

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

Andrew Miller

March 13th, 2023

## 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 | <mm/dd/yy> | <Your-Name> | <Brief description of changes in this revision> |

**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 wants a web-based game developed that can be ran on multiple platforms. “Draw It or Lose It” will be the title and it is only available for Android currently. The game will have several teams filled with multiple people going four rounds a minute each. A picture is pulled from the library and if the first team cannot guess what it is, the other team gets 15 seconds to guess.

## Requirements

* Needs one or more teams.
* Each team has multiple people
* Game and team names must be unique in order to check if name is in use or not.
* Only one instance of the game can run at any given time.
* Should run on multiple platforms.

## [Design Constraints](#_2et92p0)

* Needs one or more teams.
* Each team has multiple people
* Game and team names must be unique in order to check if name is in use or not.
* Only one instance of the game can run at any given time.
* Should run on multiple platforms.

These are requirements that we need to refer to while writing the code and developing the software. This game needs to run on multiple devices, so we need to keep that in mind while in development. This can be done by rewriting existing code or inheriting another language.

## [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)

Entity class creates a relationship between Game, Team, and Player class. Therefore, all latter classes inherit information from entity. The UML diagram shows these relationships. Every class will share references like “name” and “id”. Entity class will become a superclass. When we look at the relationship, we can see that Team and Player classes are “has a” types. In UML, we call it aggregation (HAS-A). When “has a” is used it is as instance of one class and is a reference to another instance of another class. In the UML diagram, we can see GamerService has a reference of Games, Games of Team, and Team of Player.

**"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** | Must have flexible terminal commands to access server, edit it, make changes, etc.  Offers a server-based deployment method where the website will be hosted.  Advantages: is upgradeable, multiple web hosting options.  Disadvantages: not as preferred for web hosting. | Same for Mac but more budget friendly.  Secured and more preferred.  Offers a server-based deployment method where the website will be hosted.  Advantages: security issues are usually caught before they become big, most preferred for web hosting. Is open sourced, secured, free, and cost-effective.  Disadvantages: more difficult to find applications for web hosting. Has less support compared to other OS listed. | Has more software available compared to other OS listed.  Dominant platform and more preferred. More open platform.  Offers a server-based deployment method where the website will be hosted.  Advantages: Higher resource requirements, lower loading times, more secure, offers more support compared to other OS listed.  Disadvantages: susceptibility to viruses, subpar tech support, generally more expensive compared to other OS listed. | Better if server is immobile and can be tracked to a single place. Specs are better in other devices.  More popular than other platforms, highly portable.  Offers a server-based deployment method where the website will be hosted.  Advantages: Wider reach, better compatibility, cost-effective.  Disadvantages: is very selective to smart mobile devices, poor security. |
| **Client Side** | Moderate levels of expertise and time are required. Similar cost to Windows. Is based on Unix which is a moderately easy OS to use. Is not open source which can hurt costs. | Minimum to moderate levels of expertise and time required. It is also free and open source which helps with costs and usability as well. Can be operated by command-line locally or remotely. OS is Unix-like. | Minimum expertise levels and time required. Cost is similar to Mac.  Very popular so finding resources is not hard. The open-source nature also helps reduce cost in obtaining tools and other software. | High levels of flexibility for clients and developers to access update from anywhere. Slightly more difficult to implement compared to other devices. Can be expensive or not depending on several factors such as the budget of the application and various features. |
| **Development Tools** | When running languages on Macs, Swift can be used which is a popular option. There are helpful tools such as notepad++. Macs can run languages such as HTML/CSS/JavaScript/etc. Can also support libraries to support frontend and general-purpose languages such as Java, Python, Ruby, etc.  **Desktop Application:**  Development Tool/IDE – Xcode  Programming Language – Swift (or Objective C) | Linux can run Visual Studio, Eclipse, Notepad++, etc. Linux can run languages such as HTML/CSS/JavaScript/etc. Can also support libraries to support frontend and general-purpose languages such as Java, Python, Ruby, etc.  **Desktop Application:**  Development Tool/IDE – Eclipse  Programming Language – C (default) | Is generally easier to use than Linux but runs similarly. Therefore, it can run Visual Studio, Eclipse, etc. Windows can run languages such as HTML/CSS/JavaScript/etc. Can also support libraries to support frontend and general-purpose languages such as Java, Python, Ruby, etc.  **Desktop Application:**  Development Tool/IDE – Visual Studio  Programming Language – Visual Basic (default) | We can use many apps using Android and Swift. Languages and software can be run on all three devices. Can run languages such as HTML/CSS/JavaScript/etc. Can also support libraries to support frontend and general-purpose languages such as Java, Python, Ruby, etc.  **Android:**  Development Environment – Android Studio(default)  Programming Languages – Java, Kotlin  **iOS:**  Development Tool/IDE – XCode  Programming Language – Swift |

## 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**: I strongly recommend using Windows as it will require minimal expertise and cost. There will be plenty of software and IDE choices to work with.
2. **Operating Systems Architectures**: There are three main OS architectures. These are monolithic architecture, microkernel architecture, and hybrid architecture. Monolithic architecture is where all system services and user services run in the same address space. This provides high performance but a has a high risk of system crashes. Microkernel architecture is where the OS is divided into small modules that run in different address spaces which reduces the risk of system crashes. Hybrid architecture combines facets of both monolithic and microkernel architectures.

Windows (and Linux) have a hybrid architecture. Windows provides services that are used by all Windows-based platforms and devices. These show a Graphical User Interface (GUI) while accessing system resources and more. These apps are used for multimedia, messaging, web services, development tools, etc. These can be accessed and used with an account or server. These would all be beneficial for the application we are developing.

1. **Storage Management**: Storage management involves managing the computer’s hard disk space by organizing files and directories. This helps ensure that there is ample space for new files. Different operating systems have different storage management techniques.

Windows 10 uses a helpful feature called storage sense which will help manage files on your hard drive and show the storage space being used. Another feature is being able to choose save locations on your system, making applications and files easy to find. File creation and storing large projects is very easy using Windows 10 and will help with not losing them.

1. **Memory Management**: Memory management involves allocating and deallocating memory to running processes to ensure they all have the resources required to run efficiently. It also includes virtual memory which allows the OS to utilize hard disk space for extra memory.

During the development of this game, a database and/or library with many pictures should be created. This will keep the pictures outside of the default picture folder, keeping the whole project in a more secure area of the computer. This also includes when working on your IDE and opening files during game development.

1. **Distributed Systems and Networks**: Distributed systems are groups of computers that work in tandem to perform a certain task. This is done by using networks which allow communication between the computers. Some of these networks include LANs, WANS, Wi-Fi, and Bluetooth. Distributed systems are used to increase performance, scalability, and reliability.

Develop 4 enables cross-platform game creation. This is an IDE that can be run on any device. After creation, it can be exported onto any platform whether it be iOS, Android, Mac, the web, etc. Other than that, the servers should be strong enough to support a large player base and the company should have backup power for outages.

1. **Security**: Security protects user information on and between platforms. Some methods of security include encryption, access controls, proper authentication, etc. Although Windows comes with built-in security software, using another source for data and information security is recommended. However, Windows has system scanners that detect malware, viruses, and other security threats. These processes happen in real-time, providing the highest security possible.