

<Name-of-Software-Application>

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template** 1](#_Toc115077317)

[**Table of Contents 2**](#_Toc115077318)

[**Document Revision History 2**](#_Toc115077319)

[**Executive Summary 3**](#_Toc115077320)

[**Requirements 3**](#_Toc115077321)

[**Design Constraints 3**](#_Toc115077322)

[**System Architecture View 3**](#_Toc115077323)

[**Domain Model 3**](#_Toc115077324)

[**Evaluation 4**](#_Toc115077325)

[**Recommendations 5**](#_Toc115077326)

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | <01/27/2024> | <Olorunjuwon Olutoki> | <Create a web-based version for Gaming Room’s current game.> |

**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 has requested the services of Creative Technology Solutions (CTS) to develop the web-based version of the current version of their, draw it or lose it.>

CTS recommends that the draw it or lose it game application should be developed on a Linux operating system, using a cloud-based storage and microservices architecture.

## Requirements

*<A WEB BROSWER and the right programming language for web application design such as HTML5 is required to create this project, together with any cloud-based storage available.>*

## [Design Constraints](#_2et92p0)

<The new version of the game must be a web-based versions, the game must be accessible on the most-widely used web browsers and the game must be able to handle multiple concurrent users playing the game simultaneously.>

## [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)

<Describe the UML class diagram provided below. Explain how the classes relate to each other. Identify any object-oriented programming principles that are demonstrated in the diagram and how they are used to fulfill the software requirements efficiently.>

**"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)

Development Tools

* Identify the relevant programming languages and tools (IDEs and other tools) that are used to build this type of software for deployment on each operating platform. Consider the following and articulate your findings in the software design template:
  + What impact do these technical requirements have on a development team? Consider whether multiple development teams may be needed.
  + Are there licensing costs related to the development tools?
* There are various development tools available for Linux, including those for well-known programming languages like C, C++, Python, and Java. There are several Integrated Development Environments (IDEs) available, including Eclipse, Visual Studio Code, and JetBrains IDEs (such as IntelliJ IDEA, PyCharm). All these programming languages are compatible with windows OS and Linux OS. Generally, developer tools are **free** software applications or open-source programming languages. In some cases, OS like Linux and Mac require technical knowledge to use effectively, if necessary, there can be fees for support or enterprise-level tools. The professional who works on these operating systems requires the client to pay for their expertise, also pay for their training if they need to attend one, these are all related to the licensing cost.

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 offers a Unix-based environment, which is favorable for hosting web applications, has high level of security, has high quality proprietary software, and can be used as a server for web-based software applications. The disadvantage is that Mac is less commonly used compared to windows OS and MacBooks are usually more expensive compared to other laptops. Closed system: MacBooks have a closed system and there are fewer customization options than with a Windows laptop or PC. Limited compatibility. Not all programs and applications are compatible with the Mac | Linux offers a wide range of distributions suitable for server use, with excellent stability and scalability. Linux is highly customizable and has a command line interface. Linux is the recommended choice for web-based applications.  The disadvantage is that Linux for servers is more difficult and requires more expertise than windows, including working with the command line and manually editing configuration files. | Windows Server is a commonly used platform for hosting web applications, especially for applications built using Microsoft devices. Windows is compatible with a lot of software and its user friendly.  Windows OS is a flexible operating system widely used by businesses to host applications, provide secure access to data, and manage network infrastructure. The disadvantage to this is that it’s prone to hackers, it’s not secured. It Is Not Secure as Linux/Ma Hosting. You will find some big differences when you compare the security of Linux/Mac hosting versus Windows hosting. . | Mobile devices don't typically host web applications. They are portable, They are app based, and have touch screen interface for user control Limited resources can lead to performance issues. Scalability challenges. Reliability issues like battery drain and hardware failures. Tend to be slower than mobile apps. Although mobile web apps tend to be smaller in size, they can be slower than mobile apps. ... |
| **Client Side** | <Mac OS has a good processor for easy compilation and execution of codes. Developing for Mac clients typically involves using Apple's development tools such as Xcode. Apple silicon powered MacBook has a stronger battery life there enabling client to code for a long period of time.> | One of the most significant advantages of Linux for developers is its powerful command line interface (CLI). The CLI provides a high level of control and precision, enabling them to complete a variety of tasks quickly. Not all version of Linux software has long-term support.> | <Software development for windows include usage of visual studio, core understand and execution of Java programming language, it requires the design of UX and UI for proper functioning.> | <Developing for mobile devices involves platform-specific development. Costs can vary depending on the number of platforms targeted. Most times, Android programming languages are used on mobile phones except it’s an iPhone.> |
| **Development Tools** | < Swift is a powerful and intuitive programming language for all Apple platforms. It's easy to get started using Swift, with a concise-yet-expressive syntax and modern features you'll love. AppleScript is another scripting language created by Apple Inc. that facilitates automated control over scriptable Mac applications.> | < Linux supports a wide range of programming languages. Popular IDEs include Visual Studio Code, IntelliJ IDEA, and Eclipse. > | < The Windows 10 operating system is primarily written in the C, C++, and C# programming languages. Python is also recommended for windows OS.> | The relevant programming language for mobile devices: For iOS, Swift and for android Kotlin. Swift (iOS): Ideal for iOS app development, known for its performance and safety features. 2. Kotlin (Android): The preferred choice for Android app development, offering conciseness and compatibility with Java. |

## 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 Recommend appropriate operating platform that will allow The Gaming Room to expand Draw It or Lose It to other computing environments is a web-based platform because web-based platform has a web application ecosystem for a well-integrated, stable, and fast-performing environment.  Web based program such HTML/HTML5 provides high speed performance over different networks, it can handle many users simultaneously and since there will be multiple users concurrently playing the game, it is important to use a web-based platform to avoid errors when end users try to play the game.
2. **Operating Systems Architectures**: A multi-tier architecture separates the presentation, application, and data tiers of the application, allowing for more efficient scaling and easy user access, I recommend a client-server architecture. The client-server architecture refers to a system that hosts, delivers, and manages most of the resources and services that the client requests. Examples of computer applications that use the client–server model are email, network printing, and the world wide web. Because our aim to is creating a gaming app that is accessible to everyone, regardless of whatever type of Operating system the client is using, the traditional client-server model, the server is a centralized system that serves many clients, is highly recommended.
3. **Storage Management**: Cloud storage enables you to make data available anywhere, anytime you need it. Instead of being tied to a location or specific device, people can access data from anywhere in the world from any device, provided they have an internet connection. Any cloud-based storage will be okay, either Azure or AWS. Because the game involves images that occupies a lot of storage space, I will recommend any network drive and use cloud storage as a backup storage.
4. **Memory Management**: Techniques such as garbage collection, and memory pools are commonly used to manage memory in various programming environments. Garbage Collection: Automates the process of reclaiming memory, reducing the need for manual memory deallocation by the programmer, doing this, allows the gaming app to accommodate multiple users at the same time, without experiencing slowness in the gaming app performance.
5. **Distributed Systems and Networks**: Knowing that the client would like Draw It or Lose It to communicate between various platforms, this can be achieved by using the RESTful API. RESTful APIs are widely used for web applications and enable communication over the internet using HTTP. The recommended operating platform is a web-based platform using the HTML/HTML5 web programming language, provides excellent support for RESTful APIs and makes it easy to implement.
6. **Security**: As we all know, the importance of cyber security cannot be over emphasized, securing the gaming app is to protect user identity, personal information and every other data that doesn’t require an authorized access. This can be done by using the encryption method, and secure authentication protocols such as multifactor authentication and regular security updates must be done to ensure that the game is safe and compliant with the latest security standards.