

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 09/24/2024 | Raymond Proctor | Wrote the Executive Summary, Design Constraints, and Domain Model. |
| 1.1 | 10/06/2024 | Raymond Proctor | Updated the Server Side, Client Side, and Development tools sections of the Evaluation table. |
| 1.2 | 10/20/2024 | Raymond Proctor | Updated Operating Platform, Operating Systems Architectures, Storage Management, Memory Management, Distributed Systems and Networks, and Security. |

**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)

When designing a web-based game you need to first look at the many different browsers that will be running the game. Each browser will handle the game differently and so thorough testing is needed. It is important to optimize the performance to run the best on all these browsers. Security is also a concern; you must ensure users cannot interact with the games code.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

The different browsers will affect how the game runs. This means that it can be slower on one vs. the other. Errors may only affect certain browsers and fixing them can cause errors on others. This means that the game will need to be constantly tested. Every time an update or fix is implemented, it will need to be tested across all browsers to insure it doesn’t cause more problems.

## [System Architecture View](#_ilbxbyevv6b6)

Please note: There is nothing required here for these projects, but this section serves as a reminder that describing the system and subsystem architecture present in the application, including physical components or tiers, may be required for other projects. A logical topology of the communication and storage aspects is also necessary to understand the overall architecture and should be provided.

## [Domain Model](#_8h2ehzxfam4o)

The Entity class is a superclass that the Game, Team, and Player classes all inherit from. These classes then interact with each other. The Team class has a one-to-many relationship with the Player class. While the Game class then has a one-to-many relationship with the team class. This works to build multiple teams that can have multiple players on them. The GameService class then has a one-to-many relationship with the Game class. It uses the three children-classes to create an instance of a game. Finally, the ProgramDriver Class uses the SingletonTester class to test that GameService class is using the singleton pattern.

**"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)

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.

In each cell, remove the bracketed prompt and write your own paragraph response covering the indicated information.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Mac is a much more expensive option for hosting servers. Macs are generally not built with hosting servers in mind. They are more for personal use. The cost of the hardware can be much more expensive than the other options and would be hard to scale up when needed. | Linux is an open-source operating system. It is a free option which will reduce the overhead of the project. It also provides a lot of stability for hosting the servers. | Windows has a lot of the same benefits of Linux but requires a License to use. This can drastically increase the cost of running the servers. Being a paid product windows comes with more built-in features then Linux and a customer support system. | <Evaluate Mobile Devices for their characteristics, advantages, and weaknesses for hosting a web-based software application.> |
| **Client Side** | Mac users tend to use Safari, Mac’s proprietary web browser. This makes it simpler to develop a web-based game for Mac users. Though the other major web browsers will still need to be taken into consideration. | Linux is the least used of the major different operating systems. It’s user a generally more tech savvy and can have particular tastes in their software, including web browser. It would be a good idea to investigate testing more obscure web browsers to ensure Linux users can access the game. | Windows proprietary web browser, Edge, is not as widely used in comparison to Safari on Mac. Google Chrome and Firefox are also often used. People are more likely to use different browsers depending on their PC use preferences. So, testing that major web browsers work with the game will be important. | There are many different types of mobile devices and web browsers people use on them. You also must take into consideration the screen size and touch screens when working with mobile devices. The computing power can often be much lower on these systems if someone is using a much older device. |
| **Development Tools** | Developing for macOS is free but deploying the app costs money. You will need Xcode to develop for macOS | Linux is a free open-source operating system. There are many different free options for developing code like VS Code. | Windows provides free and paid versions of Visual Studio. Though you can also use a lot of other operating systems like on Linux. These can be either paid or free development software. | When looking at developing for mobile apps you will need to decide on if you’re going to develop for Android, iPhone, or both. Like for Mac you will need Xcode to develop for iOS. Android is more like developing on Linux or Windows. There are much more options with some costing money and others being free. |

## 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**: Amason Web Services is the leading cloud-based server. They not only have the largest cloud-based service, but they are also very affordable. They have many services and will allow Draw It or Lose It to expand to many different platforms and regions. This will be the best service to use for future growth of the product.
2. **Operating Systems Architectures**: Using AWS will allow us to take advantage of EC2, AWS Lambda, and their microservices architecture. EC2 will allow testing for both Windows and Linux. AWS Lambda gives us a serverless computer service so that The Gaming Room can focus on how the game works and coding for it. Having each part of the game run on its own can help avoid bottlenecks and the game goes down when one-part breaks.
3. **Storage Management**: I would recommend using Amazon S3 to store the images needed for Draw It or Lose it. It would also be necessary to use Amazon RDS to store the game data. I think getting Amazon Backup will also be important to ensure that we do not lose important game data.
4. **Memory Management**: Amazon S3 would be used to store large images and audio files for Draw it or Lose It. This would allow us to scale based on new images added to the game. There are also services to store less used images at a cheaper price. This will allow the more used images to be easily accessible while also making the service more affordable. Scaling would also be handled by AWS and would free up resources to be used in the game. Amazon S3 also comes with Amazon CloudFront which increases the speed that it can provide the images. While Amazon RDS will be utilized to handle the game data.
5. **Distributed Systems and Networks**: Amazon API Gateway would allow for communication between multiple different platforms. This means that Draw It or Lose It would be compatible with many devices. This service also utilizes WebSocket to allow for fast communication between the players and servers. This allows Draw It or Lose It to focus on developing the game instead of developing a way for multiple platforms to use the software.
6. **Security**: AWS KMS is a built-in encryption service that would be provided with the storage services chosen above. Amazon Cognito would also provide another level of security that can be implemented, like their multi-factor authentication tool. AWS also has a Firewall that can be used, AWS Web Application Firewall. These are all built-in security services that can be utilized when choosing AWS as our cloud-based service.