

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 | 05/20/2023 | D.D. Sessions | Develop a web-based game to supplement *Draw It or Lose It* which is currently only available as an Android app. |

## 

## [Executive Summary](#_sbfa50wo7nsh)

The Gaming Room wants to develop a web-based game that serves multiple platforms based on the current game *Draw It or Lose It* which is only available as an Android application. The staff is not sure how to set up the environment. The application will pull images from a library of drawings as clues. A game consists of four one-minute rounds. Drawings are rendered steadily until the 30-second mark. If the team does not guess the puzzle before the time expires, the remaining teams can offer one guess each with a 15-second time limit.

## Requirements

* A game will have the ability to have one or more teams involved.
* Each team will have multiple players assigned to it.
* Game and team names must be unique to allow users to check whether a name is in use when choosing a team name.
* Only one instance of the game can exist in memory at any given time.

## [Design Constraints](#_2et92p0)

* Create a web-based game to allow accessibility to more users on various operating systems.
* Keep the user interface similar to the existing Android app game.
* Games built with HTML work on smartphones, tablets, PCs, and Smart TVs. Use HTML and CSS to build, style, and lay out the game’s user interface.
* Use JavaScript to write code for the game’s speed and performance.
* Create unique identifiers for each instance of a game, team, or player.

## [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 UML class diagram below demonstrates that the Game, Team, and Player classes will inherit from the Entity class. The ProgramDriver class will use SingletonTester to ensure only one instance of the game exists in memory.

**"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** | Mac is not as common as Linux or Windows for web hosting, but it is very secure and reliable. Some apps may not be compatible with Mac, and there are less support options. | Linux is open-source and widely used for web-based hosting. You can share between multiple servers. It is secure, stable, and customizable. Support is limited. | Windows is popular for hosting web-based software. It is widely used and user-friendly. Windows can be more vulnerable to malware and requires regular updates and maintenance. | Mobile devices are becoming more popular for hosting. They have smaller screens, less power, less memory, and less storage. However, they are portable, always connected to the internet, and easy to use. |
| **Client Side** | Mac web hosting is more expensive than Windows and Linux for hardware and software. Developing software on Mac may be quicker than Windows depending on the tools you use. | Linux is generally one of the cheapest types of hosting. It can be more difficult than Mac and Windows because it requires more technical knowledge. Linux is lightweight and has a large community of open-source projects so development may be faster. | Many developers are familiar with Windows, but it can be more expensive than other operating systems. | Developing software from a mobile device could be done from virtually anywhere, and these devices are simple to use. |
| **Development Tools** | You can use JavaScript, HTML5, CSS, and Node.js. Visual Studio Code is a common IDE that is Mac compatible. JetBrains products are also useful. | Linux is favorable for C#, JavaScript, PHP, Python, Ruby, and MySQL. Visual Studio Code is a common IDE that is Linux compatible. JetBrains products are also useful. | Some common languages for Windows include JavaScript, Python, Java, and C#. Visual Studio Code is a common IDE that is Windows compatible. JetBrains products are also useful. Many frameworks are supported such as Angular and Bootstrap. | Specific tools would be needed such as native mobile development, low-code platforms, and conversion tools. I’m not aware of any IDEs available for mobile devices. |

## Recommendations

1. **Operating Platform**: I would recommend using Windows. It is easy to use, and many developers are familiar with the interface. It would be cost-effective for hosting, plus it does not require much specialized expertise.
2. **Operating Systems Architectures**: Windows architecture is designed to provide a modular structure made up of simple modules. This structure makes it easier for developers to create applications. The kernel-mode components of Windows include the base operating system services such as memory management, process and thread management, security, input/output communication, networking, and interprocess communication.
3. **Storage Management**: Windows storage management supports compression, encryption, quotas, dynamic disks, and storage spaces for optimizing local drives.
4. **Memory Management**: Windows uses memory management techniques including paging, segmentation, and virtual memory. For a web-based game, the browser is the application that is running the game. The browser uses virtual memory to allocate memory for the game. This allows the operating system to use more memory than it physically has by temporarily transferring pages of data from RAM to disk storage.
5. **Distributed Systems and Networks**: You can use serverless-first architecture on AWS (cloud service) to enable connectivity from the game client to a serverless-first backend for both single- and multi-player games. Another option is to use a load balancer to efficiently distribute incoming traffic across multiple backend servers in a server pool.
6. **Security**: Protecting user information is crucial. Windows has several security features that can help with this. It uses the Trusted Platform Module to provide certificate storage and enhance computer security and privacy. The TPM helps with encryption and decryption of data and protecting authentication credentials. Microsoft 365 apps and services have built-in information protection and governance capabilities.