

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 4**

**Evaluation 5**

**Recommendations 6**

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0 | 01/21/23 | Jose Soqui | Update on the design requirements, domain models, development evaluation, UML, and reviewed recommendations. |
| 1.0 | 02/05/23 | Jose Soqui | Update to logistics within the newly appointed requirements for the app development on multiple platforms for client end interpretation. |
|  |  |  |  |

**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 plans to create a web-based game that can be played on several platforms. The game, "Draw It or Lose It," will only be accessible on Android at this time. We must make this available on both Android, iOS and Mac OS. This game's objective is to have numerous teams made up of several people to go through four rounds at a minute each. One team makes guesses up until the timer goes off when a photo is selected from a library of images. If not, until the 15 seconds have passed, each member of the opposing side may respond. This will require multiple users with a single instance of the game in existence within the memory constraints.

## 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. This can be accomplished by creating unique identifiers for each instance of a game, team, or player.

## [Design Constraints](#_2et92p0)

* Must run on multiple platforms.
* A single instance of the game can exist at any given time.
* Game and team names must be unique to allow users to check availability.
* Needs 1 or more teams involved, and each team would require multiple people.

When considering application development, it is important to keep in mind that the code must be compatible with all platforms. This means that the code must be written in a way that is compatible with all the platforms, or that existing code can be leveraged to work on other devices. By doing this, the code can be written once and then adapted to work on all platforms, making it easier to maintain and update.

## [System Architecture View](#_ilbxbyevv6b6)

Windows includes several built-in tools and utilities to help you manage and monitor the system. They include task manager, events, and device managers. These tools are essential for managing and monitoring your system. They provide detailed information about the current state of the system and can be used to troubleshoot any issues that may arise. With these tools, you can ensure that your system is running smoothly and efficiently.

## 

## [Domain Model](#_8h2ehzxfam4o)

The relationship between Game, Team and Player is an important one, as it is the basis for how the game is played. Games are the entities that provide the structure and rules for the game, while teams and players are the entities that interact within the game. In UML, this relationship can be represented by inheritance, as all three entities are inherited from the Entity superclass. This superclass contains common attributes such as id and name, which are shared by all three entities. The relationship between Team and Player is a has-a relationship, meaning that an instance of one class has a reference to an instance of another class. In the diagram below, GamesService has a reference to Games, Games has a reference to Teams, and Teams has a reference to Players. This means that one GamesService can have multiple Games, each Game can have multiple Teams, and each Team can have multiple 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.

| **Development Requirements** | **Mac** | **Linux** | **Windows** | **Mobile Devices** |
| --- | --- | --- | --- | --- |
| **Server Side** | Flexible terminal commands to configure the server, access, and make changes. Popular in web hosting with upgrading availability.  The server end will have to be scalable and handle the demand of the player base, while adhering to the security standards. | Same as Mac, more cost effective. Advantages include the security and cost effectiveness. Disadvantages It is more difficult to find applications to support the web hosting required needs.  Same as Mac for the server end of scalability and maintenance. | This would have more software to its disposal compared to other OS platforms. Advantages High resource requirements, less loading time, high comfortability  Disadvantages easy virus susceptibility.  Server would require HTTP and SQL | Specifications are better in other devices. More popular, high portability.  Advantages Have a wider reach, better compatibility, cost-effective.  Although, it is highly selective to various smart mobile devices and has poor security. |
| **Client Side** | Moderate expertise and time required. Cost like windows. | Maximum expertise and time required. Minimum cost. | Minimum expertise and time required. Cost similar to Mac. | Provides flexibility to clients or even developers to see updates at any place. Less on the implementation to other devices. |
| **Development Tools** | Swift is the clear choice for simplicity and maintenance, although it could use tools such as notepad++. Although you could use other languages as it is not limited to swift. | Linux can work with visual studio, eclipse, along with notepad++. Same as Mac, you could use other languages such as HTML/CSS/JavaScript | Languages consist of but not limited to HTML CSS/JavaScript while supporting libraries to support the frontend and general-purpose languages. IDEs same as Linux. | You can create countless apps using android and swift. Both languages and software can be run on all three machines. |

## 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**: The recommended OS would be to start on Windows due to the minimum expertise required to start and cost effectiveness. There are also many IDEs to work with on Windows including Eclipse and Visual Studio. There is also the matter of availability in the technical sense for Windows as it has gained popularity and is favored by average individuals for daily browsing.
2. **Operating Systems Architectures**: Windows provides services used by all Windows-based applications that enable applications to show a Graphical User Interface (GUI) while accessing system resources. There is also the modularity of Windows OS that allows for easy customization and scalability where components can be added or removed to allow for adaptation to hardware or software requirements.
3. **Storage Management**: Windows 10 comes with a nice feature called storage sense. This allows you to scrutinize and manage files on your hard drive, along with how much space it takes up. Other features include being able to choose to save locations for apps making them easier to find. And just like other drives, you can also use the cloud to save data. The built-in storage system allows for easy file creation and placement for large projects, so they won’t get lost or carelessly deleted. This is usually followed by a network of security keeping the storage protected from any changes without approval.
4. **Memory Management**: When creating a game, it is important to create a database or library with lots of pictures. This will help to ensure that the game looks and feels professional, as well as making it easier to access the images when needed. Memory allocation is a great way to store these images outside of the default picture folder, as it allows you to keep your project together in a more secure area on your computer. This is especially useful when you’re working with your IDE and opening files from it to create the game.
5. **Distributed Systems and Networks**: One of the most important objectives is ensuring that their servers are strong enough to support large player volumes. This is especially important for games that are played online, as they need to be able to handle many players at once. Additionally, developers should also consider having backup power in case of power outages, as this can prevent players from being able to access the game. Overall, cross-platform game development is a something that is required rather than optional, as it allows them to create games that can be enjoyed by a wider audience. This turns out to more profitability and discoverability.
6. **Security**: Windows comes with built-in security protection software that is designed to protect user data and information. This can aid in searching for malware, viruses, and other security threats in real-time, and updates automatically to keep the system and user information safe. It is also important to note that the built-in security protection software is not a substitute for good security practices. It is recommended that users supplement the built-in security protection software with additional security measures, such as anti-virus software, firewalls, and other security measures.