

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 07/24/22 | Annaliese Pintar | Filled in Executive Summary and Design Constraints |
| 2.0 | 08/06/22 | Annaliese Pintar | Completed Evaluation Table |
| 3.0 | 08/21/22 | Annaliese Pintar | 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)

Creative Technology Solutions’ client, The Gaming Room, wants to develop a web-based game that works on multiple platforms based on their Android app game, Draw It or Lose It. The game should allow one or more teams, each team with unique names, with multiple players to play. This can be done by assigning players to teams with the number of teams depending on the number of players. Once teams are assigned, the teams can each pick team names, and the program will check to make sure each name is unique. Also, only one instance of the game can exist at a time. To keep multiple instances from being created, each instance of the game, team, and player will have unique identifiers.

## [Design Constraints](#_2et92p0)

The game must be playable on multiple platforms. During development, the possible platforms need to be kept in mind to make sure that the game displays well on the different platforms. Also, internet reliance must be taken into consideration. If a player loses connection, you will want to make sure they will be able to rejoin the game. Speed is another thing to consider. Web-based games can potentially be slower.

## [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 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** | Mac is more secure than Linux and Windows. However, the licensing fee is particularly high. Mac is also more stable than windows so there will be less time your site it down. | Linux is an open-source platform so it is customizable and versatile. It does offer a server based deployment method for hosting. Also, most Linux distributions are free so it is the more affordable option for hosting. Linux can be more secure than windows, but not as secure as Mac. | Windows Server-based hosting allows you access to Windows proprietary products that you would not have access to using another operating system. But, windows is less secure and requires frequent updates which means your site will have to be down. Also, licensing costs will be high. Windows is also less secure than Mac and Linux. | Operating systems for mobile devices are not really meant for hosting web-based applications. |
| **Client Side** | Mac hardware is more expensive than the hardware required for Windows and Linux. You would need to use an emulator for the different browsers and devices to ensure things would run properly. | Because Linux is open-source, there is more that can be done when it comes to upgrades and optimization. However, with there being much more customizing to do, development might take longer. The hardware required for Linux is much cheaper than Mac and Windows.  You would need to use an emulator for the different browsers and devices to ensure things would run properly. | More people are familiar with windows, so the expertise required is less. Windows hardware is less expensive than Mac, but more expensive than hardware Linux requires.  You would need to use an emulator for the different browsers and devices to ensure things would run properly. |  |
| **Development Tools** | For Mac you would use the programming language C. The IDE I would use is Visual Studios which would cost about $1200 a year. | For Linux, you would also use the programming language C. The IDE I would use is Visual Studios which would cost about $1200 a year. | For windows you would use the programming languages C, C++, or C#. The IDE I would use is Visual Studios which would cost about $1200 a year. |  |

## 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 operating platform I would recommend for hosting Draw It of Lose It is Linux. This is because it is more cost effective than both Mac and Windows and more secure than Windows. Also, because it is open source, it is also more customizable.

1. **Operating Systems Architectures**:

The Linux system architecture is a layered system in which the layers only directly interact with layers above or below it. There are four main layers in the architecture of Linux: application programs, shell, kernel, and hardware. The hardware is the physical parts of the computer (mouse, CPU, monitor, keyboard, etc.). The kernel manages computer and hardware operations. The shell gets user input and sends instructions based off that input to the kernel (user interface). Applications are programs like test editor, web browsers, games, etc. that run on Shell. A layered system allows for layers to be modified without modifying other layers.

1. **Storage Management**:

Logical Volume Manager (LVM) is a storage management system that can be used. LVM sorts storage into logical volumes based on maximum capacity of the hard disk drive. It is scalable and flexible (capacity can be reallocated).

1. **Memory Management**:

The kernel can refer to non-existent memory called virtual memory. Virtual memory makes it seem like there is more memory than physical memory by spreading it across competing processes. Virtual address space is mapped to physical address space. This allows for only the necessary pages to take up main memory. This will be very important for Draw It or Lost It because things like the different images for each round will need to be brought to main memory during the round and then flushed after the round is over to make space for the image in the next round.

1. **Distributed Systems and Networks**:

Draw It or Lost It will be played online through a web browser which will allow multiple users to play for any device that can open a viable web browsers. Because it is online, if a user loses connection, they should be able to rejoin as soon as they regain connection. To make sure the image is optimized for different platforms, there can be multiple copies of each of the image and the browser will select only the image size best for the device. This will increase the performance of the game and ensure that the game is optimized for different platforms.

1. **Security**:

User information can be protected by making sure any sensitive information is only accessible by entering a password. Two-factor authentication is one of the best ways to protect user information with a password. It also will be important to only get necessary information from the user. Linux is open source so many developers contribute to security which can potentially increase security capabilities.