

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 | 07/18/2024 | Xavier Cheeks | <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 to turn their game “Draw It or Lose It” from an Android based application, to a web-based game on multiple platforms. They are requesting that the game is based on the Android application and need help setting up their environment and streamlining development. Each game and team name will be unique, the game will also only allow for one instance of the game to exist at a time. The game will have the ability to have one or more teams involved, and each team will have multiple players assigned. This game should be developed as a cross-platform application.

## 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 largest design constraint is developing the game on multiple platforms, so having a team that is familiar with cross-platform development would greatly benefit the project. The only other constraint would be following the requirements of the client, but with proper planning this would not be a major constraint.

## [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 the parent class of Game class, Team class, and Player class, this makes these the child classes. The child classes will inherit the attributes from the Entity class, while still being different from each other. The Game class contains a team list. The Team class contains a player list. The Player class makes sure each player has a unique ID that can be assigned to a team. Despite a player being assigned to a team, and a team having players on it, a player class does not contain a team or a game. The GameService class is used to make sure all the client’s requirements are met, this includes having a unique game, team and player name, and ensuring there is only one instance of the game at a time. The ProgramDriver class is where the main statement is located, and it uses the SingletonTester class.

**"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** | One advantage to using MacOS is the fact that it always stays consistent, anyone who has worked with it before should have no problems picking it back up. One disadvantage is that when compared to Windows or Linux, Mac products are much more expensive and harder to acquire. | One advantage to using Linux is how versatile and customizable it is, this makes some issues incredibly easy to solve compared to other Operating Systems. One disadvantage is that since Linux is the least popular of the main OS options, there is a greater likelihood for file format compatibility issues. | One advantage to using Windows is the native compatibility with Microsoft technologies and frameworks. Windows Server operating systems integrate seamlessly with Microsoft's development tools like ASP.NET, .NET Core, and Visual Studio. This can streamline the development and deployment process for applications built using these technologies, providing a cohesive environment from development through to hosting. One disadvantage is the weak security protocols, which could lead to malware in the software. | One advantage to using a Mobile Device is that it’s portable, since most people always have their mobile device on them, they can potentially adjust whenever they need to instead of the next time they’re at a computer. One disadvantage is that the main way to host on a mobile device is through cloud hosting, this could leave them most vulnerable to getting hacked. |
| **Client Side** | The advantage to MacOS is that it’s easy to learn and use. The disadvantage to this is that you must own an Apple device to even have the ability to learn or use the MacOS. | The advantage to Linux is that since it’s a free and open-source Operating System, it will be much easier and faster to solve any problems with the program. Being open-source brings disadvantages as well since there is no customer support team to call if you come across a security concern. | The advantage to Windows is that it’s easily accessible, most people can afford some kind of Windows computer. The disadvantage is that you would need to find someone who is experienced with the OS to develop the program which could increase costs. | The advantage to Mobile Devices is that they can provide a lot of flexibility to the users. The disadvantage is that there are many different Operating Systems that the developer would need to be familiar with to ensure the program works across all mobile devices. |
| **Development Tools** | The main programming language for MacOS is Swift, this was developed by Apple to work on all their devices. The official IDE for MacOS is Xcode, there is a whole suite of tools for coding, testing, and debugging projects made for MacOS. | Linux has many options to choose from when it comes to developing a project, C++ is traditionally used for system-level programming and applications on Linux due to its performance and low-level access to system resources. Visual Studio Code is an IDE that can support many languages, this makes it powerful for developing a program on Linux. | C# is a programming language that was developed by Microsoft, and is widely used for developing Windows desktop applications, enterprise software and web-services using the .NET framework. Visual Studio is the official IDE from Microsoft for developing applications for Windows, web, cloud, and mobile platforms. It supports languages like C#, C++, VB.NET, JavaScript, and Python. | Since the main mobile devices in the market are Apple or Android, the languages used, and IDE used can depend on the device. For Apple devices, Swift is the language used and Xcode is the IDE used. For Android devices, Kotlin is the modern language for Android development as a replacement to Java. Android Studio is the official IDE for Android development, based on IntelliJ IDEA, it supports Java and Kotlin. |

## 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**: The recommended Operating System for this project is Windows OS. Since the current application “Draw It or Lose It” is an Android application, Windows would be the easiest to integrate with it. Windows also has both the most users and developers of any OS meaning it will be easier to find an audience and find developers to create the project.
2. **Operating Systems Architectures**: The Windows Operating System features a straightforward Graphical User Interface (GUI) designed for intuitive navigation. It supports a diverse assortment of IDEs for application development.
3. **Storage Management**: The Windows Operating System simplifies storage management with easy-to-use configurations, this makes it easy for backend developers to navigate settings. Although owning a Windows cloud server adds an additional expense, it’s a worthwhile investment because it offers scalability for future expansions of game applications and including additional assets.
4. **Memory Management**: Windows provides diverse storage and memory management solutions, the recommended one for this project is Azure Storage. The Operating System supports virtual and physical address space for memory allocation. There are other programs that can help store and manage versions including OneDrive, Azure Cloud, or Visual Studio.
5. **Distributed Systems and Networks**: Communication between platforms can be effectively achieved through a distributed software architecture and a robust network infrastructure equipped with comprehensive redundancy and fault tolerance mechanisms.
6. **Security**: Numerous safety features are implemented, including the constant operation of the Windows Defender antivirus program on all Windows computers. There are also VPN services that are included to enhance protection of user information. Conducting daily security checks is crucial to safeguard users and prevent unauthorized access to their information. It’s also essential for developers working on the application to receive comprehensive training in securing and encrypting user data.