

<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)

**Continuing developing with project two\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_6**

**Evaluation 7**

[**Recommendations 8**](#_Toc115077326)

**Continuing system architectures with project three\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_9**

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 05/21/2023 | Lindsey DeLorenzo | Operating system comparison for best development for “Draw It or Lose It”. |
|  | 06/04/2023 | Lindsey DeLorenzo | Continuing within project two for the game “Draw It or Lose It”. |
|  | 6/18/2023 | Lindsey DeLorenzo | Continuing within project three for the game “Draw It or Lose It”. |

**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 company The Gaming Room wants to have a web-based game to be developed that can work on multiple platforms. The game that they would like to have made is called “Draw It or Lose It”. This app is currently only available on android. This game will allow multiple players to create teams and compete within four rounds of the game. This game will use stock images from a library to be used as clues for the players to solve the puzzles. If they do not guess in time the opposite team will have a chance to guess for points.

## Requirements

The requirements follow the design of the game. There must be one or more teams with multiple people on each team. The game must work on multiple devices. The game and team names must be unique and allow users to check whether a name is in use or not when choosing the team’s name.

Only one instance of the game can exist in memory at any one time.

## [Design Constraints](#_2et92p0)

The design constraints of this game lie within the requirement to be available on all platforms. This would require the code to be re-written to match devices such as swift for apple or find a way to use the current code and have it translated into other programming languages.

## [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)

Within the UML diagram, we can see that the Game, Team, and Player class have a relationship with Entity. This shows they all inherit and get information from the Entity. We can that each class share common features such as “name” and “id”. This will make Entity a superclass due to its aggregation lines. While looking at Game we see the similar games with Team and GameService. We can see that GameService has 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 must 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** | This needs flexible terminal commands to configure the server, access, or make changes. This web hosting is popular. The advantages are it has many different upgrade options with the web hosting language. The disadvantages are that it is a less preferred web hosting service. | The terminal is also flexible and is more cost friendly. The characteristics are preferred and well secured. The advantages are easily catching security issues. The disadvantages are it will be harder to support the app with the web hosting needs. | There are more software features available. The characteristics are dominant and make it a close platform. The advantages are that there are high resources, less loading time, and high comfortability. The disadvantages are that it us easy for viruses. | This is better for servers that are immobile. The characteristics have high portability and are popular. The advantages are the better compatibility and the cost-effectiveness. The disadvantages are that has poor security and is selective to various smart phones. |
| **Client Side** | The cost is similar to windows. It takes moderate time and expertise required. | The cost will be minimum. The time and expertise needed is high. | The cost is high, which is similar to mac. The time and expertise needed is low. | The cost can be low or moderate. The time and expertise is high. The mobility makes it harder to implement. |
| **Development Tools** | The relevant language is swift, though macs can run all languages. Language it runs are HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. |

## 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 recommend windows for The Game Room. Starting is easy for windows and the cost and expertise needed is low.
2. **Operating Systems Architectures**: Window provides services for all windows-based applications. They also make easy resources for graphics, multimedia, messaging, and web servicing.
3. **Storage Management**: Windows has a feature for storage called storage sense. This will allow easy file management and choose locations for apps to be saved easily. This will allow for cloud saving, making file creating and placement easy. This makes losing or deleting the files not as easy.
4. **Memory Management**: For the memory, this game will need a database or library for its stock images. This will create a easy file location for the game and for the developers to keep the project together and secure.
5. **Distributed Systems and Networks**: Because Draw It or Lose It needs to run on different devices, I believe Develop 4 is a good platform match, because of its ability to be cross-platform for game creation. The only worry the company needs to think about is that their servers can support a larger number of players with their backup power.
6. **Security**: Windows has built-in security for the software. Because it comes with protection the user can be safe with their information with the system updates.

**Continuing with Project Two**

This side of server-side configuration will allow hosting for the website and allows thousands of players. This means the ability to host needs to support a distributed environment. The network needs to be intensive and needs to be able to run the game and take inputs from the client’s side. The process needs to receive the clients request and output the rendering onto the clients’ screen.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | This is a flexible terminal that commands and configures the server, accesses the server, and makes changes to it. This has popular characteristics.  The advantages are that it is upgradable and has different web hosting options and requirements.  The disadvantages are that this is a less popular web hosting service option. | This is a flexible terminal that commands and configures the server, accesses the server, and makes changes to it, but is more cost friendly. This has the most popular characteristics.  The advantages are its efficient secure detecting and is the preferred web hosting option.  The disadvantages are it has difficult web requirement needs. | This has more software available when compared to OS. Its characteristics are dominant and is a close platform.  The advantages are it is comfortability, less loading time, and high resources.  The disadvantages are the poor tech support and easy for viruses. | This is better for servers that are immobile. The characteristics have high portability and are popular.  The advantages are the better compatibility, wider reach of clients, and the cost-effectiveness.  The disadvantages are that has poor security and is selective to various smart phones. |
| **Client Side** | The expertise needed is average and takes moderate time required to run.  The costs are similar to windows. | The expertise needed is high and the there is a high time requirement to run.  The costs are minimum. | The expertise needed is low and the time required is low.  The cost is similar to mac. | The expertise needed to implement are slightly high. Time can vary.  The cost can be higher. |
| **Development Tools** | The relevant language is swift, though macs can run all languages. Language it runs are HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. | The relevant languages can be many including HTML, CSS, JavaScript and the frontend support library can be Java, Python, PHP, and Ruby. |

## Recommendations

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room, after reviewing the host and client sides of the platform. Specifically, address the following:

1. **Operating Platform**: Again, I recommend Windows. There are many more features for the software available for the game. This can all be done with low cost and low expertise.
2. **Operating Systems Architectures**: Windows has Graphical User Interface (GUI) that will provide services for Window-based applications. The system can be used with graphics, messaging, and support user accounts.
3. **Storage Management**: Windows is equipped with a storage feature named storage sense. With this you can manage files on your hard drive and recognize the size of the space it takes. It can also save locations for easy relocation and can have cloud save data. This helps for not losing for carelessly deleting data.
4. **Memory Management**: this game needs a database or library for image storage. Memory allocation is helpful here to store and use the images for the game. This will make it easier to open files and keep the project together.
5. **Distributed Systems and Networks**: Develop 4 is good for cross-platform gaming. It can run on any device and can export easily to iOS, Android, and web. There will need to be strong servers to support the multiple players.
6. **Security**: Windows comes built in with security protection. However, user data will need to be monitored. A system scan can help this as well as viruses and threats. With evolving fraud, the system security would need to evolve as well.

**Continuing system architectures with project three**

Analyze the characteristics of and techniques specific to various systems architectures and make a recommendation to The Gaming Room, after reviewing the operating system architectures of the platform. Specifically, address the following:

1. **Operating Platform**: To expand with the popularity of the game the best operating platform would be cloud-based. A cloud platform allows the program to have centralized infrastructure. This will help the game to be able to largely grow the number users playing. They also allow for flexibility when developing features and can be operated across different devices easily.
2. **Operating Systems Architectures**: The cloud based platform should support many architectures in order to be played on different systems. Examples of this include x86 architecture and ARM architecture.
   1. The x86 architecture is good for system based from Intel x86. It provides a standard set and is supported and compatible with most operating system.
   2. The ARM architecture is good for mobile devices. It is compatible with both iOS and Android. It offers energy efficiency and is designed for low- power consumption.
3. **Storage Management**: For this program we will need a distributed file system. This will allow the data to be stored and accessed throughout the servers, scalability, and improved performance. Examples of these are Hadoop distribution file system and Amazon S3. Both are useful for the expanding game demands.
   1. Hadoop distribution file system or HDFS is made for storing and processing large amounts of datasets. It is fault tolerant and has data that is distributed in node clusters. This can handle the large storage requirements needed to run the game.
   2. Amazon S3 is Amazon simple storage system and is a cloud-based object storage service. This will provide our game with durability and scalability needed for storing and receiving data.
4. **Memory Management**: The memory for our game needs to be caching and have virtual memory. This will allow the game to have the best performance.
   1. Virtual memory is a technique that allows the system to manage and allocate data resources. The data amount is larger for the memory available to the game. This allows multitasking and better performance when accessing the data.
   2. Caching is the storing of frequently used data. This allows accessing this data quickly and reduces the need for disc space. Faster data retrieving will have better perform for the game.
5. **Distributed Systems and Networks**: A distributed system approach is needed to assist communication between the platforms and devices. Breaking down this process helps the components communicate over a network. Examples of these are API integration and message queuing.
   1. API integration is robust and specifies how the different software would communicate together to share data. This would help the game communicate between platforms.
   2. Message Queuing is asynchronous communication between components of a program. In this system information may not always be available, however this can allow offline connections.
6. **Security**: Security is crucial for the game. For security the game needs to be able to protect and maintain the user’s information. Examples pf these are user authentication, encryption, secure APIs, and Security Audits.
   1. User Authentication is when the program implements a verifying mechanism to allow access to the user’s data. This can be username and password authentication, multi-factor authentication or third-party authentication.
   2. Encryption implements techniques to protect storage and data. Secure socket layer or transport layer security can encrypt the communications between the platforms. Advanced encryption standard will make the data at rest and unreadable if the data is compromised.
   3. Secure APIs implement authentication and access controls. It uses protocols for requests of data between platforms.
   4. Security Audits need to stay updated. This can spot identity vulnerabilities and security weaknesses. This is needed for the games future emerging threats.