

<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 | <05/20/22> | <Alexander-Egelston> | <Evaluating the characteristics, advantages, and weaknesses of each operating systems to present to your client> |

**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 Gaming Room wants to develop a web-based version of their gaming app Draw it or Lose it. Currently The Gaming Room only has an android app developed for their game. The app pulls an image from a library and is slowly rendered over 30 seconds. If the team does not guess the puzzle by 30 seconds, then the other teams each have 15 seconds to guess the puzzle. The game in total lasts 4 rounds at 1 minute per round.

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.>

## [Design Constraints](#_2et92p0)

<The game requires multiple teams with differing team names. An aspect of the design needs to ensure each name is only being used once. Only one instance of the game can exist at one time which will require the use of a singleton tester to check for multiple instances. The project will also require the code to be written for different application types such as iOS and a web-based version of the android app which may require hiring different teams of developers.>

## [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 relationships with the Game, Team, and Player classes. The Game, Team, and Player classes inherit information from the Entity class. The Entity class takes the information from each of the classes such as the name and ID. These Strings are useful in creating a unique log in instance.>

**"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** | <There are many options for Mac based web hosting. Licensing for Mac based devices starts at $29.99 making it very affordable. Mac is a very popular option and easily customizable. It is more commonly used for user level projects and less so for web hosting. Many teams not oriented to the artistic aspect of development will find Mac to be an unfamiliar or inconvenient choice compared to Linux or Windows.> | <There are many options for Linux based web hosting. Linux is an open-sourced operating system and most of the necessary resources can be found for free as well so licensing costs can be as little as free. Linux is free cost friendly and secure compared to OS. The main disadvantage to Linux is the lack of availability in supportive resources. The main disadvantage for Linux is the lack of user support.> | <There are many options for Windows based web hosting. Licensing fees for Windows can be as little as $30 to well over $100 depending on the package selected. Windows is easily accessible and familiar to most making it the comfortable choice for many developers. It has a lot of community support but is notoriously susceptible to viruses which for most is its biggest drawback.> | <There are many options for mobile device-based web hosting. It does vary but many mobile devices are open source making them free for licensing much like Linux. Mobile devices are generally cost effective, portable and have higher specs. The main disadvantage is they are mostly particular to their brand of device and would require additional development for each type or brand of device.> |
| **Client Side** | <Cost is moderate for developing on a Mac based system. Time and expertise required to utilize Mac are low, especially compared to the Linux OS. This ease of use can cause the development team to complete the project in a timely fashion. There are very little to no costs associated with the development tools as many can be obtained for free.> | <Cost is low for developing on a Linux based system. Time and expertise required to utilize Linux are very high with the little community support with this OS. There are very little to no costs associated with the development tools as many can be obtained for free.> | <Cost is moderate for developing on a Windows based system. Time and expertise required to utilize Windows is minimal considering the massive community support for the OS. There are very little to no costs associated with the development tools as many can be obtained for free.> | <Cost is low for developing on a mobile device-based system project. Time and expertise required to utilize mobile development is high compared to other OS. There are very little to no costs associated with the development tools as many can be obtained for free.> |
| **Development Tools** | <Relevant programming languages and tools for Mac are Swift, HTML, JavaScript, and Notepad++. These easy-to-use development tools can help the development team to complete the project in a timely fashion. There are very little to no costs associated with the development tools as many can be obtained for free.> | < Relevant programming languages and tools for Linux are Visual Studios, Eclipse, JavaScript, Java, Python, HTML, CSS, and Notepad++. These easy-to-use development tools can help the development team to complete the project in a timely fashion. There are very little to no costs associated with the development tools as many can be obtained for free> | < Relevant programming languages and tools for Windows are Visual Studios, Eclipse, PyCharm, JavaScript, Java, Python, C++, HTML, CSS, and Notepad++. These easy-to-use development tools can help the development team to complete the project in a timely fashion. There are very little to no costs associated with the development tools as many can be obtained for free> | < Relevant programming languages and tools for Mobile Devices are Visual Studios, Eclipse, PyCharm, Java, JavaScript, CSS, HTML, Swift, Python, Objective-C, and Notepad++. These development tools are more specific to the brand/type of device and will need adjustment to accommodate different mobile device OS. This can hinder the development team in completing the project in a timely fashion and may require the help of additional teams. There are very little to no costs associated with the development tools as many can be obtained for free> |

## 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**: <An appropriate operating platform that would allow The Gaming Room to expand Draw It or Lose It to other computing environments is Windows. Windows is fairly easy to work with compared to others and has massive community support. While it is not a free operating platform, Windows is going to be more familiar to most developers and will in turn save time and effort as the comfortable choice. Windows is very user-friendly and will require less expertise than other choices as well.>
2. **Operating Systems Architectures**: <Windows provides most of its products and services through Windows-based applications. While this is an exclusive system format that forces users to purchase product keys to utilize, it is extremely easy to use once purchased. Product keys can be purchased at an individual level for single person use or at a team/company level to help accommodate the needs of projects of all sizes. There is also massive online support for all windows applications to accommodate all needs in the development process.>
3. **Storage Management**: <Windows has cloud-based saving options as well as options to choose file save locations to make managing storage easier. Windows also has a feature called storage sense that manages files on your device’s hard drive as well as tracking the amount of space each file takes to store. Windows applications also store temporary links to quickly open files you recently had opened in order to cut down on time spent looking for files.>
4. **Memory Management**: <While Windows does have a default picture storage location, you can easily set files to save in a separate location. Storing the entire library of pictures for this application on Windows is simple enough with a separate file created to store all of the pictures selected.>
5. **Distributed Systems and Networks**: <The software written for the game application can be adapted to the various operating platforms The Gaming Room wants to utilize. The different adaptations can then be connected to separate servers on the same network to keep all applications organized. If mitigating outages and loss of connectivity are large concerns then having separate servers would isolate possible outages to the servers that are affected by the outage instead of having all applications on the same server and losing all during an outage or loss of connectivity.>
6. **Security**: <Though Windows is widely known to be susceptible to viruses, its built-in software does periodically scan for viruses and malware. These device system checks will keep your project free from malware and viruses. There are also many additional firewall programs you can have installed on your device as well with Windows.>