`



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 | <01/21/2021> | April Morris | Add ability to have one or more teams involved with multiple players. Make unique team names required. Make only one game instance exist in memory at any given time. |

**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)

Draw It or Lose it is a game where players/ teams guess what is being drawn. The client requires that the game offer multi-team play with unique player and team ids and only one instance of the game in memory at a time. Our team must produce distinctive identifiers for each game, player, and team to meet our client’s needs.

## [Design Constraints](#_2et92p0)

Design constraints include developing an application that can be used across 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)

The UML diagram contains classes used for The Gaming Room application and Draw it or Lose it.

****

## [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 an excellent workstation for accessibility for server configuration.  Expensive | Challenging to navigate platform.  Simple server configuration and accessibility. Not as popular so there is a need for someone who is familiar with it to run the server.  More cost effective because it is open source and that includes a lot of resources that will be needed. | Easy to navigate platform. Probably the most popular platform so easiest to find someone to run server.  Costly. | Challenging to create the game for all mobile platiforms. May not have all of the functions that the other OS are able to run.  Inexpensive |
| **Client Side** | Mac support will be expensive but takes an average to above average amount of time to do so. | It will take much time to support users of Linux.  Minimum cost or no cost. Must find someone with a lot of experience to run it. | Windows users will require the least amount of time to support but will cost about the same as Mac. | To support mobile users with different OS it will take maximum time and proficiency in each OS. |
| **Development Tools** | Programming languages, IDEs and tools for Mac include  JavaScript, CSS, Python, Html, Eclipse, and Visual Studio.  My recommendation would be Swift. | Programing tools and languages include Visual Studio, Command prompt, Python, Java, Html, etc.  I do not have a recommendation for Linux as I am not proficient in its usage. | Languages and tools used for Windows include Java, Python, C++, Html, Eclipse, command prompt, Visual Studio, etc.  My top picks would be Eclipse and Visual Basic. | Languages and tools include Python, Html, C++, and Visual Studio, etc.  Swift could be used for Iphones and other mobile devices. |

## 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**As Windows is the most popular and commonly used platform it would be my recommendation. With a lower total cost, ability to easily find someone to run it, and the most choices for IDEs, I believe this is the way the client should go.
2. **Operating Systems Architectures**: Windows can access system recources while showing GUI.
3. **Storage Management**: <Identify an appropriate storage management system to be used with the recommended operating platform.> Windows allows users to choose save locations, manage files, on a hard drive or in cloud storage.
4. **Memory Management**: <Explain how the recommended operating platform uses memory management techniques for the Draw It or Lose It software.> There will need to be a large collection of pictures for this game, with Windows you are able to store such large items together in a chosen spot to keep it secure. Because of Windows ability to quickly access these files it also makes it a no brainer choice for this application.
5. **Distributed Systems and Networks**: <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**: <Security is a must-have for the client. Explain how to protect user information on and between various platforms. Consider the user protection and security capabilities of the recommended operating platform.> Windows does have built in security, but it isn’t the most secure so I would recommend third party security. Although it may add cost to the project, I still believe that all in all Windows is still the most cost effective option.