

Draw It or Lose It

# **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

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 3.0 | 6/07/22 | Daniel Parker | Third Draft |

**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)

New client, The Gaming Room, is looking to develop a web-based game similar to the 1980s television game *Win, Lose or Draw.* The current version, titled *Draw It or Lose It*, is currently available only on Android. The Gaming Room aims to expand onto multiple platforms. The rules of the game are as follows: four rounds of play lasting one minute each, drawings are rendered at a steady rate and are fully complete at the 30-second mark, an incorrect or late guess allows the remaining teams one more opportunity to solve the puzzle within 15 seconds.

## [Design Constraints](#_2et92p0)

* Web-based – must run on multiple platforms
* Allows one or more teams to play
* Allows each team to have multiple people
* Game and Team names must be unique (allow users to check availability)
* Only one instance of the game can exist at any time (create unique identifiers)

## [Domain Model](#_8h2ehzxfam4o)

In the UML diagram provided below, the ProgramDriver class forms an association with SingletonTester that states that the elements of a source package can be present inside the target package as well. The Entity superclass forms an inheritance relationship between the Game, Team, and Player classes. Since these classes inherit information from Entity, they will share references such as “id” and “name”. In this diagram, we can also determine the classes GameService, Game, Team, and Player all form a multiplicity relationship with one another. This simply means that every object in that every object in GameService is associated with zero or more objects in Game, and that every object in Game is associated with exactly one object in GameService. Every object in Game is associated with zero or more objects in Team, and that every object in Team is associated with exactly one object in Game. And so on from left to right of the associations in the diagram.

**"The Gaming Room UML diagram. The top of the diagram is labeled as com dot gamingroom. Test boxes are placed in two layers. The first layer has three text boxes and the second layer has four of them. In the first layer, the 'ProgramDriver' textbox points to 'SingletonTester' textbox. The 'ProgramDriver' textbox contains the text 'asterisk main round brackets.' The 'SingletonTester' textbox contains the text 'asterisk testSingleton round brackets.' The arrow between these two text boxes are labeled 'open two angle brackets uses close two angle brackets'. In the second layer, there are 'GameService', 'Game', 'Team', and 'Player' text boxes. The 'GameService' textbox has texts arranged in two layers. The first layer contains games colon List open angle bracket Game close angle bracket, nextGamesId colon long, nextPlayer Id colon long, nextTeamId colon long, and service colon GameService. The second layer contains GameService round brackets, getinstance round brackets colon GameService, addGame open parenthesis name colon String close parenthesis colon Game, getGame open parenthesis id colon long close open parenthesis colon Game, getGame open open parenthesis name colon String close open parenthesis colon Game, getGameCount round brackets colon int, getNextPlayerID round brackets colon long, and getNextTeamId round brackets colon long. The 'GameService' box is connected with the 'Game' textbox with a line labeled 'zero dot dt dot asterisk'.  The 'Game' textbox also contains text in two layers. The first layers contains the text teams colon List open angle bracket Team close angle bracket. The second layer has Game open round bracket id colon long comma name colon String close parenthesis, addTeam open parenthesis name colon String close parenthesis Team, toString round brackets colon String. The 'Game' textbox is connected with the 'Team' textbox with a line labeled 'zero dot dt dot asterisk'. The 'Team' textbox also contains text in two layers. The first layers contains the text players colon List open angle bracket Player close angle bracket. The second layer has Team open parenthesis id colon long comma name colon String close parenthesis, addPlayer open parenthesis name colon String close parenthesis colon Player, and toString round brackets colon String. The 'Team' textbox is connected with the 'Player' textbox with a line labeled 'zero dot dt dot asterisk'. It contains the text Player open parenthesis id colon long comma name colon String close parenthesis and toString round brackets colon String. The 'Game', the 'Team, and the 'Player' boxes point to the 'Entity' textbox in first layer. The 'Entity' textbox contains text in two layers. The first layer has the text id colon long and name colon String. The second layer has Entity round brackets, Entity open parenthesis id colon long comma name colon String close parenthesis, getId round brackets colon long, getName round brackets colon String, toString round brackets colon String.**

## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Characteristics: Second most popular; fast growth amongst consumers and developers  Advantages: Runs all major browsers for testing, flexible, high quality, user friendly applications and interface  Weaknesses: More expensive than Windows, not as popular for web development | Characteristics: Secured, most preferred for web server development and maintenance  Advantages: Security, popular amongst developers, many open-source projects available for use  Weaknesses: More difficult to find tech support | Characteristics: Most popular for user and web development, generally cheaper than Mac  Advantages: Numerous useful built-in programs and applications, high compatibility with programs and web development  Weaknesses: Prone to viruses, tech and customer support can be slow/tedious, less user friendly than Mac | Characteristics: Very popular, high portability, high growth potential, widely available platform  Advantages: Wide reach, extensive compatibility, cost-effective, built in authentication and biometrics  Weaknesses: Security risks vary from each device OS, can be breached , not appropriate to host server application |
| **Client Side** | Cost: High  Time: Moderate  Expertise: Moderate | Cost: Low, most cost effective  Time: Low  Expertise: High | Cost: High  Time: Moderate  Expertise: Low | Cost: Moderate  Time: Moderate  Expertise: High |
| **Development Tools** | Can run all major browsers (Internet Explorer, Chrome, Firefox, Safari). Can run all relevant programming languages (Java, JavaScript, Python, HTML, CSS). Most compatible tools are XCode, Eclipse, and Notepad++ | Can run all major browsers (Internet Explorer, Chrome, Firefox, Safari). Can run all relevant programming languages (Java, JavaScript, Python, HTML, CSS). Most compatible tools are Eclipse and Notepad++ | Can run all major browsers (Internet Explorer, Chrome, Firefox, Safari). Can run all relevant programming languages (Java, JavaScript, Python, HTML, CSS). Most compatible tools are Visual Studio, Eclipse, and Notepad++ | Can run all major browsers (Internet Explorer, Chrome, Firefox, Safari). Can run all relevant programming languages (Java, JavaScript, Python, HTML, CSS). Compatible tools are Ionic, Xamarin, React Native |

Our client, The Gaming Room, has requested a server-style configuration for hosting the website and allowing it to scale up to thousands of players. As outlined above, the ability to host the software application on each operating platform will vary. However, more users will require more server accessibility, raising potential licensing costs to the client. Each platform does offer sever-based development methods where the website will be hosted. The licensing costs are outlined above, with the most expensive being Mac and Windows, then Mobile, then Linux as the most cost effective.

Out client also wishes to expand beyond their Android-only application to support players on iOS and traditional desktop-based operating systems (Linux, Mac, Windows). The application must be delivered as a modern, responsive HTML interface running inside the web browser on each OS and be capable of communicating with the back-end web application running on the server. In order to ensure the application is compatible with all web browser platforms and mobile devices, the code will have to be tested and re-tested many times. Keeping the code simple will help with transferability of the main design, as well as including conditional comments and constantly validating to catch errors along the way.

Each desktop-based OS is capable of running the major web development programs like Eclipse, Visual Studio, and Notepad++. This will minimize the need for expertise in any single program. Each OS can also run all major browsers (Internet Explorer, Chrome, Firefox, Safari), ensuring that development and testing can be done on any platform. Mobile applications are a bit different and may require a separate development team with more expertise in that area. That being said the mobile platform would not be suitable as a development environment because it is unable to run extensive testing through programs available on the desktop OSs (Eclipse, Visual Studio). Most of the major development tools are free to use at the individual level, but require premium subscriptions for larger teams. These costs will cover a wide range of benefits to management and organization that outsourcing offers.

## 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**: For this project, I recommend using Windows for the operating platform that will allow The Gaming Room to expand “Draw It or Lose It” to other computing environments. Windows offers many resources for web development and is compatible with all major browsers and programming languages as stated above. Windows is easily compatible with PowerShell, a command-line shell built on .NET that is available on Mac and Linux for cross-platform development. Microsoft’s Azure is an excellent program for developing applications on the cloud and will be very useful for memory and storage management.
2. **Operating Systems Architectures**: Windows is by far the most popular OS, providing services used by all Windows-based applications. Windows consists of two main layers; a user mode and a kernel mode. User mode limits the system to basic operations, while kernel mode gives unrestricted access to the memory and external devices. The Windows application programming interface (API) is the programming interface that provides services used by all Windows-based applications. This API enables applications to provide a Graphical User Interface (GUI), access system resources, incorporate audio and much more. Major categories of Windows API functions include Base/Component Services, Graphics & Multimedia, Messaging, Networking, and Web Services.
3. **Storage Management**: Windows makes storage management very intuitive. Users can view how much space each file or program uses, and easily manage and organize each one. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted. Windows also allows users to merge their information to the OneDrive, Microsoft’s cloud service, to save data. Windows also boasts the Azure cloud development program which can create stateless web tier applications. “Stateless web tier means you don't store any application data in the web server memory or file system. Keeping your web tier stateless enables you to both provide a better customer experience and save money” (Anderson 2022). What this means is that the web tier sits behind a load balancer, allowing the developer to quickly respond to changes in traffic by adding or removing servers.
4. **Memory Management**: Windows has great built-in features on the .NET Framework to manage memory on the system. Common Language Runtime (CLR) is programming that manages the execution of programs and allows them to share common object-oriented classes in several supported languages. Automatic memory management is one of the services that CLR provides during Managed Execution. Since CLR’s garbage collector manages the allocation and release of memory for an application, developers don’t have to write code to perform tasks for memory management. Automatic memory management can also eliminate problems that cause memory leaks, like forgetting to free an object or attempting to access memory for an object that has already been freed.
5. **Distributed Systems and Networks**: One of the main focuses on creating “Draw It or Lose It” is to make it easily transferable to different devices for maximum exposure. A larger gaming community will require larger server and network support. Modern cloud systems will be of great benefit in this area, and Windows has great support for distributed systems. As mentioned earlier, Azure is a great resource to use for developing cloud applications on stateless web tiers. “A stateless web tier is architecturally much simpler to scale out the application. That too enables you to respond to scaling needs more quickly, and spend less money on development and testing in the process. Cloud servers, like on-premises servers, need to be patched and rebooted occasionally; and if the web tier is stateless, re-routing traffic when a server goes down temporarily won't cause errors or unexpected behavior” (Anderson 2022). Less time spent responding to and managing server demands will save costs and translate to a much smoother user experience across all web browsers.
6. **Security**: Windows comes with built-in security protection software. This system scans for malware, viruses, and security threats. Windows security is not the best compared to Linux, and is vulnerable to different threats. However, there are many ways for developers to incorporate security across all web development platforms. Ensuring that the framework you are developing on is secure and has embedded security is a good start. Using the latest version of the framework will ensure that all the built-in security features are up to date; this includes components on the client-side as well as the server-side. Proper planning for regular updates will be the best way to keep track of this plan. Utilizing encryption input/output encoding will also butter the application from attacks.

References

Anderson, R. (2022, May 9). *Web Development Best Practices (Building Real-World Cloud Apps with Azure)*. Microsoft Docs. Retrieved June 10, 2022, from https://docs.microsoft.com/en-us/aspnet/aspnet/overview/developing-apps-with-windows-azure/building-real-world-cloud-apps-with-windows-azure/web-development-best-practices