

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 | 05/21/2023 | Kendal Guizado O’riley | Initial 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)

Our client, The Gaming Room, is interested in expanding their current game Draw it or lose it, to other platforms. Currently the game is operating solely in an Android app. Our purpose will be to build a web version of this game that would be available to multiple platforms.

## Requirements

1. Expand game to multiple platforms
2. Allow for multiple teams
3. Each team must have multiple players
4. Unique names for games and teams
5. Only one instance of the game at any time

## [Design Constraints](#_2et92p0)

1. Expand game to multiple platforms
2. Allow for multiple teams
3. Each team must have multiple players
4. Unique names for games and teams
5. Only one instance of the game at any time

The first design constraint we will experience is how to expand this game that is currently on android to work on multiple platforms. Once this is addressed, we will have to face the requirements constraints that are the design of the game, and how it specifically should function.

## [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)

We first start with Entity, which creates a relationship to the Game, Team, and Players classes. This relationship will allow all these classes to inherit and get information from the Entity class through the process of inheritance. Entity is a Super Class. GameService has a reference of Game, which has a reference of Team, which has a reference of 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** | Popular for web hosting.  Mac is easy to upgrade and has many options for web hosting.  But it is the least preferred for web hosting services.  It has higher hardware requirements than Linux which could increase cost because of the use of Apple Hardware.  Licensing costs could also be added for software and tools. | Security.  It has security advantages that are easier to catch before it causes a big impact.  Not as many applications to support web hosting.  Applications can easily be scaled for number of users.  Servers are known for stability and uptime, great for apps that need to be available at any moment. | More available software  compared to mac and Linux.  Fasting loading times.  Less secure than max and Linux.  Has higher hardware requirements than Linux that can increase costs. Costs can be significant in larger scale developments. | Highly portable.  Wider reach, highly compatible.  Poor security.  Larger user base making it easier to reach a wider audience.  May not have the same computing power or resources as desktop platforms. |
| **Client Side** | Moderate cost time and expertise required.  May have to create different versions of the software for different platforms. | Maximum levels of expertise and time required. Minimal cost.  May require using cross platform development tools or creating separate versions for different platforms. | Minimal level of expertise and time required. Moderate cost.  May require cross platform development or creating separate versions for different platforms. | High level of  expertise, minimal time, and cost.  May require specialized development tools and programming languages. |
| **Development Tools** | For mac we would use swift as a highly used option. Popular mac languages would be HTML CSS and JavaScript, as well as Python, Java, and PHP, and Ruby. | For Linux we could use eclipse, and studio. As in Mac useful languages would be HTML, CSS, JavaScript, as well as Python, JAVA, PHP and Ruby. | For Windows we could use visual studio.  Useful languages would be HTML, CSS, JavaScript, as well as Python, JAVA, PHP and Ruby. | For mobile devices we could use swift and android.  Useful languages would be HTML, CSS, JavaScript, as well as Python, JAVA, 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**: Since we are considering a web-based game I would first start with a windows operating platform as it seems it would have the most available software and resources to get the project done. It has a minimal level of expertise required as well as time, and a moderate cost of implementing.
2. **Operating Systems Architectures**: Windows can allow applications to provide a Graphical User Interface that allows for the use of system resources.
3. **Storage Management**: Windows has a great storage management device that would allow for storage on local drives, as well as it is able to utilize cloud storage.
4. **Memory Management**: Since we will be processing large amounts of picture data to play the game, we will need a database of pictures. This will allow all our information to be together in one location which we can access rather quickly.
5. **Distributed Systems and Networks**: There are options available such as Develop 4 that would allow for the game to be played in multiple platforms.
6. **Security**: Windows comes with a suite of security options. We would utilize these options to ensure the game is secure. We would also have to outsource some security measures for users who are playing through the web, so that their identity, password, and personal information remain secure.