

Draw it or Lose it

# **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 | 05/14/23 | Alex Crosswhite | <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 Gaming Room wants to develop a web-based game, “Draw It or Lose It that can run on multiple platforms. The game is currently only available on android. The expectation is for the game to allow multiple teams consisting of several people. The game should last four rounds at a minute each. A picture is pulled from a library of images that will “draw” itself at a steady pace while one team guesses until time runs out. If the team is unable to guess each opposing team member gets to guess for 15 seconds.

## Requirements

*<* Please note: While this section is not being assessed, it will support your outline of the design constraints below. *In your summary, identify each of the client’s business and technical requirements in a clear and concise manner.>*

## [Design Constraints](#_2et92p0)

* Needs one or more teams involved
* Each team has multiple people
* Game and Team 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 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 Entity class creates a relationship between the Game, Team, and Player classes. This means they all inherit or get information from Entity. With a UML we can show this with inheritance. So, each class will share common references like “name” and “id”. Making Entity a superclass. When we look at their relationship, we see Team and Player is a “has a” type and have an aggregate relationship. GameService has at least one Game, each Game has at least one Team, each Team has at least one 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)

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Flexible terminal commands to access, configure, or make changes to the server.  **Characteristics:** It is popular in web hosting  **Advantages:** It is upgradeable, it has various options for different web hosting requirements  **Disadvantages:**  There are other more useful options for Web Hosting. | Linux also allows for flexible commands and is cost friendly.  **Characteristics:** Secured, most preferred.  **Advantages:** Security flaws are caught before they become an issue, it is the most preferred choice for web hosting services  **Disadvantages:** There are fewer options for Web Hosting Applications | More software available compared to other operating Systems.  **Characteristics:** It is most commonly used of the platforms. Close platform  **Advantages:** High resource requirements, less loading time, high comfortability  **Disadvantages:** High virus susceptibility, poor tech support | It's better if the server is immobile and can be tracked in a single place. Specifications are better in other devices.  **Characteristics:** More popular, high portability.  **Advantages:** Have a wider reach, better compatibility, cost-effective  **Disadvantages:** It is highly selective to various smart mobile devices Poor security |
| **Client Side** | Moderate expertise and time required. Cost similar to windows. What does the application development process need to allow the application to be compatible with all of the web browsers, platforms and mobile devices? | Maximum expertise and time required. Minimum cost. What does the application development process need to allow the application to be compatible with all of the web browsers, platforms and mobile devices? | Minimum expertise and time required. Cost similar to mac. What does the application development process need to allow the application to be compatible with all of the web browsers, platforms and mobile devices? | Provides flexibility to clients or even developers to see 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 but not limited to HTML/CSS/JavaScript while supporting libraries to support 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 frontend and general-purpose languages. These can be Java, Python, PHP, and Ruby. | Easier to use than Linux but can run the at the same efficiency. 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 are not limited to HTML/CSS/JavaScript while supporting libraries to support the frontend 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 frontend and general-purpose languages. These 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**: It is my recommendation that development should begin on windows devices first. Its varied language and IDE availability allows for easier initial development both in accessibility and in cost. This decision would also allow The Gaming Room to reach a wider audience earlier.
2. **Operating Systems Architectures**: Windows enables applications to display a Graphical User Interface (GUI) while accessing media, messages, web services and other system resources. A user account or a server can access this.
3. **Storage Management**: Windows offers the ability to choose where to save files for apps, including a cloud, making them easier to find. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted.
4. **Memory Management**: This game will need access to a database or library of pictures to draw from. The memory allocation allows for easy storage of pictures which allows you to keep your whole project together in a more secure area on your computer. This includes when you’re working with your IDE and opening files from it to create the game.
5. **Distributed Systems and Networks**:  It is important to have a reliable way to port this games files to differing operating systems. To prevent other problems like outages or connectivity issues, the company will need to make sure their servers are strong enough to support large player volumes along with backup power for power outages.
6. **Security**: Windows comes with built-in security protection software. Though, to secure user data and information it would be recommended to use another source. This system scans for malware, viruses, and security threats. Scans and system security updates should be done regularly to keep the system and user information safe.