

Completed Project

# **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 | 02/21/2024 | Brett Rush | Completed Project |

**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 customer has asked that Creative Technology Solutions (CTS) develop a web-based version of their current game, “Draw It or Lose It”.

CTS proposes to develop the web-based game using the following technologies:

**Operating Platform:** Linux

**Operating Systems Architectures:** Microservices architecture

**Storage Management:** Cloud-based storage

## [Design Constraints](#_2et92p0)

The main design constraint for this project is that the game must be developed in a web environment which can add difficulty to the project because multiple users will be accessing the game.

## [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)

Entity creates a relationship between Game, Team, and Player class. This means they all inherit their required info from Entity. With UML, it is demonstrated using inheritance. So, each class will share common references like “name” and “id”. When we look at their relationship, Team and Player are types. While Game has a Team and GameService has Games. When used with UML, it is aggregation.

**"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)

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** | Flexible terminal commands are available to configure the server, access, or enact changes, making it popular in web hosting. Its upgradability and diverse options cater to various web hosting needs, offering advantages for customization. However, it is less favored compared to other options for web hosting services due to certain limitations or drawbacks. | Characterized by its strong security measures, this hosting option is the most preferred option. Its primary advantage lies in its ability to detect and address security vulnerabilities before they escalate, making it the top choice for web hosting services. However, one downside is the challenge of finding applications that fully support the specific requirements of this hosting method. | With an abundance of software options in comparison to other operating systems, this platform is much more popular and preferred. environment. Its advantages include handling high resource demands efficiently, resulting in shorter loading times and increased user comfort. However, it is more susceptible to viruses and malware, and users may encounter challenges with technical support. | While being highly mobile, this makes for the platform to be much more susceptible to viruses. The platform is widely available to develop on since both iOS and Android are mostly open source. |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. | Maximum expertise and time required. Minimum cost due to it being extremely open-sourced. | Minimum expertise and time required. Cost similar to mac. | Provides flexibility to clients or even developers to see updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | Swift, Objective-C, Python, and more programming languages are supported by Mac OS. With functionality for both macOS and mobile app development, Xcode is the recognized integrated development environment for Mac and iOS applications.  **Impact on Development Team**:  **Explanation:**  Xcode accelerates Apple platform development, making it effective for teams aiming for macOS and iOS., it might demand Mac hardware from developers, possibly calling for specialized staff or tools.  **Licensing Costs**:  **Explanation:**  The Mac App Store offers Xcode for no charge, which lowers the cost of licensing the development environment. | **Development Tools:** Linux offers numerous development tools for well-known programming languages like C, C++, Python, Java, and more, including IDEs such as Eclipse, Visual Studio Code, and JetBrains. Linux provides package managers and command-line tools for software delivery.  **Impact on Development Team:** The variety of development tools available on Linux allows development teams to choose the ones that best fit their needs, though setting up and customizing development environments and familiarity with open-source technologies may require additional time and potential training.  **Licensing Costs:** Most Linux development tools are open-source and free | C#, C++,.NET, Python, and many other programming languages are supported by Windows. The main IDE for Windows programming is Visual Studio, which provides extensive features for desktop and web applications.  **Impact on Development Team**:  **Explanation:**  A user-friendly development environment is offered by Windows, Unless targeting numerous platforms, several development teams might not be required. There might be a need for compatibility testing on non-Windows platforms.  **Licensing Costs**:  **Explanation:**  There are several editions of Visual Studio, including the commercial Professional and Enterprise editions. | While Android Studio and Java are primarily used for Android development, Xcode and Swift/Objective-C are primarily used for iOS development. There are other cross-platform frameworks like React Native and Flutter.  **Impact on Development Team**:  **Explanation:**  Platform-specific teams are subject to language and tool restrictions. Cross-platform frameworks could eliminate the need for different teams, but they also necessitate an understanding of platform-specific issues.  **Licensing Costs**:  Downloading Xcode and Android Studio is free. When developing mobile apps, using third-party tools or cloud services may result in licensing fees. |

## 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 recommended operating platform for the game should be a web-based platform such as HTML5. This platform provides cross-platform support that allows users to access games on multiple devices and operating systems. A web-based platform will make it easier to update the game without requiring users to download and install updates.
2. **Operating Systems Architectures**: Games must be compatible with different operating system architectures. This can be achieved by using different sets of technologies such as JavaScript or Java. Also, the game must be tested on different operating systems to ensure it runs consistently and without any incompatibilities, whether it be on competing game consoles or differing computer operating systems.
3. **Storage Management**: Games must make use of cloud storage services to store user data and game progress. Cloud storage services provide scalability, reliability and availability across multiple platforms without the fear of players losing progress when upgrading or replacing their machines.
4. **Memory Management**: Games should be designed to optimize memory usage, especially for mobile devices. This can be achieved by reducing the use of large images and videos and using efficient encoding methods. Games should be designed to cache frequently accessed data to reduce load times and improve overall performance.
5. **Distributed Systems and Networks**: Games must be designed to handle distributed systems and networks to ensure secure and fast communication between servers and clients. This can be achieved by using a load balancing system and a content delivery network (CDN) to ensure that users are served by the closest server.
6. **Security**: The game must be designed with security in mind to prevent unauthorized access and data breaches. This can be achieved by using encryption and secure authentication protocols. Regular security checks and software updates must be done to ensure that the game is safe and up-to-date with the latest security standards to avoid detrimental user data leaks.