Research Manual
Central Games Platform

Kyle Hennessy
C00227463

Table Of Contents

Table of Figures

Abstract

The purpose of the Central Games Platform project is to develop a web based application intended for Windows 10 and Android that acts as a hybrid of both an e-commerce store to purchase and play video games, and an online casino, which is intended to be a one stop shop for gamers and casino players alike. Users will be able to purchase and play the latest and greatest blockbuster video game titles, or purchase casino passes to participate in casino games for a chance of winning real world currency. Video games can either be playable from the browser, or be installed to the supported device, depending on the game. Video games that are available on both platforms will be accessible to the user at no additional cost. The application will allow users to transfer winnings from casino games to a registered PayPal account. Impulsive gambling is a huge ethical issue, so a limit of 10 casino passes purchasable per day will be imposed on the user.

Central Games Platform is designed with a business to consumer/business to business approach. Users purchase games from the platform. Publishers of these games will receive the revenue the game has generated every month with a 15% cut. Casino games will be developed in-house. 100% of revenue from casino games goes straight to Central Games Platform.

Introduction

Today the video games industry is one of the highest earning and most profitable industries in the world, which is estimated to earn over €250 billion from 2.5 billion players world-wide by the end of 2025 [1]. Online gambling, which bears a lot of similarities to video games, is also an extremely profitable industry, which is expected to earn over €150 billion by the end of 2026 [2]. Many customers also play on more than one platform, over 59% of players in fact [3]. As a result, the games that customers have purchased on one platform may not be playable on another platform without buying a new copy and vice versa. Also, many traditional online gaming platforms such as the Google Play Store and Steam do not offer a safe and secure way of winning real world money, which may result in potential players turning to online casino websites that do not treat their customers with respect and feed off of those with gambling addictions.

In Ireland alone, it is estimated that around 40,000 people who participate in any form of gambling have an addiction [4]. A whale, or a high roller, is a player who places huge bets. Naturally, many casinos will prey on these whales and incentivise them to wager bets as much as possible as they contribute the most amount of money to their overall profit margin [5].

The purpose of this project is to provide an all in one solution that will appeal to both traditional gamers who have no interest in gambling and those who might be looking to enter into chance based games for a chance of winning cash prizes that can be used to purchase games on the platform or transferred to the winners bank account. There will be two different sections of the storefront. The first section will host the traditional games where only a one time purchase is necessary for the customer to play as much as they want. It will be possible to have both games playable in the browser or to download an executable for more advanced, high fidelity client-side games. The other section will have all of the gambling type games where cash prizes are given out. The games here will require a “pass” or a “ticket” to enter so for example a player will need to purchase a scratch card ticket if they wish to participate in a game of scratch card. Certain games will not require a pass such as poker, instead players will be free to join but will have to make a bet if they want to participate and be in with a chance of winning. To address the current ethical and financial concerns surrounding gambling, there will be a limit to the amount that a player can bet at any given time. Due to the gravity of dealing with customers' money and fairly distributing winnings, it is important that the integrity and security of the game must not allow for cheating or manipulation of results.

Overview of areas researched

- Programming language
- Web Development Framework
- Operating System
- Platforms
- Payment processing service
- Cheat Prevention
- Servers
- Game delivery system
- Encryption
- Data storage and management
- Source control
- MVC Pattern
- ASP.NET Core Identity
- Stripe Connect
- PayPal Payouts
- Entity Framework Core
- Repository Pattern

Programming Language

The programming language that is to be chosen for this project is very important as the entire architecture and functionality will depend on it. Because of this it is important that the most suitable programming languages are identified for the specific purpose of this application and the pros & cons of each language must be considered before settling on one.

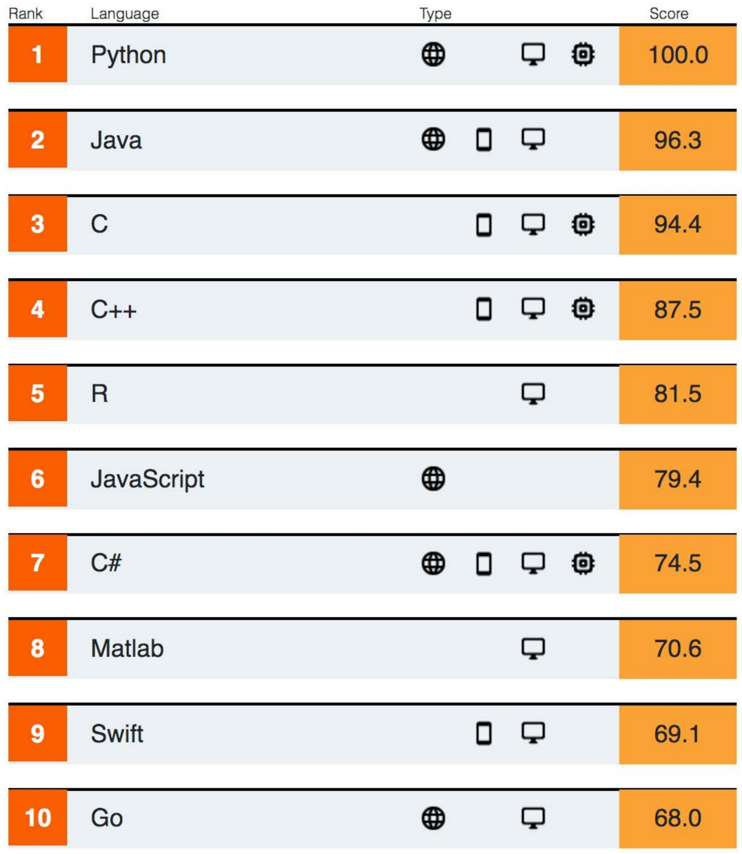
Before a search for a language begins, it is crucial that we first identify the main aspects and needs of this project in order to decide which would suit it best. The main functionality of this online games & app dashboard are that it will be cross platform with PC and mobile devices, allow customers to browse and shop in an ever increasing catalogue of games, will be able to play games directly in the browser or by downloading client applications, will keep track of customers purchases and owned games, and handle purchases and reward money to paying players.

To make our application cross platform with both PCs and mobile devices there are a number of languages that provide the tools for that. However, since this will be an e-commerce application with an ever evolving and growing catalogue of games, it would make sense to make this a web application so the catalogue can be easily updated without the customer having to update a client application to be able to access and view the newly added products.

Of course security is also a big concern for this application as it will be dealing with customers data and money. It would be a huge and potentially expensive problem if a malicious user was able to cheat at games where it would be possible to win cash prizes.

And lastly, an acceptable level of performance will need to be considered as customers will be paying for a game that will be expected to run at an acceptable frame rate, be responsive to inputs and of course load relatively fast.
The player base will also have huge potential for growth so it's important that the web application allows for scalability as its functionality and features will evolve over time.

Now that the main needs of this project is clear, it will make it much easier to narrow down the search for a language. This section will look at each main area and decide which language fits this area best. There are many programming languages out there that are capable of web development but two languages will be looked at that will fit the needs of this project from the list of most common web development languages[6]. It was decided on Python and C# for the potential languages that will be considered.



Rank	Language	Type	Score
1	Python	🌐 🖥 ⚙	100.0
2	Java	🌐 📱 🖥	96.3
3	C	📱 🖥 ⚙	94.4
4	C++	📱 🖥 ⚙	87.5
5	R	🖥	81.5
6	JavaScript	🌐	79.4
7	C#	🌐 📱 🖥 ⚙	74.5
8	Matlab	🖥	70.6
9	Swift	📱 🖥	69.1
10	Go	🌐 🖥	68.0

Figure 1 - Most popular languages for web development. Source: ieee.org

Both Python and C# are general purpose, open source languages with very large and passionate communities behind them.
Python is the most used language for web development in the world[6], being used to power websites like Youtube, Instagram and Netflix[7]. C# may not be number one in the world like Python is but it still comes in very high on the list of most used languages for web development which is used for websites like Dell, Visual Studio and Microsoft[8].

For a beginner, Python is much more easily read as it uses whitespace instead of brackets and it has a very simple syntax that is just not found in C#. However, C# is similar to most other programming languages such as Java and C++ so the learning curve may not be as steep for an experienced developer.

At first Python seemed like it would be a perfect fit for this project given that it was the most used language worldwide, had a very passionate community behind it and is easily readable with its much simpler syntax, but there is one major drawback to using Python over C# that may potentially cause our project to fail and that is Python is an interpreted language meaning that it will take much longer to respond than C# which is a compiled language.
For some applications it can take up to 44 times longer to process a request in Python than in C#[9].
This is simply unacceptable as customers will be playing games that are time sensitive, such as poker, so it is important that inputs made by them will be registered and sent to the server back-end in a fast manner.
It is for these reasons that C# has been chosen as the programming language for this project.

Web Development Framework

While C# can be used for web development, it is not the only tool needed to create a functioning web application, especially one as complex as this project. A Web Development Framework (WDF) must be used as it provides all the necessary libraries, tools and services needed to build and manage web applications and services[10].

There is a wide range of WDFs for C# but this project will require that the framework is fully stacked which means that it has all the tools and solutions, such as built-in authentication and authorization, necessary to create a web application.

By far, ASP.NET 4, which still uses the decades old closed source .NET Framework [11], is the most used WDF for C# as it allows for both back-end and front-end development. It has the added benefit that it is developed by Microsoft meaning that it is guaranteed to utilize every feature available in C# and thoroughly documented on their website[12].
In 2016 however, Microsoft released the first version of a new WDF called ASP.NET Core which is a complete redesign of ASP.NET, using the new open source .NET Core runtime, and was designed from the ground-up to be cross platform with Windows, MacOS and Linux among a list of other features not found in ASP.NET 4 such as built-in dependency injection and the ability to use multiple versions on the same machine [13].

To decide on which framework to choose from, it is important to first look at the advantages and disadvantages of both ASP.NET 4 and ASP.NET Core.

First of all, it is already known that the first version of ASP.NET Core was released in 2016 which is still a relatively new technology which means that there may be bugs or issues that have as of now not been discovered and may not be completely reliable.
Meanwhile, the first version of ASP.NET was released 17 years ago in 2002 and was at first intended to be a major improvement over ASP, which was Microsoft's very first attempt at a WDF, before outright replacing it as their main web platform[14].
While ASP.NET Core is newer and may have more features, there may be undiscovered or not very well documented issues that may cause a problem so it may make sense to use ASP.NET 4 as it has been very well tested and the limitations of it are more clear.

A clear advantage that ASP.NET Core has over ASP.NET 4 is that it allows you to develop apps for Windows, Mac, Linux and Mobile due to its use of the .NET Core runtime. This will be very important as it will allow us to target different platforms further down the line without the need of completely overhauling the underlying code. By choosing ASP.NET 4, essentially locking development to the Windows platform.
ASP.NET Core is also open source, another huge advantage as it will allow us to see the source code and manipulate it if it is necessary.
ASP.NET Core also has much better performance than ASP.NET 4, performing faster in most testing benchmarks[15].

Finally, Microsoft has slowly been phasing out .NET Framework, dropping support for versions 4, 4.5 and 4.5.1 in 2016[16]. They have also said that .NET Core is the future of .NET applications and that all new projects should be using it[17]. ASP.NET 4 may become obsolete in a few years time while ASP.NET Core will be future proofed and will become more and more utilized for years or perhaps decades to come.

ASP.NET Core	ASP.NET 4
Windows, Linux, Mac and Mobile support	Windows only
New technology, not as well tested	Well tested and documented
Open-source, allowing us to see and change source code	Closed-source, strictly no way of manipulating source code
Future proofed, actively being supported	Being phased out, not recommended for new projects

Figure 2 - **Differences between ASP.NET Core and ASP.NET 4**

After weighing the pros and cons of both of these frameworks, it is clear that our project will benefit from ASP.NET Core the most.

Operating System

It is important for us to decide on the best operating system to develop on which will allow us to take the most advantage of C# and ASP.NET Core.

ASP.NET Core has the advantage of being able to develop and build on Windows, MacOS and Linux. The most accessible and most used operating systems are Windows and Linux as of 2019 for software developers specifically with over 50% using Linux and 60% using Windows[18].

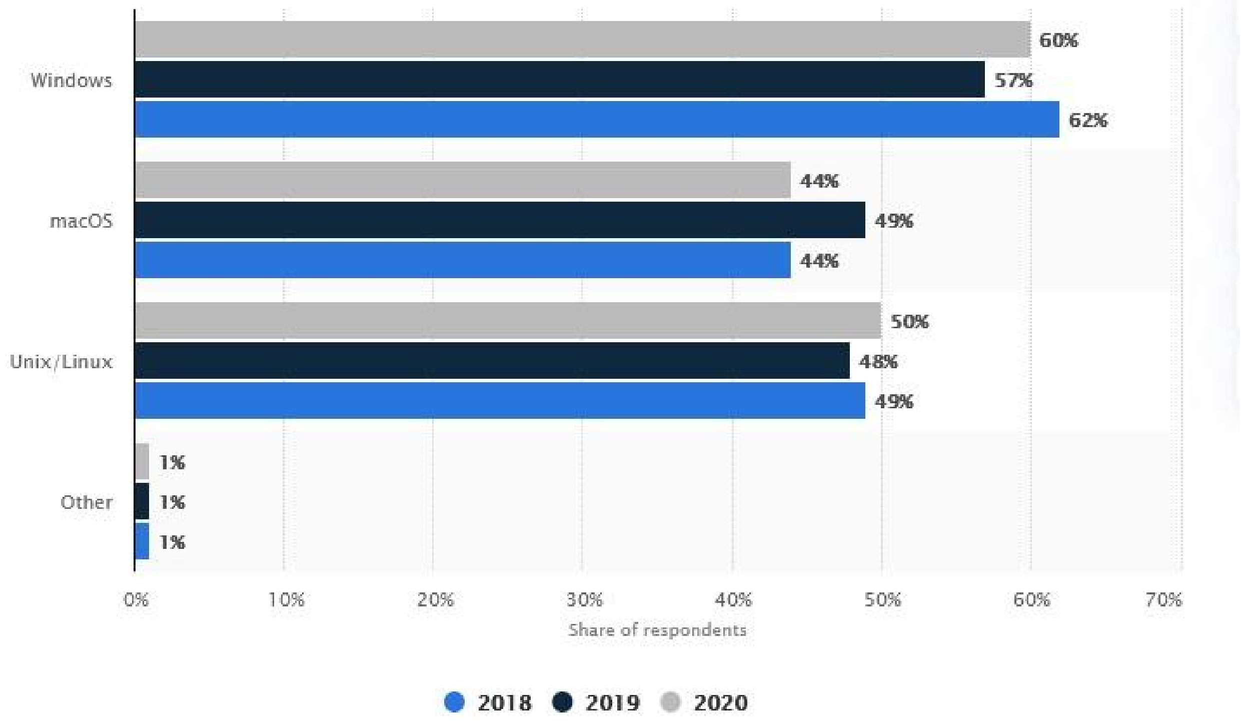


Figure 3 **- Most popular operating systems for software development. Source: statista.com**

For the purpose of this project, the pros and cons of both Windows and Linux will be looked at. The reason why MacOS will not be considered at all is because Apple requires you to own an expensive Mac computer or laptop which is more than €1000 for even the most basic models.

Linux has a clear advantage over Windows in that it is open source meaning it is free for anyone to download, use and edit to their own needs. This will also make it easier to identify and fix bugs in Linux as you can easily edit files whereas in Windows you may be dealing with a proprietary file format meaning you have no idea what is actually going on inside it with no means of editing the file.

Linux also has the bash terminal which is much better than the command prompt found on Windows. The bash terminal is a very powerful tool that allows full access to the system and can do very precise operations and commands that is just not possible using the GUI. In addition it is also used side by side with other applications and makes it very easy to see a history of what tasks you have done. The command prompt in comparison is a very basic tool that is rarely ever used and is only a means to launch applications and do basic file management.

Windows does have the advantage however of being a much more popular operating system in general, meaning that there is a lot more support for it, especially in the context that this project will be using C# and ASP.NET Core which were both created by Microsoft and were at first designed for the Windows platform earlier on before they made it cross platform. Most discussions on the internet to do with ASP.NET Core

In the context of C# development though, the huge advantage of using Windows over Linux is the fact that Visual Studio IDE only supports Windows. Visual Studio IDE is possibly one of if not the best IDE for C# given that it was also developed by Microsoft. The built-in intellisense for C# will drastically speed up development time as it provides autocomplete for variable names, references, variable types and so much more [19]. Visual Studio also comes with a compiler and debugger, making it the only piece of software needed throughout the whole coding process. If Linux were chosen for the development, it would be necessary to install individual components for compiling and debugging, increasing the complexity of the project.

For these reasons, it is clear that Windows is the best choice for the operating system for this project as it will get the most out of the C# language and ASP.NET Core, although this project is now heavily invested in the microsoft ecosystem.

Platforms

The platforms that will be the target for the application will be very important as it is desirable to have as many people playing as possible, as well as making the development process as easy as possible by avoiding any incompatibilities that may occur between different operating systems, hardware specifications etc. It is already known that the target devices will be PCs and mobile devices but it is important to distinguish which specific platforms that the project will be developed for as there may need to be design decisions made based on what platforms the project is actually targeting.

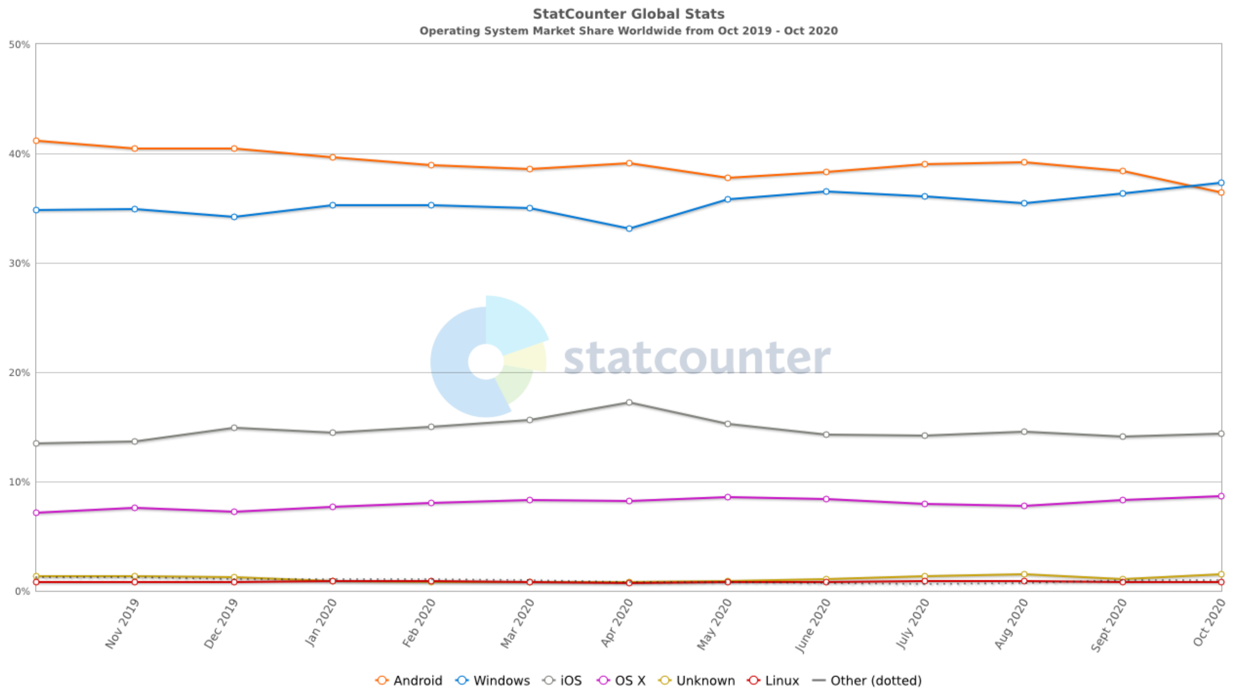


Figure 4 **- Most popular platforms worldwide as of October 2020. Source: statcounter.com**

Although it may seem like Apple has a monopoly over the mobile platform market with the iPhone, Android is actually the platform of choice for many mobile users[20].
This is no doubt due to the insane barrier of entry into the iOS ecosystem with the new iPhone 12 costing €913 directly from the Apple store, meanwhile android devices with arguably better features are available for much cheaper.
Not to mention that to also release an app for iOS devices, you must also need to own a Mac computer which was already discussed earlier in this document. All that is required for Android is a mobile device to test on making it a far cheaper alternative.

While Linux is one of the most used platforms used by software developers, for general purpose use by the greater population it doesn't even come close to Mac OS X. This could possibly be due to the higher learning curve associated with Linux and many PCs come with Windows pre-installed. It's very apparent that Windows is not only the most used PC platform, but also the most used platform in general, just narrowly beating Android.

Taking all of this into consideration, clearly the project will benefit from the two most popular platforms: Windows for PC devices and Android for mobile devices.

Payment Gateway Provider

A payment processing service for an e-commerce website is very important as it makes it possible for a customer to securely make a purchase by using a credit card, debit or any other form of electronic payment. In the context of this application, it will also allow us to send winnings to players. The big advantage of using a payment gateway provider is it takes away any security concerns with storing and handling payment data as it is handled by a reliable third party.

There are many payment processing services out there but first it must be understood how an ecommerce site works before choosing one.
Firstly, there are three actors involved. There is the customer who will be checking out after selecting some products and providing their credit/debit card details, the vendor website which will host all of the products and manage orders/transactions, and a payment gateway provider that verifies payment information and acts as a middle man which sends/receives to and from the customer's and the merchants bank[21]. Fortunately payment gateways do a lot of the communications between different banks and can successfully determine whether a payment method is valid or not.

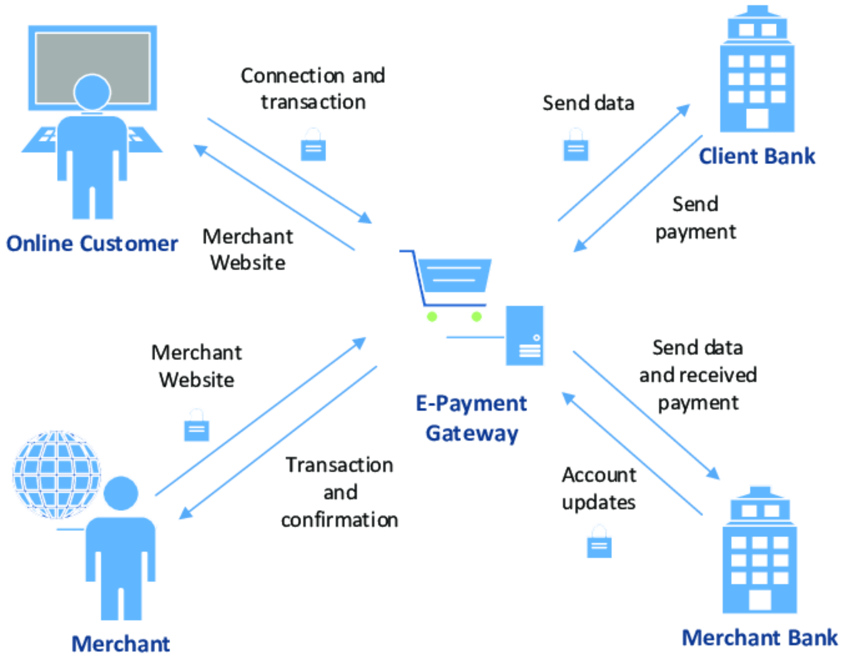


Figure 5 - **Payment gateway model. Source: researchgate.net**

Now that a brief understanding of how an e-commerce site works and how integral a payment processing gateway is to an online business, some payment processing solutions can be looked at to see which of them out there will allow the application to not only manage and handle

payments, but that will also let us set up a development environment to allow us to test payments without using actual real world money.
At the moment there are a lot of different companies offering payment processing solutions each with their own pros and cons. At its core, a payment gateway is just a way for us to send and receive money meaning that a lot of solutions out there will be offering essentially the same service.
The main thing that differentiates them is the fees associated with purchases and the integrated tools associated with the APIs they offer.
If a smaller solution was chosen for the purposes of this project, it may be cheaper per transaction but then the project will be missing out on extremely useful tools and documentation offered by more larger, established providers. It may make more sense to use a smaller solution if there was prior experience using one before and knowing exactly how it works, but in this case it would no doubt slow down the development process and may result in an unfinished end product.
Of course a key feature of this application will be the ability to send money to winning players, or payouts, too so this will be a deciding factor in which solution will eventually be chosen.

Paypal is a very attractive payment gateway provider, and arguably the most used, as it has been tried and tested due to it being around for years and is instantly recognisable to any customer who is familiar with online shopping but there is another payment gateway provider that may actually be a better fit and that is Stripe.
Stripe is a relatively new payment gateway provider which was actually designed for web developers, offering greater customization, more detailed documentation and more powerful developer tools just not found with Paypal.

There are actually many similarities between Paypal and Stripe. For one, they both offer the exact same competitive pricing of 2.9% of a purchase and a base $0.30 fee[23]. They both support the use of multiple payment types, send payouts to customers and vendors, and create automated invoices for transactions.
But there are enough differences and uses for each that will drastically affect this project. For one, Paypal is easier to set up and work with due to it being designed with convenience in mind for smaller businesses that may not have a team of developers working on it. As a result of this though it offers little customization and developer tools, with sandbox accounts being the primary way of testing.
Stripe on the other hand, as mentioned previously, was designed for developers. Stripe’s RESTful API and official libraries are available for various languages and platforms. Also the Stripe command line interface allows you to bootstrap projects, send requests to the api for testing, and allow debugging of integrations of the API.
Stripe just offers so much more freedom in the development process that is not possible with Paypal’s restricted tool set.

While it may seem like Paypal would actually speed up the development process because of its simplicity, that may not actually be the case. While there is a .NET SDK for .NET applications that is built on PayPal's REST API, it has actually recently been deprecated and "no new features or support requests will be accepted"[24]. The .NET SDK also only supports the older .NET Framework and not the newer .NET Core runtime without some workarounds that are not even guaranteed to work for an application of this scale.
Stripe on the other hand officially supports .NET and is regularly updated to support all the latest features and security updates. All of the official documentation for the Stripe API will have code snippets in C# in the context of a .NET application[25].

Given that PayPal is actually extremely basic when it comes to customization, offers less freedom over presentation of UI and doesn't officially support ASP.NET Core, it is clear that Stripe will best suit this project and its requirements.

<u>Cheat Prevention</u>

The main selling point of this project is the ability for a customer to potentially win money through pay per play games in the app. It is very important that the integrity of the games that reward real world money are maintained and that users are not able to cheat by injecting malicious code into the game or otherwise changing the outcome of the game in their favour which would result in a huge loss of money.
To prevent cheating, it is first important to identify how cheating could occur and then decide which way to go about potentially preventing it.
Since the games where it will be possible to win money in will exclusively be playable in the browser, they will of course either be coded in HTML5 or more likely JavaScript as it is capable of doing very fast logical computations that may be required in a game, has many game development frameworks and also allows for DOM manipulation[26].
However, although JavaScript is a great language for the use of browser games, there is huge potential for players to cheat at the game since all JavaScript code will be visible to the browser and in turn the end user.
There is nothing stopping a malicious user from viewing and editing the games source code which would enable them to possibly change a losing scratch card into a winning one or increase the amount of money that they have won.

Because JavaScript runs entirely in the users browser, there is no way of hiding the code from them. It is possible to add some more steps to be able to access the code by including it as an external JavaScript file. It can be brought another step further and can be obfuscating the code by removing all white space and comments from the code and shortening all variable names to make the code less readable to a human.

This is typically only done to save space but could also make it harder for anyone to understand what's happening in the code. All of these measures could be put in place but that still wouldn't stop someone with enough time and patience from reverse engineering the code. It would essentially just be buying some more time before a method of cheating is eventually discovered. In the end it would just be delaying the inevitable by attempting to hide the code from the user.

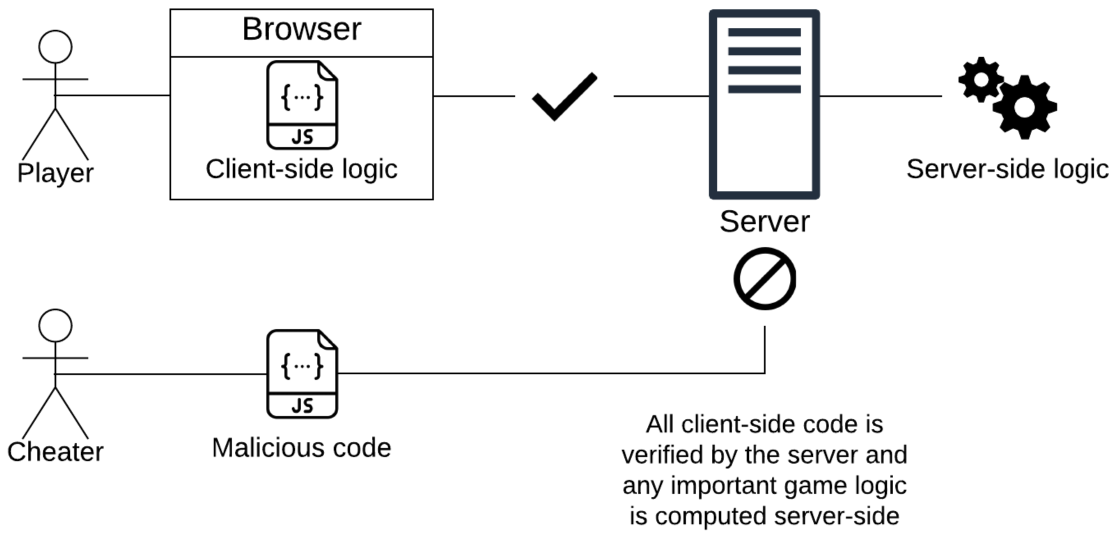

Browser
Player
Client-side logic
Server
Server-side logic
Cheater
Malicious code
All client-side code is verified by the server and any important game logic is computed server-side

Figure 6 - **How cheating could potentially be prevented**

However, there is a potential way that cheating could be prevented and that would be to handle all important game logic on the server-side, leaving the player with no actual access to any code that could potentially change the outcome. Additionally, for extra security measures any potential winners client-side code could be compared with a copy of the actual code on the server to see if there are any differences and if there are then those winnings will be disqualified and the account associated with the player will be banned.
This could be an effective way of preventing any potential cheating that may occur but it also may have some disadvantages to doing it this way. While handling all important game logic on the server side will prevent a cheater from gaining access to that code or unfairly changing their result to a winning one, it also means that there is a greater increase in load on the server and it also may actually make a game slower for players. To circumvent that, a combination of both client-side and server-side logic will be used to minimise server load and increase scalability.

Server Solution

Previously it was suggested that to verify clients game code and compute important game logic on the server, so it is important that the different types of server hosting solutions are identified to see which one would best suit this project.

One of the most popular and cost effective solutions to online gaming is the listen server model. The listen server model allows a player to utilize their own PC and turn it into a server that other players can connect to. This however would not suit this project as it would leave complete control of the servers into the players hands which would compromise the security of the game's server-side logic.
Instead a dedicated server would be required to fit the particular use case of this project. A dedicated server removes responsibility from the player, giving all control to the owner of the server.
This would benefit this project the most as the games featured on the web site will have to use server-side logic to effectively prevent cheating. Of course a dedicated server will be required for the games but an all in one server solution would be best to not only allow the hosting of games, but also to host the entire website and database.
There are two different types of server solutions that may benefit this project. These are hosted or SaaS (software as a service) solutions.
Hosted servers are servers that are owned by the creator of the application. These servers can either be rented or bought from third parties, with the owner having full control over everything on it. They allow for full customization and leave all control to the owner of the server, meaning that it is up to the developer to set up everything and to deploy their web app. This means that security such as encryption of user data will need to be handled by the developer also.
SaaS server solutions meanwhile, make it very easy to create and host web apps as there is almost no setup required at all on the developers side so deployment of web apps is quick and easy.
While hosted servers are relatively inexpensive, SaaS solutions are not. It is possible to find a wide variety of cheap or even free servers out there that would suit the needs of this project. However, although Microsoft Azure operates on the SaaS model, they do offer a free solution to deploy a web app with 1GB of space which should be more than enough to fit the needs of this project, while also handling the security aspects of the application.
Additionally, because Azure is a Microsoft product, ASP.NET Core is fully supported meaning that any .NET web apps can be deployed quickly and easily while also getting the full use out of the framework which is simply just not possible with other server solutions

Game Delivery System

The way in which games are delivered to the customer is very important as it enables them to actually play the games that they have purchased or bought a ticket for. For the traditional games purchased on the store, a download to an executable or installer could be provided on the games product page, or in the case of a browser game, be embedded directly on to the web page.
However, the games that offer monetary rewards to winners will not be as easy as just providing a download to install it on the customer's device, as this would potentially create vulnerabilities in the game making cheating more likely to happen. Instead these games will be only playable in the browser or in the mobile app, giving players as little access as possible to the game logic.

There are many ways that this can be done in the context of a browser game. One way that it could be done is to have all of the games code directly on the game page. This would be very inefficient and make the code for the web page very messy as there will be a huge amount of javascript and html code associated with the game.
Instead it would make sense to keep the games code separate from the page that the game will be played on. This can be done by embedding it into the web page using a canvas. An canvas allows you to embed documents, videos or interactive games on a main web page. It essentially lets you display a secondary web page on the main page. The game file would be stored on a separate page that will be inaccessible to the user unless they are viewing it through a canvas.
This could also be used for games created by other game developers with their games hosted on their own server, since a canvas can embed web pages from other sources and not just the server that the main website is located on, although it would not be possible to incorporate the method of cheat prevention that was decided on for this project.

Data Storage and Management

For almost any web application, especially an ecommerce one that has a big catalogue of products and a vast number of user accounts, a method to efficiently store and manage all of the raw data is essential. A database management system can be used to store and analyze all of the data associated with products, user accounts, order details, and so much more. It allows you to see relationships between all of the different categories of data.
There are two main types of database management systems currently available. These are relational databases, also known as SQL databases, or non-relational databases which are also known as NoSQL databases.

SQL databases are typically used for data that can be easily categorized and stored in a tabular format which can be joined together and cross referenced with other tables. Data in an SQL database will typically not change much at all after it is initially recorded, which would be more suitable for web applications that feature user accounts, catalogues of items or ability to order items where most if not all of the data can be easily structured. Multiple database objects are joined together into a collection, known as a schema. Schemas can be easily searched by both humans and computers.
NoSQL databases, however, store data in a non-tabular format. They typically use a key-value pair to be able to identify data, meaning it is quite simplistic and bare but it is able to read and write data much more quickly and in higher quantities. NoSQL databases work best when working with unstructured data or both a mixture of unstructured and structured. They would be best suited for a social network where a vast amount of posts are generated that contain either text or images.[27]

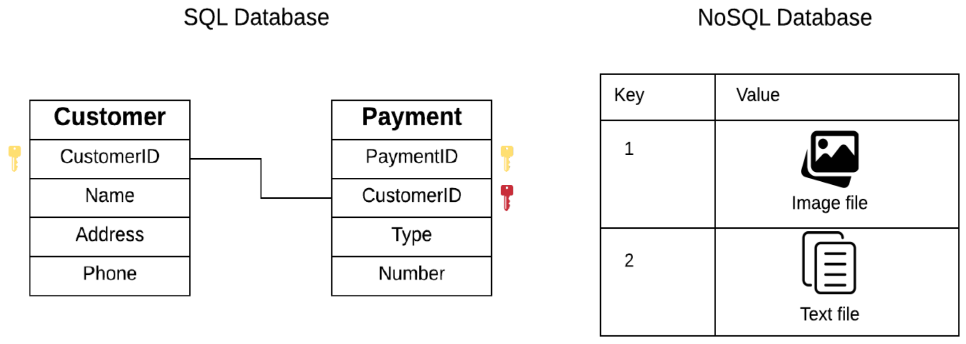


Figure 7 - Comparison of SQL and NoSQL Databases

Given that this is an ecommerce application first and foremost, it would make sense to use an SQL database as all of the data associated with it will be structured.

Since the project is going to be an ASP.NET Core application for the windows and android platforms, and the fact that it is going to be hosted on Azure's Cloud App Service, it makes sense to use Azure's SQL database solution as well. This can be directly linked to both the development local host build and the production app service build of the application. There are a number of pricing tiers available but most would easily exceed the free €100 offered to students. For that reason, the basic SQL Database tier will be used, as it offers enough storage for the needs of this project while only costing a mere €5 a month.

Source Control

While source and version and control is not entirely necessary, it would be foolish not to incorporate it into this project given the scale and complexity of it. If something were to break by implementing a new feature or to happen to the original source code such as a hard drive failure then the entire project would be lost and would have to be started again from scratch. Additionally, it would be very useful to see a timeline of every change made on the project so any new bugs can be traced back to a change in the code from implementing or removing a feature for example. The added benefit to using source control also means that more than one developer can work on the same code at the same time.

That is where source and version control comes into play. It allows you to see the entire version history of a project and see who made what changes, and create a backup of the entire project's source code. Version history is especially useful because if a new version of the project is not working as intended, it is easy to go back to the previous version to compare the new code with the old.

The most used source and version control out there is Git. Git is an actively maintained and open source project that has been around since 2005. It is a very well documented method of source and version control that a huge number of projects are using ranging from small startups to huge corporations such as Google or Netflix[28].

Git allows you to set up a local repository on your machine that will keep track of an entire project. A repository is said to be like a tree, with the latest working version of it being called the master branch. Any older versions would be behind the master branch which can then be jumped to by setting the "HEAD" to it. After a local repository has been initialized, the project would then be worked on as normal in the working directory, making changes to code or creating new files as necessary. When a new feature has been implemented or a bug has been fixed in the code, any changes made would need to be added to the staging area to be committed to the local repository. After every file has been staged, they must be committed to the local repository for these changes to actually be saved.
While a new version of the project has been created and a backup of it is now available, it is still only just a local repository meaning that it is only accessible on that machine it was created on, which still would not protect the source code by being lost from hard drive failure.

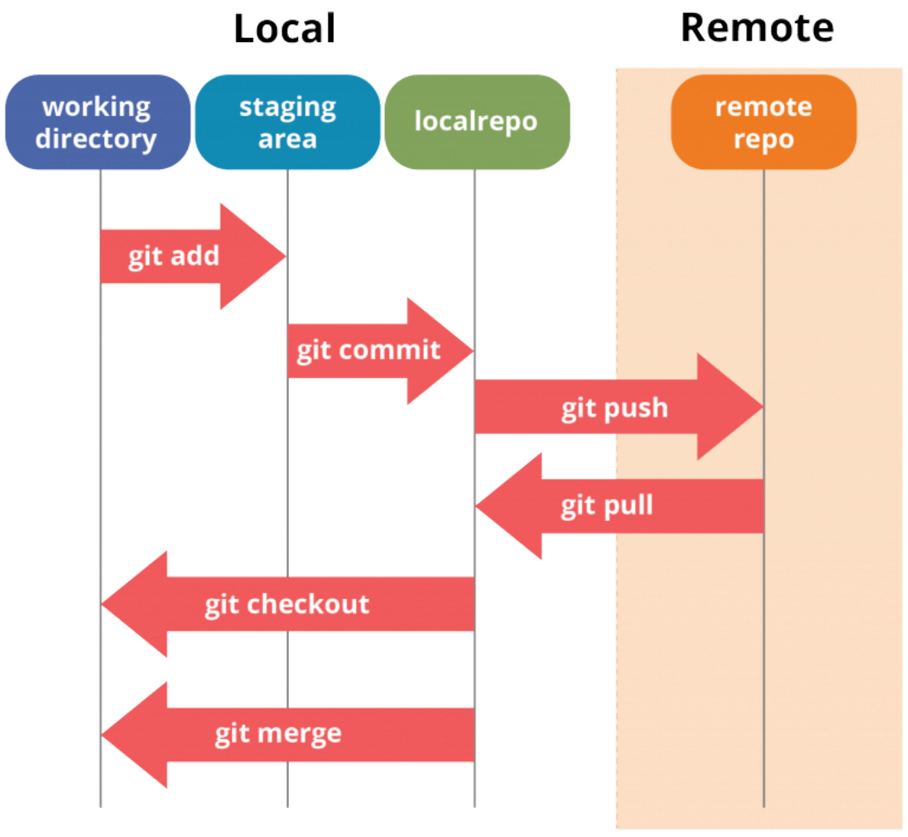


Figure 8 - **The different states in git and the commands used to swap between them. Source: edurka.co**

The repository must be pushed to a remote repository located on a website or server. While setting up a server for a remote git repository could be an option, it would make more sense to use a service like GitHub which allows you to create a remote repository which will be available for anyone to look at a project's code, making it more accessible to anyone while also making an online backup available from any device with an internet connection which completely removes the worry of losing source code due to hardware failures or user error. It also gives repository owners the option to share write access with certain developers or to make it open source entirely allowing any developers to collaborate and contribute code of their own to the project. Although this project will not be worked on by more than one developer, it still makes sense to create a remote repository to back up all of the code and to keep a history of each version.

MVC Pattern

When designing a large scale web application such as this one, it is very important that the right design pattern is used for many different reasons. A design pattern can have many different applications but the main goal of most of them is to have a structure in place that allows for easy modifications by lowering the level of coupling and increasing the level of cohesion between classes.

Additionally, the separation of concern design principle is used to separate software into each section to address one and only one concern.

That is exactly what the MVC design pattern aims to achieve. The MVC design pattern is one that is used for creating GUIs where logic is separated into 3 distinct components.

There are the views, where any information that can be displayed graphically to the user is done so here. In the case of web development, or this project in particular, the views will be the HTML pages that will be rendered on the user's browser.

Then there are the models, which is the foundation of the MVC pattern. Here is where all of the data and business logic is clearly defined. The model is used to supply data to the view which the user will be seeing.

Finally there are the controllers, which processes input sent by the user. This input can be sent through POST or GET requests via forms, or it can be as simple as redirecting a user to a web page upon pressing a button. The controller is also directly responsible for accessing or modifying any properties in a model [30].

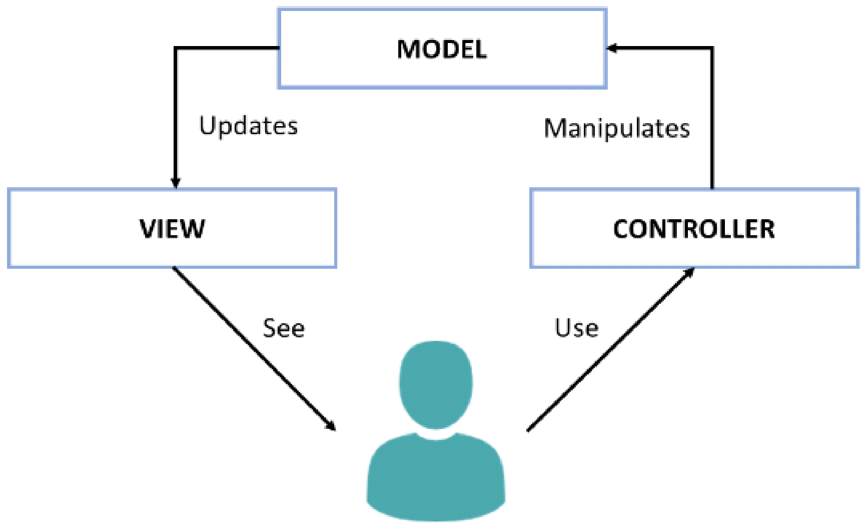


Figure 9 - Diagram illustrating the MVC pattern source: researchgate.net

ASP.NET Core has an extension that enables the use of the MVC design pattern. It contains all of the functionality of ASP.NET Core while offering some additional functionality intended to be used for the MVC pattern.
For one, it provides a Controller abstract class that is to be implemented by any other classes that you wish to use as a controller. This makes it very quick and easy to clearly define a controller to begin coding very quickly.
Model binding is a great feature that will automatically convert data sent via HTTP requests into objects so that they can be manipulated in the controller without any sort of modification. It is also possible to validate data sent in a model to ensure that there are no wrong data types or invalid data.
There are many more features in ASP.NET Core MVC but there is one that will be of great use for this project and that is the Areas feature. Areas provide the tools required to separate sections of a web app into Areas, to further obey the separation of concern design principle. This will be used to separate casino games into their own areas, keeping them away from the main functionality and logic of the web app.

ASP.NET Core Identity

For any large scale web application such as this, there will be a large number of users who will need a way to access their purchased products and only their purchased products, as well as view any details on any information related to their account. This is where authentication and authorization comes into play. In the context of any software where there are user accounts involved, authentication is a way to verify that the user is who they claim they are. This is achieved by requiring a username, email address, or password to be submitted by them before they can access their account. Authorization on the other hand, is a way of granting permission to a user for certain actions. In other words, must you be logged in to access the games library? Well yes, as this is information that is directly related to a user account, so it must only be accessible by a logged in user. Another use for authorization is to only allow specific users to access certain operations or features. This will need to be done to implement an admin page, where only users with admin privileges assigned to them will be able to access it. The admin page will allow for the creation, editing or even removal of games off of the website, so it is important that authorization is implemented [32].

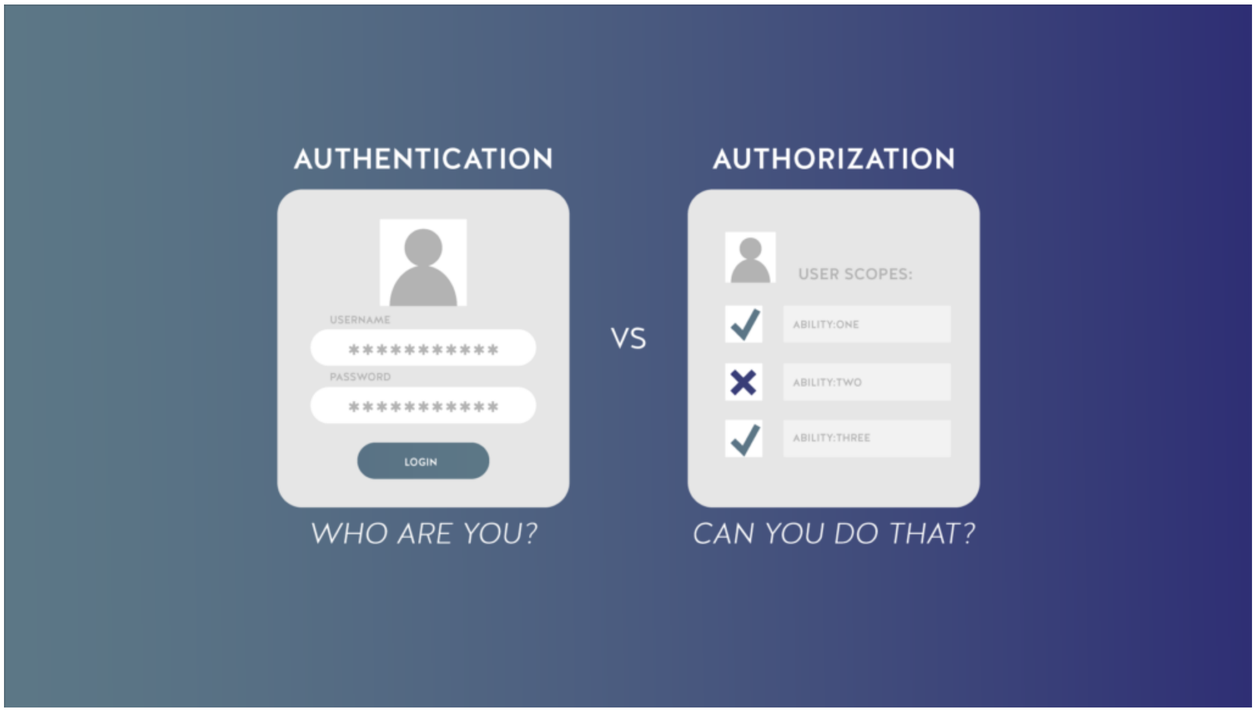


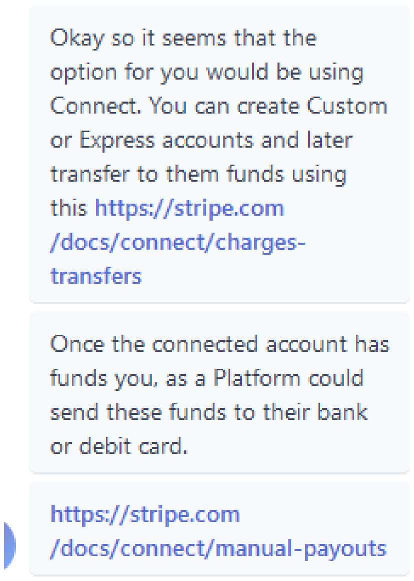
Figure 10 - Authentication vs Authorization. Source: Nordicapis.com

ASP.NET Core Identity is an API that is created by Microsoft, designed from the ground up for the ASP.NET Core framework. It contains built in tools and functionality for authentication and authorization. It is able to be scaffolded into any project which upon doing so, will generate the database migration for the necessary tables required to store user information and roles, as well as generating the user interfaces for the registration, login and account management pages into its own area that can be implemented into the project. Password hashing and salting is directly handled by the Identity API to ensure that passwords are stored securely. Passwords are stored using a key derivation function to generate a random salt and hash the password. The password is hashed using the SHA-1 hash function, which is called 1000 times to further increase the complexity of the password. This method will make it virtually impossible to brute force. The user will login through the login page where identity will look at the submitted information and will be able to authenticate that the password matches the one for the account name and, if it does, then access will be granted to the account. Identity also contains tools and operations to implement authorization into a project. There is a function that will allow you to declare a method of a controller or even an entire class to be required to only be accessible by an authenticated user. If a class or method that is marked with this operation and the user is not logged into the system, they are redirected to the login page where they can then be authenticated with the system and they will then be authorized to use that action. Another aspect of Identity is role based authorization, where users can be granted specific roles. These are mainly used to grant admin privileges to users, but it can also be used to define different types of users like for example "User" and "Developer".

The tools, operations and UI provided by Identity for authentication and authorization, along with the fact the code is open source and provides a large amount of documentation, will prove invaluable for this project.

Stripe Connect Transfers vs PayPal Payouts

The original plan to payout money to users was to use the Stripe Payouts API which it was assumed that this was used to transfer money from a business account to an individual user, but this was clearly not the case as after more in depth research had been conducted, its actual use was to transfer funds earned by a platform to their own bank account only. It was not possible to send a set amount of money to individuals bank accounts so another method needed to be used to transfer winnings to users in a way that they could use. There is a potential alternative solution offered by Stripe where individual users can sign up for a Stripe Connect account that is directly linked to the platform. This was mentioned by a Stripe Support staff member after getting into contact with them to discuss this issue.

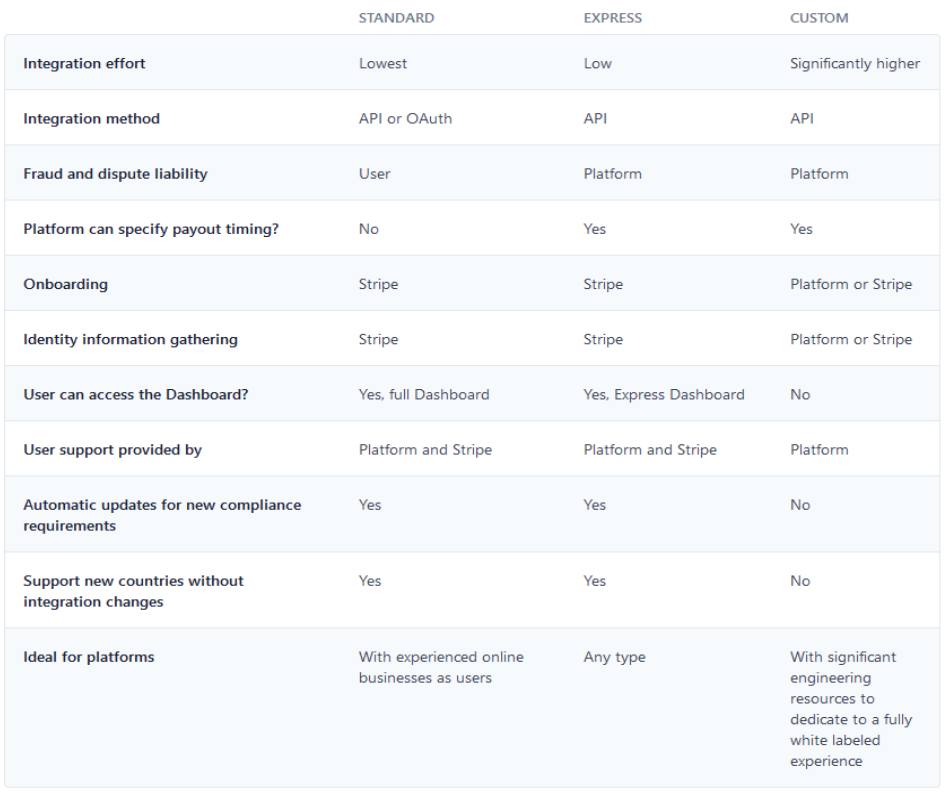

Okay so it seems that the option for you would be using Connect. You can create Custom or Express accounts and later transfer to them funds using this https://stripe.com/docs/connect/charges-transfers

Once the connected account has funds you, as a Platform could send these funds to their bank or debit card.

https://stripe.com/docs/connect/manual-payouts

Figure 11 - Stripe Connect chat log

This at first seemed like the perfect solution to transfer money to users so work had begun on implementing this into the project. The way that payouts to the user was going to work was that, if the user did not have a stripe connect account linked to Central Games Platform, they would be redirected to the Stripe Connect onboarding process where they would fill in details like their name and address, and a Stripe Connect Express account would be made. This onboarding process was straightforward enough to implement, but an issue had arisen with Express accounts in particular, and later the downsides and shortcomings of using Stripe Connect began to far outweigh the positives of the platform. When using the Express onboarding process, there is a very limited number of options available for a user when signing up, including being able to add an IBAN number to actually transfer funds to an account. Furthermore, it was not possible to add this after account registration either, as Express accounts offered a very limited dashboard for managing accounts. After once again getting into contact with Stripe Support, they suggested that this was only possible by using a Custom Stripe Connect account type, but they mentioned that it was not intended to transfer money to individuals like this, but instead was intended to be used by employees, contractors, or sellers who would receive money from the platform, similar to how JustEat would pay their delivery drivers, or Ebay would pay their sellers, so there was no guarantee that it would even work as intended without compromising the user experience. Additionally, Custom accounts required much more effort to integrate into the project, as the entire onboarding process had to be designed from scratch. As well as this, Stripe is not responsible for storing sensitive data like bank details so it is left up to the platform to manage this, adding a significant amount of development time to the project which is simply not available [34].



	STANDARD	EXPRESS	CUSTOM
Integration effort	Lowest	Low	Significantly higher
Integration method	API or OAuth	API	API
Fraud and dispute liability	User	Platform	Platform
Platform can specify payout timing?	No	Yes	Yes
Onboarding	Stripe	Stripe	Platform or Stripe
Identity information gathering	Stripe	Stripe	Platform or Stripe
User can access the Dashboard?	Yes, full Dashboard	Yes, Express Dashboard	No
User support provided by	Platform and Stripe	Platform and Stripe	Platform
Automatic updates for new compliance requirements	Yes	Yes	No
Support new countries without integration changes	Yes	Yes	No
Ideal for platforms	With experienced online businesses as users	Any type	With significant engineering resources to dedicate to a fully white labeled experience

Figure 12 - Stripe Connect account differences. Source: stripe.com

After careful consideration, a decision was made to not proceed with Stripe Connect to transfer funds to individuals. More research was conducted and similar online casinos were investigated to see how they went about transferring funds to users. A common method of transferring money was present in a lot of online casinos and that was the use of PayPal, where all a person needed was a PayPal email to transfer winnings from their account over to a usable currency.
This was possible through the PayPal Payouts API, which is a RESTful API that can be incorporated into any project. Upon searching for information on this API however, it was quickly discovered that there was almost no documentation available online on how to implement it, and only showed details on the JSON responses from API requests [35]. Furthermore, it is a relatively niche service offered by PayPal, so there were no tutorials available online and there were virtually no questions posted on StackOverflow either. This led to the implementation of the API being a long and difficult task, involving a lot of trial and error, referring to the .NET Payouts SDK source code available on GitHub [36]. The GitHub repository helped to steer the development in the right direction a little bit, but it did not demonstrate how to implement anything in the context of an ASP.NET Core MVC application, so again a lot of trial and error was involved with getting this implementation working. After many hours spent understanding how to link a business account to transfer money from and specifying an individual user's email to transfer to, it was finally implemented into the project with an easy and straightforward method for users to payout. Simply all that is required is an email address for a registered PayPal account and enough winnings in their account to facilitate the transfer.
After much confusion, trial and error, and time spent researching, it was now possible for a user to receive payouts for their winnings earned on the platform.

Entity Framework Core

As mentioned earlier in this document, an SQL database will be used to store and manage most of the data related to the application. As a result of this, there will be a need to constantly retrieve and send data to the database, and even create tables in the database from time to time. This could be done manually by writing SQL query statements and executing on the database using some kind of database management software, but this is time consuming and cannot be automated. Furthermore, to lower the complexity of the project it would not be beneficial to contain hard coded strings of SQL statements into the project, as this would also potentially present a security threat in the form of SQL injection attacks. And of course, data that is read from the database will need to be converted into an object that can be understood by the .NET runtime. Because of this, Entity Framework Core will be used to enable a code first approach to database development. A code first approach allows for a developer to not have to worry about creating a table in the database first to read and write data. This is done by creating a model of the data first and setting the properties of the model like for example creating a model for the Game object and defining the Name and the Price as its properties. Entity Framework Core will be able to recognise that this needs a table in the database and will create and execute the SQL code needed to create a table with Name and Price as the columns. This SQL code is

automatically generated and requires no intervention from a developer, giving them more time to work on more important functionality.
Furthermore, Entity Framework Core automatically converts data read from the database into .NET objects, which means that as soon as data is retrieved, it can be accessed or modified by using Language Integrated Query, or LINQ, a tool that allows you to use to use C# code to manipulate objects retrieved from a database. This ability to work with exclusively .NET objects makes it a great addition to the project's development stack as it significantly lowers both the project complexity and development time [37].

Repository Pattern

Throughout the project, there is going to be a lot of read and write functionality to the database that is going to be reused over and over again. Reused code is something that must be avoided as often as possible as repeat code will make it far more difficult to maintain as if one operation that is used 100 times throughout the project needs to be changed, a developer must individually change the code for 100 instances of this 1 operation. Instead, it would be extremely helpful if this single operation could be abstracted into a method with only 1 implementation so that other classes can then make a call to this method whenever they need to use this operation. This is what the repository pattern is used for, which is actually the design pattern Entity Framework Core was built upon [38].

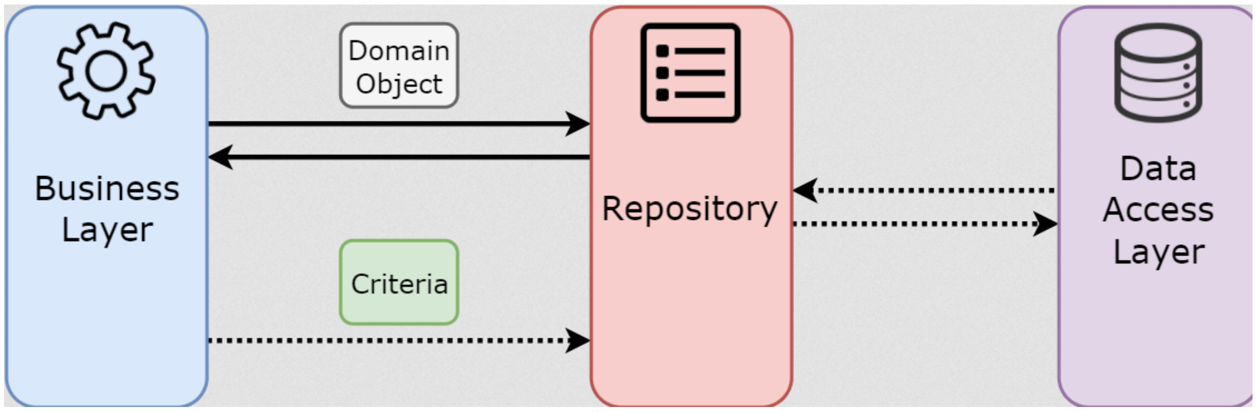


Figure 13 - Repository pattern. Source: codingsight.com

The repository acts as a bridge between the application and the database. Parameters are defined in the application, or business layer, where they are then sent to the repository through a function call. The repository can then determine if they are valid parameters and, if they are, a query is constructed using the parameters for the specific operation that was called from the business layer. The repository then sends this query to the database, or data layer, where it will then be executed and the results of the query are returned back to the repository where it is then returned to the application.
This level of abstraction will ensure that SQL queries will only need to be written out once with placeholder values, and the application needs to just call the specific query while the repository does all of the work.

Conclusion

Each area of technology was thoroughly analyzed with a critical eye, weighing both the advantages and disadvantages of each to determine which would be the perfect fit for the particular needs for this project.

C# was decided on for the programming language, being paired with ASP.NET Core as the web development as it was the best one rated for the language by the community.

Windows was chosen as the development operating system as it had the best support and community for ASP.NET Core due to both of them being owned by Microsoft.

For the platforms being targeted it was decided to aim for release on Windows PCs and Android devices as they were by far the most used platforms with the huge user bases out there.

Stripe was ultimately the best fit for the payment gateway provider as they charge no fee to set up, offer a competitive transaction fee and most importantly offer freedom of customization with their RESTful API.

To prevent cheating from happening it was decided that all important game logic was to be processed on the server to prevent the user from falsifying their results.

Microsoft Azure's Web App Service was decided to be the server hosting solution as they offer a free plan with 1GB of storage and memory, encrypt all data on the server and provide a user friendly dashboard to manage it.

Games will be delivered by either providing a downloadable executable for traditional games or by displaying the game directly in the browser using inline frames.

An SQL database is going to be used to store all of the data associated with customers, products, orders and game results.

Git along with GitHub will be used for source and version control to keep track of all of the changes in the project as well as to keep an online back-up of all the source code.

The MVC pattern will be used to separate the user interfaces, data structure, and algorithms/code into views, models and controllers respectively, which respects the separation of concern principle.

ASP.NET Core Identity will enable the use of authentication and authorization throughout the entire project, and the tables required in the database will automatically be generated, along with the user interfaces for logging in, registering, and account management.

PayPal’s Payouts API will enable users of the platform to transfer winnings from their account into their PayPal account where they are able to then use as actual money.

For data access and management, Entity Framework Core will be used to remove the need for writing SQL statements and will enable to developers to access and modify data using purely C# code.

Finally, the repository pattern will be used to abstract common data access operations so that it can be reused throughout the entire project.

By incorporating all of these various technologies into the project, it will likely greatly benefit from it and may potentially result in a successful product.

Bibliography

[1]Dobrilova, T 2020, *How Much is The Gaming Industry Worth in 2020?*, [Accessed 25 October 2020] available at: https://techjury.net/blog/gaming-industry-worth/.

[2]Miller, G 2020, *Online Gambling Market is Projected to reach USD 160 Billion by 2026,* [Accessed 26 October 2020], available at: https://europeangaming.eu/portal/latest-news/2020/09/14/77558/online-gambling-market-is-projected-to-reach-usd-160-billion-by-2026/

[3]Minotti, M 2018, *90% of US's 211.2 million gamers play on mobile, 59% are multiplatform,* viewed 26 October 2020, https://venturebeat.com/2018/09/11/npd-90-of-u-s-s-211-2-million-gamers-play-on-mobile-59-are-multiplatform/

[4]Murphy, E 2019, *How many people play online casinos in Ireland?,* viewed 26 October 2020, https://www.thecork.ie/2019/09/09/how-many-people-play-online-casinos-in-ireland/

[5]Pulley, B 1998, *Casinos Paying Top Dollar To Coddle Elite Gamblers,* viewed 26 October 2020, https://www.nytimes.com/1998/01/12/us/casinos-paying-top-dollar-to-coddle-elite-gamblers.html
[6]Cass, M 2019, *The Top Programming Languages 2019,* viewed 03 November 2020, https://spectrum.ieee.org/computing/software/the-top-programming-languages-2019

[7]Hernandez, B 2018, *10 Famous Websites Built With Python,* viewed 03 November 2020, https://www.quickstart.com/blog/10-famous-websites-built-with-python/

[8]Danylko, J 2017, *Top 10 Websites Written Using ASP.NET MVC,* viewed 03 November 2020, https://dzone.com/articles/top-10-websites-written-using-aspnet-mvc

[9]Scully, E 2020, *C# vs Python: Summary of Differences and Similarities,* viewed 03 November 2020, https://careerkarma.com/blog/csharp-vs-python/

[10]Rouse, M 2013, *web development framework (WDF),* [Accessed 03 November 2020], https://searchcontentmanagement.techtarget.com/definition/web-development-framework-WDF
[11]Altvater, A 2017, *.NET Core vs .NET Framework: How to Pick a .NET Runtime for an Application,* viewed 04 November 2020, https://stackify.com/net-core-vs-net-framework/

[12]Microsoft, 2020, *ASP.NET Documentation,* viewed 04 November 2020, https://docs.microsoft.com/en-us/aspnet/core/?view=aspnetcore-3.1

[13]Microsoft, 2020, *Choose between ASP.NET 4.x and ASP.NET Core,* [Accessed 04 November 2020], https://docs.microsoft.com/en-us/aspnet/core/fundamentals/choose-aspnet-framework?view=aspnetcore-3.1

[14]Garcia, D.J 2019, *The History of ASP.NET - Part I,* [Accessed 04 November], https://www.dotnetcurry.com/aspnet/1492/aspnet-history-part-1

[15]Lima, F.S 2019, *Benchmark ASP.NET 4.9 Vs ASP.NET Core 3.0,* [Accessed 04 November], https://www.c-sharpcorner.com/article/benchmark-asp-net-4-8-vs-asp-net-core-3-0/

[16]Microsoft, 2017, *Support Ending for the .NET Framework 4, 4.5 and 4.5.1,* [Accessed04 November 2020], https://devblogs.microsoft.com/dotnet/support-ending-for-the-net-framework-4-4-5-and-4-5-1/

[17]Microsoft, 2019, *.NET Core is the Future of .NET*, [Accessed04 November 2020], https://devblogs.microsoft.com/dotnet/net-core-is-the-future-of-net/

[18]Liu, S 2020, *PC Operating System Distribution For Software Development Worldwide in 2018 to 2020,* [Accessed05 November 2020], https://www.statista.com/statistics/869211/worldwide-software-development-operating-system/

[19]Microsoft, 2016, *C# IntelliSense*, [Accessed05 November 2020], https://docs.microsoft.com/en-us/visualstudio/ide/visual-csharp-intellisense?view=vs-2019

[20]Statcounter, 2020, *Operating System Market Share Worldwide - October 2020*, [Accessed07 November 2020], https://gs.statcounter.com/os-market-share

[21]Nagasubramanian, R. and Rajagopalan, S.P., 2012. Payment Gateway-Innovation in Multiple Payments. *International Journal of Computer Applications*, *59*(16), [Accessed09 November] 2020, http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.303.5197&rep=rep1&type=pdf

[22]Masihuddin, M., Islam Khan, B., Islam Mattoo, M. and Olanrewaju, R., 2017. A Survey on E-Payment Systems: Elements, Adoption, Architecture, Challenges and Security Concepts. *Indian Journal of Science and Technology*, [online] 10(20), pp.1-19. Available at: https://www.researchgate.net/publication/318076004_A_Survey_on_E-Payment_Systems_Elements_Adoption_Architecture_Challenges_and_Security_Concepts [Accessed 9 November 2020].

[23]Motola, C 2020*, Stripe Vs Paypal: How to Choose The Right Payment Processor For Your Online Business*, [Accessed10 November 2020], https://www.merchantmaverick.com/paypal-vs-stripe/

[24] Markus, L, 2020, *PayPal-NET-SDK Depreciation Notice,* [Accessed 10 November], https://github.com/paypal/PayPal-NET-SDK/blob/develop/README.md

[25] Nelder, A, 2017, *Official support for .NET*, viewed 10 November, https://stripe.com/blog/stripe-dotnet

[26]Beyers, J 2020, *Use of JavaScript In Online Games*, [viewed 10 November], https://dzone.com/articles/use-of-javascript-in-online-games

[27]Li, Y. and Manoharan, S., 2013. A performance comparison of SQL and NoSQL databases. 2013 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing (PACRIM), [online] Available at: https://ieeexplore.ieee.org/abstract/document/6625441 [Accessed 6 November 2020].

[28]Git. n.d, *Companies & Projects Using Git,* [online] Available at: https://git-scm.com/ [Accessed 6 November 2020].

[29]Ahmed, R 2020, *Git Tutorial - Commands And Operations In Git,* [online] Available at: https://www.edureka.co/blog/git-tutorial/ [Accessed 8 November 2020].

[30] Vyas, A., 2018. *MVC Pattern*. [online] Medium. Available at: https://medium.com/@anshul.vyas380/mvc-pattern-3b5366e60ce4 [Accessed 9 February 2021].

[31] Cloudemy: Step into the Cloud - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/Interaction-within-MVC-pattern-The-Model-component-correlates-with-all-the-data-related_fig2_328716094 [accessed 27 Apr, 2021]

[32] Sandoval, K., 2020. *What is the Difference Between Authentication and Authorization? | Nordic APIs* |. [online] Nordic APIs. Available at: https://nordicapis.com/what-is-the-difference-between-authentication-and-authorization/ [Accessed 27 April 2021].

[33] Anderson, R., 2020. *Introduction to Identity on ASP.NET Core*. [online] Microsoft.com. Available at:

https://docs.microsoft.com/en-us/aspnet/core/security/authentication/identity?view=aspnetcore-5.0&tabs=visual-studio [Accessed 27 April 2021].

[34] Stripe. n.d. *Choose your account type*. [online] Available at: https://stripe.com/docs/connect/accounts [Accessed 28 April 2021].

[35] PayPal. n.d. *Payouts*. [online] Available at: https://developer.paypal.com/docs/api/payments.payouts-batch/v1/ [Accessed 28 April 2021].

[36] GitHub. 2020. *paypal/Payouts-DotNet-SDK*. [online] Available at: https://github.com/paypal/Payouts-DotNet-SDK [Accessed 28 April 2021].

[37] Microsoft. 2020. *Overview of Entity Framework Core - EF Core*. [online] Available at: https://docs.microsoft.com/en-us/ef/core/ [Accessed 28 April 2021].

[38] Panyushkin, D., 2017. *Entity Framework: Repository Pattern*. [online] {coding}Sight. Available at: https://codingsight.com/entity-framework-antipattern-repository/ [Accessed 28 April 2021].

DECLARATION

*I declare that all material in this submission e.g. thesis/essay/project/assignment is entirely my/our own work except where duly acknowledged.

*I have cited the sources of all quotations, paraphrases, summaries of information, tables, diagrams or other material; including software and other electronic media in which intellectual property rights may reside.

*I have provided a complete bibliography of all works and sources used in the preparation of this submission.

*I understand that failure to comply with the Institute's regulations governing plagiarism constitutes a serious offence.

Student Name: (Printed)KYLE HENNESSY_______________________________________

Student Number(s): C00227463_______________________________

Signature(s): Kyle Hennessy____

Date: 30/04/2021____

Please note:
The Institute regulations on plagiarism are set out in Section 10 of Examination and Assessment Regulations published each year in the Student Handbook.