

**Draw It or Lose It Web-Based Game**

# **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 | <08/19/2021> | <Justin-Haby > | <Brief description of changes in this revision> |

**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 task is to create a web-based version of the Draw It or Lose It game with the objective of expanding the range of platforms through which the game can be played. The game is to be played in four rounds, with each round taking a maximum of 1 (one) minute. A player is to belong to one of several unique teams. A player has 30 seconds to draw the correct puzzle with the help of a library of stock drawings that provide clues. Expiry of this time (30 seconds) means that the other remaining teams have 15 seconds to each give a guess for the puzzle. Teams, players, as well as game instances will be assigned unique ids so that a player can only belong to one team at any given time and so that there is only a single player playing the game at any given time.

## [Design Constraints](#_2et92p0)

Among the design constraints brought upon by the development of the gaming application on a web-based platform is coming up with a simplistic user interface design for the game. The goal is to enhance the user(gamer) experience while playing the game. Part of enhancing user experience is coming up with a highly interactive user interface.

Another design constraint is performance and speed. The game has to respond quickly and adequately to user inputs, primarily because the essential operation of the game is solely based on specific time constraints.

Scalability is another design constraint. The game user interface has to be designed such that it is highly responsive. This means that its display should adjust accordingly based on the size of the device being used to access the game from the web. In addition, devices with different operating systems should be able to access the game without any difficulties.

The **implication** of the above design constraints is that the developer can narrow down to the most critical aspects of the web-based gaming application to achieve maximum client satisfaction. As such, any existing uncertainties are removed.>

## [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 class diagram diagrammatically shows the classes that are to be used in the gaming application and their relationships. The entity class represents a gaming instance. It **includes** the game class, team class, and player class. Therefore, it can be stated that the entity class **encapsulates** the game, team, and player classes. These classes can also **inherit**specific methods from the entity class.

One game can have many teams. Each team can subsequently be composed of many players hence the **one-to-many** relationships used in the UML diagram. Each class is uniquely identified by an id, as shown in the class diagram.

The game service class represents a complete gaming activity or instance. For this class to be established, it requires the game, team, and player classes. This exhibits a relationship that applies **polymorphism.** A single game service is composed of the game, team, and player details.

****

## [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** | < **Characteristics-** Complex code; it is a closed platform; hence software and updates require payments.  **Advantages-** More secure, there is more flexibility (in commands used in the terminal).  **Disadvantages**- Dependent on the availability of Mac hardware which is expensive.> | < **Characteristics-** Open source  **Advantages-** High security levels, flexibility in terminal commands usage, cheaper since it is free, requires few resources.  **Disadvantages-**Associated with migration problems.> | < **Characteristics-** Often requires updates, associated with software purchases (it is not freely available)  **Advantages-** provides higher scalability making it easier to integrate the software application with a wide range of other applications.  **Disadvantages-** frequent updates may pose a convenience and speed problem, relatively insecure, performance dependent on specific hardware components such as Ram and processor.> | < **Characteristics-** Portable (can be used to access software applications anywhere and at any time)  **Advantages-** Faster speeds(downloading), better user engagement, more flexible and cost-effective.  **Disadvantages-** Limited screen size (smaller display of mobile devices), different specifications depending on the user.> |
| **Client Side** | < **Cost-** Expensive due to the cost of hardware and also due to the fact that it is subject to software purchases.  **Time-** an average amount of time required to support multiple clients.  **Expertise-** requires a moderate to high level of expertise/skills to navigate through the OS.> | < **Cost-** minimum costs required.  **Time-**a maximum amount of time is needed to support multiple clients on Linux systems.  **Expertise-** Requires a high level of expertise.> | < **Cost-** more expensive compared to Linux systems.  **Time-** little time required by the client for understanding how to support windows set up.  **Expertise-** Minimum levels of expertise required.> | < **Cost-** Flexibility provided has the end result of making it more cost-effective.  **Time-** Little amount of time is required.  **Expertise-** Users do not need high levels of expertise but are slightly more difficult to implement than other devices.> |
| **Development Tools** | < For the **frontend,** languages consist of HTML, CSS, and JavaScript with supporting libraries. **Development tools**- include eclipse, visual studio, GitHub, PyCharm, notepad++ and online development tools. The **IDE’s-** can be PHP, Ruby and Python.> | < **Frontend-** HTML, CSS and JavaScript with libraries to support the front end. **Development tools-** include visual studio, GitHub, PyCharm, notepad++ and online development tools.  **IDE’s-** can be eclipse, IntelliJ, PHP, Ruby and Python.> | < **Frontend-** HTML, CSS, bootstrap, React JS, Angular, jQuery and JavaScript.  **Development tools-** GitHub, GitLab, Docker, Jenkins, Stack overflow, among others.  **IDE’s**- eclipse, NetBeans, IntelliJ.> | < **Front end-** include but is not limited to HTML, CSS and JavaScript.  **Development tools-** GitHub, GitLab, Docker, Jenkins.  Programming languages- Python, Java, C++, android.  **IDE’s-** eclipse, NetBeans, IntelliJ, notepad++.> |

## 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 advise Gaming Room to consider windows operating platform for the “Draw It or Lose It” game. This is especially because of the wide range of users that have windows operating systems on their computers. Most users also find windows operating system easier to use. A windows server also offers support to various IDE’s and provides support to many other server roles like application server, database server, mail server and file server.
2. **Operating Systems Architectures**: Windows server operating system provides support to a wide range of IDE’s (scalability), supports more network connections, and provides support to a wide range of functionalities like file management, database management and application management.
3. **Storage Management**: The use of a windows server operating system enables the software application to be stored directly into the memory of a user’s device. This gives means that the application operates at high speeds as it is processed faster. A windows operating system server also offers an opportunity for cloud storage which offers more storage space.
4. **Memory Management**: Windows operating system provides support to dynamic memory allocation, which makes it possible to allocate more memory to virtual memory than the size of the host device’s physical memory. The result is a more flexible memory since the system can change the amount of memory while the machine is running. Applications are therefore able to run efficiently.
5. **Distributed Systems and Networks**: The use of windows operating system on the server allows different computer systems using more than one computer to run the game software. Support to distributed processing means that a single program can run simultaneously at different locations. As such, it is possible for the game to team up different players who are in different areas such that they are playing the same game together.
6. **Security**: Windows operating system on the server allows users to:

* Create user accounts that are password protected for authentication and authorization purposes.
* There are various Intrusion Detection and Prevention systems in windows operating system. For instance, windows firewalls help detect and remove potential threats in the network. In addition, antispyware components like the windows defender can help detect and remove viruses and trojan horses.
* Support for Virtual Private Network service protects user accounts from being accessed by unauthorized personnel.