

<Name-of-Software-Application>

# **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 | <07/17/2020> | <Cory Daniel> | <Evaulate the operating systems for the client program Draw it or Lose it> |

**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 wants to create more fluidity in the gaming application of various operating system platforms, with different patterns in the software distributed in different environments. For a solution to the design problem, we will evaluate the different operating systems and see how to perform the game best on each platform.

## [Design Constraints](#_2et92p0)

Some of the design constraints for the game application have to do with how to integrate users in the various operating platforms and how they function. A design constraint for web-based distributed environments would be tailoring the project and coding for the development to the game for each operating system, making sure we utilize each web-based environments on said system. We want to make sure we can add more then one team and have multiple people per team, so we must design the gam application to be able to add multiple players.

## [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 UML diagram has the class attributes that are inherited from the entity class, which holds the strings for to get names and team IDs. We can see with the game, team, and player classes, that they have a “has” relationship with the entity class, due to the way the arrow is empty. We can also see from our game service class the UML diagram we edited on a future assignment, showing a reflective association with the game class.

****

## [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 has advantages in terms of having easy terminal commands to make changes to web based software applications | Linux runs on most devices, so it has the advantage in this department over Mac. | Larger library of software for the web based software, giving more variety for the operating system in terms of the others. | Mobile devices are at a disadvantaged as the serves are technically moving rather then begin stationary. Washed down version of other operation systems with less specifications. |
| **Client Side** | A little bit of expertise is needed to navigate the various apps and docks on the platform, along with understanding certain commands as well. Could be pricey depending on what equipment. | Minimal cost but the hardest of the operating systems to learn, even downloading files can be tricky with using this operating system. | Some moderate expertise is needed, but this is the most commonly used and learned on operating system for most of the population. Could be pricy on the client side depending on what equipment is being used. | Provides flexibility and options for clients on the go. Could have some trouble learning different devices and how apps and software run on each different mobile device. |
| **Development Tools** | Macs can run all languages such as HTLS/CSS/JAVA and general libraries. The libraries consist of python, ruby, and the mentioned above. | Linux can run with the eclipse IDE that we use within this course, as well as other languages | Windows Macs can run all languages such as HTLS/CSS/JAVA and general libraries. The libraries consist of python, ruby, and the mentioned above. | Android and iPhone devices and run all the programs from said machines in previous cells. |

## 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 system I will recommend for The Gaming Room would be windows, as the expertise needed, and the cost efficiency would best benefit the company. We would be able to scale it across multiple computing environments.
2. **Operating Systems Architectures**: Windows can provide GUI (graphical user interface) and the system resources to help service the windows based applications. These applications have to do with web services, messaging, multimedia throughout the platform, and can be tailored to certain user preferences.
3. **Storage Management**: Windows storage can be adjusted using various hard or solid state drive models, as well as using applications such as OneDrive. These files can be organized in folders and managed through a storage settings system tailored to delete programs and larger files.
4. **Memory Management**: With windows, we will eb able to use the memory management to create large folders of pictures that will be used for the program. The accessibility and easy storage of these files makes it an advantage for windows as the program gains advantages for its user-friendly adaptability. <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.>
5. **Distributed Systems and Networks**: One of the ways to accomplish Draw It or Lose It to communicate on various platforms would be the accessibility of the files. A way this could be achieved would be through DropBox, a user friendly and multi-platform cloud sharing software that will help us share files across multiple platforms. <Knowing that the client would like Draw It or Lose It to communicate between various platforms, explain how this may be accomplished with distributed software and the network that connects the devices. Consider the dependencies between the components within the distributed systems and networks (connectivity, outages, and so on).>
6. **Security**: When using windows, we already have built- in security through the operating system. It constantly updates to help us protect user information within the platform and throughout other platforms. Windows also helps us backup files and fix problems that relate to security issue we might have with a program. It might also benefit the program to protect user information by using a third party software that can help us create security across the platforms .