

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 6**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 5/21/24 | Stephanie Fowler | Initial software requirements/design |

**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 “The Gaming Room” wants to develop a web-based game that serves on multiple platforms. It currently is only available in an Android application and is called Draw It or Lose It. The client has requested that the game will have the ability to have one or more teams, each team will have multiple players, the game/team names must be unique (users can check to see if the name is in use already), and only one instance of the game can exist in the memory at a time. Draw It or Lose it contains a large library of stock drawings that will be used as clues. Also, the staff at The Gaming Room do not know 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)

* Android, IOS, and web-based applications all have different software development designs
* The API needs to be able to work with 3 different platforms
* The API need to allow one or more teams from any of the three platforms
* Game and team names need to be unique
* Limit instances of the game to only one by using unique identifiers (game, team, and player)
* Allow the ability to inform the user that a name already exists, and they need to pick something different.
* The staff at The Gaming Room do not know how to set up the environment

## [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 Program Driver class contains the main method. ProgramDriver uses directed association with SingletonTester and it tests to see if there is already an instance of GameService. Entity class is the parent class to Game, Team, and the Player classes. Where Game, Team, and Player class all inherit Entity’s required attributes. A Player can not have a Team, but a Team can have a Player. A Team can not have a Game, but a Game can have a Team. A Game can’t have a GameService, but a GameService can have a Game. GameService can only have one instance of each game running at a time, each Game can only have one unique Team at a time, and each Team can only have one of each individual Player at a time.

**"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: Can be used as a server for web-based software applications, but not as common as Linux or windows.  Advantages: Unix-based system that provides a great amount of security and stability. Also, includes features such as Apache we server.  Weaknesses: there are limited hardware choices for server grade Macs. It will be more expensive compared to Linux based solutions. | Characteristics: It is a popular choice for hosting web-based software applications. Offers a wide variety of distributions that are suitable for server use. Has excellent stability and scalability.  Advantages: It is cost-effective, easily customizable, and offers a wide variety of server software options. It is well known for its security and being reliable.  Weaknesses: Will require a lot more technical knowledge to set up and manage compared to Mac or Windows | Characteristics: Windows server is a very commonly used platform especially for applications that were built using Microsoft technologies.  Advantages: Has excellent integration with Microsoft tools and technologies. Offer web server options such as IIS. Great support for using .NET applications.  Weaknesses: Licensing costs can be very high. May not be well suited for open-source software development. | Characteristics: Mobile devices don’t normally host web applications. Instead, servers are used to serve mobile app data. Mac, Linux, and Windows can be used for mobile app backends.  Advantages: Data travels a shorter distance for reduced latency. Can be cost-efficient for different use cases.  Weaknesses: limited resources sometimes will lead to issues with performance. Scalability can be challenging. Reliability issues such as drained battery, hardware failures, and security risks. Potentially having limited bandwidth. |
| **Client Side** | Software development considerations: Developing for their clients involves using Apple’s development tools like Xcode. Costs will be in the mid-range and knowing Swift and Objective-C will most likely be required. | Sodtware development considerations: developing for them can vary, it all depends on the distribution and the desktop environment, Costs are low most of the time but having experience may be required for distribution specific considerations. | Software development considerations: Developing for them often involves using Visual Studio. The cost can vary, and the experience needed using .NET languages like C# will most likely be required. | Software development considerations: developing for them involves using platform-specific development. Costs will vary depending on the platforms that are being used. Experiences in languages like Swift for IOS, Kotlin for Android, and cross platform tools such as Flutter may be used or even required. |
| **Development Tools** | Programming Languages and Tools: Xcode is the primary IDE for Mac Applications. Supports Swift, Objective C, and C++ | Programming Languages and Tools: Supports a wide variety of programming languages. Popular IDEs are Visual Studio Code, IntelliJ IDEA, and Eclipse. | Programming Languages and Tools: Visual Studio is used as the primary IDE for Windows development, and supports language such as C#, C++, and many more other programming languages. | Programming Languages and Tools: When using IOS, Xcode and Swift/Objective C are used. Android uses Android Studio and Kotlin/Jave are most used. Cross platform tools such as Flutter (Dart) can be a streamline development for both. |

## 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**: As we move forward, I recommend using Windows throughout the next phase. I feel like it would be the easiest to use and will allow for less risks as we go.
2. **Operating Systems Architectures**: Windows requires less experience and lower costs to make the game. Windows also provides a great Graphical User Interface for all web-based applications.
3. **Storage Management** Storage sense will help maintain space on the hard drive in Windows. You will be able to manage files and keep track of how much space is being taken up. You will also be able to choose the location where things will be stored, and it is easy to look up.
4. **Memory Management**: An advantage of Windows is memory allocation, all the images you will be storing and managing will be put together in a secure area on the computer along with any code that is part of the project.
5. **Distributed Systems and Networks**: Portable Class Libraries and Net Standard Libraries are available and can be shared across the different operating systems. .net standard is most likely going to be used most.
6. **Security**: Windows has built-in security and can be used along with other security software. This will allow for double the protection. This will allow your computer to stay safe from malware and other viruses. Having double the protection and making sure it stays up to date is important so your system will be less vulnerable and wont get impacted.