

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 | 01/24/24 | Alex Heang | Modifications were implemented on the cover page, the document revision history, the executive summary, design constraints, system architecture view, domain model, and recommendations. |

**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 objective of the Gaming Room project is to create a web-based game that can be accessed on various platforms. It is inspired by the existing mobile game "Draw It or Lose It," currently available only on Android. The game involves teams guessing drawings from a library of images within a time limit. Design constraints require unique game and team names to facilitate users in checking name availability. The system architecture view and domain model illustrate the interaction and flow of the game, wherein multiple teams engage in four rounds, each lasting a minute. If a picture is not guessed correctly within the time limit, the opposing team has the opportunity to answer within a 15-second timeframe.

## [Design Constraints](#_2et92p0)

*● Compatibility with various platforms is essential.*

*● Each team needs to consist of multiple players.*

*● The game should be restricted to only one active instance at any given time.*

## [Domain Model](#_8h2ehzxfam4o)

The Entity establishes a connection between the Game, Team, and Player classes, indicating that they all inherit or derive information from the Entity class. In UML, this relationship is represented through inheritance, designating Entity as the superclass. Examining their associations, it becomes evident that Team and Player exhibit a "has a" relationship. Specifically, Game has a Team, and GameService has Games. In UML terminology, this is referred to as aggregation (HAS-A), signifying that a user "has a" relationship involves an instance of one class having a reference to an instance of another class. In this diagram, we observe that GameService holds a reference to Games, Games holds a reference to Teams, and Team holds a reference to Players.

**"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** | Mac offers convenient access and configurable server settings, featuring an easily navigable graphical user interface and versatile terminal commands. | Affordable but challenging to navigate, the platform provides a command shell for straightforward server configuration and accessibility. | The server-side comes at a higher cost but offers a user-friendly graphical user interface along with a command prompt. | Specifications excel in other devices, and the specifications of mobile devices vary based on individual user preferences. |
| **Client Side** | Costly for users.  Moderate time and expertise are necessary. Precise skills are required to navigate the operating system. | Considerable expertise and time investment are necessary. Proficiency in Linux data is a prerequisite for operating system usage. It comes with a high cost for Linux users. | Pricier compared to Linux systems, but it is user-friendly with an easy learning curve for supporting a Windows setup. Requires minimal expertise. | Offers flexibility for clients and developers to view updates from any location. Slightly more challenging to implement than on other devices. |
| **Development Tools** | Utilizes languages such as HTML, CSS, and JavaScript, along with libraries supporting frontend development. Other development tools include PyCharm, GitHub, Visual Studio, and more. | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. Other. Linux systems include JavaScript, Ruby, PHP and Python | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. Developer tools include Eclipse, command prompt, PyCharm, Eclipse | Languages that consist of HTML, CSS and JavaScript. Libraries to support frontend and languages. Developer tools include Eclipse, command prompt, PyCharm, Eclipse |

## Recommendations

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

**Operating Platform**

The most suitable operating platform for expanding Draw It or Lose It to various computing environments is Windows. Windows stands out due to its abundance of Integrated Development Environments (IDEs), providing a versatile environment for development.

**Operating Systems Architectures**

Microsoft Windows, a graphical operating system developed by Microsoft, is recommended. It offers comprehensive functionalities such as file storage, software execution, gaming, video playback, and internet connectivity.

**Storage Management**

Windows 10 introduces a beneficial feature called Storage Sense, facilitating the scrutiny and management of files on the hard drive. This feature is advantageous for overseeing the storage of Draw It or Lose It photos and game-related data, ensuring efficient use of space.

**Memory Management**

Leveraging Windows 10's Storage Sense, efficient memory management can be achieved. It allows for the storage and organization of Draw It or Lose It photos and game-related information within a secure and consolidated memory space.

**Security**

While Windows comes with built-in security protection, it is advisable to augment user data and information security by employing additional security measures from external sources.

**Distributed Systems and Networks**

For network-based multi-user interaction systems like network games, incorporating a shared database among physically distributed players is crucial. Currently, developers often need to build the shared database and inter-player communication from scratch. Consider exploring existing solutions or frameworks to streamline the implementation of shared databases and communication in network games, reducing development effort and enhancing efficiency.