

<DRAW IT or LOSE IT!>

# **CS 230 Project Software Design Template**

Version 3.0

## 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 | 01/24/24 | Daniel Jones | First revision, inserted requirements and addressed concerns |
| 2.0 | 02/11/24 | Daniel Jones | Second Revision, evaluate the characteristics, advantages, and weaknesses of Linux, Mac, Windows, and mobile platforms when developing based on large server requirements, as well as costs, time, and development know how. |
| 3.0 | 3/2/24 | Daniel Jones | Updated final recommendations to client for clarity. |

**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 aims to create a cross-platform web game called “Draw It or Lose It.” While the current version is exclusive to android, the goal is to expand its availability across multiple platforms. The game involves multiple teams, each comprising of several contestants competing in four rounds lasting a minute each. During each round, a team guesses a picture randomly selected from a library of images, attempting to identify it before time expires. If a picture remains unanswered, members of the opposing teams have ethe opportunity to provide their guesses within a 15-second time frame.

## Requirements

These are the requirements needed to follow while writing the code and software. While this is only the game aspect, we still need to look at application development. The Gaming Room would like this to run on all devices. This means we already have it on android but need to work it into another mobile device. Along with machines like Windows, Linux, and Apple. To do this we will need to find a way to either re-write the code in swift for (Apple devices) or come up with a way to use existing code to be run on other devices by inheriting other languages. Like when we use multiple computer languages together to make stronger code.

## [Design Constraints](#_2et92p0)

* One or more teams need to be involved.
* Each team is made up of multiple people.
* Team names and game names must be unique to allow users to check whether the name is in use or free.
* Only one instance of the game can exist at any time.
* Must be able to run on multiple 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 creation of a relationship is established by the Entity among the Game, Team, and Player classes. This relationship includes either inheritance or information retrieval from the Entity class by Game, Team, and Player. Utilizing UML, we can visually represent this through inheritance, signifying that each class, namely Game, Team, and Player, shares common attributes such as "name" and "id" inherited from the Entity class, making Entity the superclass.

Analyzing the relationship, we observe that Team and Player exhibit a "has a" type connection. Specifically, Game has a Team, and GameService has Games. In UML, this is denoted as aggregation, representing a "HAS-A" relationship. In the context of a user, "has a" implies that it is an instance of one class while also holding a reference to an instance of another class.

Examining the UML diagram, we discern that GameService possesses a reference to Games, Games has a reference to Team, and Team has a reference to Player.

**"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 let the company configure the server, access, and make changes.  Characteristics: it is popular in web hosting  Advantages: it can be upgraded; it has multiple options for different web hosting requirements  Disadvantages: It is less preferred for web hosting services | Flexible terminal commands let the company configure the server, access, and make changes.  Characteristics: It is secure and most preferred.  Advantages: Security flaws are caught before they have a chance to become an issue, it is the most preferred choice for web hosting services  Disadvantages: it is more difficult to find applications to support the web hosting requirements. | Has the most software available compared to the other operating systems.  Characteristics: it is the dominant platform.  Advantages: High resource requirements, less loading time, high comfortability  Disadvantages: easy virus susceptibility, poor tech support | More lightweight than other OS’s Specs are better in other devices.  Characteristics: Most popular device, high portability.  Advantages: Has a wider reach, better compatibility, more cost-effective, larger audience.  Disadvantages: It is highly selective to various smart mobile devices, and many have poor security. |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. What are the needs of the app development process to ensure the application is compatible with all web browser platforms and mobile devices? | Maximum expertise and time required. Minimum cost required. What are the needs of the app development process to ensure the application is compatible with all web browser platforms and mobile devices? | Minimum expertise and time required. Cost similar to windows. What are the needs of the app development process to ensure the application is compatible with all web browser platforms and mobile devices? | Provides flexibility to clients and developers to access updates at any place. Slightly more difficult to implement than other devices. |
| **Development Tools** | When running languages on macs we can run swift the more popular option. While mixing in nice tools like notepad++. Though Macs can run all languages. Languages consist of HTML /CSS /JavaScript while supporting the frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Linux can work with visual studio, eclipse, along with notepad++ for a nice and easy-to-use tool. Along with many more languages and tools. Languages consist of but not limited to HTML/ CSS/ JavaScript while supporting libraries to support the front-end and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linuz but can run the same as it. So visual studio, eclipse to name a few of the many languages. And with multiple tools notepad++ is a simple to use the tool. Languages consist of but not limited to HTML/ CSS/ JavaScript while supporting libraries to support the front-end and general-purpose languages. These can be Java, Python, PHP, and Ruby. | You can create countless apps using android and swift. Both languages and software can be run on all three machines. Languages consist of but not limited to HTML/ CSS/ JavaScript while supporting libraries to support the front-end and general-purpose languages. These can be Java, Python, PHP, and Ruby. |

**Project Two addition**

**Web Based Application Support Chart**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Platform** | **Characteristics** | **Advantages** | **Weaknesses** | **Server-Based Deployment** | **Potential Licensing Costs** |
| **Mac** | **High user experience and reliability** | **High user experience, advanced security features** | **Smaller user based and hardware options compared with windows** | **Yes, with a MacOS server** | **Moderate** |
| **Linux** | **High scalability and stability** | **Cost-effective, more flexibility for server environments, and open-sourced** | **Steep learning curve for developers, software has limited support** | **Yes, second most used for servers** | **Low cost** |
| **Windows** | **Most users, wide compatibility, user friendly** | **Largest user base, high levels of software support, high customization** | **Licensing costs tend to be higher, higher possibility of security concerns** | **Yes, windows servers are the most used servers** | **Moderate to high** |
| **Android** | **Largest mobile device market share, high mobile optimization** | **Access to largest mobile user base, near seamless integration with Android devices** | **Older versions lack support, split between devices** | **No, app deployed through the Google Play store though backend can be hosted on a server** | **Low to moderate for google play store** |
| **iOS** | **Like MacOS known for user-friendly environment, security, and reliability** | **User experience is high quality with strong security features** | **Limited hardware options, strict guidelines from Apple** | **No server-based deployment, as App is deployed through the apple app store. The server can host the backend of application.** | **Low to moderate for Apple App store** |

**Does each of the operating platforms offer a server-based deployment method where the website will be hosted?**

Windows Mac and Linux offer direct Server-Based deployment, but mobile platforms only allow back end server hosting

**What are the potential licensing costs to the client, The Gaming Room, for the server operating system?**

The financial impact of licensing fees depends on the choice of development tools and platforms. Certain development tools and Integrated development environments off free or open-source alternatives, which can substantially reduce licensing expenses. For example, opting for an open-source web development tool like visual studio code or sublime text can lead to cost savings compared to using paid ide’s.

On the other hand, specific platforms or frameworks may entail licensing fees. For instance, windows development utilizing Visual studio might require licenses for particular features and distributing applications on apps stores such as Google Play or Apple App Store could also involve charges related to memberships and updates. To effectively manage licensing costs, the development team should meticulously assess the functionality and requirements of the chosen tools and platforms while comparing cost-efficient alternatives wherever feasible.

**What is required of the application development process to ensure the application is compatible with all web browser platforms and mobile devices?**

Cost: Similar to the licensing costs stated above. Depending on what development requirements there are, mobile platforms may require membership and update fees to keep the application up to date and compatible with devices.

Time: The development timeline will vary platform to platform as each platform has its own set of challenges and requirements to have the same results when compared to another platform. Trying to develop across multiple platforms at once will require more complex project management and longer development times.

Expertise: Supporting Multiple platforms requires a de team with expertise in different operating systems and development environments. Each platform will have its own unique set of best practices and the teams should be well versed in these differences to ensure the efficient flow of integration and development.

## Recommendations

Project Three Addition

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 suggest that The Gaming Room initiates it projects on Windows device due to the abundance of available software, couple with minimal expertise and cost required to kickstart projects. Additionally, working on windows ensures a wide array of IDEs(Integrated Development Environments) to choose from, providing flexibility and ample options for development.
2. **Operating Systems Architectures**: Windows offers essential services utilized by all Windows-based applications, facilitating the display of Graphical User Interface (GUI) while accessing system resources and more. These applications also involve Graphics and Miltimedia, messaging, and web services. Users can utilize these services through either a user account or a dedicated server.
3. **Storage Management**: Windows 10 includes a useful feature known as Storage Sense, enabling users to examine and oversee files on their hard drive, including monitoring the amount of space they have used and what comprises the used space. Additional features involved the option to designate preferred save locations for apps, simplifying their retrieval. Similar to other storage devices, windows 10 supports cloud storage for data backup. The built-in storage system facilitates seamless file creation and organization for significant projects, reducing the risk of misplacement or accidental deletion.
4. **Memory Management**: In the process of developing this game, it’s essential to establish a database or library containing an extensive collection of images. The memory allocation feature enables convenient storage of pictures outside the default picture folder. This setup ensures that your entire project is kept in a designated area on your computer. This arrangement remains intact even when working with your Integrated Development Environment (IDE) and accessing files from it to build the game.
5. **Distributed Systems and Networks**: Due to the variations among operating systems, I explored methods to delay the game across various devices. My research led me to discover Develop 4 which enables cross-platform game creation. This IDE is compatible with any device. After creating the game, it’s straightforward to export the game files to the web, iOS, Android, and other platforms, facilitating cross-play. This approach aids in managing dependencies. To mitigate potential issues such as outages or connectivity problems, the company must ensure robust server capabilities to accommodate high player volumes including provisions for backup power during power outages.
6. **Security**: Windows includes inherent security protection software. However, for enhanced safeguarding of user data and information. It is advisable to explore alternative security measures. Nevertheless, concerning the built-in features on the system, Windows comes equipped with protection capabilities. This system actively scans for malware, viruses, and security threats in real-time. To adapt to evolving threats, the system undergoes automatic updates to ensure ongoing safety for both the system and user information.