

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 | <March 31, 2024> | Kaleb Vaughn | I have created an executive summary, design constraints, and a domain model. With revised code. |

**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)

<Write a summary to introduce the software design problem and present a solution. Be sure to provide the client with any critical information they must know in order to proceed with the process you are proposing.>

**The Gaming Room company is looking to develop a web-based game that can be used on multiple platforms, that is based off their most current game. This game is called, “Draw it or Lose it” this can currently only be played on Android devices. It is a team-based game, that consists of four rounds, where a team must guess the puzzle before the time expires. They will be guessing the puzzle based off a large library of stock drawings as clues, where their drawings are rendered at smooth pace and that are completed at the 30-second mark. Also, if the teams fail to guess the puzzle before the time expires, then the opposing teams will have the opportunity to make a guess each with a time limit of 15 seconds. The Gaming Room needs assistance with the streamlining development since the staff at The Gaming Room are not familiar with setting 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)

<Identify the design constraints for developing the game application in a web-based distributed environment and explain the implications of the design constraints on application development.>

**Needs to have multiple rounds built into the game. Needs multiple teams, with unique team names. Needs a timer, for the drawings, the team to answer, and for the opposing teams to steal. Needs to be built in a programming language that can be used for web based. Needs to generate a large library of drawings for teams to guess. Needs to follow the policies of the multiple different platforms.**

## [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.>

**Game, Team, and Player all inherit from the Entity class. GameService, Game, Team, and Player, have association of one-to-many relationship between one another. ProgramDriver and SingleTester have an association that uses each other. Game, Team, and Player are subclasses, while Entity is a superclass.**

**"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 does offer web hosting for server-based deployment. Maintaining a server on Mac can become quite expensive so that is a weakness. Mac has great flexibility and powerful tools to expand the server. Unix is also great for web hosting. | Linux does offer web hosting for server-based deployment. Specifically, Linux hosting has lower prices than its competitors. Since Linux is an open-source licensing will be under something like GNU or GPU. It has very stable running and runs quite quickly. A weakness of Linux it can be hard for new users to use because of how many barriers it contains, so learning Linux could take a lot of time. | Windows does offer web hosting for server-based deployment. Since it is proprietary software, licensing will cost. A weakness is the licensing can become quite expensive. Windows is very easy to use and understand. Contains a lot of support services and tools to help. Majority of software is compatible with Windows. | Mobile devices can be used to access web applications, but it is quite tricky to do web hosting. A weakness is how limited it is when it comes to server hosting. Oracle cloud is a great service for building big data applications with outstanding performance. Oracle database helps manage devices and mobile applications keeping them synchronized, and organized. |
| **Client Side** | The cost of MacOS depends on the hardware you get, while the MacOS updates are free, the Apple Hardware can be quite pricey. Developing expertise requires knowledge on OS, web technologies, with an average amount of time spent. | The cost Linux is free to use and is an open-source operating system. A lot of time required with expertise knowledge on Linux data to properly use the operating system. Testing is need for compatibility reasons. | The cost depends on the version you get, it can be either $139 for Windows 10 or $199 for Windows 10 Pro. Average to low time is needed. Relatively easy to understand so expertise can be low. Knowledge of standard windows versions. | The cost depends all on the hardware you choose to get between apple, and android. Can be quite difficult to implement but is very flexible. Expertise in mobile app development for IOS and Android will be required. Time will be high, since dealing with multiple platforms. |
| **Development Tools** | Programming languages used are Objective-C, C++, Java, HTML, CSS, Swift, and tools to build this type of software are Visual Studios, Eclipse, PyCharm, WebStorm, NetBeans, Rider. | Programming languages used Python, C, C++, Java, JavaScript, and tools to build this type of software are PyCharm, WebStorm, Visual Studio, CLion, Atom, Qt, GTK, Geany. | Programming languages used are JavaScript, HTML, CSS, Python, C++, C# and tools to build this type of software are Eclipse, PyCharm, Visual Studio, WebStorm, Notepad++, Xcode, WinUI. | Programming languages used are Java, C#, Objective-C, JavaScript, Python, and tools to build this type of software Android Studio, and Xcode. |

## 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**:

**I would recommend Windows, the reason being is not only is it the most widely used operating system, but it is very easy to use and can be easily integrated with the current Draw it or Lose it Android application. It offers a lot of supporting services, tools, and it is very compatible.**

1. **Operating Systems Architectures**:

**Windows architectures can use the applications kernel mode, which gives a layered approach of services. This includes managing memory, security, processes, and privileges. It is one of the most trusted for operating systems like Windows and it has multiple rings of protection.**

1. **Storage Management**:

**Windows offers its own storage management system that offers several features. Storage spaces offers to combine multiple hard drives together into one logical drive, which provides you a storage pool that can be managed. It also has a feature that provides a tool that can clear up unnecessary files, called Disk Cleanup. There are many other features and services that can help maintain storage.**

1. **Memory Management**:

**Windows has a built-in memory management, with also a virtual address space. Which is great for Draw It or Lose it because this gives us fast access to data, and with a great transferring processes between memory to disk. Also, it keeps track of all memory locations, and can effectively maintain the status of the memory, like the game images files for Draw It or Lost It.**

1. **Distributed Systems and Networks**:

**Using a client-server system is needed here, because clients can request the server for data or an objective, and the server can respond by assigning the information/tasks and sending them back to the client. This game relies on multiple users on at once and client-server provides strong multiple server networks, with great connectivity, that give clients communication and efficient maintenance if needed.**

1. **Security**:

**The way to protect user information is to ensure that users do not have access to information they shouldn’t have access to. Personal information is not shared and only access by that given user. Encrypted data can be extremely helpful to prevent hackers from accessing information, and if they by chance did, they would not be able to decrypt the information. Also, we would want it to be antivirus compatible. Something like McAfee would work, even though it can be quite expensive, it is user-friendly to applications, has a VPN, compatible for all devices, effective detection, offers more services than many other softwares, and does not take up performance for systems.**