

Draw It or Lose It

# **CS 230 Project Software Design Template**

Version 1.0

## Table of Contents

[**CS 230 Project Software Design Template**](#_l6ti7uoag22u)1

[**Table of Contents**](#_30j0zll)2

[**Document Revision History**](#_grjogdjh5fi8)2

[**Executive Summary**](#_sbfa50wo7nsh)3

[**Design Constraints**](#_2et92p0)3

[**System Architecture View**](#_ilbxbyevv6b6)3

[**Domain Model**](#_8h2ehzxfam4o)3

[**Evaluation**](#_2o15spng8stw)3

[**Recommendations**](#_m8aleynsvzvc)5

## [Document Revision History](#_grjogdjh5fi8)

| Version | Date | Author | Comments |
| --- | --- | --- | --- |
| 1.0  1.1  1.2  1.3 | 09/19/21  09/25/21  10/04/21  10/16/21 | Robert O’Hanley  Robert O’Hanley  Robert O’Hanley  Robert O’Hanley | Initial software design for “Draw It or Lose It”  Updated Domain Model information  Created evaluation chart for multiple platforms  Making recommendations for task |

**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 called “Draw It or Lose It.” The game is currently only on Android and would like the game to be available on multiple platforms. The game will consist of four rounds of play that last one minute each. Drawings will be rendered from a large library stock and teams will be given 30 seconds to guess the puzzle. Once the time expires, the remaining teams will have one guess each to solve the puzzle with a 15 second time limit. I recommend using Java as the programming language because of its ability to create multi-platform applications.

## [Design Constraints](#_2et92p0)

* The game must be able to run on multiple platforms.
* Game and Team names must be unique and must perform checks for duplicates.
* Only one instance of the game can run at a given time.
* The game should have the ability to have one or more teams involved.
* Each team must be able to have multiple players.

When we move forward to create this application, we will need to take into account that the game must run on multiple platforms. Java is an incredible at making multi-platform applications and is a strong candidate for this project. The game and team names having to be unique require us to run checks for duplicates and catch them before they cause confusion within the game. To ensure that only one instance of a game is running at a given time, we will need to also implement a check in order to ensure that only one instance of a game gets started. The game having the ability to support one or multiple teams, as well as the players on those teams, will require the creation of lists to hold all of the data.

## [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)

Our Entity class creates a solid base for our Game, Team, and Player classes which each of those inherit information from. Those three classes have access to the id and name attributes in the Entity class. This ensures that our code is free of duplication making the code easier to read and less cluttered. When starting at our GameService class, we can see that there is a reference to our Game class. Then looking on we can see that our Game class has an instance of Team, and then Team has an instance of player within their methods. Finally, when looking at our ProgramDriver class, we can see that the main function exists inside of this and it also uses our SingletonTester class.

**"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** | A popular and easy to use server-based deployment method would be the Mac OS X server. Advantages would be high security and disadvantages would be that is not as widely supported as other options. Potential costs could be high depending on if the company does not have MacOS products. | Linux is an open source operating system that has a strong following and community. Linux is one of the most widely used operating systems to host websites. Advantages are that cost is the lowest out of the options. Disadvantages are that the operating system can be challenging to use. | Windows has the most amount of third party software available. Some advantages are ease of use and performance capabilities. A disadvantage would be its vulnerability to viruses. Another disadvantage would be multiple software requirements along with configuration to ensure things run smoothly. | Hosting a web-server on a mobile device gives you the power to always have access to your server at your fingertips. Some disadvantages would be possible instability of your network can have a negative impact on the stability of the server. Deploying a server on this platform is possible, however, it will be more difficult than our other options. |
| **Client Side** | Software development considerations using Mac OS would be that the systems are generally more expensive, it will take a moderate amount of time and expertise. | Considerations for using the Linux OS would be that it requires the lowest cost, but comes with a higher amount of time and expertise to support multiple clients as they pertain to Linux. | Cost wise, Windows is moderately expensive, around the same expertise as Mac OS, and a low amount of time required. | The cost for a mobile web-server would be low. The amount of time needed would be high and expertise would also be high due to the short comings that come from programming on a mobile device. |
| **Development Tools** | Popular IDEs for Mac OS would be Eclipse, Atom, and Xcode. Languages used would be Python, PHP, Java, HTML, CSS, Javascript, and more. | Relevant IDEs for Linux would be Eclipse, notepad++, and others. Relevant programming languages would be HTML, CSS, Javascript, Java, PHP, Python, and more. | Popular IDEs for the Windows would be VS code, Eclipse, Atom, and many more. Programming language that will be used are Java, HTML, CSS, Java script, Python, SQL, and more. | Some mobile device languages would be Java and Kotlin for Android. For Mac OS, swift is a language of choice. You can also find support for HTML, CSS, and Java script. IDEs that could be used would be Android Studio and xCode. We also have a number of IDEs that we can find on app stores to fit our needs. |

## 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 operating platform that I would recommend for The Gaming Room to expand Draw It or Lose It would be the Windows operating system. This operating system requires minimum expertise at a reasonable cost and has many options for development including many IDEs and programming languages that can be used.
2. **Operating Systems Architectures**: The use of a graphical user interface (GUI) enables ease of use when it comes to development. Windows also offers two different architectures, 32 bit and 64 bit. If you want to develop software compatible with older software, using the 32 bit system would be better suited, otherwise, you gain more performance and utility when using the 64 bit system.
3. **Storage Management**: The most useful storage system I feel would be to make use of Cloud storage. This will ensure that you can access your data from anywhere and ensure its safety. Making use of Cloud storage ensures that information will be backed up and that it is easy to access across different development platforms.
4. **Memory Management**: When it comes to memory, Windows is going to use a few techniques to ensure that memory is properly managed. Windows will be able to move needed data from storage and put it into memory such as the random access memory (RAM) and the cache. This will be done using virtual memory and only the information that is needed will be transferred into memory to ensure speed and stability.
5. **Distributed Systems and Networks**: Cloud computing can help ensure that all devices are able to communicate with each other. In order to ensure that connectivity is maintained and that there are no outages, you will need to ensure that the servers that have been chosen to be used are reliable and that you also have a reliable back up just incase there is an outage to ensure that there is no data lost and connectivity has minimal disruption.
6. **Security**: Windows has a variety of security features implemented in the operating system. Built in, Windows has a firewall that is effective against unverified access, some systems support biometrics such as a fingerprint scanner or face/iris scanner, and they also use a username and password based authentication. Inside of the game, the use of REST API will ensure that there is proper authentication for users as well as determine role permissions to help ensure application and data security for both the users and also the developers.