

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 | 3/19/2023 | Camrin Stilwell | Design a software summary with the client’s requirements for a gaming application. |
| 1.1 | 4/2/2023 | Camrin Stilwell | Edited and updated Development Tools |
| 1.2 | 4/15/2023 | Camrin Stilwell | Reviewed and updated Recommendations |

**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*, requests a web-based game that will serve multiple platforms based on their current game, *Draw It or Lose It*, only available on Android devices currently. The game will be loosely like the popular 1980s game show, *Win, Lose or Draw*.

## Requirements

* Game must run on multiple platforms.
* Multiple teams will include multiple players.
* Unique game and team names will be needed and checked.
* Only one instance of the game can exist in memory at any time.
* Programs will be written using the Java language.

## [Design Constraints](#_2et92p0)

The game must be made available to users on multiple platforms, using what is present from the current application on Android, this code may require more developers to write and change it from its Android system, to be supported elsewhere.

## [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 Game, Team, and Player classes will be inherited directly from the Entity class which will help prevent any repetitive code being written in every class. The SingletonTester and ProgramDriver will have a usage relationship, the ability to execute the application, and test it using the main class. The GameService, Game, Team, and Player classes have the association of zero to many of each other based on the model.

**"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** | An advantage is having a flexible command terminal to create a server, give access, and make alterations when needed. It has different technologic implementations and fewer restraints. A weakness is the licensing, expenses, and lack of popularity compared to other OS. | An advantage is also having a flexible command terminal. It is also cost-effective compared to other options, saving from cheaper capacity size and lack of licensing costs. A weakness is having minimal technical restraints. | An advantage is the numerous amount of software that is available. Due to its popularity, more software exists for this OS than other systems. Its licensing is also cheaper than Mac. | An advantage is the scalability to web-host, however its limitation is the requirement of having a device with internet connectivity. In addition to its advantages, it also has inexpensive licensing. |
| **Client Side** | Needs moderate amount of time and experience but at a high cost. This also supports current browsers, has secure authentication, and protocols to prevent hacking. The downside is having minimal support when it comes to media types. | Needs a high amount of time and experience but at a low cost due to the OS free accessibility to the public. Limited media types. Limited number of devices can run this OS well. | Needs the least amount of time and experience. The cost is greater than Linux but less than Mac. | Flexible developing base for clients, real-time updates can be monitored. This is a bit more difficult to implement than other devices. Has a wide range of accessibility to smartphones and tablets. |
| **Development Tools** | Most likely would be using Xcode IDE (free) with Swift language as this is only used on Mac devices. This would be the only requirement for the Dev. Team to learn how to use the IDE and language to create the software. | Most likely would be using Eclipse (free) or similar IDEs for development and cross-platforming, using the most common language on Linux, Java. The Dev. Team would need to know java and how to use the IDEs to create the software. | Most likely would be using a popular IDE such as Visual Studio (free – paid subscription), using a more common language on Windows, Java, C, or C++. More flexibility for the Dev. Team on what language to use, and they can find an IDE that is free or pay to use the professional subscription version of Visual Studio. | IOS and Android use different programming languages. IOS would similarly use Swift, while Android would most likely use Java or C++. Similar situation to both Mac and Windows, the Dev. Team would need to split into groups to create applications meant for the specific device. |

## 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**: It would be recommended to *The Gaming Room* to use Windows OS to develop the *Draw It or Lost It* application. Windows requires the least amount of experience, has a vast library of software, and are all provided at a reasonable cost.
2. **Operating Systems Architectures**: The Windows architecture interface gives services used by all Window-based applications such as access to system resources, Graphical User Interface, memory, audio, etc. This also allows applications to use the computer’s operating systems without directly interfering with the systems. Example: Access to memory that makes up the application without affecting how the operating system works.
3. **Storage Management**: The application can be stored in a user’s device which supports fast loading time. Windows also supports cloud storage which gives a large amount of storage space if needed. Another Windows system utility helps to maintain storage by deleting unimportant files that are taking up space.
4. **Memory Management**: Windows has a Memory Management system built in, which allows the system to store and retrieve data from a different storage to use in the main memory. A database for the game’s image files would be needed as it will allow easy access for the application to retrieve.
5. **Distributed Systems and Networks**: Windows OS can provide communication and data sharing between applications. Each client application will be dependable on a single server application for the game. However, a strong server network would also be needed, as the game itself needs to depend on multiple clients to connect to the server to play one game together.
6. **Security**: Windows Security provides antivirus protection. This would be included as data for the game would need to be encrypted to protect users of the application and prevent other threats.