

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 2.0

## 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 | 07/18/2025 | Cameron Beck | Provided a summary for the project and described how the UML table operates. |
| 2.0 | 08/01/2025 | Cameron Beck | Evaluated multiple platforms for hosting a web-based software application. |
| 3.0 | 8/14/2025 | Cameron Beck | Analyzed the characteristics and techniques specific to various systems architectures to make a recommendation to The Gaming Room. |

**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 client in question, The Gaming Room, wants to develop a web-based game that is available on multiple platforms based on their current game, *Draw It or Lose It*, which is currently available only on the Android app. The game itself consists of a mix of games and it’s a team-based game consisting of multiple players on both parties. The problem the client has is that the staff has no clue on how to set up the environment.

## 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 app must be functional to all other platforms. Because of that, it must come with different software development kits to suit each platform; cross play must be one of them.

The game and team names must be unique to check whether a name is in use when choosing a team name.

The game must be able to add one or more teams from any given platform.

Since the game and team names are required to be unique, the instance of the game must be limited to one.

## [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)

<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 ProgramDriver class contains the main function, and it calls the SingletonTester class to test out any instance that exists within the GameService class. The three classes: Game, Team, and Player, all inherit the Entity class since they all contain the Entity class’s required attributes. Across the three inheritance classes, the Game cannot have GameService, but GameService can have many games in its class. The team can’t have a Game, but a Game can have a Team. Lastly, the player can’t have many teams, but the Team can have many Players. All of these are demonstrations of One-To-Many relationships within the diagram. Each team can only have one unique name at a time as well and each individual player must have one unique name.

**"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** | Characteristics: Mac is a unix-based operating system that contains a user-friendly interface.  Advantages: Mac contains strong security features necessary to hosting a web-based application. Mac also offers high quality hardware ensuring both stability and performance  Weaknesses: Both Mac’s hardware and software are more expensive than other platforms. Also, there may be some software and tools that are not compatible with Mac. | Characteristics: Linux is open source, which allows users to access and modify its source code.  Advantages: Linux servers are known for being stable and their continuous uptime without constant reboots. This is necessary for web applications.  Weaknesses: Linux has limited support and resources unlike other platforms such as Mac OS and Windows. | Characteristics:  Windows can run on multiple software simultaneously in comparison to other platforms. It's the most popular and preferred platform for webhosting.  Advantages:  Windows web hosting is consistent and flexible. It uses a Net framework and ASP which makes one’s website work faster. Windows can detect any cyber threats towards security.  Weaknesses:  Windows may suffer compatibility issues. It’s security for the Window’s web server is average; If the server malfunctions, then the Windows server will operate slowly and be prone to cyberattacks. | Characteristics:  Mobile devices are portable and can be used anywhere they go so long as they have internet connection whether Wi-Fi or 5G connection.  Advantages:  Mobile devices can easily allow their uses to access their game from any location. People use smartphones 24/7 so the potential player base can be huge.  Weaknesses:  The game’s performance relies heavily on the quality of the user’s internet. Mobile devices have very limited resources since the battery is constantly draining which strongly affects the resource-intensive apps in a very negative way. |
| **Client Side** | Cost: Both Macs development tools and hardware are rather expensive.  Time: Developing systems with Mac tends to be time consuming  Expertise: Since software development with Mac is time consuming, it would be reasonable to suggest it requires an expert level of understanding to us Mac for software development. | Cost: Linux has no licensing fees since it’s open source. It’s considered a budget-friendly option for developers and businesses  Time: Linux can be time-consuming since its users may need to manage its drivers to ensure the necessary software is installed in the system.  Expertise: There is a steep learning curve for linux, but it has a community of people to help out with that curve. | Cost: The cost, unlike Linux, isn’t cheap as upgrades and licenses tend to cost extra.  Time: It requires little time to have the game up and run because many people have interacted with Windows OS before. A lot of PC games are Windows Compatible which makes the running time fast.  Expertise: Windows servers typically have a User-Friendly interface which makes having a GUI among the Windows servers easy to use. | Cost: The cost of development may vary depending on what tools is needed to design cross-play features on mobile apps.  Time: Implementing cross-play to mobile games is a difficult task which will require time to implement.  Expertise:  Since mobile devices must support multiple types of mobile users, this requires expertise in developing gaming apps that has cross-play features. |
| **Development Tools** | Programming Languages: The programming languages used for Mac OS are Objective C and C++  IDEs: CLion - Code C and C++ apps  Microsoft Visual Studio and Android Studio | Programming Languages: Java, C, C++, Python, Ruby, JavaScript, etc. Linux supports a wide range of programming languages.  IDEs: Eclipse IDE, Visual Studio Code, Intellij IDEA | Programming Languages: C#, C++, Java, HTML, JavaScript OSA, TypeScript.  IDEs: Visual Studio Code, IntelliJ IDEA, Eclipse, Komodo IDE, NetBeans | Programming Languages: Java, Kotlin, Swift, JavaScript, Dart, C#, Python.  IDEs: Visual Studios, QT IDE, CppDroid, XCode, Arduino, Jetbrains Rider |

## 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**:
   1. The operating platform that it is recommended to allow The Gaming Room to expand *Draw it or Lose it* is Windows. Windows is the most common operating system used in comparison to other operating system choices because it has a user-friendly interface which makes the system easier to use.
2. **Operating Systems Architectures**:
   1. Windows comes in two modes which are Kernel and User mode. Kernel mode is responsible for providing low level access to both the hardware and system resources. Meanwhile, User mode is responsible for hosting both applications and subsystems making sure that they are isolated from Kernel mode for stability and security. Windows also utilizes a priority-based preemptive scheduling algorithm to allocate CPU time to threads making sure both multitasking and responsiveness is efficient.
3. **Storage Management**:
   1. An appropriate storage management system that is recommended for Windows is Azure Storage Explorer from Microsoft. You don’t have to pay a hefty price for data storage and management. You can store terabytes of data in the cloud only for a few bucks a month. Azure Storage Explorer allows you to download, upload, and manage Azure Storage blobs, files, queues, and tables. On top of that, you can configure storage permissions, access controls, tiers, and rules. All the features listed will be useful for managing storage for The Game Room.
4. **Memory Management**:
   1. Windows memory management’s responsibility involves controlling or maintaining the main memory and transferring all its processes from the primary memory to disk during its execution. Its memory management keeps tabs of all the memory’s location whether the processes use them. To add, Windows has its very own virtual address space for each 32-bit process which allows up to 4 gigabytes of memory to be viewed.
5. **Distributed Systems and Networks**:
   1. Since The Game Room is a network-based game, it would be reasonable to incorporate a database that keeps track of all the players’ name and platform the player is accessing the game from. As long as all the players are within that database and connected to within the server, players can communicate to each other whether it is through in game chat or messaging.
6. **Security**:
   1. With security playing an important role for the client to ensure protection of user information from any potential cyber-attacks, multiple platforms can start by using robust encryption protocols. The client will have every user from every platform information stored within the client’s database so if the robust encryption protocol is at play, then any data that is collected and stored will always be encrypted. Multiple systems can utilize multi-factor authentication and secure data storage pertained to each platform.