

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 |
| --- | --- | --- | --- |
| 1.3 | 06/15/2023 | Brandon Goller | Added to security measures and recommendations section. |

**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)

The Gaming room project is to develop a web-based game that serves multiple platforms based on the current game, draw it or lose it, which is only available on android. The purpose of the game is to have multiple teams going four rounds at a minute each. When a picture is pulled from a library of images, one team guesses until the time runs out. If not answered, each opposing team gets fifteen seconds to guess.

## Requirements

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.
* Includes large library of stock drawings for clues.
* Each game includes four rounds of play.
* If a team cannot guess the image within 30 seconds, another team has 15 seconds to guess.

## [Design Constraints](#_2et92p0)

Must run on multiple platforms

●Each team should have multiple players

●Only one instance of the game can exist at any time.

●Game and team names must be unique to allow users to check whether a name is in use when

choosing a team name

* Must run on multiple platforms.
* Must render images at a 30 second mark.
* Should improve upon the original game design.

## [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 creates a relationship between Game, Team, and Player class. This means they all inherit or get information from Entity. If we look at their relationship, we see Team and Player are “has a” type. While Game has a Team and GameService has Games. When using UML, it’s called aggregation (HAS-A). This means an instance of one class has a reference to an instance of another class. When we look at this diagram, we see GameService has a reference of Games, Games a reference of Teams, and Team a reference of Players.

**"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 has easy accessibility, easy to use graphical user interface, and flexible terminal commands. Highest level of server security. | Cost friendly. Platforms are difficult to navigate. Command shell for simple server configuration and accessibility. | Windows is known for its user-friendly interface and ease of use.  Windows is compatible with a wide range of software applications.  Windows may not perform as well as other operating systems and is more vulnerable to security threats. | Most susceptible to security breaches. Specifications vary for each user. Updates require more planning. |
| **Client Side** | Macs are generally more expensive than other computers. Developing software for multiple clients takes longer on Mac and requires some expertise. | Linux is open-source and free to use so it is less expensive. Developing for multiple clients can take longer and require more expertise. | Windows is a proprietary operating system that requires licensing fees. Windows has a large community of developers and users who provide support and resources. | Provides the most flexibility for clients and developers. Takes longer to implement updates. |
| **Development Tools** | Swift  Objective-C  Xcode  NetBeans: An extensible editor that provides plug-in support for programming languages like C/C++, XML, HTML, PHP, JSP and JavaScript. | C  C++  Java  Python  Eclipse: An integrated development environment (IDE) that is widely used for developing applications in Java and other programming languages.  PyCharm IDE: Used for Python programming. Supports web-development with Django. | C++  C#  Java/JavaScript  Python  Eclipse IDE  PyCharm IDE  Visual Studio: An integrated development environment (IDE) that is widely used for developing applications in C++, C#, and other programming languages1.  Windows Deployment Services (WDS): A server role that enables you to remotely deploy Windows operating systems2. | Java  C++  C#  Swift  Objective-C  Xcode  Android Studio: An IDE that is designed specifically for developing applications for Android devices. |

## 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**: Windows has the most to offer The Gaming Room to meet their requirements. There is no shortage of IDEs to work with and needs the least amount of expertise. There is also extensive support for clients and developers that will make the transition easy and smooth.
2. **Operating Systems Architectures**: Microsoft Windows is a graphical operating system developed and published by Microsoft. It provides a way to store files, run software, play games, watch videos, and connect to the internet.
3. **Storage Management**: Windows Disk Management is a built-in tool that comes with Windows. It allows you to create, delete, format, and resize partitions on your hard drive.
4. **Memory Management**: The memory manager in Windows is responsible for allocating and deallocating memory to processes as needed. It also manages virtual memory, which is a technique that allows the operating system to use more memory than is physically available by temporarily transferring data from RAM to disk1. The Windows Memory Manager implements a demand-paged virtual memory subsystem. This means that it is a lazy allocator and does not launch the entire application and appropriate DLLs into physical memory when you launch an application such as Notepad1. Instead, it loads only the parts of the application that are needed into memory as they are needed.
5. **Distributed Systems and Networks**: Distributed software systems are designed to work across multiple platforms and devices. They are made up of multiple components that work together to provide a seamless experience for the user. These components can include servers, clients, databases, and other software applications. To communicate between various platforms, distributed software systems use network protocols such as TCP/IP, HTTP, and FTP. These protocols allow data to be transmitted between devices over a network connection. However, there are several dependencies between the components within distributed systems and networks that must be considered. For example, connectivity issues can cause data transmission to fail or be delayed. Outages can also occur due to hardware or software failures. To ensure that distributed software systems are reliable and performant, it is important to design them with these dependencies in mind. This can include using load balancers to distribute traffic across multiple servers, implementing redundancy to ensure that data is not lost in the event of an outage, and using monitoring tools to detect issues before they become critical.
6. **Security**: Windows Defender Antivirus - This is a built-in antivirus program that helps protect your device from malware and other threats.

Windows Firewall - This is a built-in firewall that helps protect your device from unauthorized access.

BitLocker - This is a built-in encryption feature that helps protect your data from unauthorized access.

SmartScreen - This is a built-in feature that helps protect you from phishing attacks and other malicious websites.

Player login – Login should be encrypted with a bot detector and password requirements should be strong enough to prevent hacking access.