Remember, if the feature you are trying to implement is something many people want then there probably is a solution for it on Google.
* Compare your code with a working sample to see what is different.
* If the program crashes at a complex instruction and you can't fix it, try breaking the instruction into several simpler instructions and then step through them.
* If your code suddenly stops working and you can't fix it, comment out calls to half of the code in the suspect area and run it again. If the problem still exists, comment calls to half of the remaining code in the suspect area. Repeat this process until your program begins to work. Then, slowly add in your code and test until you find the break point. This process allows you to literally narrow in on the problem at an exponentially increasing rate.
* For tough issues write a simple little program that only implements the feature you are trying to implement so you can focus on baby steps. Don't be afraid to simplify and then transfer your knowledge into your larger program.
* If your program suddenly stops working and you still can fix it, create a new project, and build it up incrementally with your old code while testing it along the way.
* Ask for help if you are stuck for more than 25 to 45 minutes. This may be the only way for me to know you are stuck in a timely manner. I can give hints and guidance in your code to help if you ask. Remember, when you ask for help, I will want to see all your code so please upload it to the appropriate drop box for that deliverable and let me know. Before you hear back, please work on some other aspect of the material, so you do not lose precious time.