

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 | 03/21/2024 | Matthew Marsh | Changes were made to the cover page, the executive summary, design constraints, system architecture view, domain model and recommendation. |
| 2.0 | 04/01/2024 | Matthew Marsh | Characteristics, Advantages, and Weaknesses added to the Development Requirements table for the Server Side, Client Side, and Development Tools. |
| 3.0 | 04/17/2024 | Matthew Marsh | Update the recommendations more in depth. |

**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, is asking us to develop a web-based game called “Draw It or Lose It” that will be able to be used on multiple platforms. As of now, it is only available as an Android app. The app, based on the popular television game show, consists of four rounds of play, each round is for one minute.

## Requirements

* The game must have the ability to have one or more teams involved.
* Each of the teams will have multiple players assigned to it.
* Game and team names must be unique to allow users to see if a name is in use when choosing their team’s name.
* Only one game can be in the memory. Will need to create unique identifiers for each instance of a game, team, or player.

## [Design Constraints](#_2et92p0)

* Must be able to run on multiple platforms.
* Each team must have multiple players.
* Only one game can be played at the same time.
* The game and team names must be unique to allow the users to check and see if those names are already in use.

## [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 creates a relationship between the Game, Team, and Player classes. The arrow indicates that all these classes inherit an attribute from a super class. The programDriver class, which is pointing to the singletonTester class, shows us that the programDriver will use the singletonTester class to test the code to make sure it is working. This helps to ensure that only one game is stored in the memory at the same time. The numbers between the lines between the classes lets us know the number of associations within each class. We will use this diagram to create the game.

**"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’s servers offer a more reliable and stable environment for the game to run on.  **Advantages:**  Mac is Unix based, which is great for web hosting.  **Weaknesses:**  The mac has very limited software options in comparison to Linux. | **Characteristics:**  The Linux servers are a more powerful and efficient platform.  **Advantages:**  It has multiple choices for server software and the cost makes it economical.  **Weaknesses:**  The Linux system requires that you have knowledge of Linux to get it set up. | **Characteristics:**  The multitude of web hosting tools makes Windows more user friendly.  **Advantages:**  Windows makes the administrator interface easy to use.  **Weaknesses:**  Windows products and licensing are more expensive. | **Characteristics:**  Mobile devices can access web apps, but you would need to have the cloud service to host them.  **Advantages:**  Mobile devices are highly scalable through cloud hosting.  **Weaknesses:**  You have limited control over the server hosting environment. |
| **Client Side** | **Considerations:**  You need to have knowledge of many web-based technologies such as: HTML, CSS, and JavaScript. You will need to do testing on Safari to ensure it works with the Apple systems. | **Considerations:**  As with the Mac systems, you need to have knowledge of web-based technologies. You must also take cost and time into consideration. You would need to run tests throughout the Linux systems to ensure they are compatible. | **Considerations:**  Most of the windows client development follows all the standard web practices. The cost and time are minimal, which saves the company money. Ensuring that you test your product with all the different window versions is crucial. | **Considerations:**  You need to have knowledge in mobile app development for both the iOS and android devices. Cost and time can be high due to the different needs of each different platform. You also need to know how to do proper coding for the different web browsers to ensure that they work with the different platforms. |
| **Development Tools** | Languages that support HTML, CSS, and JavaScript. Many libraries assist with support and development. Can be used in programs like PyCharm and Visual Studios. | Languages that support HTML, CSS, and JavaScript. Many libraries assist with support and development. Linux systems include JavaScript, Ruby, PHP, and Python. | Languages that support HTML, CSS, and JavaScript. Many libraries assist with support and development. The Developer tools include Eclipse, PyCharm, etc. | Languages that support HTML, CSS, and JavaScript. Many libraries assist with support and development. IDE’s for programming include HTML, C++ and Python |

## 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 appropriate operating platform which will allow The Gaming Room to expand on Draw It or Lose It is windows. Windows has many users, which will allow for a better distribution of the game to a larger audience. The plethora of IDE’s makes windows the best option for the game. With these options, this will allow for the cost of the game to remain within the constraints that the customer is asking for.
2. **Operating Systems Architectures**: Windows is an operating system created by Microsoft. Its wide range of operations allows it to run software, store files, watch videos, and so much more. Windows users can develop and set up the environment the way that they want with an interactive GUI that is easy to use and well established. If the users have any issues with the game, Windows has great technical support and a plethora of documents that can help users tackle almost any issue that they come across.
3. **Storage Management**: I believe that the Windows storage sense would allow you to be able to manage your files better on your hard drive and allow you to better manage storage space. Windows also has the options to use Dropbox and OneDrive to save files to, so they don’t have to use their own hard drive space to do it.
4. **Memory Management**: Windows storage sense would allow better management of the photos and game player for the game. It would also allow you the ability to keep them all in one location for better access. Windows also gives you the ability to specify how much space you want to reserve for any application. Many users prefer to have control over what the applications are doing to their computers.
5. **Distributed Systems and Networks**: With now having the option of cloud servers, this allows us to run servers to allow the users to call the application on whatever platform they are using at the time. The ability to configure the requests and respond with different instances of the game will allow them to communicate more effectively and easily.
6. **Security**: Windows comes with built-in security software solutions. You can also secure data and information from other software sources to help keep the app secure. You also can implement firewalls and different types of encryptions. This allows the company to respond to hackers more effectively and help to prevent any cyber-attacks. You can also rely on the support staff at Microsoft to help with any support that is needed to help secure your servers and game effectively.