

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 |
| --- | --- | --- | --- |
| 1.2 | 08/11/2021 | Ryan Edwards | Analyze and make recommendations to The Gaming Room |

## [Executive Summary](#_sbfa50wo7nsh)

The client wishes to expand their gaming application onto various other platforms, with different and distinct software. As for solutions for the design problems, we will have to evaluate all of the different operating systems being considered for the program.

## [Design Constraints](#_2et92p0)

There are a few main design constraints presented. The main one is developing the application in a manner to allow users to use the program from various operating systems and platforms. The other is that the client wishes for user authentication methods to be implemented so that user information as well as user access can remain secure.

## [System Architecture View](#_ilbxbyevv6b6)

States nothing is required here. Can update and resubmit if necessary.

## [Domain Model](#_8h2ehzxfam4o)

The UML diagram shown below details the classes implemented into the gaming room from their game Draw It or Lose It. In it the object oriented class attributes are inherited from the Entity class. It also shows how the Game, Team and Player classes are related to the Entity class via a has a relationship. Lastly encapsulation is shown in regards to the Entity class as it secures information by restricting access to the methods held in the application, thus protecting its user data.



## [Evaluation](#_2o15spng8stw)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | A nice graphical interface for users. Provides ease of use as well as easy accessibility. | There can be difficulties navigating OS, however has command shell for easy configuration. | Very friendly GUI for user as well as plenty of options for software. Command Prompt is quite useful as well. | Has most variation in regards to user. Will be require more effort to create game that works with multiple mobile platforms. |
| **Client Side** | Will require an average amount of time as well as require decent skill for navigation. | Will require more time and higher skill proficiency compared to other OS. Requires knowledge of Linux to operate. | Does not require too much time or proficiency from users. Easy to learn. | Will require a lot more time as well as skill to support mobile devices. Can be difficult to use applications designed for other systems and platforms. |
| **Development Tools** | Possible IDE for Mac include JavaScript, CSS, HTML, and Python. Possible tools include Eclipse and Visual Studio. | Possible IDE for Linux include Python, Java, HTML, and Ruby on Rails. Possible tools include GitHub, Visual Studio, and nodejs. | Possible IDE for Window OS include Python, HTML, and C++. Possible tools include Command Prompt, Eclipse, as well as a plethora of others. | Programming languages supported consist of Python, C++, and HTML. Possible tools include Command Prompt, GitHub, and Visual Studio. |

## Recommendations

1. **Operating Platform**: For Draw it or Lose it, Windows OS would provide the best results for further expansion. Windows OS offers reliability in helping client with specific requests for computers running on a particular network. Windows OS also includes multiple server roles as well as being designed to run on their hardware.
2. **Operating Systems Architectures**: Windows OS architecture includes a large variety of systems and concepts that would allow applications to utilize the windows platforms processes without negatively affecting the processes making the platform itself work. This would allow the user to utilize a computer to suit their needs while not negatively affecting everything else.
3. **Storage Management**: Windows allows for routine reallocation of information for servers, suiting the users need. Also provides proper disk management and storage sense to handle more advanced storage tasks while removing unnecessary files. Can ensure machines have the required storage space available for modification as well as run smoothly.
4. **Memory Management**: Windows OS contains RAM as well as both a physical and virtual address space for memory management. Would need to create a database to store the image files for the application to access and utilize.
5. **Distributed Systems and Networks**: Utilizing networking support in distributed systems will help with implementing software onto those systems. This offers simple communication between multiple workstations. Will also allow tasks to be divided and processed among the user and server for workstations on the same network.
6. **Security**: Windows OS Windows Defender Application. The provided security helps with protecting sensitive date as well as preventing breaches. It will block harmful attacks detected as well as increase the overall security. Includes built in protection against malware and corruption.