

*Draw It or Lose It*

# **CS 230 Project Software Design Template**

Version 1.1

## Table of Contents

[CS 230 Project Software Design Template](#l6ti7uoag22u) 1

[Table of Contents](#j0zll) 2

[Document Revision History](#grjogdjh5fi8) 2

[Executive Summary](#sbfa50wo7nsh) 3

[Design Constraints](#et92p0) 3

[System Architecture View](#ilbxbyevv6b6) 3

[Domain Model](#h2ehzxfam4o) 3

[Evaluation](#o15spng8stw) 3

[Recommendations](#m8aleynsvzvc) 5

## [D](#grjogdjh5fi8)[ocument Revision History](#grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/20/22 | Zachary Huff | Began work on Executive summary And Design Constraints |
| 1.1 | 01/28/22 | Zachary Huff | Finishing up Software Design Template |
| 1.2 | 02/10/22 | Zachary Huff | Finishing Recommendations |
| 1.3 | 2/19/22 | Zachary Huff | Made edits to Recommendations |

**Instructions**

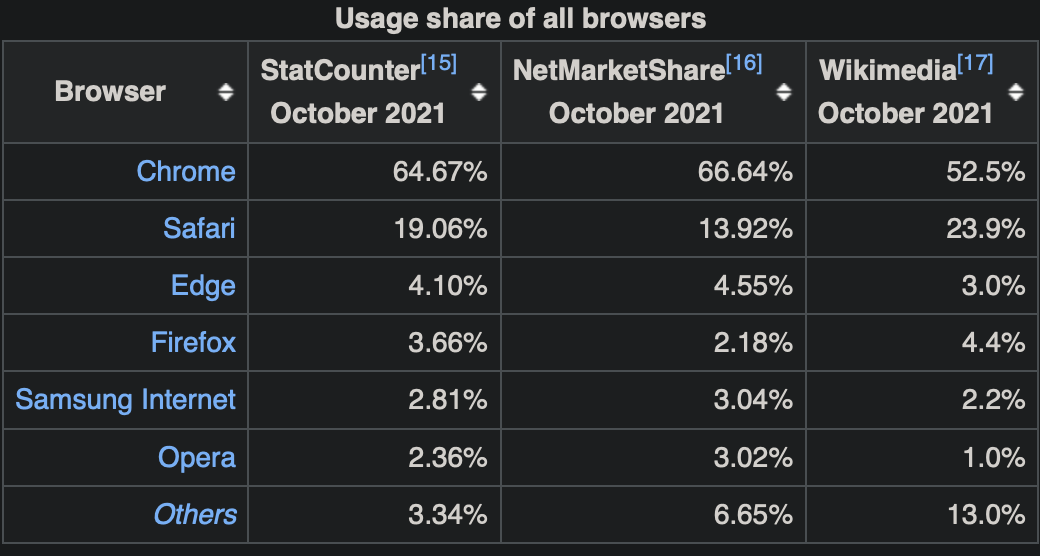
Fill in all bracketed information on page one (the cover page), in the Document Revision History table, and below each header. Under each header, remove the bracketed prompt and write your own paragraph response covering the indicated information.

## [E](#sbfa50wo7nsh)[xecutive Summary](#sbfa50wo7nsh)

We are to create a game like the television show Win, Lose, or Draw. Where our application will render an image over the span of 30 seconds. During this time team 1 will be able to guess what the image is if they get it wrong or run out of time team 2 will have 15 seconds to steal their points. Each team can have multiple players, game and team names must be unique, and only one instance of a game can exists in memory at any given time.

## [Design Constraints](#et92p0)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

First, we will need to make sure the game can run on all modern browsers.

<https://en.wikipedia.org/wiki/Usage_share_of_web_browsers>

Google Chrome is by far the most widely used internet browser out there, with Safari coming in second, Safari being mostly used on mobile platforms such as iPhones and iPads. We will have to ensure the game can work on mobile devices as well as typical desktop computers. Also, to reach the widest array of users we can have to also ensure the application will run on as many internet browsers as we can.

HTML, CSS, and Javascript will be essential in creating a web-based application such as Draw It or Lose It.

## [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.

## [D](#h2ehzxfam4o)[omain Model](#h2ehzxfam4o)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

**"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.**

Game, Team, and Player are all super classes of the Entity Class. Because these classes inherit from Entity this makes them a “Is-A” relationship.

The program driver “uses” the singletonTester to ensure there is only one instance of the game in memory at any given time.

The Team class has Players so the relationship between these two classes would be a “has-a” relationship. This is because an instance of the Player class is used in the Team class. Likewise, the Game class has-a relationship with team because it can add teams using the Team class. Finally, the GameService class can create games called the Game class. This is also a “has-a” relationship indicated by the “0…\*” zero or more.

The GameService can create multiple game instances, the Game Class can have multiple teams, and the Team class can have multiple players. (0…\*)

<https://www.geeksforgeeks.org/what-is-is-a-relationship-in-java/>

**https://creately.com/blog/diagrams/class-diagram-relationships/**

## [E](#o15spng8stw)[valuation](#o15spng8stw)

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 must 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** | Much like Linux has an easy to use command line terminal which is used to make changes and access files in the server. It has less hardware than can run the OS compared to Linux/Windows. | Linux is cheap and efficient to use, along with having an excellent terminal which is used to enter commands for making changes and configuring the server. | Windows server is the most widely used server OS in the market today. Windows along with Linux are the two most widely used when it comes to web hosting services. | There is software created specifically for mobile devices. Mobile Web Server for instance was created specifically for smart phones and uses open-source software allowing users to edit servers to work with mobile applications. There are also many third-party companies that allow users to purchase servers for mobile applications. |
| **Client Side** | MacOS is only used on Apple computers, which not everyone has or has access to. Also, Mac computers tend to cost more money on average than windows PCs. Although with virtualization anyone can have access to MacOS. | Linux is free and open-source software. Making it by far the most cost-efficient OS on this list. Although the usefulness of Linux comes with a caveat as it is the most difficult to navigate, as the user must know many different commands. | Windows is the most widely used OS out there. It is flexible and easy to use, and most third-party applications work on this OS compared to Mac or Linux. Although the OS itself is not free it has low system requirements and can be run on a wide array of machines. | More difficult to implement than the other Operating Systems. But it must be an advantage of being mobile, which gives the user access at any time or place. |
| **Development Tools** | Xcode for macOS is used for Swift and C# development. This application is only available for macOS. Other than this macOS can be used for all front/backend development. | Can be used for all common languages and general purpose / backend development. | Microsoft Visual Studio this is a great IDE for C++ and other backend development. Windows can also be used for all front end/general purpose languages. | There are few tools for mobile development, but they are out there. Python, Java, C++ and other common languages can be built and ran on mobile. Notepad applications can be used for other front-end development. |

## 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**:

Draw it or lose it should be first developed as a web-based application. This will have to be done using both front and backend development. When the application is up and running on web browsers it can then be ported over to the mobile market where it is sure to gain the most popularity.

1. **Operating Systems Architectures**:

The x86 architecture is among the most widely used and most popular. It will run on both AMD and intel processors. Although the arm architecture will need to be implemented later.

1. **Storage Management**:

Storage will need to be HDD, although hard disks are slower, they are much more cost effective. On the other hand, if money is not an option, we can use a solid-state drive (SSD) which will provide a much faster upload/download to the drive at a cost.

1. **Memory Management**:

Because our specific game will not need too much memory, we could potentially use 32-bit system. Although most processors today are 64 bits with access to much more than 4 GB of memory. So, memory will not be much of an issue with today's technology.

1. **Distributed Systems and Networks**:

We will use a database that stores all needed user information. This way we can implement other platforms in the future to derive data from the central database. Many online games today are cross platforms, it will take more time to develop but it is a process that can be done. Along with the database itself we will need servers that multiple users can play on using different devices.

1. **Security**:

Security is a major factor when creating any application, game or otherwise. It should be in the back of the developers' minds while they are designing the game from the ground up. User data is of the upmost importance especially their log in credentials. Also, we will have to combat cheaters who wish to hack into our game and cheat to beat other players. Most games today hire a third-party company to help combat cheating because it has become such a giant marketplace. If cheating becomes a major issue for a game this is the best way to combat, it.