

Heather Johnson

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

**Resources** 6

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/23/2021 | Heather Johnson | Client based game development |

**Instructions**

Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [Executive Summary](#_sbfa50wo7nsh)

Creating a new web based game for clients can be an exciting and elevating adventure. However, challenges can arise during the development process but with some fore thought many issues can be preventive. Today, we will discuss a few issues that may arise in our journey and how to avoid or correct them along the way. When creating a multi-platform game, the software should easily swap from platform to platform without crashing. To do this, the software architecture should be built with that in mind. It is necessary to develop the code in a manner where certain attributes can be adjusted for the platform’s specific specifications. In the realm of console game development, this is not a new problem. One website states, “How does a game identify a player across Steam, Epic, Xbox, PlayStation, and Switch? "This is done by something called account linking," Arifianto said. "With account linking, you want to be able to link your common account with the different varying platform accounts. You've probably already seen some of these... If you have a Ubisoft Uplay account, you go to their website and that's where you actually link the different accounts. Same thing with Epic Games.” (Valentine, 2020). Similarly, we will do this with our program. Behind the scenes our code with interlock platforms and avoid disruptions or unmanageable code.

Budgetary concerns should be addressed in advance so that the company is prepared for the cost and prepared for any happenstances along the way. Depending on the depth of the game’s code, how any users involved and any licenses or copyrights involved, web based games can be expensive. One source states, “Some very well-known games Angry Birds fall under this category. These mobile games involve different levels, characters, story, and good graphics. The variety and depth in the content improve the replayability and engages players for longer durations. It may cost you anywhere from $250,000 to $700,000 to develop such games” (Georgiou, 2019). Our suggestion is to discuss this projected budget with your accounting team and secure the funds in advance to avoid any future constraints.

## [Design Constraints](#_2et92p0)

Design constraints are an everyday occurrence that developers must face. Some examples are phone capabilities, the platform’s capabilities and the applications user friendliness. The code behind the face of our cell phones differs wildly between Samsung, iPhones and various others. How does a developer prepare for this? One way is to develop on a cross platform framework: essentially a developer tool that talks to multiple platforms. One author states, “Native app development eschews the complexity of creating a sustainable product that spans multiple platform app development and focuses on generating a competent design that stays close to the target platform–Android, iOS, etc. Cross-platform frameworks seek to generate an app that reaches out to as many followers of your brand as possible by covering a wide number of end devices during the programming and creation process” (Manchanda, 2020). In addition, by utilizing a cross platform framework, the program will aid in creating the program across platforms that have differing battery life, storage space and video qualities.

But what keeps a user coming back and using a web-based game time after time? User friendly and engaging applications prove to survive the test of time. One website states, ““OMG, this thing is sooooo slow!” As a mobile web app developer, those are probably the very last words you ever want to hear from one of your users. You must therefore think carefully about how to reduce and optimize each byte and server transfer to reduce the user’s wait time. It’s unrealistic to expect that transfers will always be done over a WiFi network, and you should know that 60% of mobile web users say they expect a site to load on their mobile phone in 3 seconds or less” (Agrimbau, 2020). Other areas to consider are the ease in which the user can select objects on the screen, the font size and color patterns that may be unattractive to the user.

## [System Architecture View](#_ilbxbyevv6b6)

As we stated earlier, we will need to construct our design team to build within their framework, code that is easily updatable. Web based games as well as their allocated devices (i.e., iPhones, Samsung Galaxy etc.. are continuously updated to impose new features and promote efficiency. We do not wish to have an update available which then causes the app to crash on an iPhone but not a Samsung device.

## [Domain Model](#_8h2ehzxfam4o)

By implementing the singleton method, we utilize encapsulation so that outside processes do not access certain classes. Instead, these classes are single instances of themselves. Our base class titled “Entity” will have three classes that inherit attributes. The Game, Team and Player class extend our Entity class and inherit such variables as id and name. This shows that in any given instance the game, team and player classes with have an instance of the class Entity. Each team, player and game will have their own unique items such as team name, player name and each team with have its own ID. This makes it easier to detect if a certain team or player has already been entered into the system. We do not want to create duplicates because it creates confusion and takes up unnecessary memory in the game.

****

## [Evaluation](#_2o15spng8stw)

Using your experience to evaluate the characteristics, advantages, and weaknesses of each operating platform (Linux, Mac, and Windows) as well as mobile devices, consider the requirements outlined below and articulate your findings for each. As you complete the table, keep in mind your client’s requirements and look at the situation holistically, as it all has to work together.

| **Development Requirements** | **Mac** | | **Linux** | | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- | --- | --- |
| **Server Side** | * Consistency of the Mac’s layouts, data structures and navigation. * Mac has ever evolving technology. * Migrating to silicon. * Contain eccentricities does as closed source code and proprietary products. | * Innumerable different applications * Linux is open source which makes it great for the company on a budget * Linux does not have the capital that Mac and Windows possess * Limited in graphics, security and functionality. | | * Windows has generational establishment * Top technology, developers and resources back by dominating the computer industry. * There is a Windows machine for you. * Microsoft has long embraced Touch-screen computing and emerging technology with Windows, a feature not found in Macs anywhere” * Because there are so many more Windows users, malware, spyware, and ransomware are far, far more prevalent attacks” (Gewirtz, 2019). | | * Portability and easy access. * While they wait at the DMV, during a treacherously long meeting or in the comfort of their own bed, the game will be there. * Public interest access which can be a problem for security. * Memory storage, graphics abilities, battery consumption and even data usage. |
| **Client Side** | * Mac users tend to pay more on average than other users for their devices. * Numerous memes have been created to joke about how any additional pieces or software can cost the average Mac user | | * Client already running various apps developed with Linux is high. * Development teams are usually independent and therefore may be difficult to tie down to upgrade on a regular basis. | | * The Windows platform has the largest development usage and programmers of any other iOS in the world. * Programmer levels are inconsistent and vary greatly | * Mac users with iPhones will need specifications than Windows phones powered by Android. * An iPhone user may be able to share with their Mac but an Android user may not have the same access. |
| **Development Tools** | * A to share with ease to other devices that run the same iOS. * MacOS is tightly integrated with iOS, * Mac and an iPhone or use an iPhone to take a picture that drops directly into a Mac application * Apple’s products are very Closed off to outside development tools and contain proprietary IDE’s. | | * Ability to work with Raspberry Pi, OctoPi, Debian and centOS * Lack of centralization. If a team develops a iOS they may not aid or service others | | * Microsoft’s iOS, Windows, brings with it 8 generations * Constantly evolving it’s tools and aiding fledgling * Multiplatform IDE’s such as Visual Studio * Various tools, plug ins, work shops and support | * A program that runs on your computer may not run the same on your mobile device. * Memory, battery life, graphics and video abilities, on average, are less desktops. * Proprietary version of developmental tools. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room. Specifically, address the following:

1. **Operating Platform**: In our opinion, Windows would be the best recommendation for an operating platform. By developing the app in Windows we are still able to develop a cross platform game utilizing one of Microsoft’s framework IDE’s. In addition, by investing our efforts in Microsoft we will have access to their virtual app stores along with customer and technical support. Microsoft posses “the platform with the largest market share, Windows also offers the most applications” (Gewirtz, 2019).
2. **Operating Systems Architectures**: With software packages much like Microsoft Office and Visual Studio, we will have the tools necessary to work with a framework across multiple platforms. In addition, Microsoft offers a package using the environment with .NET that is also multi-platform. .NET will not only serve us for our back-end development but also encompasses UI (user interface) development.
3. **Storage Management**: Cloud storage is the wave of the future. Microsoft allots some cloud storage, and the company will have to pay for additional storage. However, with cloud storage the worries of servers being physically destroyed in one area will be minimal. The server will store our product in multiple locations with access points not anchored by a central location.
4. **Memory Management**: In our code we used what we in the industry call the “singleton method”. This type of programming calls only what it needs in the moment and does not allow processes to languish in the background. By creating single instances of the part of the program that we need at any given time, we reduce redundancy in our code creating less memory usage.
5. **Distributed Systems and Networks**: If the team develops with a cross platform framework, the issue of whether the application can run on multiple devices. With mobile devices the problem of connectivity and outages will also be a problem that has to be addressed. One way to avoid such pitfalls would be to have a downloadable version of the game. This would limit new pictures but would give the client some time of play. Regardless, we may need to do additional research and provide warnings for users that use a network that is renowned for outages and lacking connectivity.
6. **Security**: By using Windows, we are exposing our application to a large population of developers. This will come with a larger mount of hackers and malware. However, Windows has accounted for this and provided methods to secure your code and additional software that manages potential threats. One such service is DevSecOps, this service provided by Microsoft serves to protect our framework from malware and phishing attempts.

**Resources**

Agrimbau, T. (2014, March 11). Developing Mobile Web Applications: When, Why, and How. Toptal Engineering Blog. <https://www.toptal.com/android/developing-mobile-web-apps-when-why-and-how>.

Georgiou, M. (2020, December 10). Game App Development: How Much Does It Really Cost to Create a Mobile... RSS. <https://www.imaginovation.net/blog/mobile-game-app-development-cost/#:~:text=It%20may%20cost%20you%20anywhere,%24700%2C000%20to%20develop%20such%20games.&amp;text=Modern%2Dday%20businesses%20are%20taking,improve%20their%20team%2Dbuilding%20efforts>.

Gewirtz, D. (2019, April 8). Windows, Mac, or Linux? We compare the pros and cons of these computing platforms. ZDNet. https://www.zdnet.com/article/windows-mac-or-linux-we-compare-the-pros-and-cons-of-these-computing-platforms/.

Manchanda, A. (2021, May 12). Cross-Platform App Frameworks in 2021. Insights - Web and Mobile Development Services and Solutions. <https://www.netsolutions.com/insights/cross-platform-app-frameworks-in-2019/>.

Valentine, R. (2020, March 25). Tips for developers making a cross-platform game. GamesIndustry.biz. <https://www.gamesindustry.biz/articles/2020-03-20-tips-for-developers-making-a-cross-platform-game>.